Compressed Self-Attention for Deep Metric Learning with Low-Rank Approximation

Ziye Chen[1] , **Mingming Gong**[2*] , **Lingjuan Ge**[1] and **Bo Du**[1*]
[1]School of Computer Science, Institute of Artificial Intelligence, and National Engineering Research Center for Multimedia Software, Wuhan University, China
[2]School of Mathematics and Statistics, University of Melbourne, Australia
{ziyechen, 2014301500136, remoteking}@whu.edu.cn, mingming.gong@unimelb.edu.au

Abstract

In this paper, we aim to boost the performance of deep metric learning by using the *self-attention (SA)* mechanism. However, due to the pairwise similarity measurement, the cost of storing and manipulating the complete attention maps makes it infeasible for large inputs such as images. To solve this problem, we propose a *compressed self-attention with low-rank approximation (CSALR)* module, which significantly reduces the computation and memory costs without sacrificing the accuracy. In CSALR, the original attention map is decomposed into a landmark attention map and a combination coefficient map with a small number of landmark feature vectors sampled from the input feature map by average pooling. Thanks to the efficiency of CSALR, we can apply CSALR to high-resolution shallow convolutional layers and implement a multi-head form of CSALR, which further boosts the performance. We evaluate the proposed CSALR on person re-identification which is a typical metric learning task. Extensive experiments shows the effectiveness and efficiency of CSALR in deep metric learning and its superiority over the baselines.

1 Introduction

Metric learning aims to construct well-structured distance metrics, which can be used to perform various tasks, such as k-NN classification, clustering, and information retrieval. Recently, deep metric learning with CNNs has shown a large improvement in learning embedding features that have small intra-class and large inter-class distance. Deep metric learning has a wide range of application in computer vision, such as person re-identification [Ye *et al.*, 2018; Wang *et al.*, 2018c], face recognition [Deng *et al.*, 2019; Liu *et al.*, 2018], and keypoint descriptor learning [Mishchuk *et al.*, 2017; Xu *et al.*, 2019].

However, due to various geometric and photometric changes such as scale change, viewpoint change, and illumination change, and the limited receptive field of convolutional kernels, the learned embedding features are not discriminative enough to ensure the intra-class compactness and the inter-class discrepancy, which would affect the performance of deep metric learning.

To tackle this problem, we enhance the discriminative power of CNNs with *self-attention (SA)* mechanism [Vaswani *et al.*, 2017], which can capture long-range contextual dependencies adaptively. SA calculates the response at each position as a weighted sum of all the feature vectors, where the weights (i.e., attention maps) are determined by the pairwise similarities among all the feature vectors. However, the pairwise similarity measurement in SA leads to high computation and memory costs, making it infeasible for large inputs. Although [Chen *et al.*, 2020] proposed a *compressed self-attention (CSA)* module, it lacks the theoretical guarantee for the accurate reconstruction of the original attention maps.

Therefore, we propose a *compressed self-attention with low-rank approximation (CSALR)* module, which reduces the computation and memory costs greatly without sacrificing the accuracy compared with the original SA. Taking advantage of the property that the feature vectors in a feature map are redundant, especially for the spatially adjacent ones, we decompose the complete attention map into a landmark attention map and a combination coefficient map with a small number of landmark feature vectors, which are sampled from the input feature map by average pooling. Then we apply the landmark attention map to the input feature map to produce the landmark output, and then apply the combination coefficient map to the landmark output to obtain the output feature map.

The high efficiency of CSALR brings additional benefits. First, we can apply CSALR to high-resolution shallow convolutional layers which lack long-range dependencies. Second, we can implement a multi-head form of CSALR where we partition the feature maps into several groups along the

*Corresponding Author. This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, 62041105, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, the Natural Science Foundation of Hubei Province under Grants 2018CFA050, the Fundamental Research Funds for the Central Universities under Grant 413000092 and 413000082. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

channel dimension and apply CSALR in each group independently, which makes the features more discriminative. We evaluate the proposed CSALR on person re-identification, which is a typical metric learning task. Extensive experiments validate the significance of CSALR in deep metric learning.

2 Related Work

2.1 Deep Metric Learning

Deep metric learning employs deep convolutional neural networks (CNNs) to learn embedding features which have small intra-class and large inter-class distance. However, due to the limited receptive field of CNNs and the challenging geometric and photometric changes, the learned embedding features are not discriminative enough. Recently, some researchers focus on designing a more effective loss function to force the networks to learn more representative embedding features. For example, [Fan *et al.*, 2019] proposed a modified softmax function to learn a discriminative hypersphere manifold embedding for person re-identification. [Deng *et al.*, 2019] proposed an additive angular margin loss to enhance the discriminative power of the network for face recognition.

