Supplementary Material for
LoOp: Looking for Optimal Hard Negative Embeddings for Deep Metric Learning

Bhavya Vasudeva[1*] Puneesh Deora[1*] Saumik Bhattacharya[2] Umapada Pal[1] Sukalpa Chanda[3]
[1]Indian Statistical Institute, Kolkata, India [2]Indian Institute of Technology, Kharagpur, India
[3]Østfold University College, Halden, Norway

1. Introduction

This supplementary material includes the results for validation of the spherical-homoscedasticity assumption (Section 2), proof of Proposition 1 (Section 3), the proofs for obtaining the optimal hard negatives (Section 4), the toy examples demonstrating the effectiveness of using optimal hard negatives (Section 5), the t-SNE plots for visualizing the training process (Section 6), some results demonstrating the effect of network architecture (Section 7), and results for train-validate-test split (Section 8).

2. Validation of Spherical-homoscedastic Distributions Assumption

To validate the spherical-homoscedasticity assumption, we project the embeddings (of samples belonging to the same class) to a 3D space using principal component analysis (PCA). We need to show that the eigenvalues of the covariance matrices of the data distributions of different classes are close to one another. Fig. 1 shows the eigenvalues for four randomly picked classes. It is seen that these eigenvalues are very close indeed for all the three datasets, indicating a similar shape of data distributions (spherical-homoscedasticity).

Further, we calculate the mean and standard deviation (×100) of the three eigenvalues obtained by using all the classes in each dataset. They are 7.02±1.84, 4.84±1.05, 3.84±0.84 for the CUB-200-2011 dataset, 3.67±1.17, 2.16±0.60, 1.56±0.41 for the Cars196 dataset, and 12.24±5.25, 6.63±2.98, 3.93±2.04 for the SOP dataset. The values of the standard deviations are small, which again validates the assumption.

3. Proof of Proposition 1

Proposition 1. *If the pairs of points $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{y_1}$, $\mathbf{y_2}$ (from two different classes) belong to spherical-homoscedastic distributions, then the points on the curves $\widehat{\mathbf{x_1x_2}}$ and $\widehat{\mathbf{y_1y_2}}$ have a higher probability of belonging to the same classes as $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{y_1}$, $\mathbf{y_2}$, respectively, than the other classes.*

Proof. The definition of spherical-homoscedastic distributions says that such distributions are separated by decision boundaries that are hyperplanes. In such a case, a region between two points (say $\mathbf{x_1}$, $\mathbf{x_2}$) of the same class will also belong to that class with a higher probability than the other classes. Thus, any points sampled from that region will have the highest probability of belonging to the same class as that of $\mathbf{x_1}, \mathbf{x_2}$. As we assign class belongingness of an unknown point based on the maximum probability or likelihood, the points lying on the curves $\widehat{\mathbf{x_1x_2}}$ and $\widehat{\mathbf{y_1y_2}}$ will belong to the same classes as $\mathbf{x_1}, \mathbf{x_2}$ and $\mathbf{y_1}, \mathbf{y_2}$, respectively. □

4. Proofs for Finding Optimal Hard Negatives

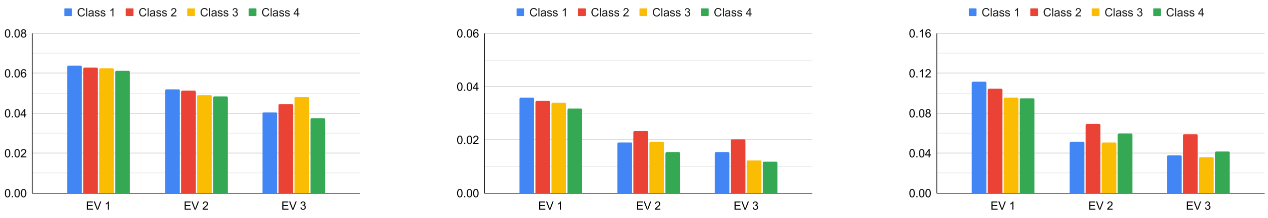
In this section, we present the proofs for finding the optimal points $\mathbf{p_1}$ and $\mathbf{p_2}$, such that the distance between a pair of positives and a pair of negatives is minimized. Section 4.1 presents the case where the optimal points are l_2-normalized and lie on the hypersphere, whereas Section 4.2 presents the case where optimal points are not l_2-normalized.

4.1. Proof for solution of optimal points with l_2-normalization

First, the expression for $\mathbf{p_1}$ is obtained as follows:

$$
\begin{aligned}
\mathbf{p_1} &= \mathbf{Rx_1} = \mathbf{Rn_1}\\
&= (\mathbf{I} + \sin\alpha(\mathbf{n_2 n_1}^T - \mathbf{n_1 n_2}^T)\\
&\quad - (1-\cos\alpha)(\mathbf{n_1 n_1}^T + \mathbf{n_2 n_2}^T))\mathbf{n_1}\\
&= \mathbf{n_1} + (\mathbf{n_2}\cancelto{1}{\mathbf{n_1^T n_1}} - \mathbf{n_1}\cancelto{0}{\mathbf{n_2^T n_1}})\sin\alpha\\
&\quad - (\mathbf{n_1}\cancelto{1}{\mathbf{n_1^T n_1}} + \mathbf{n_2}\cancelto{0}{\mathbf{n_2^T n_1}})(1-\cos\alpha)\\
&= \mathbf{n_1} + \mathbf{n_2}\sin\alpha - \mathbf{n_1}(1-\cos\alpha)\\
&= \mathbf{n_1}\cos\alpha + \mathbf{n_2}\sin\alpha,
\end{aligned}
$$

[*]Equal contribution



(a) CUB-200-2011 dataset.

