Improving Visual Embeddings using Attention and Geometry Constraints

Pengfei Fang

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

August 2022

Except where otherwise indicated, this Thesis is my own original work.
The nature and extent of collaboration have been outlined in this Thesis.

Pengfei Fang
27 August 2022

To my parents.

Declaration

My doctoral studies have been conducted under the guidance and supervision of Dr. Lars Petersson and Dr. Mehrtash Harandi. This Thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Most of the results in this Thesis have been published at refereed international journals or conferences. Some of these results have been achieved in collaboration with other researchers. Co-authored publications as a result of parts of my Thesis research are listed as below: (*Note: * indicates that the Thesis contains material of this paper.*)

0.1 Published Papers

- *__P. Fang__, J. Zhou, SK. Roy, L. Petersson and M. Harandi. Bilinear Attention Networks for Person Retrieval. Proc. of the IEEE International Conference on Computer Vision (ICCV'19), Seoul, Korea, 2019.
- *__P. Fang__, P. Ji, J. Zhou, L. Petersson and M. Harandi. Channel Recurrent Attention Networks for Video Pedestrian Retrieval. Proc. of the 15^{th} Asian Conference on Computer Vision (ACCV'20), Virtual Kyoto, Japan, 2020.
- *__P. Fang__, P. Ji, L. Petersson and M. Harandi. Set Augmented Triplet Loss for Video Person Re-Identification. Proc. of the IEEE Winter Conference on Application of Computer Vision (WACV'21), Virtual conference, 2021.
- *__P. Fang__, J. Zhou, SK. Roy, P. Ji, L. Petersson and M. Harandi. Attention in Attention Networks for Person Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2021.
- *__P. Fang__, M. Harandi and L. Petersson. Kernel Methods in Hyperbolic Spaces. Proc. of the IEEE International Conference on Computer Vision (ICCV'21), Virtual conference, 2021.

0.2 Non-lead Author Papers

- J. Zhou, SK. Roy, __P. Fang__, M. Harandi and L. Petersson. Cross-Correlated Attention Networks for Person Re-Identification. Image and Vision Computing (IVC), 2020.
- J. Hong, __P. Fang__, W. Li, T. Zhang, C. Simon, M. Harandi and L. Petersson. Reinforced Attention for Few-Shot Learning and Beyond. Proc. of the IEEE

Conference on Computer Vision and Patterson Recognition (CVPR'21), Virtual conference, 2021.

- A. Cheraghian, S. Rahman, **P. Fang**, SK. Roy, L. Petersson and M. Harandi. Semantic-aware Knowledge Distillation for Few-Shot Class-Incremental Learning. Proc. of the IEEE Conference on Computer Vision and Patterson Recognition (CVPR'21), Virtual conference, 2021.

- A. Cheraghian, S. Rahman, S. Ramasinghe, **P. Fang**, C. Simon, L. Petersson and M. Harandi. Synthesized Feature based Few-Shot Class-Incremental Learning on a Mixture of Subspaces. Proc. of the IEEE International Conference on Computer Vision (ICCV'21), Virtual conference, 2021.

- R. Ma, **P. Fang**, T. Drummond and M. Harandi. Adaptive Poincaré Point to Set Distance for Few-Shot Classification. Proc. of the 36^{th} AAAI Conference on Artificial Intelligence (AAAI'22), Virtual conference, 2022.

- J. Zhou, T. Zhang, **P. Fang**, M. Harandi and L. Petersson. Feature Correlation Aggregation: on the Path to Better Graph Neural Networks. Under review.

Acknowledgements

I have so many people to thank for their help during my doctoral study in the Australian National University (ANU) and Commonwealth Scientific and Industrial Research Organisation (CSIRO)! I can never make it without the help from them.

Firstly and foremost, I would like to convey my sincere gratitude to my primary supervisor, Dr. Lars Petersson, who has been incredibly supportive since I first met him. At the PhD application stage, Lars provided me full support in the scholarship application and suggestions in the future research directions, which paved a smooth way for me to start my PhD journey. Throughout my doctoral study, Lars's continuous support in all aspects, *e.g.*, idea discussion, computing resources, academic writing, *etc.*, makes my PhD adventure easier. I really enjoy working in such an amazing research environment that Lars created for us. I appreciate all his supports and help. Importantly, his supervision has been the cornerstone of this Thesis.

I also wish to express my heartfelt thanks to my chair supervisor, Dr. Mehrtash Harandi, for his great supervision in my research journey. I first met Mehrtash in his data analytics course at ANU, where I found the beauty of machine learning and determined to pursue a PhD degree in this field. During my doctoral study, he would always like to sacrifice his time to discuss my research topics/ideas, answer mathematical questions, or share his valuable comments. Further, his critical thinking, rigour in mathematical problems, and very keen sense in research directions have influenced me a lot and will be a life-long treasure for my future career.

I also take this opportunity to thank Dr. Pan Ji for his kind help and collaboration throughout these years. The discussions with him on different research topics are always very stimulating and have been a source of fancy ideas. He also shared his experience in both research and industry, which are useful in my future career.