Other researchers focus on designing a more robust network architecture. For instance, [Sun *et al.*, 2018] proposed a body partition strategy and a partition refinement method for person re-identification. [Kalayeh *et al.*, 2018] adopted semantic parsing strategy to extract the features of human body parts for person re-identification. There are also some works applying SA in deep metric learning, such as [Si *et al.*, 2018] and [Han *et al.*, 2018]. However, due to the high computation and memory costs of SA, SA is only applied to the part level or global level features, which is difficult to make full use of SA mechanism. [Chen *et al.*, 2020] proposed a compressed form of self-attention, however, it cannot guarantee that the reconstructed attention maps can actually approximate the original attention maps, which may degrade the performance.

2.2 Self-Attention Mechanism

Self-attention (SA) mechanism was originally proposed for machine translation [Vaswani *et al.*, 2017], then it has been successfully applied in computer vision, such as video analysis [Wang *et al.*, 2018b], image segmentation [Fu *et al.*, 2019], and image generation [Zhang *et al.*, 2018]. SA also has a close relationship to community search[Fang *et al.*, 2020b; Fang *et al.*, 2020a]. The implementation of SA mechanism for CNNs is as follows.

As shown in Figure 1, given an input feature map $\mathbf{I} \in \mathbb{R}^{C\times H\times W}$, where C, H and W represent the channels, height and width of $\mathbf{I}$, respectively, we first map $\mathbf{I}$ to two embedding feature maps $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{C'\times H\times W}$, and a new feature map $\mathbf{V} \in \mathbb{R}^{C\times H\times W}$ by 1×1 convolutions. Then, the attention map $\mathbf{A} \in \mathbb{R}^{HW\times HW}$ between $\mathbf{Q}$ and $\mathbf{K}$ is computed as follows:

$$A_{ij} = \frac{exp(Q_i^\intercal K_j)}{\sum_{l=1}^{HW} exp(Q_i^\intercal K_l)}, \tag{1}$$

where $Q_i, K_j \in \mathbb{R}^C$ denotes the feature vectors at the i-th and j-th position of $\mathbf{Q}$ and $\mathbf{K}$, respectively, and A_{ij} is a pairwise similarity measuring the j-th position's impact on the i-th position. Then, we apply the attention map $\mathbf{A}$ to the feature map $\mathbf{V}$, and perform a gated residual connection with the input $\mathbf{I}$ to obtain the output $\mathbf{O} \in \mathbb{R}^{C\times H\times W}$ as follows:

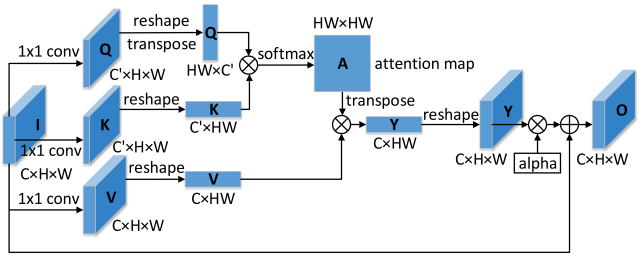

Figure 1: The illustration of the original self-attention mechanism.

$$O_i = I_i + \alpha \sum_{j=1}^{HW} A_{ij} V_j, \tag{2}$$

where α is a learnable scale parameter which is initialized to 0 and gradually learns its value during training.

In contrast to convolutions, SA can capture long-range contextual dependencies by aggregating all the feature vectors based on the pairwise similarities among them. However, the pairwise similarity measurement leads to massive consumption of computation and memory, especially for a large input, making it challenging to utilize SA mechanism fully.

3 Compressed Self-Attention with Low-Rank Approximation

In this section, we first propose a low-rank approximation to any square matrix. Then we present a compressed self-attention with low-rank approximation (CSALR) module. Finally, we introduce a general framework for deep metric learning with the CSALR modules.

3.1 Low-Rank Approximation via Sampling

Given a square matrix $A_{n,n} \in \mathbb{R}^{n\times n}$, we propose to approximate it by randomly sampling m rows and m columns of $A_{n,n}$ without replacement, and then setting $A_{n,n} \approx A_{n,m}A_{m,m}^{-1}A_{m,n}$, where $A_{n,m}$ is the $n \times m$ block of $A_{n,n}$, and with similar definitions for other blocks. The proposed method is similar to the Nyström method [Baker, 1977], however, it does not require $A_{n,n}$ to be a positive (semi-)definite matrix. We give the proof as follows.

