# Unsupervised Deep Metric Learning with Transformed Attention Consistency and Contrastive Clustering Loss Supplementary Material

Yang Li[1][0000−0002−8372−1481], Shichao Kan[2][0000−0003−0097−6196], and Zhihai He[1][0000−0002−2647−8286]

[1] University of Missouri, Columbia, MO, USA
yltb5@mail.missouri.edu, hezhi@missouri.edu
[2] Beijing Jiaotong University, Beijing, China
16112062@bjtu.edu.cn

## 1 Detail of Our Baseline System

In this work, we adapt the recent state-of-the-art multi-similarity (MS) loss [2] from supervised metric learning to unsupervised metric learning using $k$-means clustering to assign pseudo labels. To the best of our knowledge, multi-similarity (MS) loss [2], published in CVPR 2019, is the first pair-based method which considers three major similarities (self-similarity, negative relative similarity, and positive relative similarity) to explore informative pairs and achieve the current state of the art performance in supervised deep metric learning. This method first defines the pairwise similarity $S_{ij}$ between two image samples in the current batch during the network training process. Assume $x_i$ is an anchor sample, the set $P_i$ of positive pairs $\{x_i, x_p\}$ are selected from the current batch according to

$$S_{i,p} > \max_{y_j \neq y_i} S_{ij} + \epsilon. \tag{1}$$

and the set $N_i$ of negative pairs $\{x_i, x_n\}$ are selected by,

$$S_{i,n} > \min_{y_j = y_i} S_{ij} - \epsilon, \tag{2}$$

The MS loss can be then formulated as,

$$L_{MS} = \frac{1}{L} \sum_{i=1}^{L} \left\{ \frac{1}{\alpha} \log[1 + \sum_{j \in P_i} e^{-\alpha(S_{ij} - \lambda)}] + \frac{1}{\beta} \log[1 + \sum_{j \in N_i} e^{\beta(S_{ij} - \lambda)}] \right\}. \tag{3}$$

We follow the same experimental setting of MS loss [2] and use the same parameters $\epsilon$, $\alpha$, $\beta$, and $\lambda$.

The original MS algorithm was developed for supervised deep metric learning. In this work, we adapt it to unsupervised deep metric learning as our baseline

**Table 1.** The performance comparison between MS [2] loss with $k$-means clustering and our baseline system on CUB dataset.

| | **CUB** | | | |
|---|---|---|---|---|
| | R@1 | R@2 | R@4 | R@8 |
| MS with $k$-means | 52.6 | 64.9 | 76.4 | 85.4 |
| + Memory Bank (Our Baseline) | **53.9** | **66.2** | **76.9** | **85.8** |

system to implement our proposed new approaches. We use the $k$-means clustering to assign pseudo labels. To further improve its performance, we extend this similarity analysis in MS loss from the current batch to the whole training set using the approach of memory bank [3]. The features of all training samples generated by the network are stored in the memory bank by the enqueue-dequeue method. When the memory bank is full, the features and corresponding labels of the oldest mini-batch are removed by the dequeue method. In this way, MS loss can explore informative pairs based on the whole training set to achieve improved performance. Using this approach, we can compute the similarity scores between all samples in the mini-batch and all samples in the training set. Our experimental results in Table 1 demonstrate that the memory bank improves the performance of MS loss for unsupervised metric learning.

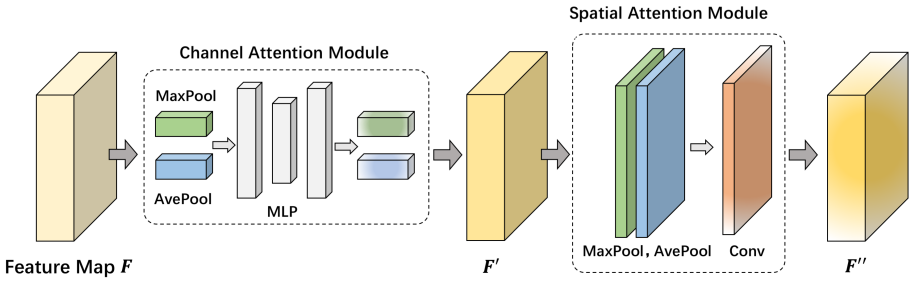



**Fig. 1.** Overview of the attention module. $F$ is the input feature map, $F^{'}$ is the channel attention refined feature map, and $F^{''}$ is the spatial attention refined feature map.

## 2 Attention Module

In our TAC-CCL, we adapt the convolutional block attention module (CBAM) [4] as our attention module, whose structure is shown in Fig. 1. It contains two modules, channel attention and spatial attention, focusing on *what* and *where*, respectively. Given an input feature $F$, the channel attention forwards the

average-pooled and max-pooled features to a multi-layer perceptron (MLP) and applies the element-wise summation to these two features. The spatial attention module follows the channel attention module to explore the most informative part of the feature map. The max-pooling and average-pooling layers are then applied to the refined feature $F^{'}$ by the channel attention module. These two features are then concatenated and processed by a convolution layer. $F^{''}$ is the output feature map of the attention module.