My time as a PhD candidate at ANU and CSIRO was full of fun by accompanied by many nice friends and colleagues. They are Jieming (Jim) Zhou, Jie Hong, Dr. Weihao Li, Shi Qiu, Yan Han, Christian Simon, Dr. Soumava Kumar Roy, Changkun Ye, Peipei Song, Hao Zhu, Junlin Han, Dr. Jing Zhang, Dr. Ali Cheraghian, Dr. Shafin Rahman, Dr. Russell Tsuchida, Dr. Yiqing Guo, Dr. Moshiur Farazi, and many others. Without them, my PhD life at Canberra would surely be of much less fun! I indeed miss the BBQ "banquets" near the Lake Burley Griffin with them. I am also fortunate to visit Monash University and received lots of help from Rongkai Ma and Xuelian Cheng.

My old friends in Canberra should also have a place in my acknowledgement. I studied for my master's degree in ANU and worked on my individual research project in the Networked System group at the Research School of Engineering. The people in the group are very friendly and brilliant. They taught me so much knowl-

edge on mathematics and multi-agent systems. Thank you to all the members of the group: Dr. Zhiyong Sun, Dr. Qingchen Liu, Dr. Na Huang, Dr. YonHon Ng, Zhixun Li, Dr. Yun (Gavin) Hou, Dr. Junming (Jamie) Wei, Dr. Xiaolei (Eric) Hou, Dr. Mengbin (Ben) Ye, Dr. Jiahu Qin, and many others. I especially thank Dr. Qingchen Liu, for his supervision in my individual research project. I can never understand the complex mathematical theorems without his explanation.

Lastly but also most importantly, I wish to express my heartfelt thank to my parents for their unconditional love, support, and encouragement. This Thesis is dedicated to my parents.

Abstract

Learning a non-linear function to embed the raw data (*i.e.*, image, video, or language) to a discriminative feature embedding space is considered a fundamental problem in the learning community. In such embedding spaces, the data with similar semantic meaning are clustered, while the data with dissimilar semantic meaning are separated. A number of practical applications can benefit from a good feature embedding, *e.g.*, machine translation, classification/recognition, retrieval, any-shot learning, *etc*. In this Thesis, we aim to improve the visual embeddings using attention and geometry constraints.

In the first part of the Thesis, we develop two neural attention modules, which can automatically localise the informative regions within the feature map, thereby generating a discriminative feature representation for the image. An *Attention in Attention* (AiA) mechanism is first proposed to align the feature map along with the deep network, by modelling the interaction of inner attention and outer attention modules. Intuitively, the AiA mechanism can be understood as having an attention inside another, with the inner one determining where to focus for the outer attention module. Further, we employ explicit non-linear mappings in Reproducing Kernel Hilbert Spaces (RHKSs) to generate attention values, leading the channel descriptor of the feature map to own the representation power of second-order polynomial kernel and Gaussian kernel. In addition, the *Channel Recurrent Attention* (CRA) module is proposed to build a global receptive field to the feature map. The existing attention mechanisms focus on either the channel pattern or the spatial pattern of the feature map, which cannot make full use of the information in the feature map. The CRA module can jointly learn the channel and spatial patterns of the feature map and produce attention value per every element of the input feature map. This is achieved by feeding the spatial vectors to a recurrent neural network (RNN) sequentially, such that the RNN can create a global view of the feature map.

In the second part, we investigate the superiority of geometry constraint for embedding learning. We first study the geometry concern of the set as an embedding for a video clip. Usually, the video embedding is optimised using triplet loss, in which the distance is calculated between clip features, such that the frame feature cannot be optimised directly. To this end, we model the video clip as a set, and employ the distance between sets in the triplet loss. Tailored for the *set-aware triplet loss*, a new set distance metric is also proposed to measure the hard frames in a triplet. Optimising over set-aware triplet loss leads to a compact clip feature embedding, improving the discriminative of the video representation. Beyond the flat Euclidean embedding space, we further study a curved space, *i.e.*, hyperbolic spaces, as image embedding spaces. In contrast to Euclidean embedding, hyperbolic embedding can encode the hierarchical structure of data, as the volume of hyperbolic space increases

exponentially. However, performing basic operations for comparison in hyperbolic spaces is complex and time-consuming. For example, the similarity measure is not well-defined in hyperbolic spaces. To mitigate this issue, we introduce the *positive definite (pd) kernels for hyperbolic embeddings*. Specifically, we propose four pd kernels in hyperbolic spaces in conjunction with a theoretical analysis. The proposed kernels include hyperbolic tangent kernel, hyperbolic RBF kernel, hyperbolic Laplace kernel, and hyperbolic binomial kernel.