Given the eigensystem of a full matrix $A_{n,n} \in \mathbb{R}^{n\times n}$:

$$A_{n,n}U_{n,n} = U_{n,n}\Lambda_{n,n}, \tag{3}$$

where $U_{n,n}, \Lambda_{n,n} \in \mathbb{R}^{n\times n}$ are the eigenfunction matrix and the diagonal eigenvalue matrix of $A_{n,n}$, respectively.

Following the Nyström method [Baker, 1977], we approximate (3) by randomly sampling m columns of $A_{n,n}$ as follows:

$$\frac{n}{m}A_{n,m}U_{m,n} \approx U_{n,n}\Lambda_{n,n}, \tag{4}$$

Then we match (4) against (3), and arrive at

$$\frac{n}{m}A_{n,m}U_{m,n} \approx A_{n,n}U_{n,n}, \tag{5}$$

randomly choose m rows of $A_{n,m}$ in (5), and arrive at

$$\frac{n}{m} A_{m,m} U_{m,n} \approx A_{m,n} U_{n,n}, \tag{6}$$

multiply both sides of (6) by $A_{m,m}^{-1}$, $A_{n,m}$ (from left) and $U_{n,n}^{-1}$ (from right) successively, and arrive at

$$\frac{n}{m} A_{n,m} U_{m,n} U_{n,n}^{-1} \approx A_{n,m} A_{m,m}^{-1} A_{m,n}, \tag{7}$$

match (4) against (7), and arrive at

$$U_{n,n} \Lambda_{n,n} U_{n,n}^{-1} \approx A_{n,m} A_{m,m}^{-1} A_{m,n}, \tag{8}$$

match (8) against (3), and arrive at

$$A_{n,n} \approx A_{n,m} A_{m,m}^{-1} A_{m,n}. \tag{9}$$

Equation (9) is the proposed low-rank approximation to the square matrix $A_{n,n}$. It doesn't require $A_{n,n}$ to be positive (semi-)definite or symmetric. We can sample different rows and columns of $A_{n,n}$ to form the block $A_{m,m}$, as long as $A_{m,m}$ is full rank.

3.2 CSALR Module

In this section, we propose a *compressed self-attention with low-rank approximation (CSALR)* module, where we compress the original attention maps with the proposed low-rank approximation to reduce both the computation and memory costs. We observe that the feature vectors in a feature map are inherently redundant, especially those spatially adjacent to each other. Thus we can decompose the complete attention map into a landmark attention map and a combination coefficient map with a small number of landmark feature vectors sampled from the input feature map by average pooling.

As shown in Figure 2, given an input feature map $\mathbf{I} \in \mathbb{R}^{C \times H \times W}$, we first obtain the feature maps **Q**, **K** and **V** as the original SA mechanism in Figure 1 (Section 2.2). Same with (1), the complete attention map $\mathbf{A}_{n,n}$ is computed as follows:

$$\mathbf{A}_{n,n} = \left[\frac{exp(Q_i^\intercal K_j)}{\sum_{l=1}^{n} exp(Q_i^\intercal K_l)} \right]_{n \times n}, \tag{10}$$

where $n = H \times W$ is the total number of the feature vectors in **Q** or **K**. We consider the normalization with softmax in the following low-rank approximation for numerical stability.

Then we approximate the complete attention map $\mathbf{A}_{n,n}$ via sampling based on Eq.(9). Leveraging the spatial redundancy of the feature vectors, we sample the landmark feature map $\mathbf{Q'}, \mathbf{K'} \in \mathbb{R}^{C' \times H' \times W'}$ by applying average pooling to **Q** and **K**, respectively, where the $kernel\_size$ and the $stride$ of the pooling layer both equal to $\left(\frac{H}{H'}, \frac{W}{W'}\right)$.

Similar to (10), the partial attention maps $\mathbf{A}_{n,m}$ between **Q** and **K'**, $\mathbf{A}_{m,m}$ between **Q'** and **K'**, and $\mathbf{A}_{m,n}$ between **Q'** and **K** are computed as follows:

$$\mathbf{A}_{n,m} = \left[\frac{exp(Q_i^\intercal K'_j)}{\sum_{l=1}^{m} exp(Q_i^\intercal K'_l)} \right]_{n \times m}, \tag{11}$$

$$\mathbf{A}_{m,m} = \left[\frac{exp(Q'^\intercal_i K'_j)}{\sum_{l=1}^{m} exp(Q'^\intercal_i K'_l)} \right]_{m \times m}, \tag{12}$$