## 3 Ablation Studies on the Transformed Attention Consistency

We conduct ablation studies on the attention module and cross-images transformed attention consistency in our proposed transformed attention consistency (TAC) module. From the results shown in Table 2, when we couple our baseline system with the attention module, the Recall@1 rate is 54.8% on the CUB dataset. The cross-image transformed attention consistency loss improves the Recall@1 rate from 54.8% to 56.5%. We can see that the cross-image transformed attention consistency has much more significant contributions than the attention module only.

**Table 2.** Recall@$K$ (%) performance analysis of transform attention consistency on CUB dataset.

| Methods | **CUB** | | | |
|---|---|---|---|---|
| | R@1 | R@2 | R@4 | R@8 |
| Baseline | 53.9 | 66.2 | 76.9 | 85.8 |
| + Attention Module | 54.8 | 66.6 | 77.6 | 85.8 |
| + Transformed Attention Consistency (Cross-Images) | **56.5** | **68.4** | **78.4** | **86.3** |

## 4 Pseudo Code of Our TAC-CCL Algorithm

The pseudo code is detailed in Algorithm 1.

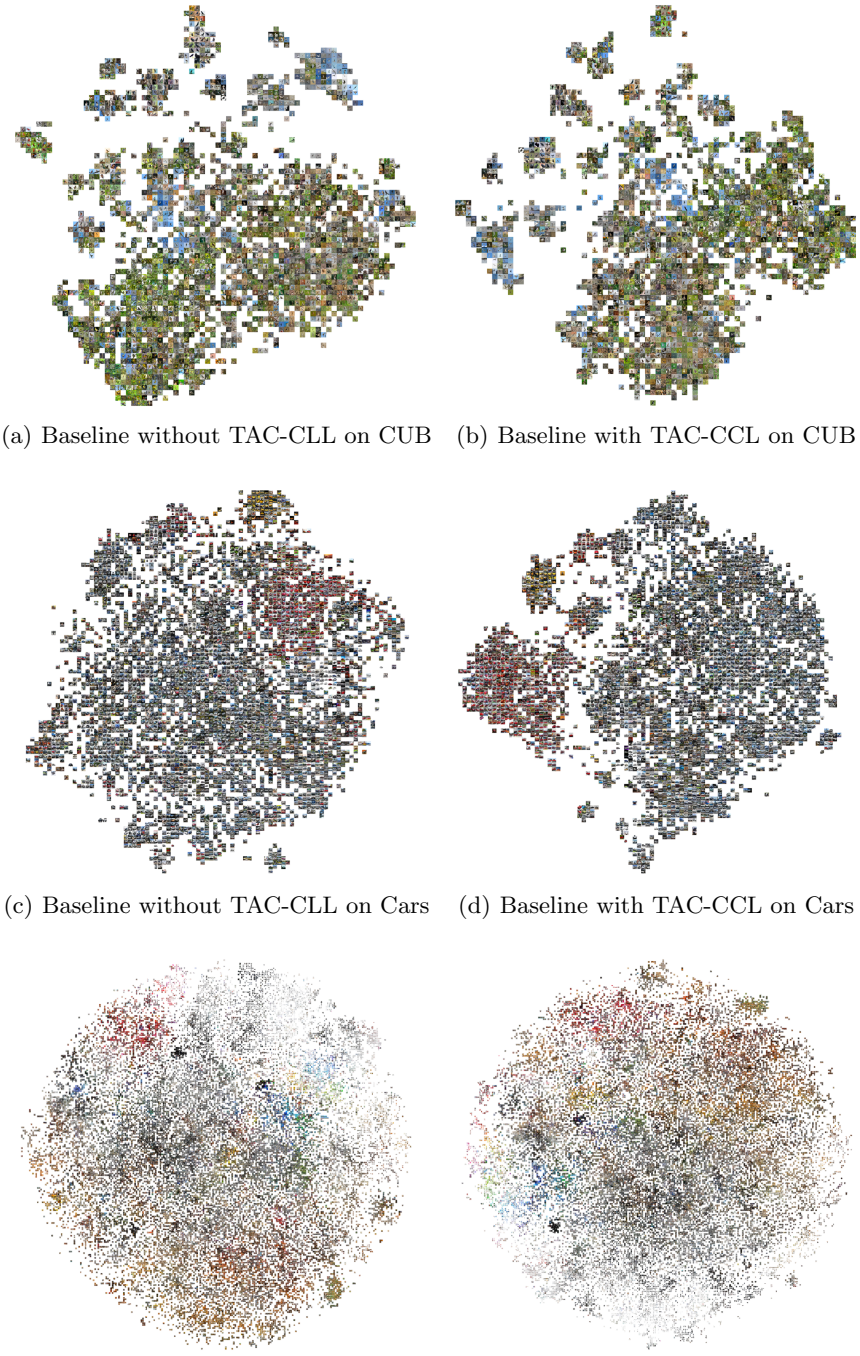
**Algorithm 1** Summary of Procedures: Training Phase