We demonstrate the effectiveness of the proposed methods via a image or video person re-identification task. We also evaluate the generalisation of hyperbolic kernels by few-shot learning, zero-shot learning and knowledge distillation tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In the computer vision (CV) community, one fundamental problem is to enable machines the capacity to encode visual data, *e.g.*, image or video, to a latent embedding space. In such an embedding space, the intra (inter)-class distance of objects is minimised (maximised) as shown in Fig. 1.1. For example, in the raw image space, the images are distributed randomly. The image embedding techniques (*i.e.*, SIFT [Lowe, 2004], neural networks [LeCun et al., 2015]) are supported to encode the images, such that the images of the same classes are clustered, while images of different classes are separated. Many practical applications rely on image embedding techniques, such as image classification/recognition, metric learning, image retrieval, object detection, semantic segmentation, to name a few. In the last decades, many efforts have been made to develop the embedding techniques, however, it remains a dominant problem to create a discriminative embedding space. This Thesis studies the embedding method so as to improve the embedding quality of the visual data.

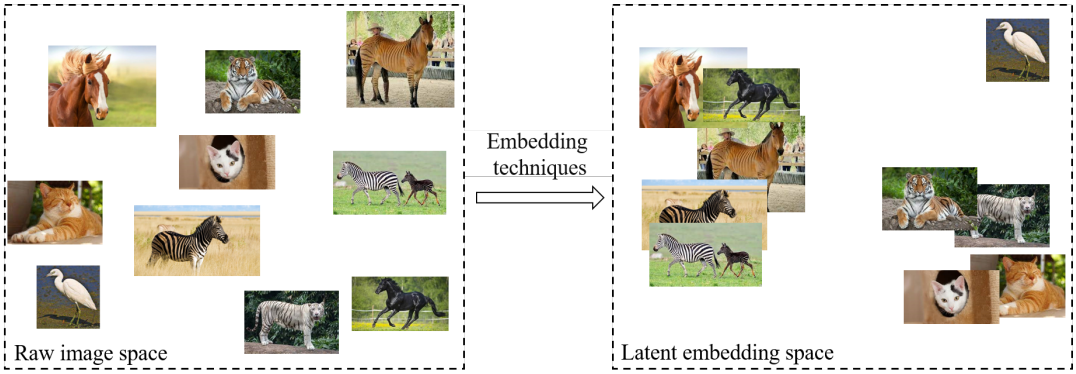


Figure 1.1: The pipeline of image embeddings. In the embedding space, the images of the same classes are clustered, while the images of the different classes are separated.

In the modern CV community, a discriminative and robust feature embedding for the image is extracted by convolutional neural networks (CNNs), with the earliest works proposed by LeCun *et al*. [LeCun et al., 1998]. A CNN contains multiple convolutional layers between the input (*e.g.*, raw data) and output (*e.g.*, classifier

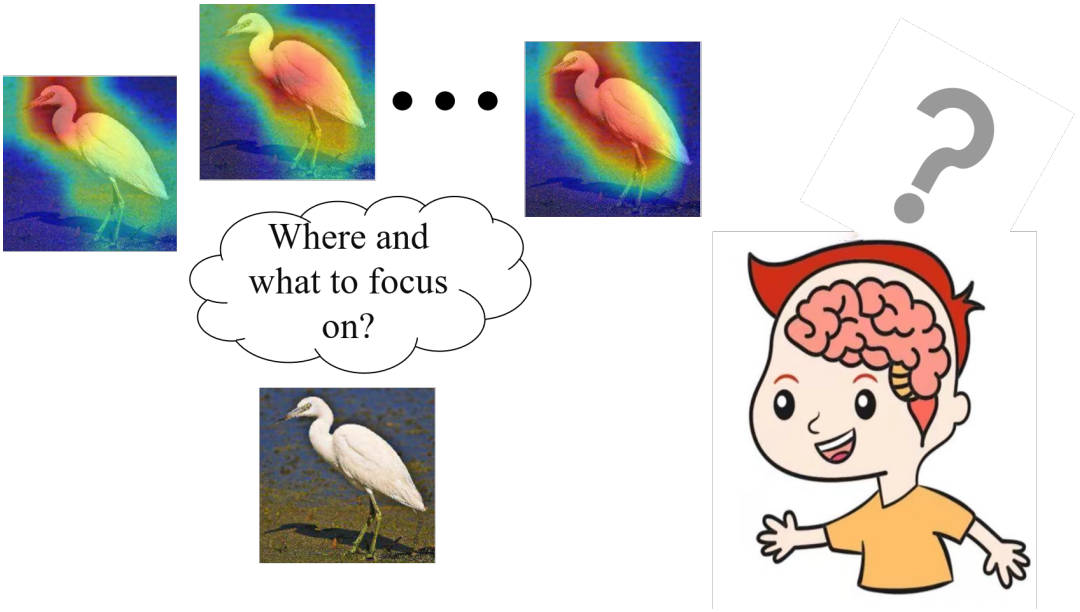


Figure 1.2: The visual attention mechanism mimics the human's perception process. The attention mechanism can localise the informative regions within the image. It thereby helps the deep network to encode a discriminative feature embedding for the input image. In the heat map, the response increases from blue to red. Best viewed in colour.

or feature embedding), with each layer having many convolutional kernels and an activation function. The hierarchical architecture of CNNs is inspired by an animal's visual perception [Hubei and Wiesel, 1962]. Generally speaking, the artificial neurons and convolutional kernels mimic biological neurons and receptors from an animal's visual perception. The function that visual perception can transmit the neural electric signal exceeding a threshold to the next layer of neurons is achieved by the activation function after the convolutional layer. In the past few years, many effective and efficient CNN architectures have been proposed to improve the recognition accuracy [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016; Huang et al., 2017; Sandler et al., 2018; Ma et al., 2018; Chen et al., 2020b], and became the standard tool to encode images in a discriminative embedding space.

