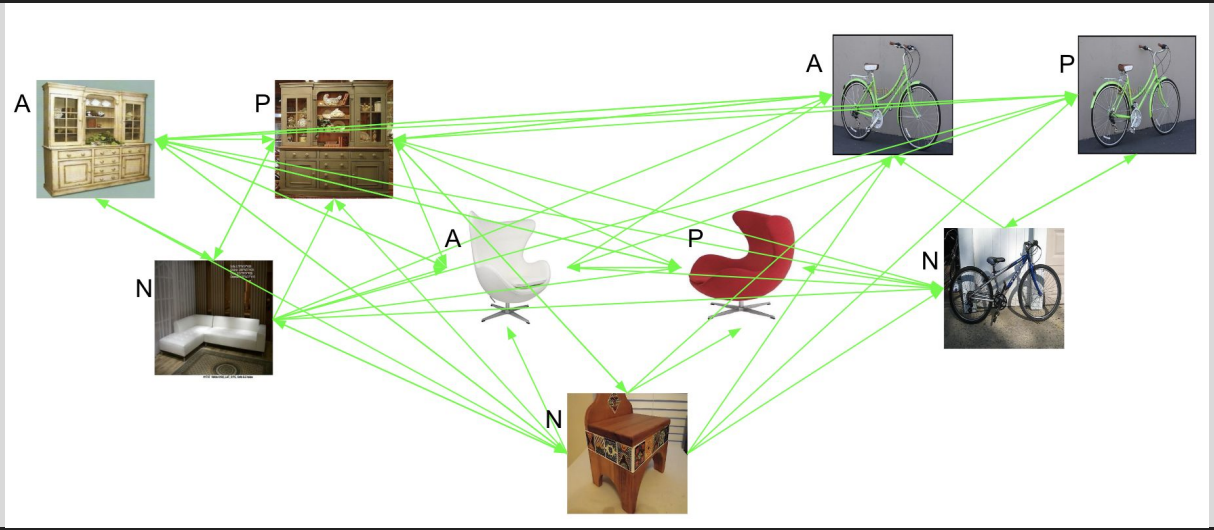
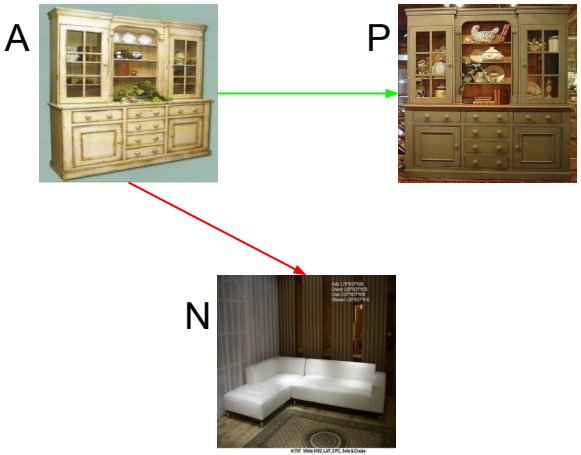
DML II: Beyond Pairs and Triplets - Contextual classification losses



Ismail Elezi

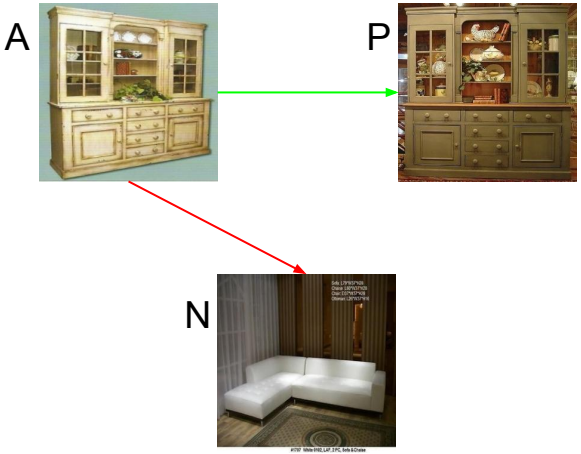
Technical University of Munich

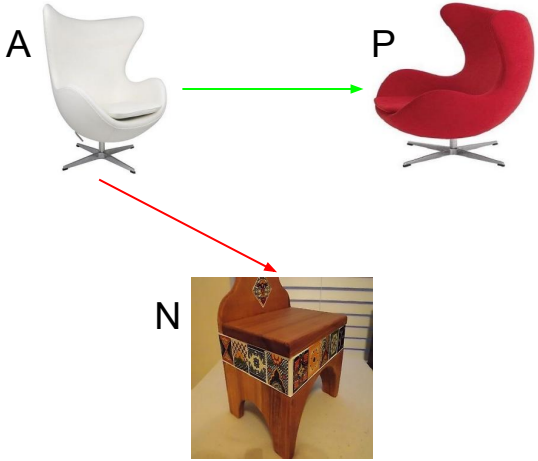
The Problem with the Triplet Loss

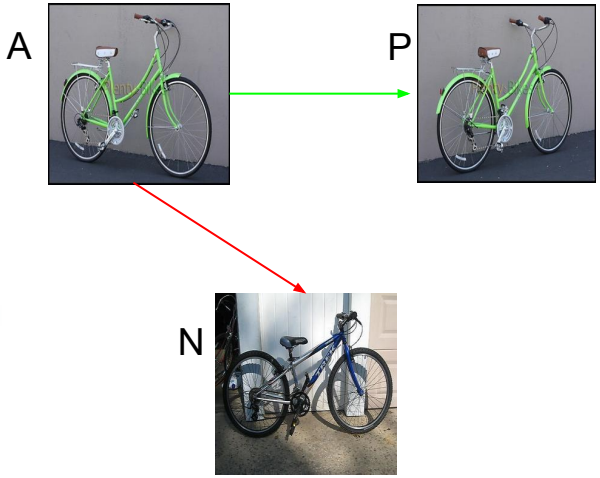


For a triplet, the triplet loss uses 2 relations: 1 between the anchor and the positive, and one between the anchor and the negative.

The Problem with the Triplet Loss

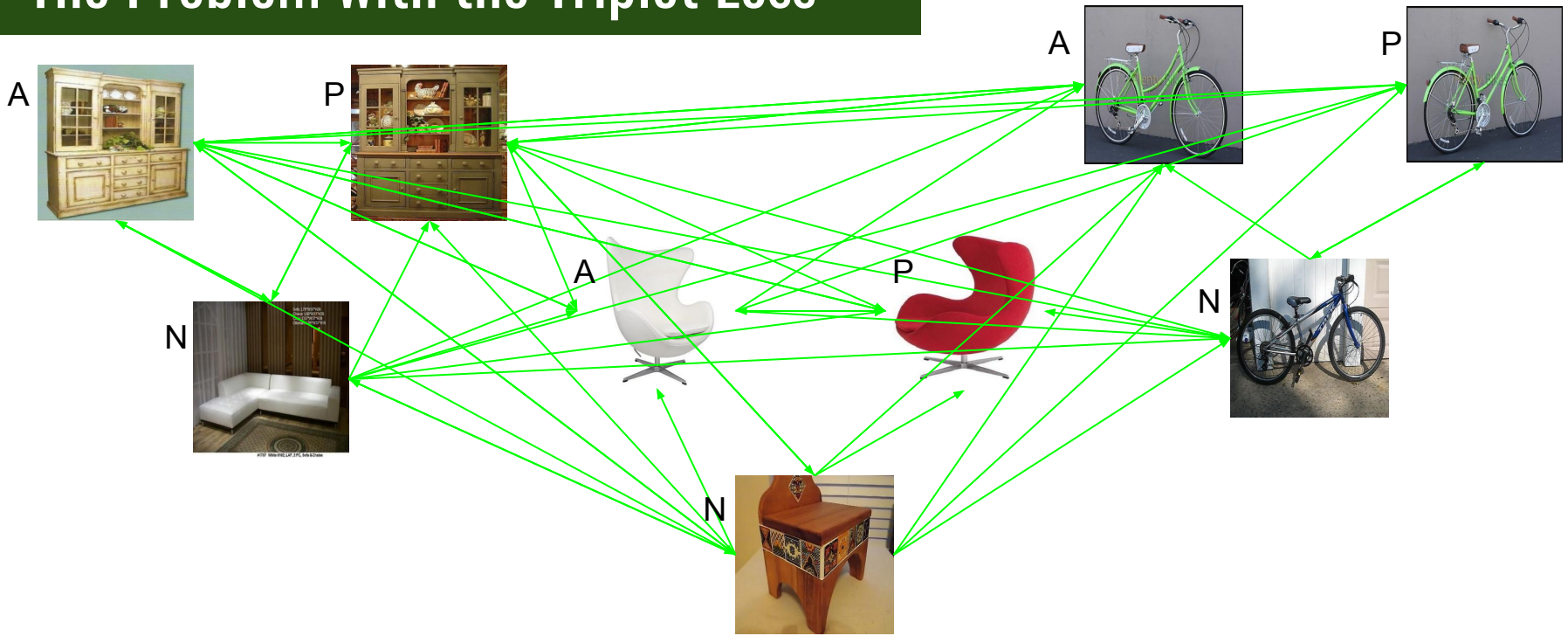






For a mini batch consisting of N triplets, the number of relations that triplet loss uses is 2N/3.