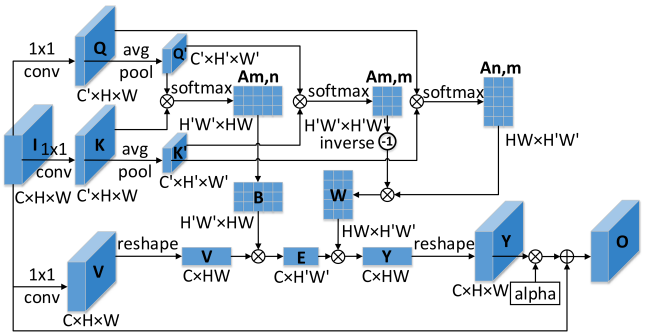

Figure 2: The illustration of the proposed compressed self-attention mechanism.

$$\mathbf{A}_{m,n} = \left[\frac{exp(Q'^\intercal_i K_j)}{\sum_{l=1}^{n} exp(Q'^\intercal_i K_l)} \right]_{m \times n}, \tag{13}$$

where Q'_i and K'_j are the landmark feature vectors at the i-th and j-th position of the landmark feature map **Q'** and **K'**, respectively, $m = H' \times W'$ is the total number of the feature vectors in **Q'** or **K'**. It is worth noting that since the normalization is considered, $\mathbf{A}_{n,m}$ is just a rescaled approximations of the corresponding $n \times m$ block of $\mathbf{A}_{n,n}$ (same for $\mathbf{A}_{m,m}$), which is reasonable considering the low-rank property of $\mathbf{A}_{n,n}$.

Then, according to (9), the complete attention map $\mathbf{A}_{n,n}$ can be approximated with the partial attention maps $\mathbf{A}_{n,m}$, $\mathbf{A}_{m,m}$ and $\mathbf{A}_{m,n}$ as follows:

$$\mathbf{A}_{n,n} \approx \mathbf{A}_{n,m} \mathbf{A}_{m,m}^{-1} \mathbf{A}_{m,n} = \mathbf{W}\mathbf{B}, \tag{14}$$

where $\mathbf{B} = \mathbf{A}_{m,n} \in \mathbb{R}^{m \times n}$ represents the landmark attention map between the landmark feature vectors and all the feature vectors, and $\mathbf{W} = \mathbf{A}_{n,m} \mathbf{A}_{m,m}^{-1} \in \mathbb{R}^{n \times m}$ represents the combination coefficient map based on the relationship between all the feature vectors and the landmark feature vectors. The complete attention map $\mathbf{A}_{n,n}$ is reconstructed by applying the combination coefficient map **W** to the landmark attention map **B**.

Since the adjacent feature vectors in space are more similar, the sub attention map of each feature vector is more similar to that of the landmark feature vector corresponding to the same pooling patch, whose combination coefficient should be larger. Thus we transform $\mathbf{A}_{n,m}$ and $\mathbf{A}_{m,m}$ as follows:

$$\mathbf{A}_{n,m} = \left[\frac{exp(Q_i^\intercal K'_j)}{\sum_{l=1}^{m} exp(Q_i^\intercal K'_l)} + \mathbb{1}(Q_i, K'_j) \right]_{n \times m}, \tag{15}$$

$$\mathbf{A}_{m,m} = \left[\frac{exp(Q'^\intercal_i K'_j)}{\sum_{l=1}^{m} exp(Q'^\intercal_i K'_l)} + \mathbb{1}(Q'_i, K'_j) \right]_{m \times m}, \tag{16}$$

where $\mathbb{1}(Q_i, K'_j)$ denotes that if the feature vector Q_i and the landmark feature vector K'_j correspond to the same pooling patch, its value is 1, otherwise, its value is 0 (same for $\mathbb{1}(Q'_i, K'_j)$). It is worth noting that (16) guarantees that $\mathbf{A}_{m,m}^{-1}$ always exists.

Then we apply the landmark attention map $\mathbf{B}$ to the feature map $\mathbf{V}$ to produce the landmark output $\mathbf{E} \in \mathbb{R}^{C \times H'W'}$, and then apply the combination coefficient map $\mathbf{W}$ to the landmark output $\mathbf{E}$, and perform a gated residual connection with the input $\mathbf{I}$ to obtain the output feature map $\mathbf{O} \in \mathbb{R}^{C \times H \times W}$ as follows:

$$O_i = I_i + \alpha \sum_{k=1}^{H'W'} W_{ik} \sum_{j=1}^{HW} B_{kj} V_j = I_i + \alpha \sum_{k=1}^{H'W'} W_{ik} E_k, \tag{17}$$

where α is a learnable scale parameter which is initialized to 0 and gradually learns its value during training.