**Initialization:** We use the feature vectors of the training dataset provided by the network $G$ with pre-trained GoogLeNet backbone and $k$-means clustering to assign pseudo labels to the training dataset. The memory bank is denoted as $B$. The warm up threshold is denoted as $I_w$ and the cluster center update step is denoted as $S_u$.
**Input:** Random sampling 5 images per class for each mini-batch $X$. The transformation function is denoted as $T$.
**Output:** The feature embedding
Calculate $\{C_k\}$, $1 \leq k \leq K$ ▷ calculate the $k$-means clustering
**for** $i$ *iterations* **do**
    $X^{'} = T(X)$
    $F, M \leftarrow G(X)$
    $F^{'}, M^{'} \leftarrow G(X^{'})$
    Update $B$ ▷ update memory bank
    $L_{TAC} = \sum_{(u,v)} |M(u,v) - M'(T_u(u,v), T_v(u,v))|^2$
    **if** $iterations > I_w$ **then**
        $L_{CC} = \mathcal{E}_F \left\{ \frac{||F - C_+(F)||_2}{||F - C_-(F)||_2} \right\}$
        $Loss = L_{MS} + \alpha * L_{TAC} + \beta * L_{CC}$ ▷ update network G
    **else**
        $Loss = L_{MS} + \alpha * L_{TAC}$ ▷ update network G
    **end if**
    **if** $iterations \% S_u == 0$ **then**
        update $\{C_k\}$, $1 \leq k \leq K$
        Reset $B$ ▷ reset memory bank
    **end if**
**end for**

## 5 The t-SNE Visualization and Additional Retrieval Examples

Our unsupervised deep metric learning algorithm aims to aggregate the samples of the same classes into compact clusters in the high-dimensional feature space while separating samples from different classes from each other. To demonstrate this property, we use Barnes-Hut t-SNE [1] to visualize the images of the CUB, Cars, and SOP datasets in the feature space with features extracted by the baseline system with and without the TAC-CCL approach, as shown in Fig. 2. We can see that using the TAC-CCL, the obtained feature clusters are more compact within each class and better separated from each other between classes.

We also provide additional retrieval results of more example queries on CUB, Cars, and SOP datasets in Fig. 3, 4, and 5. We can see that the top a few number of retrieval results are very accurate. It should be noted that some classes in these datasets have very few samples.



(a) Baseline without TAC-CLL on CUB (b) Baseline with TAC-CCL on CUB

(c) Baseline without TAC-CLL on Cars (d) Baseline with TAC-CCL on Cars

(e) Baseline without TAC-CLL on SOP (f) Baseline with TAC-CCL on SOP

**Fig. 2.** The Barnes-Hut t-SNE visualizations of the CUB, Cars, and SOP test datasets with and without the TAC-CCL method.

**Fig. 3.** Retrieval results of some example queries on CUB dataset. The query images and the negative retrieved images are highlighted with blue and red.

**Fig. 4.** Retrieval results of some example queries on Cars dataset. The query images and the negative retrieved images are highlighted with blue and red.

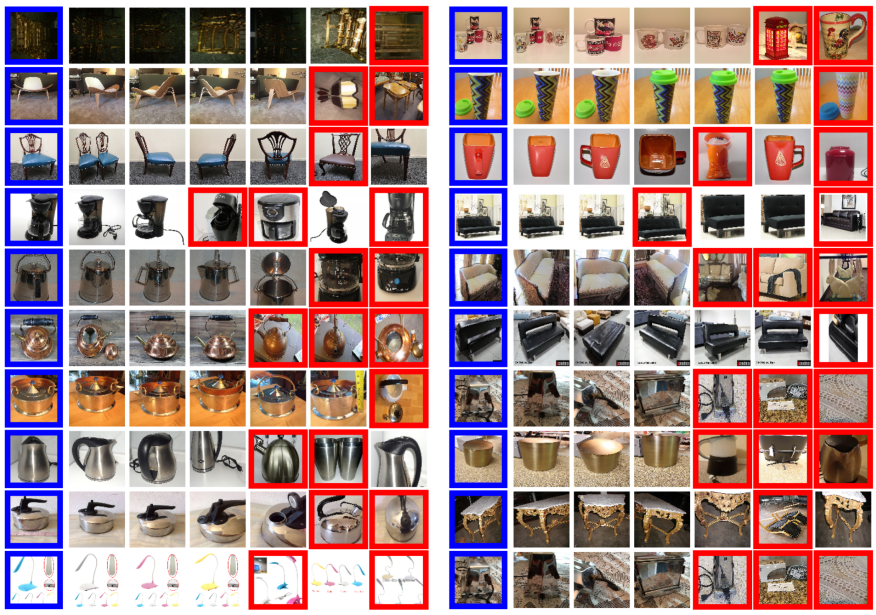


**Fig. 5.** Retrieval results of some example queries on SOP dataset. The query images and the negative retrieved images are highlighted with blue and red.

## References

1. Van Der Maaten, L.: Accelerating t-sne using tree-based algorithms. The Journal of Machine Learning Research **15**(1), 3221–3245 (2014)
2. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5022–5030 (2019)
3. Wang, X., Zhang, H., Huang, W., Scott, M.R.: Cross-batch memory for embedding learning. arXiv preprint arXiv:1912.06798 (2019)
4. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)