(b) Cars196 dataset.

(c) SOP dataset.

Figure 1: Representation of eigenvalues of the covariance matrices of four randomly picked classes for (a) CUB-200-2011, (b) Cars196, and (c) SOP datasets. The embeddings have been projected to a 3D space using PCA. The model is trained using the proposed approach with triplet loss. The values from the same position on the diagonal have been plotted together.

where the dot products are reduced to 1 or 0 as $\mathbf{n_1}$ and $\mathbf{n_2}$ are orthonormal vectors.

The KKT conditions to obtain the solution for the problem listed in Section 2.2 of the paper are mentioned again for reference:

$$\frac{\partial L}{\partial \alpha} = \frac{\partial f}{\partial \alpha} + \lambda_1 - \lambda_2 \tag{1}$$

$$= a\cos\alpha\sin\beta - b\sin\alpha\sin\beta + c\cos\alpha\cos\beta$$

$$- d\sin\alpha\cos\beta + \lambda_1 - \lambda_2 = 0, \tag{2}$$

$$\frac{\partial L}{\partial \beta} = \frac{\partial f}{\partial \beta} + \lambda_3 - \lambda_4 \tag{3}$$

$$= a\sin\alpha\cos\beta + b\cos\alpha\cos\beta - c\sin\alpha\sin\beta$$

$$- d\cos\alpha\sin\beta + \lambda_3 - \lambda_4 = 0, \tag{4}$$

$$\lambda_i g_i = 0\,;\, i = 1,2,3,4, \tag{5}$$

$$\lambda_i \leq 0\,;\, i = 1,2,3,4, \tag{6}$$

$$g_i \leq 0\,;\, i = 1,2,3,4. \tag{7}$$

The solutions for the 9 possible cases to be considered are discussed below.

Note: The solutions to the equations involving α, β are denoted by $\hat{\alpha}, \hat{\beta}$.

Case 0: $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0.$

Using (2), (4), we obtain:

$$a\cos\alpha\sin\beta - b\sin\alpha\sin\beta + c\cos\alpha\cos\beta$$

$$-d\sin\alpha\cos\beta = 0, \tag{8}$$

$$a\sin\alpha\cos\beta + b\cos\alpha\cos\beta - c\sin\alpha\sin\beta$$

$$-d\cos\alpha\sin\beta = 0. \tag{9}$$

Simplifying these, we get:

$$\cos\alpha(a\sin\beta + c\cos\beta) = \sin\alpha(b\sin\beta + d\cos\beta), \tag{10}$$

$$\cos\alpha(b\cos\beta - d\sin\beta) = \sin\alpha(c\sin\beta - a\cos\beta). \tag{11}$$

Dividing both sides in (10), (11) by $\cos\alpha\cos\beta$, and then dividing (10) by (11) we get:

$$\frac{a\tan\beta + c}{b - d\tan\beta} = \frac{b\tan\beta + d}{c\tan\beta - a}.$$

This can be simplified to get:

$$\tan^2\beta - \frac{a^2 + b^2 - c^2 - d^2}{ac + bd}\tan\beta - 1 = 0,$$

$$\therefore \hat{\beta} = \tan^{-1}\left(\frac{B \mp \sqrt{B^2 + 4(ac+bd)^2}}{2(ac+bd)}\right).$$

For $\hat{\beta} < 0$, we consider $\hat{\beta} + \pi$ as a possible solution due to the constraint g_3.

Similarly, using (8), (9), we can form a quadratic equation in $\tan\alpha$:

$$\tan^2\alpha - \frac{a^2 - b^2 - c^2 + d^2}{ab + cd}\tan\alpha - 1 = 0,$$

$$\therefore \hat{\alpha} = \tan^{-1}\left(\frac{A \pm \sqrt{A^2 + 4(ab+cd)^2}}{2(ab+cd)}\right).$$

Similar to the argument for $\hat{\beta}$, $\hat{\alpha}+\pi$ is considered as a possible solution when $\hat{\alpha} < 0$. (In the aforementioned equations, $A = a^2 - b^2 + c^2 - d^2$ and $B = a^2 + b^2 - c^2 - d^2$.)