The low-rank decomposition of the original attention maps in (14) significantly reduces the computation and memory costs, and the inherent redundancy of the feature vectors avoids performance degradation caused by compression. In experiments, we find that setting $m \ll n$ does not decrease the accuracy.

The high efficiency of CSALR allows us to implement a multi-head form of CSALR. We partition the feature maps into several groups along the channel dimension and apply CSALR in each group independently. Then we fuse the different groups with 1×1 convolution. In this way, each group can handle a specific relationship between the features alone, which helps to diversify the long-range interactions and makes the features more discriminative.

3.3 CSALR in Deep Metric Learning

The proposed CSALR module can be easily applied to the existing backbones to capture long-range contextual dependencies efficiently, providing a general framework for deep metric learning. The computational and memory efficiency of CSALR allows us to apply CSALR to high-resolution shallow convolutional layers and implement a multi-head form of CSALR.

As shown in Figure 3, the framework includes a backbone network, CSALR modules, and deep supervision branches. We apply the CSALR module to each stage of the backbone network. In this way, the high-resolution shallow convolutional layers, whose receptive filed is limited, can benefit more from the long-range interactions. To diversify the long-range interactions, we implement the multi-head form of CSALR. To assist the learning of attention maps, we apply deep supervision to each CSALR module. It's worth noting that the residual connection of CSALR is drawn out of the CSALR module in Figure 3.

In each deep supervision branch, we first perform an element-wise multiplication between the input feature map and the output of CSALR module and obtain a new feature map, then we apply a global average pooling layer to the feature map and obtain a feature vector, and then we project the feature vector linearly with a fully connected layer and apply a loss function to the result. In experiments, we find that the element-wise multiplication is better than the element-wise summation, so we implement deep supervision before the residual connection. The final loss function of the framework is as follows:

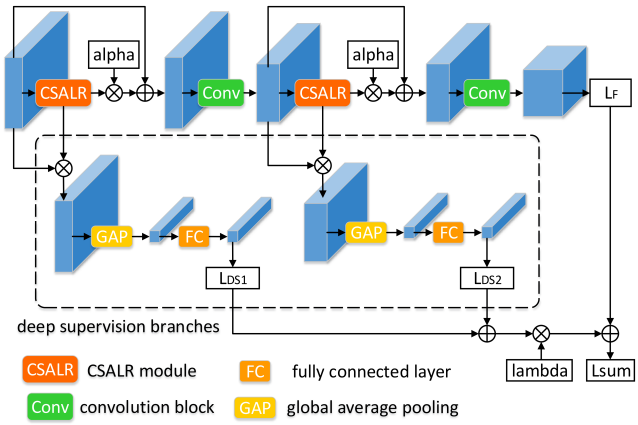

Figure 3: The general framework of applying our CSALR module in deep metric learning.

$$L_{sum} = L_F + \lambda \sum_{i=1}^{N} L_{DSi}, \tag{18}$$

where L_F is the loss of the main branch, and L_{DSi} is the loss of the i-th deep supervision branch. N is the total number of CSALR modules. λ is a balance factor between L_F and L_{DS}.

When applying the proposed framework to boost the performance of the specific metric learning algorithms, we just modify the backbone networks of the original models without changing the remaining settings. Thus, the loss L_F is the same as the original algorithm. The loss L_{DS} is cross entropy in our experiments. When comparing CSALR with SA or CSA, we just replace the CSALR module with the SA or CSA module in the framework, without changing the deep supervision branches or the corresponding loss functions.

4 Experiments

We validate the proposed CSALR on person re-identification, which is a typical metric learning task. We choose two representative methods in person re-identification, *i.e.*, **PCB** [Sun *et al.*, 2018] and **PCB-RPP** [Sun *et al.*, 2018], as our baselines. We modify the backbone networks of the baseline models without changing other settings. We provide both qualitative and qualitative comparisons to demonstrate the effectiveness and efficiency of CSALR in deep metric learning.