Along with the CNN architectures, many other strategies are further studied to improve the CNN's embedding capacity, including data pre-processing Zhong et al. [2017b], optimiser Kingma and Ba [2014], loss function Schroff et al. [2015], learning trick He et al. [2019], *etc*. In this Thesis, we aim to improve the quality of the visual embedding from two perspectives, namely, visual attention and geometry constraints. Those two advanced solutions study the embedding method in two lines, one for the feature extractor and another for the embedding space.

The attention mechanism, inspired by the human's perception process to recognise objects, is studied extensively in the visual community. The main goal of the attention mechanism is to help the deep network to attend the informative regions

in the image. As shown in Fig. 1.2, when a human sees a "heron", one needs to attend to/focus on the most important area to recognise the "heron" correctly. To incorporate this attentive ability in the deep network, many attention mechanisms have been developed. The attention module is designed to automatically select the meaningful parts of images, and is trained in a weakly supervised manner (*i.e.*, no explicit label information is given to identify the areas to attend to). Broadly speaking, the attention mechanism, as an additional block in the deep network, can generate data-dependent weights, to weigh the importance of features. In other words, it is a method to distinguish the importance of the data, which helps the network to encode discriminative feature embeddings.

The hard attention, whose attention map contains a binary value, is first proposed in the early works [Mnih et al., 2014; Xu et al., 2015; Jaderberg et al., 2015]. The seminal work in [Mnih et al., 2014] divides the image into portions, which are then sequentially fed to the glimpse network. In this modelling, the network can glimpse the image multiple times and adaptively select the useful areas. However, training such an attention block requires reinforcement learning, which is complex, slow and resource-intensive. Xu *et al*. model the attention to aligning the visual and language concepts via recurrently comparing the feature maps and language embeddings, generating the image captions [Xu et al., 2015]. In [Jaderberg et al., 2015], another kind of hard attention, termed spatial transformer network (STN), aligns the image via localising and cropping the object in the feature map.

However, the hard attention can only select some regions within the image, without weighing the importance of local features[1]. Thus another category of attention mechanism, soft attention, is introduced to learn data-dependent attention weights, thereby emphasising/attenuating the significant/insignificant local features in the feature map. To achieve so, many studies have been made to learn attention weights [Bahdanau et al., 2015; Hu et al., 2018; Woo et al., 2018; Wang et al., 2018b]. This type of attention mechanism is first studied in the machine translation Bahdanau et al. [2015]. The following years also witness its success in the computer vision field. Hu *et al*. use the global feature[2] to re-weigh each slice of the feature map. Instead of weighing the importance channel-wise, the convolutional block attention module (CBAM) also learns the regional interaction spatially [Woo et al., 2018]. To capture long-distance dependencies, Wang *et al*. propose to aggregate the local feature via non-local operations [Wang et al., 2018b]. Many attention mechanisms that followed have been developed to use the channel or spatial information in the feature map [Wang et al., 2018a; Li et al., 2018c; Jetley et al., 2018; Chen et al., 2019b], which motivates us to develop attention mechanisms, aiming to use more information efficiently from the feature map.

In this Thesis, we propose two attention mechanisms, namely the Attention in Attention (AiA) mechanism and the Channel Recurrent Attention mechanism. The

[1] In this Thesis, the local feature refers to the patch descriptor at each spatial position in the feature map.

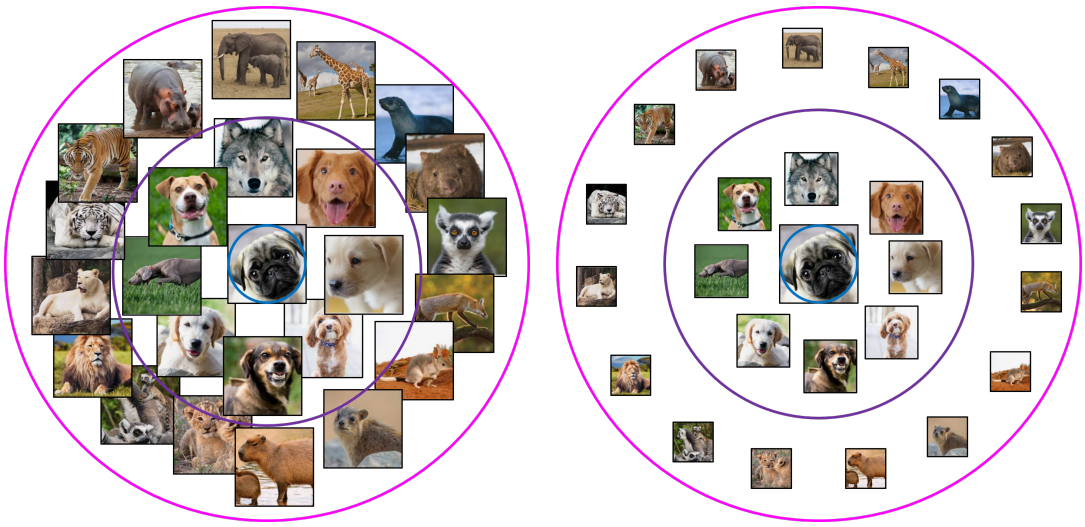
[2] In this Thesis, the global feature refers to the channel descriptor, which is aggregated from the feature map via global average pooling, global max pooling, *etc*.