Case 1: $g_1 = 0\ (\hat{\alpha} = 0), \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0.$

$$\frac{\partial L}{\partial \beta} = b\cos\beta - d\sin\beta = 0, \implies \tan\beta = \frac{b}{d},$$

$$\therefore \hat{\beta} = \tan^{-1}\left(\frac{b}{d}\right). \tag{12}$$

Using (1), (2), (12), we get:

$$\lambda_1 = -\left.\frac{\partial f}{\partial \alpha}\right|_{\alpha=0,\beta=\hat{\beta}}$$

Case 2: $g_2 = 0\ (\hat{\alpha} = \alpha_0), \lambda_1 = 0, \lambda_3 = 0, \lambda_4 = 0.$

Using (4), we obtain:

$$\begin{aligned}\frac{\partial L}{\partial \beta} &= a\sin\alpha_0\cos\beta + b\cos\alpha_0\cos\beta - c\sin\alpha_0\sin\beta \\ &- d\cos\alpha_0\sin\beta = 0, \\ &\implies \tan\beta = \left(\frac{a\sin\alpha_0 + b\cos\alpha_0}{c\sin\alpha_0 + d\cos\alpha_0}\right), \\ &\therefore \hat{\beta} = \tan^{-1}\left(\frac{a\sin\alpha_0 + b\cos\alpha_0}{c\sin\alpha_0 + d\cos\alpha_0}\right). \end{aligned} \tag{13}$$

Using (1), (2), (13), we get:

$$\lambda_2 = \left.\frac{\partial f}{\partial \alpha}\right|_{\alpha=\alpha_0,\beta=\hat{\beta}}$$

Case 3: $g_3 = 0$ $(\hat{\beta} = 0), \lambda_1 = 0, \lambda_2 = 0, \lambda_4 = 0$. Using (2), we obtain:

$$\begin{aligned} &c\cos\alpha - d\sin\alpha = 0, \implies \tan\alpha = \left(\frac{c}{d}\right), \\ &\therefore \hat{\alpha} = \tan^{-1}\left(\frac{c}{d}\right). \end{aligned} \tag{14}$$

Using (3), (4), (14), we get:

$$\lambda_3 = -\left.\frac{\partial f}{\partial \beta}\right|_{\alpha=\hat{\alpha},\beta=0}$$

Case 4: $g_4 = 0$ $(\hat{\beta} = \beta_0), \lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$. Using (2), we obtain:

$$\begin{aligned} &a\cos\alpha\sin\beta_0 - b\sin\alpha\sin\beta_0 + c\cos\alpha\cos\beta_0 \\ &- d\sin\alpha\cos\beta_0 = 0, \\ &\implies \tan\alpha = \frac{a\sin\beta_0 + c\cos\beta_0}{b\sin\beta_0 + d\cos\beta_0}, \\ &\therefore \hat{\alpha} = \tan^{-1}\left(\frac{a\sin\beta_0 + c\cos\beta_0}{b\sin\beta_0 + d\cos\beta_0}\right). \end{aligned} \tag{15}$$

Using (3), (4), (15), we obtain:

$$\lambda_4 = \left.\frac{\partial f}{\partial \beta}\right|_{\alpha=\hat{\alpha},\beta=\beta_0}$$

Case 5: $g_1 = 0$ $(\hat{\alpha} = 0)$, $g_3 = 0$ $(\hat{\beta} = 0), \lambda_2 = 0, \lambda_4 = 0$.

Using (1), (2), we have:

$$\begin{aligned}\frac{\partial L}{\partial \alpha} &= \frac{\partial f}{\partial \alpha} + \lambda_1 = 0 \\ \lambda_1 &= -\left.\frac{\partial f}{\partial \alpha}\right|_{\alpha=0,\beta=0}\end{aligned}$$

Similarly using (3), (4), we get:

$$\begin{aligned}\frac{\partial L}{\partial \beta} &= \frac{\partial f}{\partial \beta} + \lambda_3 = 0 \\ \lambda_3 &= -\left.\frac{\partial f}{\partial \beta}\right|_{\alpha=0,\beta=0}\end{aligned}$$

For cases 6-8, we can easily obtain the respective λ_i values in a similar fashion to case 5. These have already been listed in Table 1 in the paper. For the aforementioned cases, we consider $\hat{\alpha}(\hat{\beta}) + \pi$ as a possible solution when $\hat{\alpha}(\hat{\beta}) < 0$ due to the constraint $g_1(g_3)$.

4.2. Proof for solution of optimal points without l_2-normalization

In this case, f is obtained as:

$$\begin{aligned} f(k_1, k_2) &= ||\mathbf{p_1} - \mathbf{p_2}||_2^2 \\ &= ||k_1(\mathbf{x_2} - \mathbf{x_1}) - k_2(\mathbf{y_2} - \mathbf{y_1}) + \mathbf{x_1} - \mathbf{y_1}||_2^2 \\ &= ||k_1\mathbf{u} - k_2\mathbf{v} - \mathbf{w}||_2^2. \end{aligned}$$

The Lagrangian function is given by:

$$L(k_1, k_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4) = f(k_1, k_2) - \sum_{i=1}^{4} \lambda_i g_i.$$

The constraints are mentioned below for reference:

$$\begin{aligned} &g_1 = -k_1 \leq 0\,;\, g_2 = k_1 - 1 \leq 0, \\ &g_3 = -k_2 \leq 0\,;\, g_4 = k_2 - 1 \leq 0. \end{aligned} \tag{16}$$