4.1 Datasets

We use three datasets for evaluation, *i.e.*, **Market-1501** [Zheng *et al.*, 2015], **DukeMTMC-reID** [Ristani *et al.*, 2016; Zheng *et al.*, 2017], and **CUHK03-NP** [Zhong *et al.*, 2017]. Market-1501 contains 751 training IDs with 12,936 images and 750 query IDs with 3,368 query images and 19,732 gallery images, which are captured by 6 cameras. DukeMTMC-ReID contains 702 training IDs with 16,522 images and 702 query IDs with 2,228 query images and 17,661 gallery images, which are captured by 8 cameras. CUHK03-NP contains 767 training IDs with 7,365 images and 700

Models	Market-1501				DukeMTMC-reID				CUHK03-NP			
	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP
DuATM [Si *et al.*, 2018]	91.4	97.1	—	76.6	81.8	90.1	—	64.6	—	—	—	—
Mancs [Wang *et al.*, 2018a]	93.1	—	—	82.3	84.9	—	—	71.8	65.5	—	—	60.5
SphereReID [Fan *et al.*, 2019]	**94.4**	98.0	98.7	83.6	83.9	90.6	92.4	68.5	—	—	—	—
PCB [Sun *et al.*, 2018]	92.3	97.2	98.2	77.4	81.7	89.7	91.9	66.1	59.7	77.7	85.2	53.2
PCB + SA	93.8	97.7	98.5	82.2	85.3	92.7	94.8	73.4	65.9	82.3	88.4	62.7
PCB + CSA [Chen *et al.*, 2020]	93.7	**98.2**	98.8	82.3	85.5	92.8	94.5	73.5	67.2	83.9	88.9	63.7
PCB + CSALR	93.9	97.8	98.6	82.4	**86.1**	93.2	**95.1**	**74.0**	67.6	82.4	88.4	64.0
PCB-RPP [Sun *et al.*, 2018]	93.8	97.5	98.5	81.6	83.3	90.5	92.5	69.2	62.8	79.8	86.8	56.7
PCB-RPP + SA	93.3	97.5	98.7	83.2	84.9	92.8	94.7	72.9	66.1	84.0	89.0	64.6
PCB-RPP + CSA [Chen *et al.*, 2020]	93.9	97.8	**98.8**	83.5	85.4	93.1	94.5	73.1	67.4	83.6	**89.1**	65.0
PCB-RPP + CSALR	94.0	98.1	**98.8**	**83.8**	85.4	**93.3**	95.0	73.8	**68.8**	**84.4**	**89.1**	**65.3**

Table 1: Comparison of the models with and without the proposed CSALR. SA means the original self-attention.

query IDs with 1,400 query images and 5,332 gallery images, where each identity is captured by 2 disjoint cameras. It offers two kinds of bounding boxes, one is hand-labeled, the other is DMP-detected, and we use the latter.

For evaluation, we use the cumulative matching characteristic (CMC) for rank-1, rank-5, rank-10 and the mean average precision (mAP) as our metrics. We evaluate all the methods under the single-query mode. For simplicity, we do not use re-ranking [Zhong *et al.*, 2017] for post-processing, which can improve mAP by a large margin.

4.2 Implementation Details

We use a similar setting to [Sun *et al.*, 2018]. The backbone is ResNet-50 with pre-trained weights from ImageNet. We use the proposed framework with CSALR to enhance the backbone. We apply the CSALR module to each residual block of the backbone except the last one, leading to 3 CSALR modules in total. In Eq. (9), the loss L_F is the same as PCB and PCB-RPP, and the loss L_{DS} is cross entropy. The balance factor λ is set to 1.0. We implement the multi-head form of CSALR. For each CSALR module, the number of sampled points in each group is set to 96, and the number of groups is set to 2. When comparing CSALR with SA or CSA, we replace the CSALR module with the SA or CSA module in the framework without changing other settings. It's worth noting that we do not implement the multi-head form of SA, since it costs too much computation and memory resources.

For data processing, the input images are all resized to 384×128. We apply random horizontal flipping with 0.5 and normalization as data augmentation. For training, we set the batch size to 64 and the total number of training epochs to 100. The optimizer is set to Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 10^{-4}. The base learning rate is initialized to 0.1 and decayed by 0.1 after every 40 epochs, and the learning rate for the backbone network is set to $0.1\times$ the base learning rate. For PCB-RPP, we first train PCB for 40 epochs, and then train RPP for another 60 epochs with weights initialized from PCB.

4.3 Performance Comparison

As shown in Table 1, the proposed framework with CSALR modules can effectively boost the performance of all the original algorithms on Market-1501, DukeMTMC-ReID and CUHK03-NP with respect to both rank-1 accuracy and mAP. On the three datasets, for PCB with CSALR, the increase in rank-1 accuracy is $+1.6\%$, $+4.4\%$, and $+7.9\%$, respectively; the increase in mAP is $+5.0\%$, $+7.9\%$, and $+10.8\%$, respectively; for PCB-RPP with CSALR, the increase in rank-1 accuracy is $+0.2\%$, $+2.1\%$, and $+6.0\%$, respectively; the increase in mAP is $+2.2\%$, $+4.6\%$, and $+8.6\%$, respectively. This suggests that CSALR can effectively capture long-range contextual dependencies and make the extracted features more discriminative. It's worth noting that the improvement for mAP is larger than that for rank-1 accuracy. In reality, rank-1 accuracy represents the ability to retrieve the easiest match in the gallery, while mAP represents the ability to find all the matches, which indicates that CSALR helps to find more challenging matches.