AiA mechanism contains two components, *i.e.*, inner attention and outer attention. The inner attention builds the interaction among the local and global features, while the outer attention preserves the spatial structural information of the feature map. The second attention mechanism aims to create a global receptive field to the feature map. This is achieved by learning the channel and spatial patterns jointly by a recurrent neural network (RNN). Our developments are the first attempt that both channel attention and spatial attention are jointly optimised to select useful features. Extensive experiments on image/video person re-identification demonstrate the superior performance of proposed attention mechanisms.

In the second part of this Thesis, we study the advance of other geometry constraints for image embeddings. The Euclidean space has been considered a default space to encode images since it is a natural generalisation of the 3-D space we live in and provides straight operators and measurements. However, the Euclidean space has difficulty encoding the complex structural relationships between data. For example, the Euclidean space will cause distortions when encoding graph data [Liu et al., 2019a], as the Euclidean distance cannot reveal the graph distance between two the connected nodes. Having such an issue in mind, the geometry constraint is introduced to encode the data, thereby better revealing the structures between data. The embedding in the subspace, also known as a Grassmannian manifold, is robust against outliers and other variations including pose, illumination [Zhang et al., 2020a; Simon et al., 2020; Basri and Jacobs, 2003]. Modelling the points as a set can be tolerant to the order of data [Yu et al., 2018; Zaheer et al., 2017; Ribera et al., 2019]. Beyond the Euclidean space, the curved spaces, *e.g.*, hyperbolic or hyperbolic spaces, can encode complex structural information among the data, increasing the discriminative power of embedding spaces [Zhang et al., 2020b; Liu et al., 2017b; Ganea et al., 2018; Khrulkov et al., 2020]. For example, the volume of hyperbolic spaces increases exponentially, thereby being able to encode hierarchical information in the data (see the difference between image embeddings in Euclidean spaces and hyperbolic spaces in Fig. 1.3[3].).

In this Thesis, we study the advance of exploring sets and hyperbolic geometry as embedding spaces. We first study video data embedding. Existing studies directly optimise clip features, aggregated from a set of frame features, to learn a video embedding. Such training protocol ignores the distribution of video frames and leads to sub-optimal learning of video embedding spaces. That said, the existing pipeline for the video data embedding learning ignores optimising the difficult frame features. To bridge this gap, we propose to model the video clip as a set and employ the distance between sets to optimise the triplet loss. We also define a hybrid set distance metric, which reveals the distance between hard positive frames and hard negative frames separately in a video triplet, such that the hard frame features can be optimised explicitly. Hyperbolic geometry can be characterised as a Riemannian manifold with a negative sectional curvature. Several recent studies suggest that embedding words, graphs, or images using hyperbolic geometry can be beneficial

[3]This simple illustration is inspired by the work in [Gulcehre et al., 2019]



(a) Euclidean embedding in 2D space (b) Hyperbolic embedding in 2D space

Figure 1.3: Illustration of how images might be embedded in a Euclidean space and a hyperbolic space in the 2D case. The location of the embedding indicates the distance between each image and that of a pug (in the centre). In the case that the number of objects within a given semantic distance from the central object grows exponentially, the Euclidean space is not likely to encode such structures (*e.g.*, tree-like or graph structure). In hyperbolic space, the volume grows exponentially, thereby giving sufficient areas to embed the images. For visualisation, we have shrunk the images in the Euclidean diagram.

compared to the common practice of using Euclidean geometry or hypersphere [Tifrea et al., 2019; Gulcehre et al., 2019; Khrulkov et al., 2020; Cho et al., 2019]. However, working in hyperbolic spaces is not without difficulties as a result of its curved geometry and some operations (*i.e.*, addition, similarity measurement) are complex. For example, computing the mean of a set of points in hyperbolic spaces requires an iterative algorithm [Lou et al., 2020]. We take a step further to kernelize the embedding in hyperbolic spaces, such that the embedding can enjoy the rich structure of Hilbert spaces as well as simplify some operations involving hyperbolic data. We propose four kernel functions for hyperbolic spaces including the hyperbolic tangent kernel, hyperbolic RBF kernel, hyperbolic Laplace kernel, and hyperbolic binomial kernel, and evaluate the power of the proposed kernels on several tasks.