The Problem with the Triplet Loss



What if we could take advantage of all N^2 (or in case of symmetry $N(N-1)/2$) relations between the samples in the minibatch?

Two ways of doing so

1) Proxy losses.

2) Information propagation losses.

Possible to combine them.

Proxy losses

But before that a classification loss

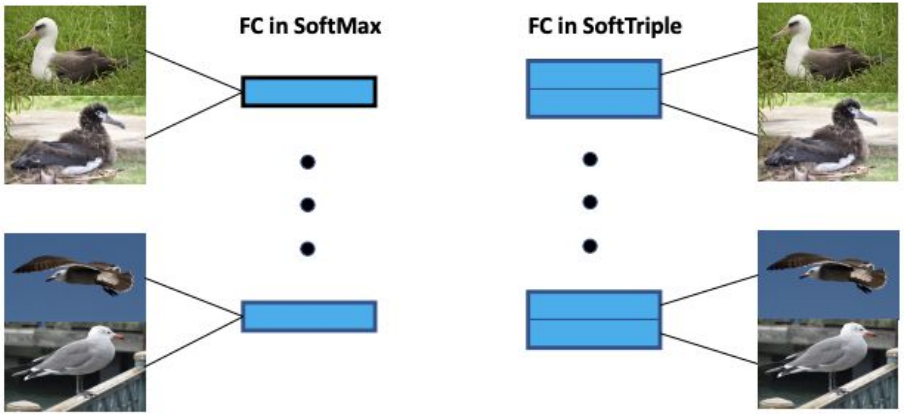


Figure 1. Illustration of the proposed SoftTriple loss. In conventional SoftMax loss, each class has a representative center in the last fully connected layer. Examples in the same class will be collapsed to the same center. It may be inappropriate for the real-world data as illustrated. In contrast, SoftTriple loss keeps multiple centers (e.g., 2 centers per class in this example) in the fully connected layer and each image will be assigned to one of them. It is more flexible for modeling intra-class variance in real-world data sets.

Qian et al., SoftTriple Loss: Deep Metric Learning Without Triplet Sampling, ICCV 2019

But before that a classification loss

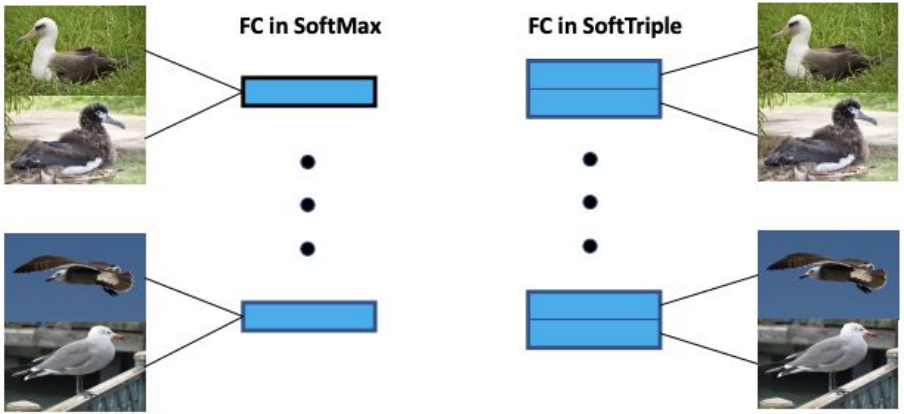


Figure 1. Illustration of the proposed SoftTriple loss. In conventional SoftMax loss, each class has a representative center in the last fully connected layer. Examples in the same class will be collapsed to the same center. It may be inappropriate for the real-world data as illustrated. In contrast, SoftTriple loss keeps multiple centers (e.g., 2 centers per class in this example) in the fully connected layer and each image will be assigned to one of them. It is more flexible for modeling intra-class variance in real-world data sets.

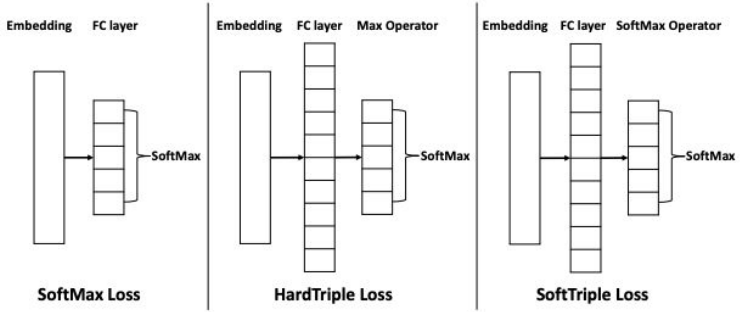


Figure 2. Illustration of differences between SoftMax loss and proposed losses. Compared with the SoftMax loss, we first increase the dimension of the FC layer to include multiple centers for each class (e.g., 2 centers per class in this example). Then, we obtain the similarity for each class by different operators. Finally, the distribution over different classes is computed with the similarity obtained from each class.

Qian et al., SoftTriple Loss: Deep Metric Learning Without Triplet Sampling, ICCV 2019

The need for proxies

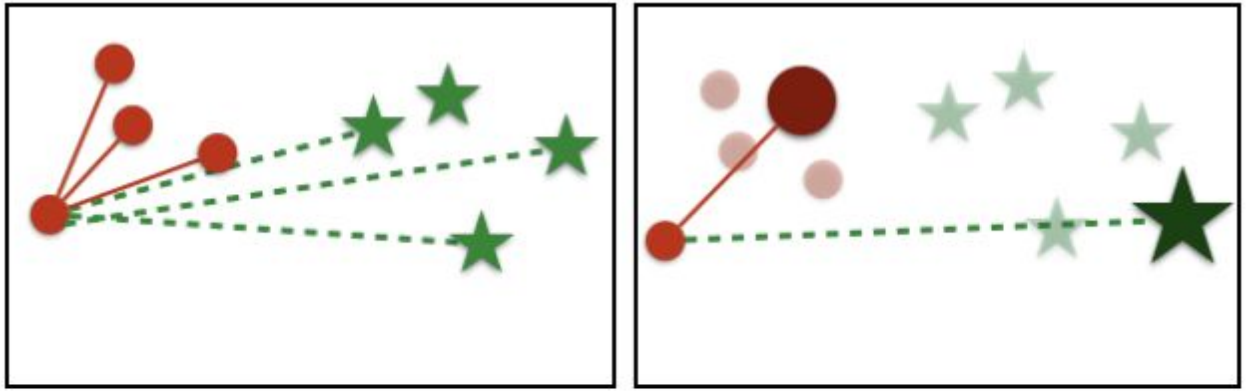


Figure 2: Illustrative example of the power of proxies. [Left panel] There are 48 triplets that can be formed from the instances (small circles/stars). [Right panel] Proxies (large circle/star) serve as a concise representation for each semantic concept, one that fits in memory. By forming triplets using proxies, only 8 comparisons are needed.

Proxy-NCA

Algorithm 1 Proxy-NCA Training.

Randomly init all values in θ including proxy vectors.
for $i = 1 \ldots T$ **do**
  Sample triplet (x, y, Z) from D
  Formulate proxy triplet $(x, p(y), p(Z))$
  $l = -\log\left(\frac{\exp(-d(x,p(y)))}{\sum_{p(z)\in p(Z)} \exp(-d(x,p(z)))}\right)$
  $\theta \leftarrow \theta - \lambda \partial_\theta l$
end for

Y is the positive proxy,
Z is the set of proxies.