The partial derivatives for the KKT conditions are given as follows:

$$\frac{\partial L}{\partial k_1} = \frac{\partial f}{\partial k_1} + \lambda_1 - \lambda_2 \tag{17}$$

$$= \mathbf{u} \cdot (k_1\mathbf{u} - k_2\mathbf{v} - \mathbf{w}) + \lambda_1 - \lambda_2 = 0, \tag{18}$$

$$\frac{\partial L}{\partial k_2} = \frac{\partial f}{\partial k_2} + \lambda_3 - \lambda_4 \tag{19}$$

$$= \mathbf{v} \cdot (-k_1\mathbf{u} + k_2\mathbf{v} + \mathbf{w}) + \lambda_3 - \lambda_4 = 0. \tag{20}$$

The rest of the KKT conditions are given by (5), (6), and (7), where g_i's are the ones mentioned in (16). In a simplified form, (18) and (20) can be written as:

$$\frac{\partial L}{\partial k_1} = ak_1 + bk_2 + c + \lambda_1 - \lambda_2 = 0, \tag{21}$$

$$\frac{\partial L}{\partial k_2} = a'k_1 + b'k_2 + c' + \lambda_3 - \lambda_4 = 0, \tag{22}$$

respectively, where $a = \mathbf{u} \cdot \mathbf{u}$, $b = -\mathbf{u} \cdot \mathbf{v}$, $c = -\mathbf{u} \cdot \mathbf{w}$, $a' = -\mathbf{v} \cdot \mathbf{u}$, $b' = \mathbf{v} \cdot \mathbf{v}$, $c' = \mathbf{v} \cdot \mathbf{w}$. The solutions for the 9 possible cases are discussed below.

Note: The solutions to the equations involving k_1, k_2 are denoted by $\hat{k_1}, \hat{k_2}$.

Case 0: $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$.
Using (21), (22) we get:

$$\begin{aligned} ak_1 + bk_2 + c = 0, \\ a'k_1 + b'k_2 + c' = 0. \end{aligned}$$

Solving for k_1, k_2 we get:

$$\hat{k_1} = \frac{b'c - bc'}{a'b - ab'}, \quad \hat{k_2} = \frac{ac' - a'c}{a'b - ab'}.$$

Case 1: $g_1 = 0 \ (\hat{k_1} = 0), \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$.
Using (22), we get:

$$\begin{aligned} b'k_2 + c' = 0, \\ \therefore \hat{k_2} = \frac{-c'}{b'}. \end{aligned} \tag{23}$$

Using (21), (17), (23), we get:

$$\lambda_1 = -\left.\frac{\partial f}{\partial k_1}\right|_{k_1=0, k_2=\hat{k_2}}$$

Case 2: $g_2 = 0 \ (\hat{k_1} = 1), \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$.
Using (22), we get:

$$\begin{aligned} a' + b'k_2 + c' = 0, \\ \therefore \hat{k_2} = -\frac{a' + c'}{b'}. \end{aligned} \tag{24}$$

Using (21), (17), (24), we obtain:

$$\lambda_2 = \left.\frac{\partial f}{\partial k_1}\right|_{k_1=1, k_2=\hat{k_2}}$$

Similar to the aforementioned 2 cases we can obtain the following results for case 3, and case 4.

Case 3: $g_3 = 0 \ (\hat{k_2} = 0), \lambda_1 = 0, \lambda_2 = 0, \lambda_4 = 0$.
Using (21), (22), (19), we get:

$$\hat{k_1} = -\frac{c}{a} \implies \lambda_3 = -\left.\frac{\partial f}{\partial k_2}\right|_{k_1=\hat{k_1}, k_2=0}$$

Case 4: $g_4 = 0 \ (\hat{k_2} = 1), \lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$.
Using (21), (22), (19) we get:

$$\hat{k_1} = -\frac{b + c}{a} \implies \lambda_3 = \left.\frac{\partial f}{\partial k_2}\right|_{k_1=\hat{k_1}, k_2=1}$$

Case 5: $g_1 = 0 \ (\hat{k_1} = 0), g_3 = 0 \ (\hat{k_2} = 0), \lambda_2 = 0, \lambda_4 = 0$.
Using (21), (17), and (22), (19), we get:

$$\begin{aligned} \lambda_1 &= -\left.\frac{\partial f}{\partial k_1}\right|_{k_1=0, k_2=0} \\ \lambda_3 &= -\left.\frac{\partial f}{\partial k_2}\right|_{k_1=0, k_2=0} \end{aligned}$$

Case 6: $g_1 = 0 \ (\hat{k_1} = 0), g_4 = 0 \ (\hat{k_2} = 1), \lambda_2 = 0, \lambda_3 = 0$.
Using (21), (17), and (22), (19), we get:

$$\begin{aligned} \lambda_1 &= -\left.\frac{\partial f}{\partial k_1}\right|_{k_1=0, k_2=1} \\ \lambda_4 &= \left.\frac{\partial f}{\partial k_2}\right|_{k_1=0, k_2=1} \end{aligned}$$