We also compare CSALR with SA and CSA. As shown in Table 1, it's interesting that although the attention maps are compressed in CSALR, the performance of CSALR is better than that of SA. There are two reasons for this phenomenon. On the one hand, the inherent redundancy of the feature vectors guarantees an accurate approximation of the original attention maps. On the other hand, the multi-head form of CSALR helps to diversify the long-range interactions, which compensates for the performance degradation caused by compression. We can also see that the performance of CSALR is better than that of CSA, which suggests that the approximation to the attention maps in CSALR is more reasonable and more accurate than that in CSA.

4.4 Different Number of Sampled Points

We fix the number of groups to 2 and vary the number of sampled points in each group to see its influence. As shown in Table 2, the performance rises as the number of sampled points increases. The reason behind this is that the more the sampled points, the more accurate the approximation to the original attention maps and the less the performance degradation. However, the performance improvement is slight when the number of sampled points increases from 96 to 192. This further demonstrates the significant redundancy of the feature vectors in a feature map, so that we can approximate the complete attention maps accurately with only a small number of landmark feature vectors.

Models	Sample Num	Market-1501		DukeMTMC-reID	
		R-1	mAP	R-1	mAP
PCB + CSALR	24	93.6	82.1	85.5	73.5
PCB + CSALR	48	93.8	82.3	85.8	73.8
PCB + CSALR	96	**93.9**	82.4	86.1	74.0
PCB + CSALR	192	**93.9**	**82.6**	**86.3**	**74.2**

Table 2: Comparison of the models with different number of sampled points in each group of CSALR.

4.5 Different Number of Groups

We fix the number of sampled points to 96 and vary the number of groups to see its influence. As shown in Table 3, the performance is best when the number of groups is 2. The performance drops when the number of groups is smaller or larger than 2, indicating that the multi-head form of CSALR helps to diversify the long-range interactions and makes the extracted features more discriminative. However, too many groups and too few channels in each group will lead to insufficient expression of the specific semantics, which harms the performance of CSALR.

Models	Group Num	Market-1501		DukeMTMC-reID	
		R-1	mAP	R-1	mAP
PCB + CSALR	1	93.6	82.0	85.8	73.8
PCB + CSALR	2	**93.9**	**82.4**	**86.1**	**74.0**
PCB + CSALR	4	93.7	82.3	86.0	73.8
PCB + CSALR	8	93.6	82.1	85.7	73.6

Table 3: Comparison of the models with different numbers of groups in CSALR.

4.6 CSALR on Different Layers

As shown in Table 4, applying the CSALR modules to both the shallow and deep layers of the backbone leads to the best performance. Applying the CSALR modules to shallow layers is better than deep layers, which demonstrates that the shallow layers lack long-range dependencies extremely. So it's more beneficial to apply SA mechanism to shallow layers, and our CSALR module is very efficient considering the large size of the shallow layers.

Models	Attn Stages	Market-1501		DukeMTMC-reID	
		R-1	mAP	R-1	mAP
PCB	no	92.3	77.4	81.7	66.1
PCB + CSALR	1st stage	93.3	81.4	84.8	73.0
PCB + CSALR	2nd,3rd stages	92.9	80.9	84.4	72.3
PCB + CSALR	1st,2nd,3rd stages	**93.9**	**82.4**	**86.1**	**74.0**

Table 4: Comparison of the models with CSALR on different layers of the backbone.

4.7 Resource Costs Comparison

As shown in Table 5, for SA and CSALR with 1, 2, 4, 8 groups, compared with the original method, the relative increase in memory cost is $+118.45\%$, $+19.88\%$, $+39.67\%$, $+61.27\%$, and $+85.08\%$, respectively; the decrease in FPS is -54.17%, -12.70%, -20.86%, -28.10%, and -34.52%, respectively. We can see that the methods with CSALR consume much less computation and memory resources than the methods with SA, even if the number of groups is set to 8. Thus, we can apply CSALR with more flexibility.

Models	Group Num	Memory (MB/image)	FPS
PCB	—	137.64	581.81
PCB + SA	1	300.68	266.67
PCB + CSALR	1	165.00	507.93
PCB + CSALR	2	192.25	460.43
PCB + CSALR	4	221.97	418.30
PCB + CSALR	8	254.75	380.95

Table 5: Comparison of the speed and memory cost of the models with and without SA or CSALR.