Movshovitz-Attias et al., No Fuss Distance Metric Learning using Proxies, ICCV 2017

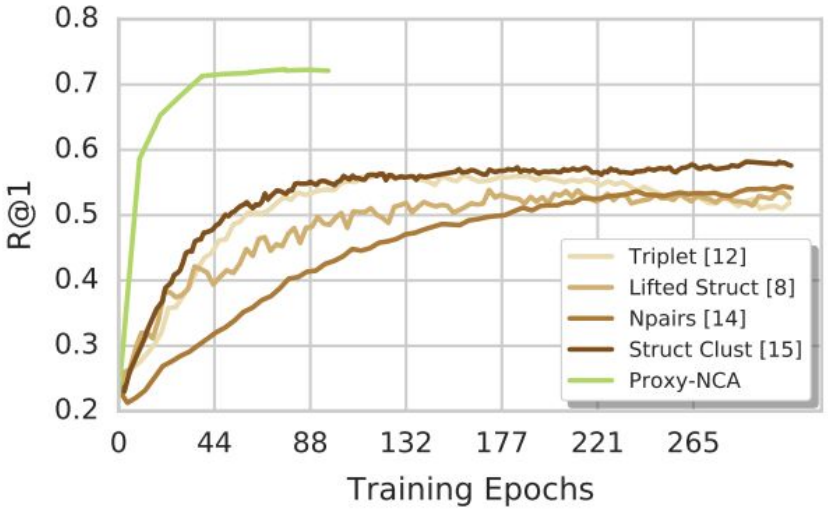
Results and convergence speed

	R@1	R@2	R@4	R@8	NMI
Triplet Semihard [12]	51.54	63.78	73.52	81.41	53.35
Lifted Struct [8]	52.98	66.70	76.01	84.27	56.88
Npairs [14]	53.90	66.76	77.75	86.35	57.79
Proxy-Triplet	55.90	67.99	74.04	77.95	54.44
Struct Clust [15]	58.11	70.64	80.27	87.81	59.04
Proxy-NCA	**73.22**	**82.42**	**86.36**	**88.68**	**64.90**

Table 1: Retrieval and Clustering Performance on the Cars196 dataset. Bold indicates best results.

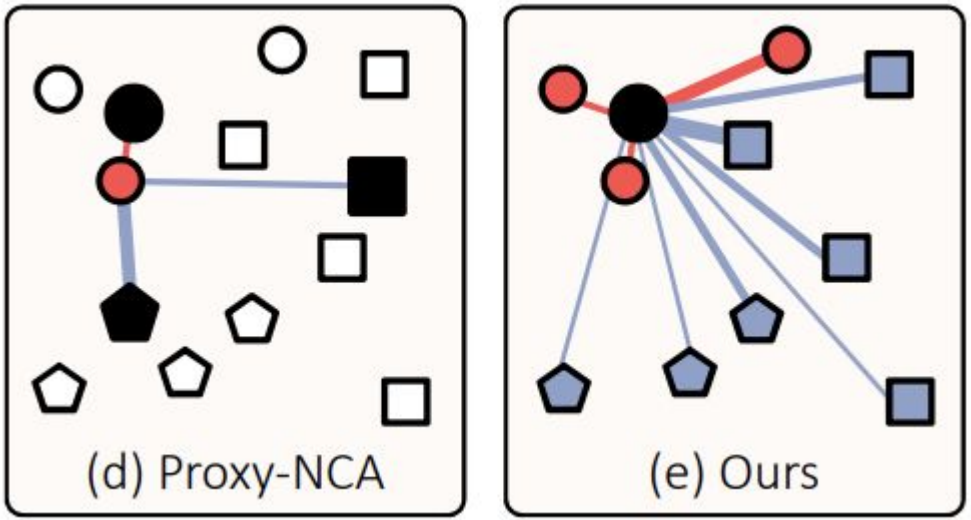
	R@1	R@2	R@4	R@8	NMI
Triplet Semihard [12]	42.59	55.03	66.44	77.23	55.38
Lifted Struct [8]	43.57	56.55	68.59	79.63	56.50
Npairs [14]	45.37	58.41	69.51	79.49	57.24
Struct Clust [15]	48.18	61.44	**71.83**	**81.92**	59.23
Proxy NCA	**49.21**	**61.90**	67.90	72.40	**59.53**

Table 2: Retrieval and Clustering Performance on the CUB200 dataset.



Using proxies as anchors

In Proxy-NCA, the anchors is a data point, while positive and negative samples are proxies.



In Proxy-Anchor, the anchor is a proxy, while positive and negative samples are data points..

Kim et al., Proxy Anchor Loss for Deep Metric Learning, CVPR 2020

Roughly the same loss as Proxy-NCA

$$\ell(X) = \frac{1}{|P^+|} \sum_{p \in P^+} \log \left(1 + \sum_{x \in X_p^+} e^{-\alpha(s(x,p)-\delta)} \right) + \frac{1}{|P|} \sum_{p \in P} \log \left(1 + \sum_{x \in X_p^-} e^{\alpha(s(x,p)+\delta)} \right),$$

where $\delta > 0$ is a margin, $\alpha > 0$ is a scaling factor, P indicates the set of all proxies, and P^+ denotes the set of positive proxies of data in the batch. Also, for each proxy p, a batch of embedding vectors X is divided into two sets: X_p^+, the set of positive embedding vectors of p, and $X_p^- = X - X_p^+$.

Fast convergence

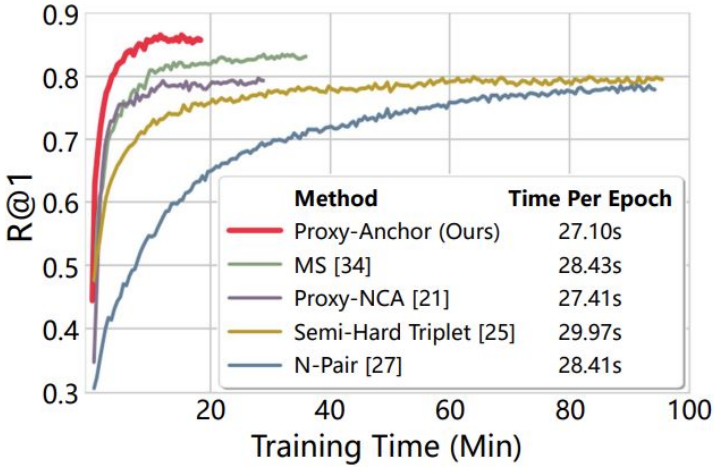
Type	Loss	Training Complexity
Proxy	Proxy-Anchor (Ours)	$O(MC)$
	Proxy-NCA [21]	$O(MC)$
	SoftTriple [23]	$O(MCU^2)$
Pair	Contrastive [2, 4, 9]	$O(M^2)$
	Triplet (Semi-Hard) [25]	$O(M^3/B^2)$
	Triplet (Smart) [10]	$O(M^2)$
	N-pair [27]	$O(M^3)$
	Lifted Structure [29]	$O(M^3)$

Table 1. Comparison of training complexities.

Fast convergence

Type	Loss	Training Complexity
Proxy	Proxy-Anchor (Ours)	$O(MC)$
	Proxy-NCA [21]	$O(MC)$
	SoftTriple [23]	$O(MCU^2)$
Pair	Contrastive [2, 4, 9]	$O(M^2)$
	Triplet (Semi-Hard) [25]	$O(M^3/B^2)$
	Triplet (Smart) [10]	$O(M^2)$
	N-pair [27]	$O(M^3)$
	Lifted Structure [29]	$O(M^3)$