Case 7: $g_2 = 0 \ (\hat{k_1} = 1), g_3 = 0 \ (\hat{k_2} = 0), \lambda_1 = 0, \lambda_4 = 0$.
Using (21), (17), and (22), (19), we get:

$$\begin{aligned} \lambda_2 &= \left.\frac{\partial f}{\partial k_1}\right|_{k_1=1, k_2=0} \\ \lambda_3 &= -\left.\frac{\partial f}{\partial k_2}\right|_{k_1=1, k_2=0} \end{aligned}$$

Case 8: $g_2 = 0 \ (\hat{k_1} = 1), g_4 = 0 \ (\hat{k_2} = 1), \lambda_1 = 0, \lambda_3 = 0$.
Using (21), (17), and (22), (19), we get:

$$\begin{aligned} \lambda_2 &= \left.\frac{\partial f}{\partial k_1}\right|_{k_1=1, k_2=1} \\ \lambda_4 &= \left.\frac{\partial f}{\partial k_2}\right|_{k_1=1, k_2=1} \end{aligned}$$

5. Analysis of Triplet Loss with Optimal Hard Negatives

Given two pairs of points in the embedding space: $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{y_1}$, $\mathbf{y_2}$, belonging to two different classes. The triplet loss is formulated as follows:

$$\mathcal{L}_{Tri} = \left[d(\mathbf{x_1}, \mathbf{x_2}) - d(\mathbf{x_1}, \mathbf{y_1}) + m \right]_+ . \tag{25}$$

The modified triplet loss obtained by incorporating the proposed approach is given by:

$$\mathcal{L}'_{Tri} = \left[d(\mathbf{x_1}, \mathbf{x_2}) - d(\mathbf{p_1}, \mathbf{p_2}) + m \right]_+ , \tag{26}$$

where $\mathbf{p_1}$ and $\mathbf{p_2}$ lie on arcs $\widehat{\mathbf{x_1}\mathbf{x_2}}$ and $\widehat{\mathbf{y_1}\mathbf{y_2}}$, respectively. The gradients with respect to the four samples are given by:

$$\frac{\partial \mathcal{L}'_{Tri}}{\partial \mathbf{x_1}} = 2\mathbb{1}_{\mathcal{L}'_{Tri} > 0} \left[(\mathbf{x_1} - \mathbf{x_2}) - (\mathbf{p_1} - \mathbf{p_2}) \cdot \left(\frac{\partial \mathbf{p_1}}{\partial \mathbf{x_1}} \right) \right], \tag{27}$$

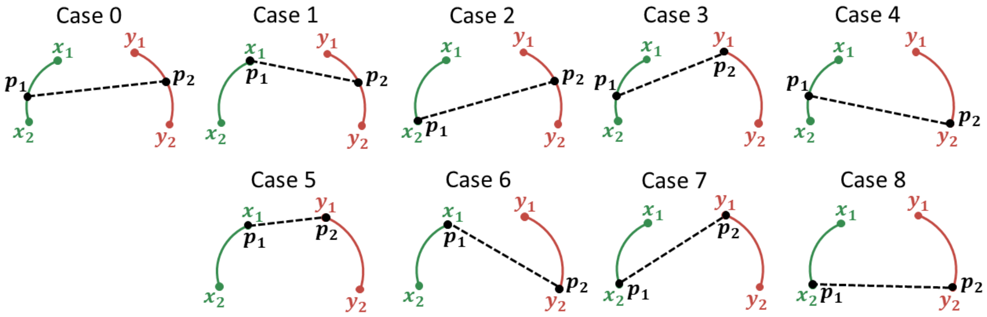


Figure 2: Illustrations of the 9 cases in which the KKT conditions for finding the minimum distance between arcs $\widehat{\mathbf{x_1x_2}}$ and $\widehat{\mathbf{y_1y_2}}$ can be satisfied.

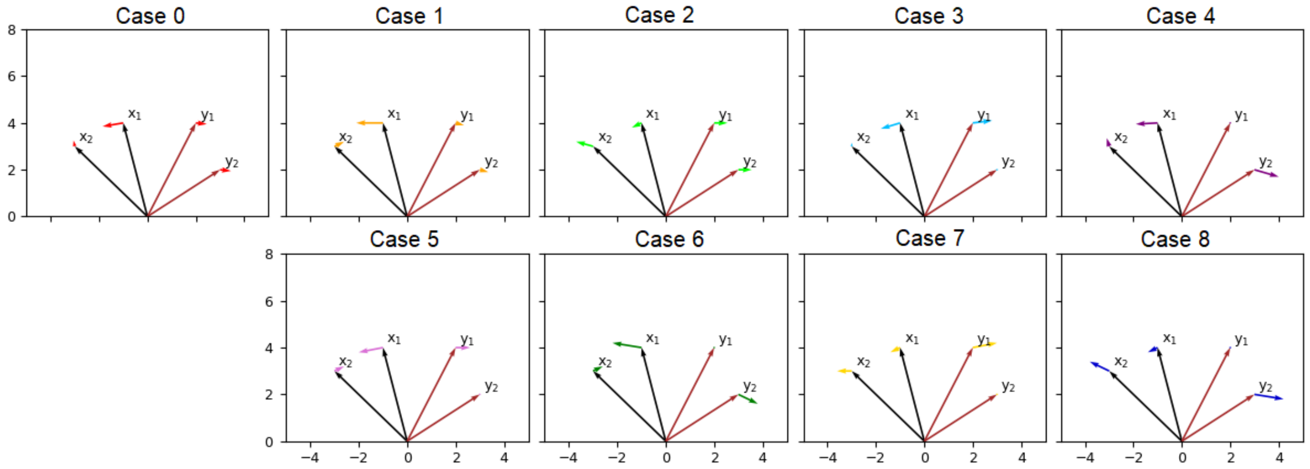


Figure 3: An example showing four samples and their updated positions obtained by using the gradients for different choices of $\mathbf{p_1}$ and $\mathbf{p_2}$. The samples have been selected such that $\mathbf{x_1}$ and $\mathbf{y_1}$ are the hardest negatives.