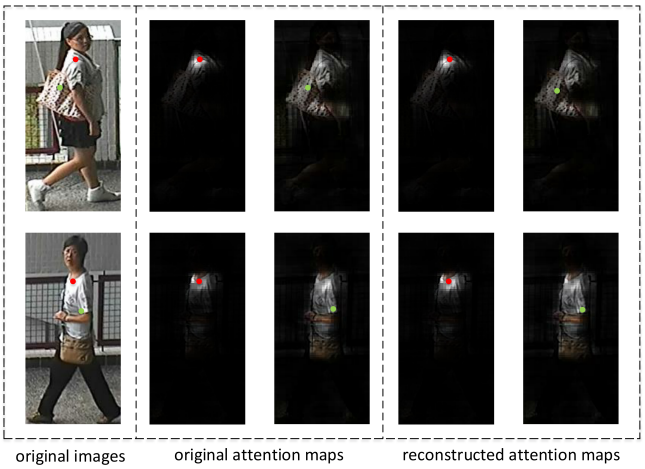

Figure 4: Visualization of the original attention maps and the reconstructed attention maps.

4.8 Analysis of Attention Maps

As shown in Figure 4, for each selected point (marked in a specific color) in the image, the corresponding attention map focus on the most relevant parts in the whole image, which indicates that the SA mechanism can capture long-range contextual dependencies adaptively. We can also see that the reconstructed attention maps are very similar to the original attention maps, which indicates that the feature vectors in a feature map have significant redundancy. We can approximate the complete attention maps accurately with only a small number of landmark feature vectors.

5 Conclusion

In this paper, we aim to boost the performance of deep metric learning with *self-attention (SA)* mechanism, which can capture long-range contextual dependencies adaptively. Accordingly, we propose a *compressed self-attention with low-rank approximation (CSALR)*, which significantly reduces the computation and memory costs without sacrificing the accuracy. Qualitative and quantitative experiments demonstrate the significance of CSALR in deep metric learning.

References

[Baker, 1977] Christopher TH Baker. The numerical treatment of integral equations. 1977.

[Chen *et al.*, 2020] Ziye Chen, Yanwu Xu, Mingming Gong, Chaohui Wang, Bo Du, and Kun Zhang. Compressed self-attention for deep metric learning. 2020.

[Deng *et al.*, 2019] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[Fan *et al.*, 2019] Xing Fan, Wei Jiang, Hao Luo, and Mengjuan Fei. Spherereid: Deep hypersphere manifold embedding for person re-identification. *Journal of Visual Communication and Image Representation*, 60:51–58, 2019.

[Fang *et al.*, 2020a] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. A survey of community search over big graphs. 29(1):353–392, 2020.

[Fang *et al.*, 2020b] Yixiang Fang, Yixing Yang, Wenjie Zhang, Xuemin Lin, and Xin Cao. Effective and efficient community search over large heterogeneous information networks. 13(6):854–867, 2020.

[Fu *et al.*, 2019] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[Han *et al.*, 2018] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-aware attention model for fine-grained representation learning. pages 2040–2048, 2018.

[Kalayeh *et al.*, 2018] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1062–1071, 2018.

[Liu *et al.*, 2018] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *Advances in Neural Information Processing Systems*, pages 6222–6233, 2018.

[Mishchuk *et al.*, 2017] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[Ristani *et al.*, 2016] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[Si *et al.*, 2018] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5363–5372, 2018.

[Sun *et al.*, 2018] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2018a] Cheng Wang, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 365–381, 2018.

[Wang *et al.*, 2018b] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[Wang *et al.*, 2018c] Zheng Wang, Xiang Bai, Mang Ye, and Shinichi Satoh. Incremental deep hidden attribute learning. pages 72–80, 2018.

[Xu *et al.*, 2019] Yanwu Xu, Mingming Gong, Tongliang Liu, Kayhan Batmanghelich, and Chaohui Wang. Robust angular local descriptor learning. *arXiv preprint arXiv:1901.07076*, 2019.

[Ye *et al.*, 2018] Mang Ye, Zheng Wang, Xiangyuan Lan, and Pong C Yuen. Visible thermal person re-identification via dual-constrained top-ranking. pages 1092–1099, 2018.

[Zhang *et al.*, 2018] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[Zheng *et al.*, 2015] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[Zheng *et al.*, 2017] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3754–3762, 2017.

[Zhong *et al.*, 2017] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017.