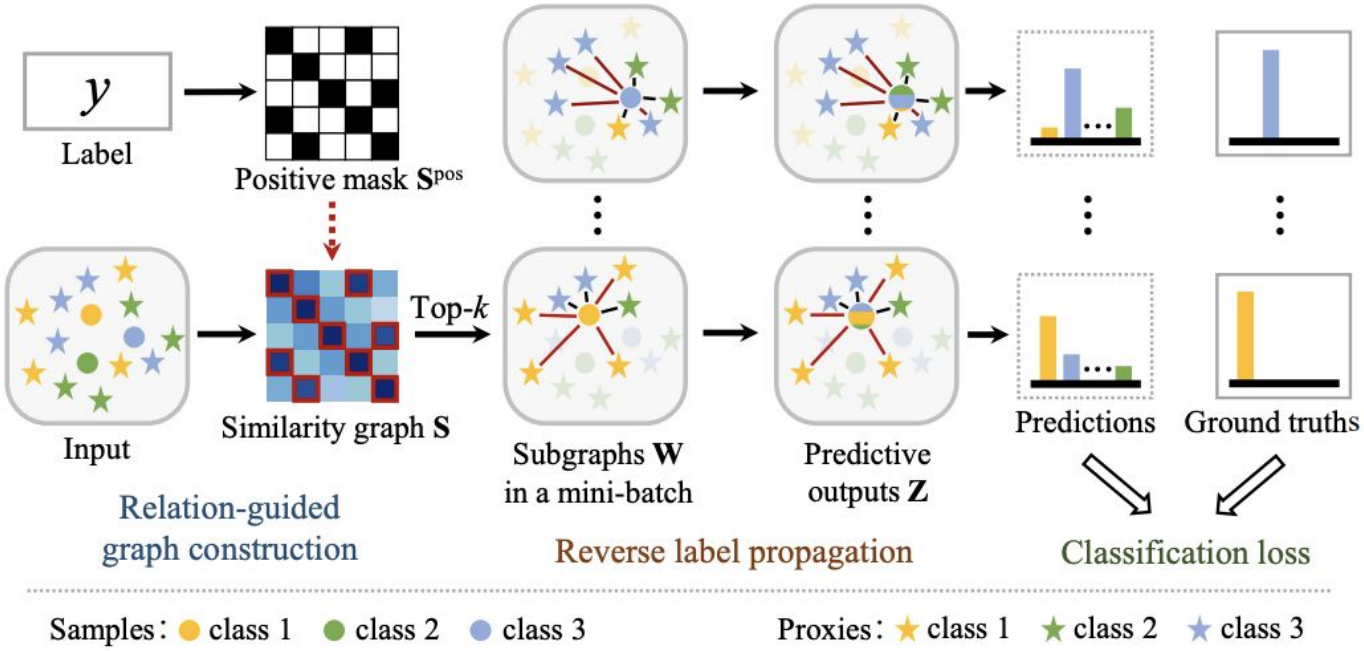
Table 1. Comparison of training complexities.



Results

Recall@K		CUB-200-2011				Cars-196			
		1	2	4	8	1	2	4	8
Clustering^{64} [28]	BN	48.2	61.4	71.8	81.9	58.1	70.6	80.3	87.8
Proxy-NCA^{64} [21]	BN	49.2	61.9	67.9	72.4	73.2	82.4	86.4	87.8
Smart Mining^{64} [10]	G	49.8	62.3	74.1	83.3	64.7	76.2	84.2	90.2
MS^{64} [34]	BN	57.4	69.8	80.0	87.8	77.3	85.3	90.5	94.2
SoftTriple^{64} [23]	BN	60.1	71.9	81.2	88.5	78.6	86.6	91.8	95.4
Proxy-Anchor^{64}	BN	**61.7**	**73.0**	**81.8**	**88.8**	**78.8**	**87.0**	**92.2**	**95.5**
Margin^{128} [37]	R50	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1
HDC^{384} [40]	G	53.6	65.7	77.0	85.6	73.7	83.2	89.5	93.8
A-BIER^{512} [22]	G	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1
ABE^{512} [15]	G	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1
HTL^{512} [7]	BN	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7
RLL-H^{512} [35]	BN	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1
MS^{512} [34]	BN	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5
SoftTriple^{512} [23]	BN	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9
Proxy-Anchor^{512}	BN	**68.4**	**79.2**	**86.8**	**91.6**	**86.1**	**91.7**	**95.0**	**97.3**

Combining proxies with label propagation



Zhu et al., Fewer is More: A Deep Graph Metric Learning Perspective Using Fewer Proxies, NeurIPS 2020

Key is the label propagation

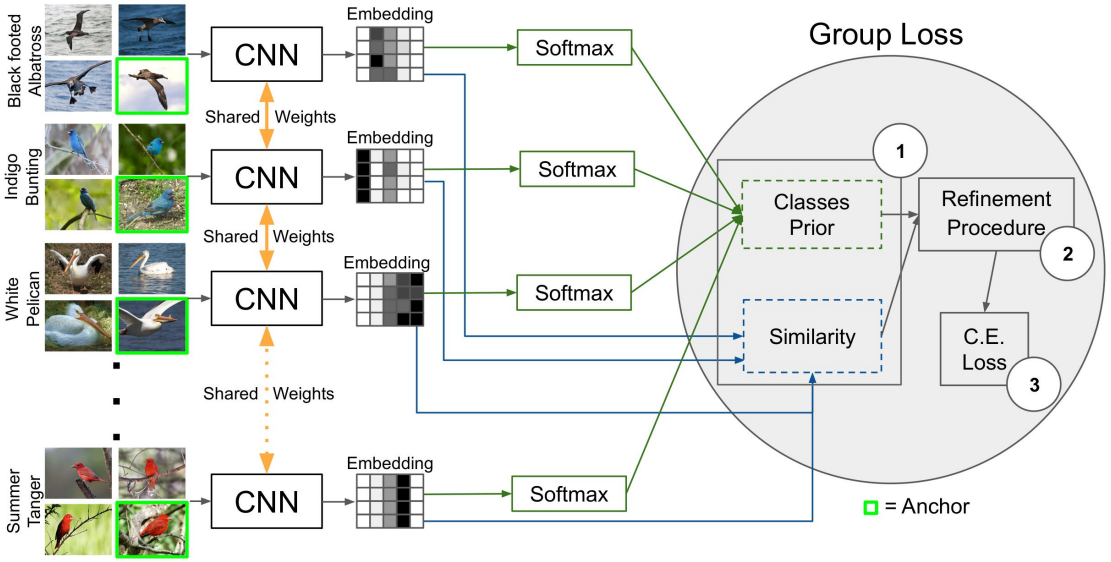
Figure 3: Subgraph evolution during the loss back-propagation process. Meanings of colors and shapes are the same as Fig. 2.

Results

Method		CUB-200-2011				Cars196				Stanford Online Products			
		NMI	R@1	R@2	R@4	NMI	R@1	R@2	R@4	NMI	R@1	R@10	R@10
SemiHard^{64} [29]	BN	55.4	42.6	55.0	66.4	53.4	51.5	63.8	73.5	89.5	66.7	82.4	91.9
Clustering^{64} [24]	BN	59.2	48.2	61.4	71.8	59.0	58.1	70.6	80.3	89.5	67.0	83.7	93.2
LiftedStruct^{64} [25]	G	56.6	43.6	56.6	68.6	56.9	53.0	65.7	76.0	88.7	62.5	80.8	91.9
ProxyNCA^{64} [23]	BN	59.5	49.2	61.9	67.9	64.9	73.2	82.4	86.4	90.6	73.7	–	–
HDC^{384} [45]	G	–	53.6	65.7	77.0	–	73.7	83.2	89.5	–	69.5	84.4	92.8
HTL^{512} [5]	BN	–	57.1	68.8	78.7	–	81.4	88.0	92.7	–	74.8	88.3	94.8
DAMLRRM^{512} [40]	G	61.7	55.1	66.5	76.8	64.2	73.5	82.6	89.1	88.2	69.7	85.2	93.2
HDML^{512} [46]	G	62.6	53.7	65.7	76.7	69.7	79.1	87.1	92.1	89.3	68.7	83.2	92.4
SoftTriple^{512} [26]	BN	69.3	65.4	76.4	84.5	70.1	84.5	90.7	94.5	**92.0**	**78.3**	90.3	95.9
MS^{512} [35]	BN	–	65.7	77.0	86.3	–	84.1	90.4	94.0	–	78.2	90.5	96.0
ProxyGML^{64}	BN	65.1	59.4	70.1	80.4	67.9	78.9	87.5	91.9	89.8	76.2	89.4	95.4
ProxyGML^{384}	BN	68.4	65.2	76.4	84.3	70.9	84.5	90.4	94.5	90.1	77.9	90.0	96.0
ProxyGML^{512}	BN	**69.8**	**66.6**	**77.6**	**86.4**	**72.4**	**85.5**	**91.8**	**95.3**	90.2	78.0	**90.6**	**96.2**

Information propagation losses

The Group Loss



① **Initialization**: Initialize X, the image-label assignment using the softmax outputs of the neural network. Compute the $n \times n$ pairwise similarity matrix W using the neural network embedding.

② **Refinement**: Iteratively, refine X considering the similarities between all the mini-batch images, as encoded in W, as well as their labeling preferences.

③ **Loss computation**: Compute the cross-entropy loss of the refined probabilities and update the weights of the neural network using backpropagation.

The goal of the loss function is to refine the soft-labels predicted by a neural network, using an iterative procedure based on the similarity between the images in the minibatch.