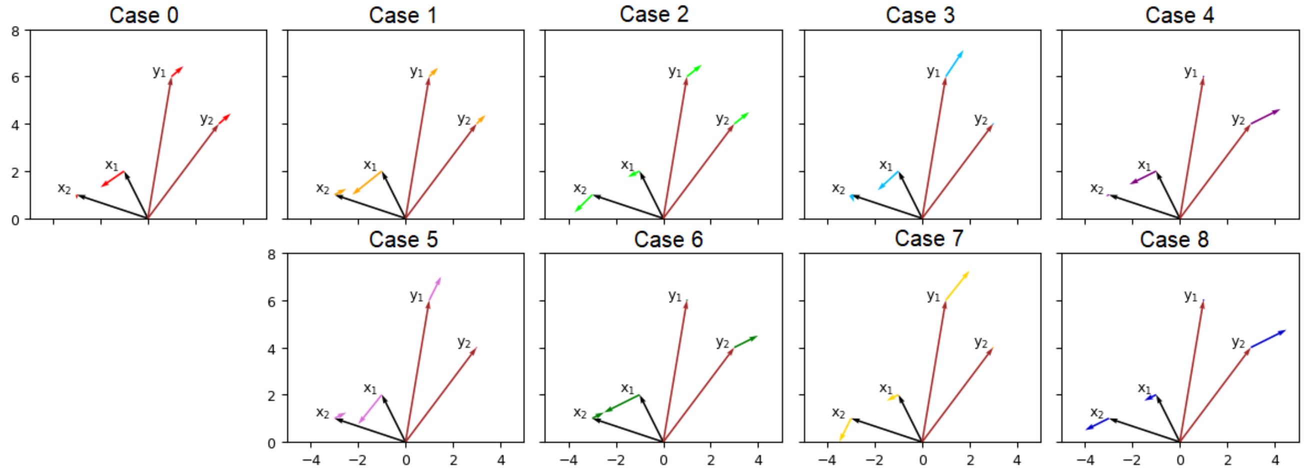


Figure 4: An example showing four samples and their updated positions obtained by using the gradients for different choices of $\mathbf{p_1}$ and $\mathbf{p_2}$. The samples have been selected such that $\mathbf{x_1}$ and $\frac{\mathbf{y_1}+\mathbf{y_2}}{2}$ are the hardest negatives.

$$\frac{\partial \mathcal{L}'_{Tri}}{\partial \mathbf{x_2}} = 2\mathbb{1}_{\mathcal{L}'_{Tri}>0}\left[-(\mathbf{x_1}-\mathbf{x_2}) - (\mathbf{p_1}-\mathbf{p_2})\cdot\left(\frac{\partial \mathbf{p_1}}{\partial \mathbf{x_2}}\right)\right], \tag{28}$$

$$\frac{\partial \mathcal{L}'_{Tri}}{\partial \mathbf{y_1}} = 2\mathbb{1}_{\mathcal{L}'_{Tri}>0}\left[(\mathbf{p_1}-\mathbf{p_2})\cdot\left(\frac{\partial \mathbf{p_2}}{\partial \mathbf{y_1}}\right)\right], \tag{29}$$

$$\frac{\partial \mathcal{L}'_{Tri}}{\partial \mathbf{y_2}} = 2\mathbb{1}_{\mathcal{L}'_{Tri}>0}\left[(\mathbf{p_1}-\mathbf{p_2})\cdot\left(\frac{\partial \mathbf{p_2}}{\partial \mathbf{y_2}}\right)\right]. \qquad (30)$$

Fig. 2 depicts the 9 possible ways in which $\mathbf{p_1}$ and $\mathbf{p_2}$ can be selected. Figs. 3 and 4 show two examples where the four samples and their updated positions obtained by using the gradients for different choices of $\mathbf{p_1}$ and $\mathbf{p_2}$ are shown. In these Figs., when $\mathbf{p_1}$ and $\mathbf{p_2}$ are not one of the end points, they are the midpoints of the arcs on which they lie.

The objective of metric learning is to enable samples from the same class move towards each other and away from the samples of different classes. We use this criteria to analyze the updated samples in each case of Figs. 3 and 4.