To this end, this thesis studies a challenging research problem - the embedding method for the visual data, *e.g.*, images. Aiming to create discriminative embeddings for the visual data, we investigate two possible solutions, *i.e.*, attention mechanism and geometry constraints. Those two advanced solutions solve the research question in two lines, one developing novel neural architectures to extract features, and another investigating the embedding spaces to understand the underlying distribution of data.

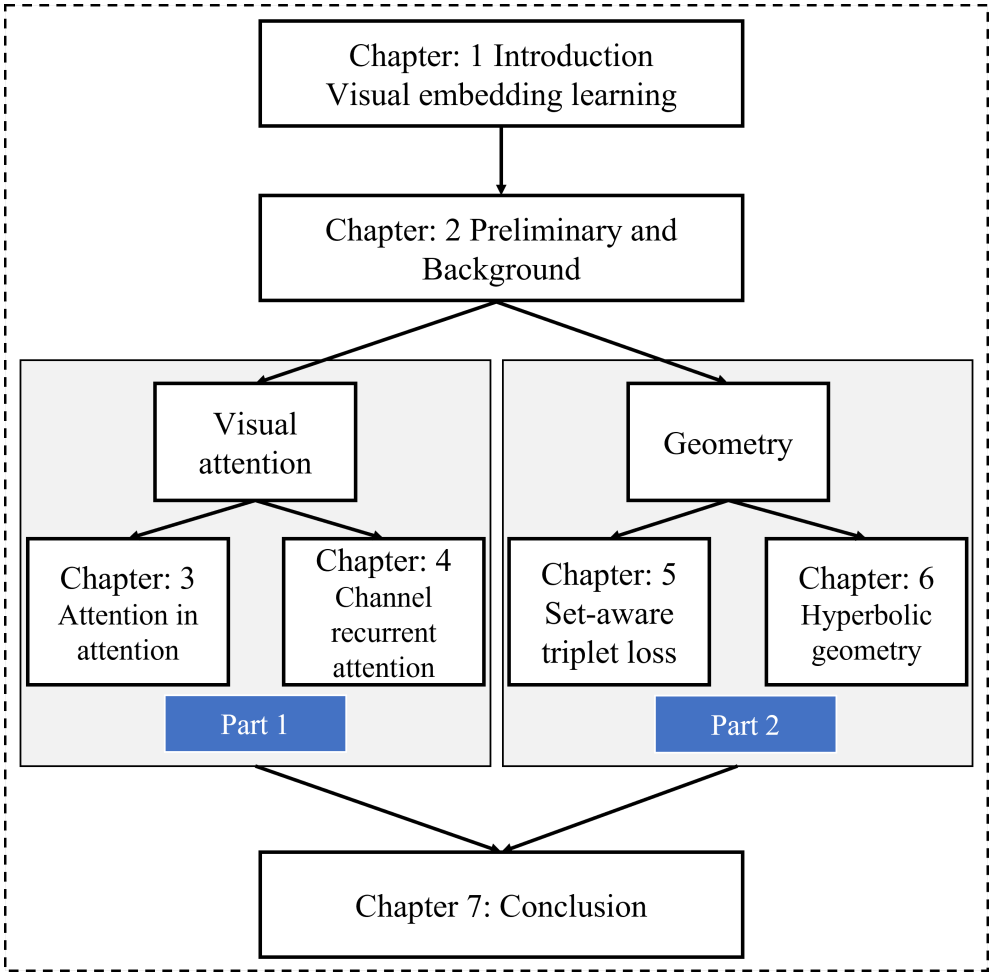


Figure 1.4: The outline of this Thesis.

1.2 Thesis Outline

The remaining chapters of the Thesis (see Fig. 1.4) are organised as follows:

Chapter 2 - Preliminary and Background: This chapter first defines the standard notation used across the Thesis, following the preliminary and background knowledge for this Thesis, *i.e.*, CNNs, RNNs, visual attention, geometry *etc*.

Chapter 3 - Attention in Attention Networks: In this chapter, we first propose the Attention in Attention (AiA) mechanism, which models the explicit interaction between global and local features to mitigate the misalignment issue of the feature map. We also generalise the AiA mechanism by making use of higher-order statistics, explicitly encoded by non-linear kernel mappings within the AiA framework, to generate an attention map. To showcase the effectiveness of the proposed attention mechanism, we perform extensive experiments on a person re-identification task. This chapter presents the contribution of the published works [Fang et al., 2021c,

2019].

Chapter 4 - Channel Recurrent Attention Networks: In this chapter, we present a full attention block to build a global receptive field to the feature map. The main attention unit, termed channel recurrent attention, identifies the attention map by jointly leveraging spatial and channel patterns via a recurrent neural network, such that the attention module can learn the spatial and channel information jointly. We show that the proposed attention can bring significant performance gain over both video/image person re-identification tasks. This chapter presents the contribution of the published work [Fang et al., 2020].