Elezi et al., The Group Loss for Deep Metric Learning, ECCV 2020

The Group Loss - 1) Initialization

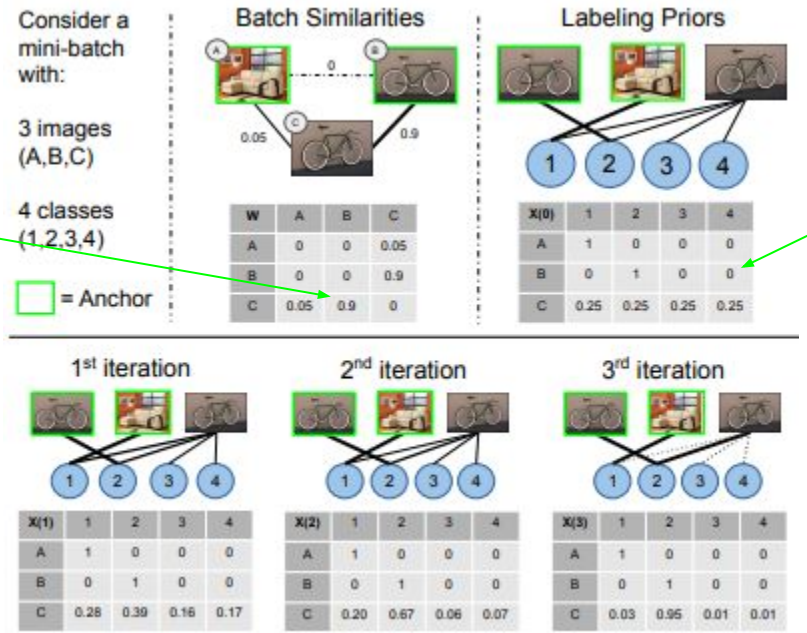
1) The local information is represented as a probability matrix given by the softmax layer of the neural network. Some of the entries (called *anchors*) are set to one-hot labeling in order to propagate noiseless information. These entries do not contribute to the loss function, instead they guide the remaining samples towards their correct labeling.

2) A measure of similarity is computed between all pairs of embeddings in the minibatch to generate a similarity matrix. We compute the similarity between embeddings *i* and *j* using Pearson's correlation:

$$\omega(i,j) = \frac{\mathrm{Cov}[\phi(I_i), \phi(I_j)]}{\sqrt{\mathrm{Var}[\phi(I_i)]\mathrm{Var}[\phi(I_j)]}}$$

The Group Loss - 2) Refinement - toy example

B and C are much more similar than A and B, or A and C, so the entry is set to a much higher value.



Maybe the net is untrained and so the probabilities are initialized to an uniform distribution.

The Group Loss - 2) Refinement - Replicator Dynamics

Iterate t + 1 times

Propagate information

$$x_{i\lambda}(t+1) = \frac{x_{i\lambda}(t)\pi_{i\lambda}(t)}{\sum_{\mu=1}^{m} x_{i\mu}(t)\pi_{i\mu}(t)}$$

Lambda is the class

Normalize in order to stay in the standard simplex

The Group Loss - 2) Refinement - Replicator Dynamics

Iterate t + 1 times

Propagate information

$$x_{i\lambda}(t+1) = \frac{x_{i\lambda}(t)\pi_{i\lambda}(t)}{\sum_{\mu=1}^{m} x_{i\mu}(t)\pi_{i\mu}(t)}$$

Lambda is the class

Normalize in order to stay in the standard simplex

From the similarity matrix

$$\pi_{i\lambda} = \sum_{j=1}^{n} w_{ij} x_{j\lambda}$$

This measures the support that the current mini-batch gives to image i belonging to class lambda

The Group Loss - 3) Loss Function

1) Compute cross-entropy over the refined probabilities.

2) Backpropagate over the entire net.

- Group Loss has no parameters to learn, but it propagates the gradients over the network.

- This is very different to softmax cross-entropy loss.

The Group Loss - Algorithm

Algorithm 1: The Group Loss

Input: input : Set of pre-processed images in the mini-batch $\mathcal{B}$, set of labels y, neural network ϕ with learnable parameters θ, similarity function ω, number of iterations T

Compute feature embeddings $\phi(\mathcal{B}, \theta)$ via the forward pass
Compute the similarity matrix $W = [\omega(i, j)]_{ij}$
Initialize the matrix of priors $X(0)$ from the softmax layer
for $t = 0, \ldots, T\text{-}1$ **do**
    $Q(t) = \mathrm{diag}([X(t) \odot \Pi(t)]\, \mathbb{1})$
    $X(t+1) = Q^{-1}(t)\, [X(t) \odot \Pi(t)]$
Compute the cross-entropy $J(X(T), y)$
Compute the derivatives $\partial J / \partial \theta$ via backpropagation, and update the weights θ

Results

	CUB-200-2011					CARS 196					Stanford Online Products			
Loss	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@10**	**R@100**	**NMI**
Triplet [35]	42.5	55	66.4	77.2	55.3	51.5	63.8	73.5	82.4	53.4	66.7	82.4	91.9	89.5
Lifted Structure [39]	43.5	56.5	68.5	79.6	56.5	53.0	65.7	76.0	84.3	56.9	62.5	80.8	91.9	88.7
Npairs [37]	51.9	64.3	74.9	83.2	60.2	68.9	78.9	85.8	90.9	62.7	66.4	82.9	92.1	87.9
Facility Location [38]	48.1	61.4	71.8	81.9	59.2	58.1	70.6	80.3	87.8	59.0	67.0	83.7	93.2	89.5
Angular Loss [43]	54.7	66.3	76	83.9	61.1	71.4	81.4	87.5	92.1	63.2	70.9	85.0	93.5	88.6
Proxy-NCA [23]	49.2	61.9	67.9	72.4	59.5	73.2	82.4	86.4	88.7	64.9	73.7	-	-	90.6
Deep Spectral [18]	53.2	66.1	76.7	85.2	59.2	73.1	82.2	89.0	93.0	64.3	67.6	83.7	93.3	89.4
Classification [52]	59.6	72	81.2	88.4	66.2	81.7	88.9	93.4	96	70.5	73.8	88.1	**95**	89.8
Bias Triplet [50]	46.6	58.6	70.0	-	-	79.2	86.7	91.4	-	-	63.0	79.8	90.7	-
Ours	**65.5**	**77.0**	**85.0**	**91.3**	**69.0**	**85.6**	**91.2**	**94.9**	**97.0**	**72.7**	**75.7**	**88.2**	94.8	**91.1**

Results - Ensembles

	CUB-200-2011					CARS 196					Stanford Online Products			
Loss+Sampling	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@10**	**R@100**	**NMI**
Samp. Matt. [25]	63.6	74.4	83.1	90.0	69.0	79.6	86.5	91.9	95.1	69.1	72.7	86.2	93.8	90.7
Hier. triplet [10]	57.1	68.8	78.7	86.5	-	81.4	88.0	92.7	95.7	-	74.8	88.3	94.8	-
DAMLRRM [54]	55.1	66.5	76.8	85.3	61.7	73.5	82.6	89.1	93.5	64.2	69.7	85.2	93.2	88.2
DE-DSP [7]	53.6	65.5	76.9	61.7	-	72.9	81.6	88.8	-	64.4	68.9	84.0	92.6	89.2
RLL 1 [50]	57.4	69.7	79.2	86.9	63.6	74	83.6	90.1	94.1	65.4	76.1	89.1	95.4	89.7
GPW [51]	65.7	77.0	86.3	91.2	-	84.1	90.4	94.0	96.5	-	78.2	90.5	96.0	-
Teacher-Student														
RKD [31]	61.4	73.0	81.9	89.0	-	82.3	89.8	94.2	96.6	-	75.1	88.3	95.2	-
Loss+Ensembles														
BIER 6 [29]	55.3	67.2	76.9	85.1	-	75.0	83.9	90.3	94.3	-	72.7	86.5	94.0	-
HDC 3 [57]	54.6	66.8	77.6	85.9	-	78.0	85.8	91.1	95.1	-	70.1	84.9	93.2	-
ABE 2 [19]	55.7	67.9	78.3	85.5	-	76.8	84.9	90.2	94.0	-	75.4	88.0	94.7	-
ABE 8 [19]	60.6	71.5	79.8	87.4	-	85.2	90.5	94.0	96.1	-	76.3	88.4	94.8	-
A-BIER 6 [30]	57.5	68.7	78.3	86.2	-	82.0	89.0	93.2	96.1	-	74.2	86.9	94.0	-
D and C 8 [39]	65.9	76.6	84.4	90.6	69.6	84.6	90.7	94.1	96.5	70.3	75.9	88.4	94.9	90.2
RLL 3 [50]	61.3	72.7	82.7	89.4	66.1	82.1	89.3	93.7	96.7	71.8	**79.8**	**91.3**	**96.3**	90.4
Ours 2-ensemble	65.8	76.7	85.2	91.2	68.5	86.2	91.6	95.0	97.1	**91.1**	75.9	88.0	94.5	72.6
Ours 5-ensemble	**66.9**	**77.1**	**85.4**	**91.5**	**70.0**	**88.0**	**92.5**	**95.7**	**97.5**	**74.2**	76.3	88.3	94.6	**91.1**

Results - Robustness Analysis

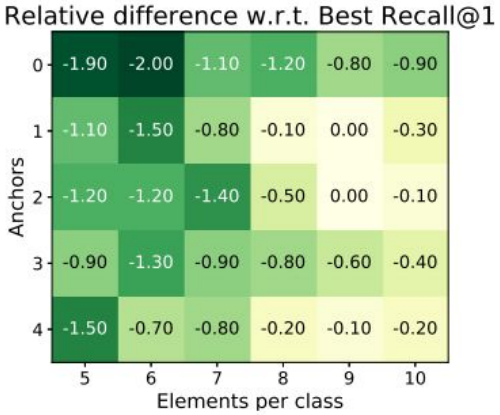


Figure 4: The effect of the number of anchors and the number of samples per class.