In Fig. 3, $\mathbf{x_1}$ and $\mathbf{x_2}$ are not moving towards each other in cases 2, 7, 8, whereas in cases 4, 6, 8, $\mathbf{y_1}$ and $\mathbf{y_2}$ are moving away from each other. Further, the distance between $\mathbf{y_1}$ and $\mathbf{y_2}$ remains the same in cases 0-2. Among cases 3 and 5, the distance between updated samples $\mathbf{x_1}$ and $\mathbf{x_2}$ is closer in case 5, and hence the optimal set of negatives $(\mathbf{p_1}, \mathbf{p_2})$ is $(\mathbf{x_1}, \mathbf{y_1})$. This can also be seen visually, as $\mathbf{x_1}$ and $\mathbf{y_1}$ are the closest points between the two curves.

In Fig. 4, $\mathbf{x_1}$ and $\mathbf{x_2}$ are not moving towards each other in cases 2, 7, 8, whereas in cases 3-8, $\mathbf{y_1}$ and $\mathbf{y_2}$ are moving away from each other. Among cases 0 and 1, the distance between updated samples $\mathbf{x_1}$ and $\mathbf{x_2}$ is closer in case 1, and hence the optimal set of negatives $(\mathbf{p_1}, \mathbf{p_2})$ is $(\mathbf{x_1}, \frac{\mathbf{y_1}+\mathbf{y_2}}{2})$. This can also be seen visually, as $\mathbf{x_1}$ and $\frac{\mathbf{y_1}+\mathbf{y_2}}{2}$ are the closest points between the two curves.

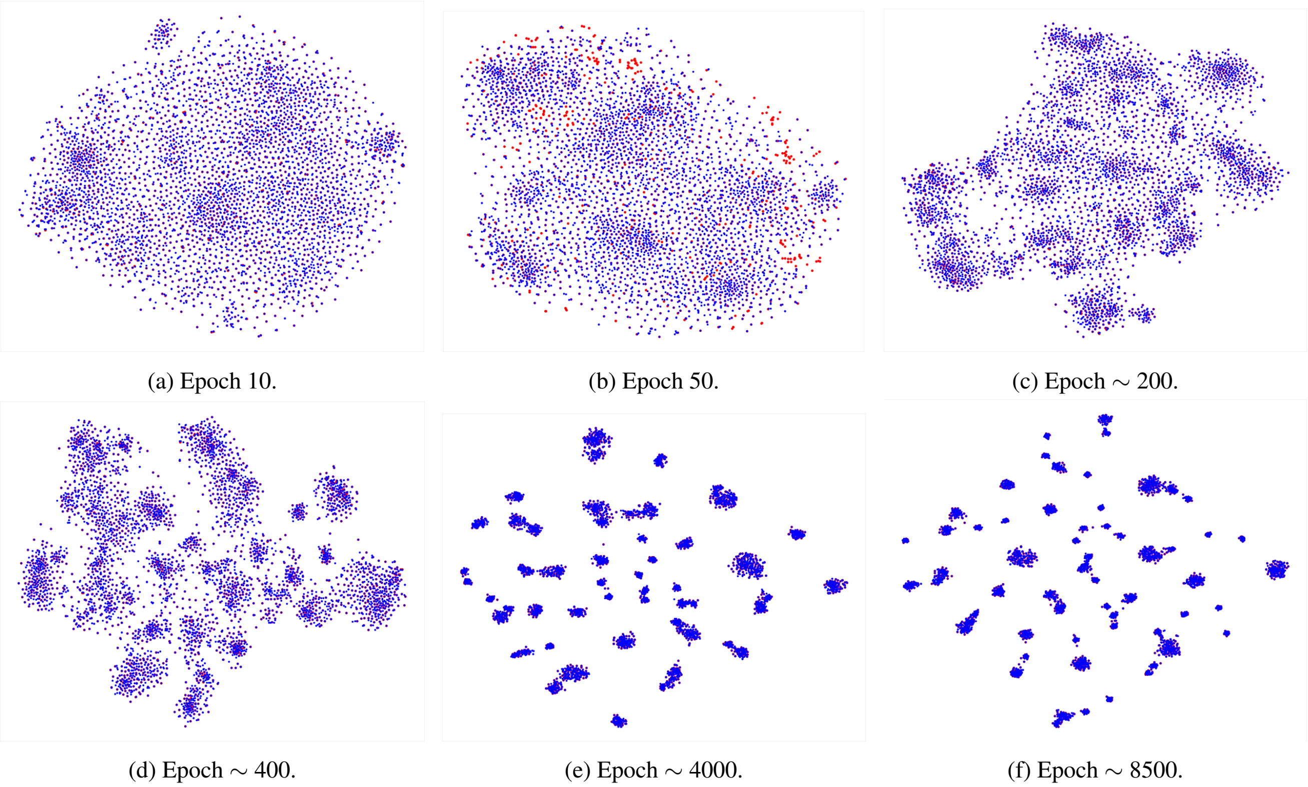
6. t-SNE Visualization of the Embedding Space

We visualize the embedding space using the Barnes-Hut t-Distributed stochastic neighbor embedding (t-SNE) technique [15]. Fig. 5 shows the t-SNE plots when the combination of LoOp and triplet loss is used for training the Cars196 dataset. It can be seen that with increasing epochs, the clusters become more distinct. Further, as training progresses, the synthetic samples (red) lie within the area occupied by the original samples (blue).