Chapter 5 - Set Augmented Triplet Loss: This chapter proposes to model the video clip as a set and instead studies the distance between sets in the triplet loss. In contrast to the distance between clip representations, the distance between sets considers every pair-wise distance in two sets, thereby making better use of frame features in sets. We further propose a hybrid distance metric between sets, tailored for the set-aware triplet loss. Thorough experiments on video person re-identification verify the advantage of the proposed method. This chapter presents the contribution of the published work [Fang et al., 2021b].

Chapter 6 - Kernel Methods in Hyperbolic Spaces: This chapter studies the kernelization for the hyperbolic representations. We first develop a family of pd kernels in the curved hyperbolic spaces, in conjunction with their theoretical analysis. The proposed kernels include the powerful universal ones, such as the hyperbolic RBF kernel. Then we comprehensively evaluate the representation power of the proposed kernels on various tasks including few-shot learning, zero-shot learning, person re-identification and knowledge distillation. This chapter presents the contribution of the work [Fang et al., 2021a].

Chapter 7 - Conclusion: In the final chapter of this dissertation, we summarise the overall contributions and discuss some future works, *i.e.*, open problems and directions.

Chapter 2

Preliminary and Background

This chapter introduces the preliminary and background materials that have been used in this Thesis. We first define the notation.

2.1 Notation

The notation used in this Thesis are fairly standard. Formally, we use $\mathbb{R}^n$, $\mathbb{R}^{h\times w}$, $\mathbb{R}^{c\times h\times w}$ and $\mathbb{R}^{t\times c\times h\times w}$ to denote the n-dimensional Euclidean space, the real matrix space (of size $h \times w$), and the image and video spaces, respectively. We also use $\mathbb{H}^n$ and $\mathcal{H}$ to denote the n-dimensional hyperbolic space and the Hilbert space. Throughout the Thesis, the matrices/tensors and vectors are denoted by bold capital letters (*e.g.*, $\boldsymbol{X}$) and bold lower-case letters (*e.g.*, $\boldsymbol{x}$), respectively. The transpose of a matrix (*e.g.*, $\boldsymbol{X}$) or a vector (*e.g.*, $\boldsymbol{x}$) is denoted by the superscript $\top$, *e.g.*, $\boldsymbol{X}^\top$ or $\boldsymbol{x}^\top$. We use $e(\cdot)$ and $\exp(\cdot)$ interchangeably as exponential function in this Thesis.

The symbol $\otimes$ and $\oplus$, represent the Hadamard product (*i.e.* element-wise multiplication) and element-wise summation. $\text{Sigmoid}(\cdot) : \mathbb{R} \to [0, 1], \text{Sigmoid}(x) := \frac{1}{1+\exp(-x)}$ is the sigmoid function. The softmax normalisation is defined as $\text{Softmax}(\cdot) : \mathbb{R} \to [0, 1]$, $\text{Softmax}(\boldsymbol{x}_i) := \frac{\exp(\boldsymbol{x}_i)}{\sum_{j=1}^{n}\exp(\boldsymbol{x}_j)}$. $\text{BN}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n, \text{BN}(\boldsymbol{x}) := \gamma\frac{\boldsymbol{x}-\text{E}[\boldsymbol{x}]}{\sqrt{\text{Var}[\boldsymbol{x}]}} + \beta$ and $\text{ReLU}(\cdot) : \mathbb{R} \to \mathbb{R}_{\geq 0}, \text{ReLU}(x) := \max(0, x)$ refer to batch normalisation and rectified linear unit. $\tanh(\cdot) : \mathbb{R} \to \mathbb{R}, \tanh(x) := \frac{\exp(2x)-1}{\exp(2x)+1}$ refers to the hyperbolic tangent function.

Please note that in later chapters, we may use additional notation that will be solely used in that particular chapter. This will be made clear in the corresponding chapter.

2.2 Convolutional Neural Networks

In recent years, convolutional neural networks (CNNs) have become the "workhorse" to encode the image/video data because of their rich capacity to extract features. CNNs are first proposed by LeCun in 1998 and able to achieve high precision to classify handwritten digits in binary images [LeCun et al., 1998]. Beyond the classification task, other fundamental CV tasks *i.e.*, semantic segmentation, object detection,

are also benefiting from CNNs [He et al., 2016, 2017; Fu et al., 2019a; Long et al., 2015]. A typical deep CNN (see Fig. 2.1) is stacked by different types of components, including a convolutional block, a pooling layer, a fully-connected (FC) layer, *etc*. By stacking many of these layers, deep CNNs can extract features of the raw images and subsequent feature maps by convolutional filters, whose parameters are learned by back-propagation [LeCun et al., 2015]. In the remainder of this section, we will briefly introduce the building blocks of CNNs.

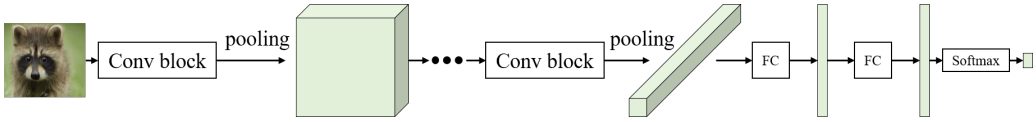


Figure 2.1: Illustration of a deep convolutional neural network.