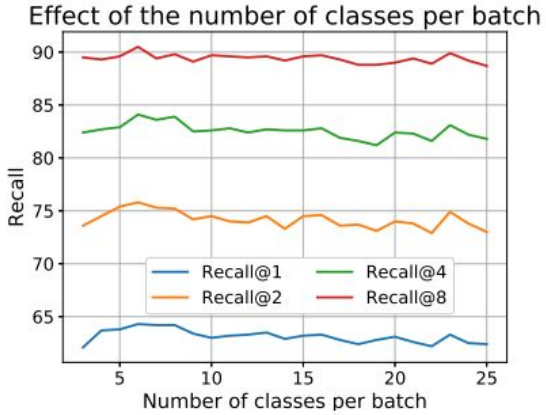


Figure 5: The effect of the number of classes per mini-batch.

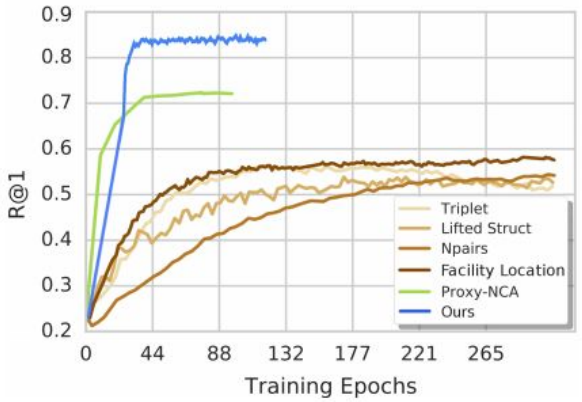
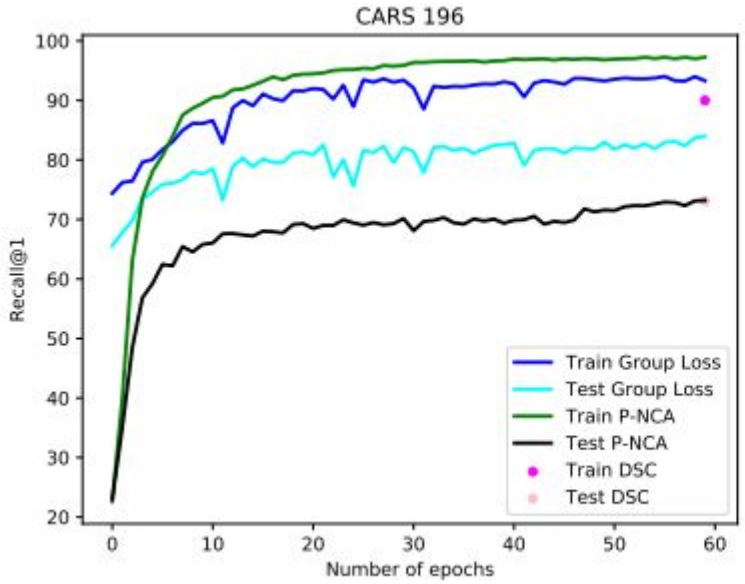


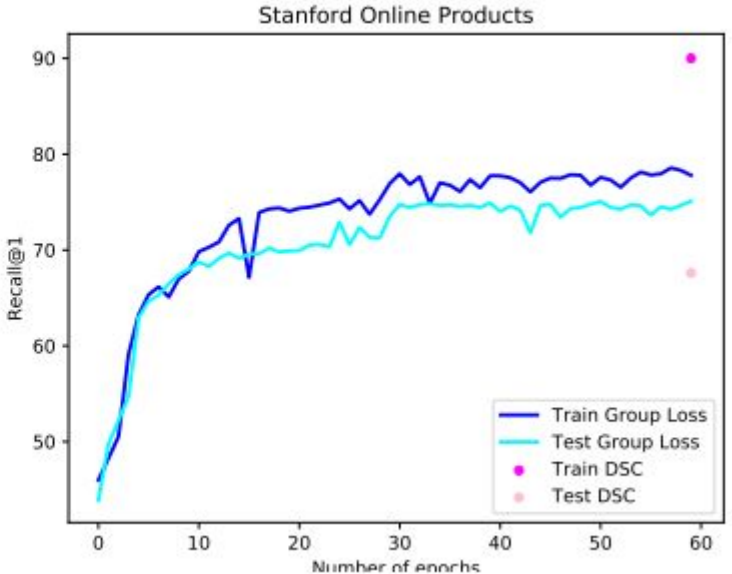
Figure 6: Recall@1 as a function of training epochs on Cars196 dataset. Figure adapted from [18].

One epoch takes 14% less time in CUB, and 8% less time in CARS

Proxy NCA

Less overfitting and implicit regularization?