It is important to note that although the epoch numbers in Fig. 5 seem large, the iterations per epoch, given by $\frac{\text{Number of classes}\times\text{Samples per class}}{\text{Batch size}}$, are small in number.

7. Effect of Network Architecture

In order to observe the effect of the network architecture on the proposed approach, we deploy two architectures,



(a) Epoch 10.

(b) Epoch 50.

(c) Epoch $\sim$ 200.

(d) Epoch $\sim$ 400.

(e) Epoch $\sim$ 4000.

(f) Epoch $\sim$ 8500.

Figure 5: t-SNE visualization of LoOp with triplet loss using CARS196 dataset showing the embeddings of the train data for different epochs. Blue samples are the original training samples, while Red samples are the synthetic samples generated by LoOp. The marker size of synthetic samples is bigger to improve visibility. As we move from (a) to (f), the red samples become less and less visible as they lie in the same region as the original samples.

Method	NMI	R@1	R@2	R@4
Triplet Semi-hard [11]	55.4	42.6	55.0	66.4
StructClustering [13]	59.2	48.2	61.4	71.8
Proxy NCA [8]	59.5	49.2	61.9	67.9
Binomial Deviance [14]	-	50.3	61.9	72.6
N-pair [12]	60.4	51.0	63.3	74.3
DVML [7]	61.4	52.7	65.1	75.5
Histogram [14]	-	52.8	64.4	74.7
ECAML [1]	60.1	53.4	64.7	75.1
Angular [16]	61.0	53.6	65.0	75.3
HDC [17]	-	53.6	65.7	77.0
EE [6]	59.9	55.0	67.3	77.6
HTL [2]	-	57.1	68.8	78.7
BIER [10]	-	57.5	68.7	78.3
HTG [18]	-	59.5	71.8	81.3
ABE [5]	-	60.6	71.5	79.8
LoOp-IBN (Ours)	**66.0**	60.4	72.1	81.4
LoOp-R50 (Ours)	64.4	**61.1**	**72.5**	**81.7**

Table 1: Comparison of clustering and retrieval performance with SOTA methods for CUB-200-2011 dataset. **Bold** numbers indicate the best values. - indicates not reported. IBN: Inception-BN, R50: ResNet-50.

namely Inception-BN [4] and ResNet-50 [3] as the feature extractors, pre-trained on the ImageNet ILSVRC dataset. In both the cases, the embeddings are l_2-normalized and the batch normalization layers are frozen. The learning rate is set as 10^{-5}, and a weight decay multiplier of 4×10^{-4} is used. The rest of the parameters are kept the same as mentioned in the paper. We report the NMI and Recall@K values to measure the performance.

Table 1 presents the results of our approach and comparisons with state-of-the-art (SOTA) methods, like deep variational metric learning (DVML) [7], energy confused adversarial metric learning (ECAML) [1], hierarchical triplet loss (HTL) [2], boosting independent embeddings robustly (BIER) [10], hard triplet generation (HTG) [18], attention-based ensemble (ALE) [5], as well as metric learning-based loss functions, like triplet with semi-hard mining [11], proxy NCA [8], N-pair [12], histogram [14], angular [16]. To show that improvements carry over to R-50, we run EE [6] with R-50 and compare the increase in (NMI, R@1, R@2, R@4) for LoOp vs. EE with triplet loss: (4.2, 6.0, 6.1 6.3) for GoogLeNet, (4.5, 6.1, 5.2, 4.1) for R-50. It can be seen that LoOp outperforms all the other methods for both clustering and retrieval tasks.

8. Results for Train-Validate-Test Split

Table 2 shows the comparison of results for train-validate-test [9] and train-test splits. These results are obtained using the same settings in Section 7, using ResNet-50 architecture and CUB-200-2011 dataset.

Split	NMI	R@1	R@2	R@4
Train-Validate-Test	59.8	56.4	68.6	78.9
Train-Test	64.4	61.1	72.5	81.7

Table 2: Comparison of results for train-validate-test and train-test splits using CUB-200-2011 dataset.

References

[1] B. Chen and W. Deng. Energy confused adversarial metric learning for zero-shot image retrieval and clustering. In *AAAI*, 2019.

[2] W. Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

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

[6] B. Ko and G. Gu. Embedding expansion: Augmentation in embedding space for deep metric learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7255–7264, 2020.

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

[8] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] K. Musgrave, S. J. Belongie, and S. Lim. A metric learning reality check. In *European Conference on Computer Vision (ECCV)*, 2020.

[10] M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Deep metric learning with BIER: Boosting independent embeddings robustly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):276–290, 2020.

[11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[12] K. Sohn. Improved deep metric learning with multi-class N-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[13] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Learnable structured clustering framework for deep metric learning. *CoRR*, abs/1612.01213, 2016.

[14] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 4170–4178. Curran Associates, Inc., 2016.

[15] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[16] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[17] Y. Yuan, K. Yang, and C. Zhang. Hard-aware deeply cascaded embedding. In *IEEE International Conference on Computer Vision (ICCV)*, pages 814–823, 11 2017.

[18] Y. Zhao, Z. Jin, G. Qi, H. Lu, and X. Hua. An adversarial approach to hard triplet generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.