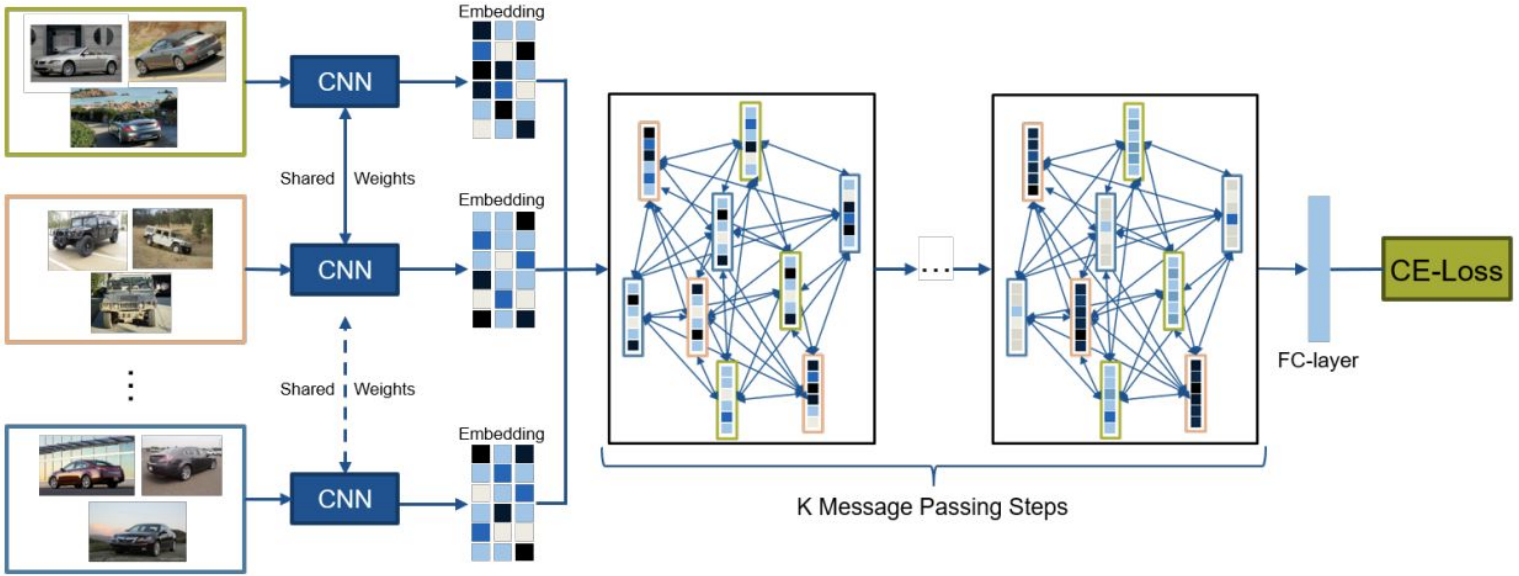




But this is Deep Learning

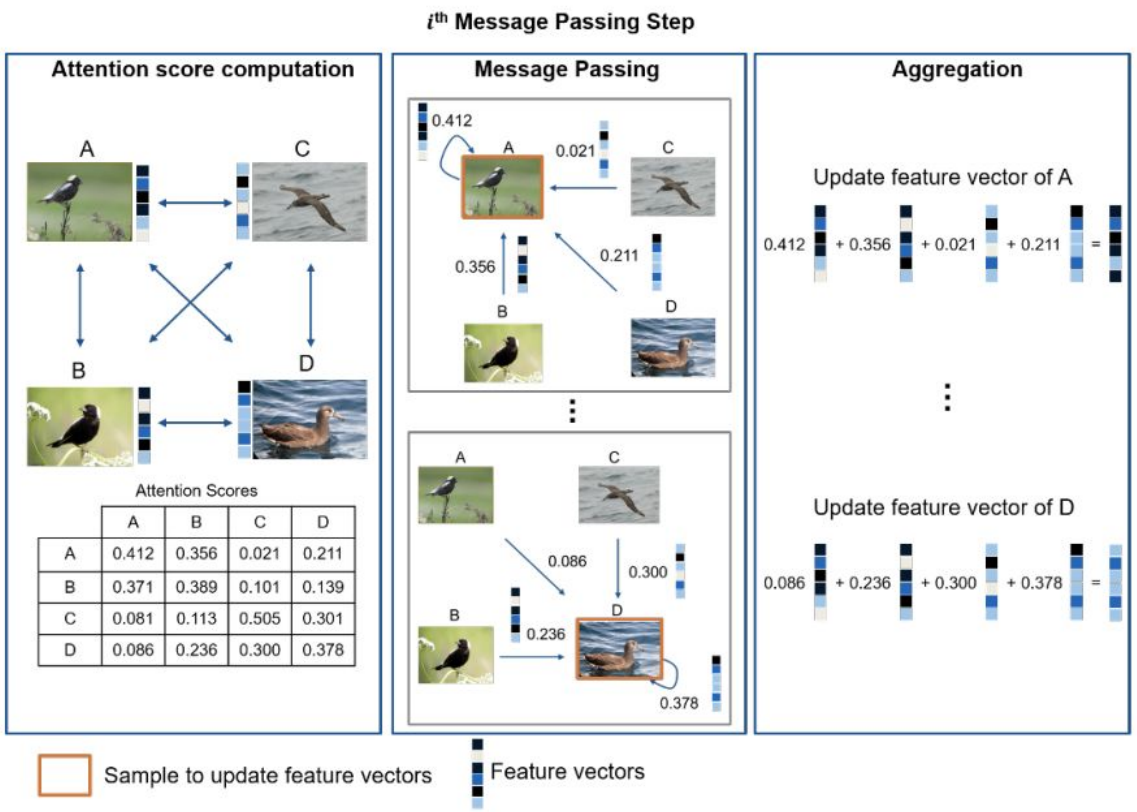
And we want to learn as much as possible!

Message Passing Framework

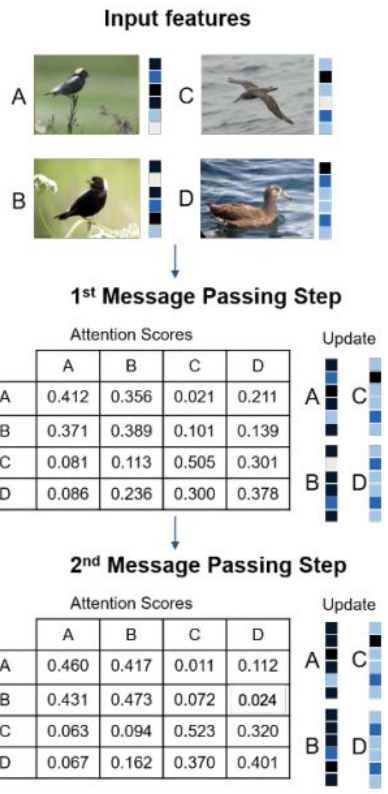


Seidenschwarz et al., Learning intra-batch connections for Deep Metric Learning, ICML 2021

Message Passing Framework



	A	B	C	D
A	0.412	0.356	0.021	0.211
B	0.371	0.389	0.101	0.139
C	0.081	0.113	0.505	0.301
D	0.086	0.236	0.300	0.378



	A	B	C	D
A	0.412	0.356	0.021	0.211
B	0.371	0.389	0.101	0.139
C	0.081	0.113	0.505	0.301
D	0.086	0.236	0.300	0.378

	A	B	C	D
A	0.460	0.417	0.011	0.112
B	0.431	0.473	0.072	0.024
C	0.063	0.094	0.523	0.320
D	0.067	0.162	0.370	0.401

Results

Method	BB	CUB-200-2011					CARS196					Stanford Online Products			
		R@1	R@2	R@4	R@8	NMI	R@1	R@2	R@4	R@8	NMI	R@1	R@10	R@100	NMI
Triplet^{64} (Schroff et al., 2015) *CVPR15*	G	42.5	55	66.4	77.2	55.3	51.5	63.8	73.5	82.4	53.4	66.7	82.4	91.9	89.5
Npairs^{64} (Sohn, 2016) *NeurIPS16*	G	51.9	64.3	74.9	83.2	60.2	68.9	78.9	85.8	90.9	62.7	66.4	82.9	92.1	87.9
Deep Spectral^{512} (Law et al., 2017) *ICML17*	BNI	53.2	66.1	76.7	85.2	59.2	73.1	82.2	89.0	93.0	64.3	67.6	83.7	93.3	89.4
Angular Loss^{512} (Wang et al., 2017) *ICCV17*	G	54.7	66.3	76	83.9	61.1	71.4	81.4	87.5	92.1	63.2	70.9	85.0	93.5	88.6
Proxy-NCA^{64} (Movshovitz-Attias et al., 2017) *ICCV17*	BNI	49.2	61.9	67.9	72.4	59.5	73.2	82.4	86.4	88.7	64.9	73.7	-	-	90.6
Margin Loss^{128} (Manmatha et al., 2017) *ICCV17*	R50	63.6	74.4	83.1	90.0	69.0	79.6	86.5	91.9	95.1	69.1	72.7	86.2	93.8	90.7
Hierarchical triplet^{512} (Ge et al., 2018) *ECCV18*	BNI	57.1	68.8	78.7	86.5	-	81.4	88.0	92.7	95.7	-	74.8	88.3	94.8	-
ABE^{512} (Kim et al., 2018) *ECCV18*	G	60.6	71.5	79.8	87.4	-	85.2	90.5	94.0	96.1	-	76.3	88.4	94.8	-
Normalized Softmax^{512} (Zhai & Wu, 2019) *BMVC19*	R50	61.3	73.9	83.5	90.0	69.7	84.2	90.4	94.4	96.9	74.0	78.2	90.6	96.2	91.0
RLL-H^{512} (Wang et al., 2019b) *CVPR19*	BNI	57.4	69.7	79.2	86.9	63.6	74	83.6	90.1	94.1	65.4	76.1	89.1	95.4	89.7
Multi-similarity^{512} (Wang et al., 2019a) *CVPR19*	BNI	65.7	77.0	86.3	91.2	-	84.1	90.4	94.0	96.5	-	78.2	90.5	96.0	-
Relational Knowledge^{512} (Park et al., 2019a) *CVPR19*	G	61.4	73.0	81.9	89.0	-	82.3	89.8	94.2	96.6	-	75.1	88.3	95.2	-
Divide and Conquer^{1028} (Sanakoyeu et al., 2019) *CVPR19*	R50	65.9	76.6	84.4	90.6	69.6	84.6	90.7	94.1	96.5	70.3	75.9	88.4	94.9	90.2
SoftTriple Loss^{512} (Qian et al., 2019) *ICCV19*	BNI	65.4	76.4	84.5	90.4	69.3	84.5	90.7	94.5	96.9	70.1	78.3	90.3	95.9	92.0
HORDE^{512} (Jacob et al., 2019) *ICCV19*	BNI	66.3	76.7	84.7	90.6	-	83.9	90.3	94.1	96.3	-	80.1	91.3	96.2	-
MIC^{128} (Brattoli et al., 2019) *ICCV19*	R50	66.1	76.8	85.6	-	69.7	82.6	89.1	93.2	-	68.4	77.2	89.4	95.6	90.0
Easy triplet mining^{512} (Xuan et al., 2020b) *WACV20*	R50	64.9	75.3	83.5	-	-	82.7	89.3	93.0	-	-	78.3	90.7	96.3	-
Group Loss^{1024} (Elezi et al., 2020) *ECCV20*	BNI	65.5	77.0	85.0	91.3	69.0	85.6	91.2	94.9	97.0	72.7	75.1	87.5	94.2	90.8
Proxy NCA++^{512} (Teh et al., 2020) *ECCV20*	R50	66.3	77.8	87.7	91.3	71.3	84.9	90.6	94.9	97.2	71.5	79.8	91.4	96.4	-
DiVA^{512} (Milbich et al., 2020) *ECCV20*	R50	69.2	79.3	-	-	71.4	87.6	92.9	-	-	72.2	79.6	-	-	90.6
PADS^{128} (Roth et al., 2020) *CVPR20*	R50	67.3	78.0	85.9	-	69.9	83.5	89.7	93.8	-	68.8	76.5	89.0	95.4	89.9
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	BNI	68.4	79.2	86.8	91.6	-	86.1	91.7	95.0	97.3	-	79.1	90.8	96.2	-
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	R50	69.7	80.0	87.0	92.4	-	87.7	92.9	95.8	97.9	-	80.0	91.7	96.6	-
Proxy Few^{512} (Zhu et al., 2020) *NeurIPS20*	BNI	66.6	77.6	86.4	-	69.8	85.5	91.8	95.3	-	72.4	78.0	90.6	96.2	90.2
Ours^{512}	R50	**70.3**	**80.3**	**87.6**	**92.7**	**74.0**	**88.1**	**93.3**	**96.2**	**98.2**	**74.8**	**81.4**	91.3	95.9	**92.6**

The effect of MPN

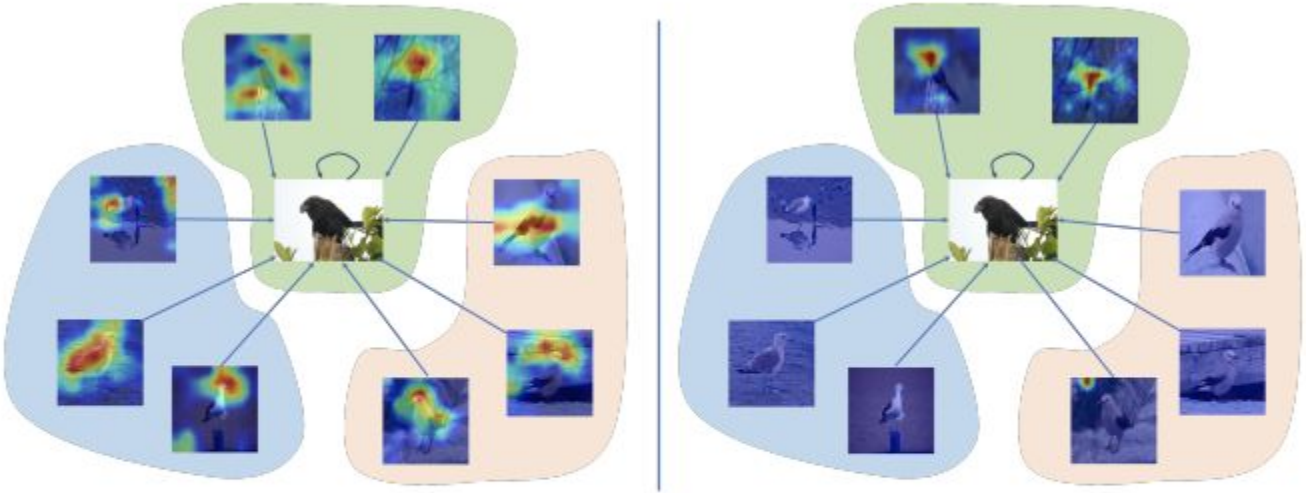


Figure 7. Comparison of the embeddings of a given batch after one epoch of training without and with MPN.

Regularization effect

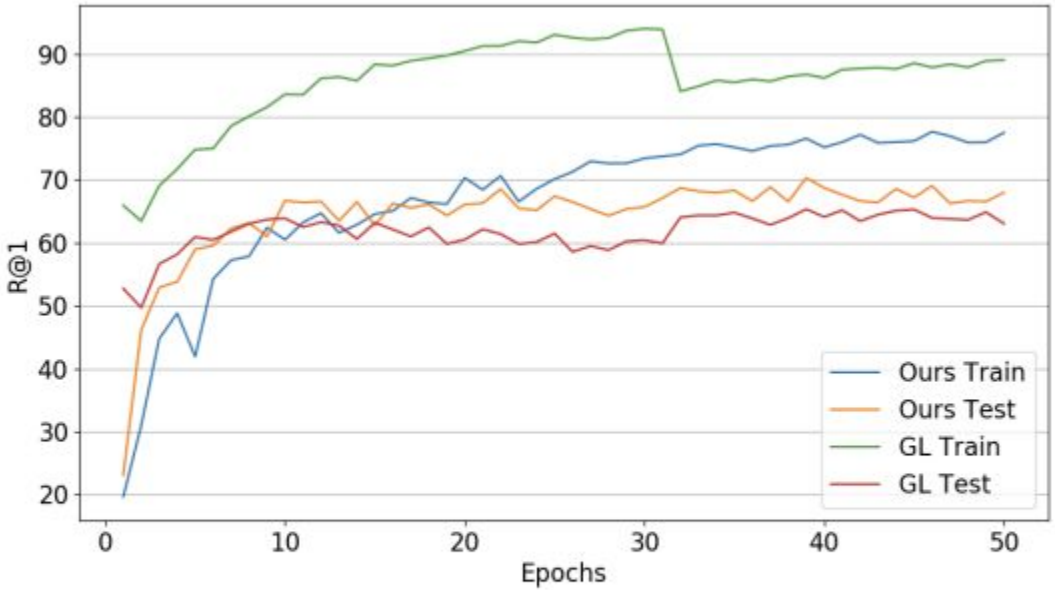


Figure 8. Performance on training and test data of CUB-200-2011 compared to Group Loss (Elezi et al., 2020).

More results

		CUB-200-2011		CARS196	
Training Losses	Test Time Embeddings	**R@1**	**NMI**	**R@1**	**NMI**
Cross-Entropy	Backbone Embeddings	67.5	69.8	84.2	68.7
MPN Loss	Backbone Embeddings	68.1	72.0	87.2	72.1
MPN Loss + Auxiliary Loss	Backbone Embeddings	70.3	74.0	88.1	74.8
MPN Loss + Auxiliary Loss	MPN Embeddings	**70.8**	**74.5**	**88.6**	**76.2**

Table 3. Performance of the network with and without MPN during training and testing time. We achieved all results using embedding dimension 512.

More results

		CUB-200-2011		CARS196	
Training Losses	Test Time Embeddings	**R@1**	**NMI**	**R@1**	**NMI**
Cross-Entropy	Backbone Embeddings	67.5	69.8	84.2	68.7
MPN Loss	Backbone Embeddings	68.1	72.0	87.2	72.1
MPN Loss + Auxiliary Loss	Backbone Embeddings	70.3	74.0	88.1	74.8
MPN Loss + Auxiliary Loss	MPN Embeddings	**70.8**	**74.5**	**88.6**	**76.2**

Table 3. Performance of the network with and without MPN during training and testing time. We achieved all results using embedding dimension 512.

	CUB-200-2011		Cars196		Stanford Online Products		In-Shop Clothes
	R@1	**NMI**	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**
GL	65.5	69.0	85.6	72.7	75.7	91.1	-
Ours	70.3	74.0	88.1	74.8	81.4	92.6	92.8
GL 2	65.8	68.5	86.2	72.6	75.9	91.1	-
Ours 2	72.2	74.3	90.9	74.9	81.8	92.7	92.9
GL 5	66.9	70.0	88.0	74.2	76.3	91.1	-
Ours 5	**73.1**	**74.4**	**91.5**	**75.4**	**82.1**	**92.8**	**93.4**

Table 4. Performance of our ensembles and comparisons with the ensemble models of (Elezi et al., 2020).

Thank you!

- Thanks for attending and sorry for the talk being virtual.
- Many other metric learning methods could not have been covered because of the timing.
- Some of the losses presented here could be massively improved by adding a few simple tricks (Group Loss -> Group Loss ++, Proxy NCA -> Proxy NCA ++), will be covered by Jenny in the next talk.