The basic unit of the convolutional block is known as the 2D convolution operation. Formally, given an image $\boldsymbol{I}$ and a convolution kernel $\boldsymbol{c}$, the output $\boldsymbol{O}$ at position (i, j) can be calculated as:

$$\boldsymbol{O}(i,j) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} I(i-n, j-m)c(n,m). \tag{2.1}$$

Fig. 2.2 provides a toy example of a standard 2D convolution operation and output $\boldsymbol{O}(1,1)$ is obtained as $1 \times 5 - 2 \times 1 - 3 \times 1 - 2 \times 1 - 3 \times 1 = -5$.

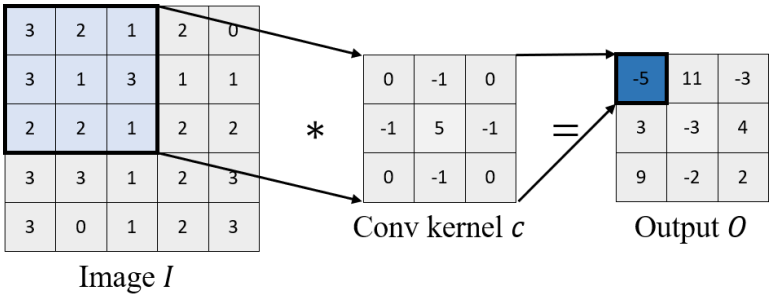


Figure 2.2: A standard 2D convolution operation. The symbol $*$ indicates convolution.

Having the convolution operator at hand, different architectures of convolutional blocks are proposed to learn a better image embedding. The vanilla convolutional block simply stacks several convolutional layers, illustrated in Fig. 2.3(a). Suppose the input of a two layer convolutional block is denoted by $\boldsymbol{X}$, then the output, $\boldsymbol{Y}$, can be calculated by:

$$\boldsymbol{Y} = \mathrm{Conv}\big(\mathrm{ReLU}(\mathrm{Conv}(\boldsymbol{X}))\big) = f(\boldsymbol{X}). \tag{2.2}$$

Here we use f to indicate a two layer convolutional block as shown in Fig. 2.3(a). In the convolutional block, the ReLU function provides a non-linearity to the CNN. Thus deep CNNs can theoretically regress complex non-linear functions. The deep network composed of such an architecture will suffer from the issue of vanishing

gradients during training. To mitigate this issue, other variants, namely, residual connection blocks (see Fig. 2.3(b)) and dense connection blocks (see Fig. 2.3(c)), are developed. The residual connection block and dense connection block are formulated as:

$$\boldsymbol{Y} = f(\boldsymbol{X}) + \boldsymbol{X} \tag{2.3}$$

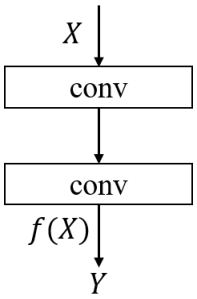
and

$$\boldsymbol{Y} = \text{Concat}\big(f(\boldsymbol{X}), \tilde{\boldsymbol{X}}, \boldsymbol{X}\big), \tag{2.4}$$

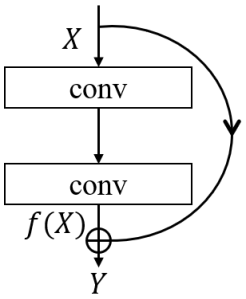
where $\tilde{\boldsymbol{X}} = \text{ReLU}(\text{Conv}(\boldsymbol{X}))$ indicates the feature map in the intermediate layer of the convolutional block.

These convolutional blocks avoid the issue of vanishing gradients by reusing preceding feature maps ($\boldsymbol{X}$ in Fig. 2.3). Also, the inception block (see Fig. 2.3(d)) increases the learning capacity of CNNs by means of using the various receptive fields of convolutional kernels in parallel, given by:

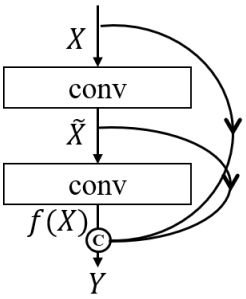
$$\boldsymbol{Y} = \text{Concat}\big(f_1(\boldsymbol{X}), \ldots, f_4(\boldsymbol{X})\big). \tag{2.5}$$



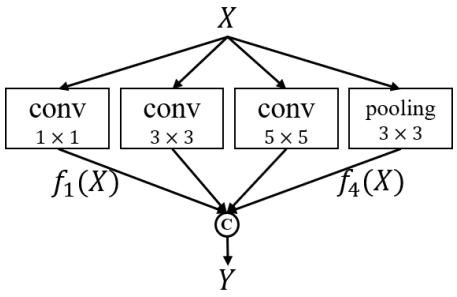
(a) Vanilla convolutional block [Krizhevsky et al., 2012]



(b) Residual connection block[He et al., 2016]



(c) Dense connection block [Huang et al., 2017]



(d) Inception block [Szegedy et al., 2015]

Figure 2.3: Illustration of different convolutional blocks.

2.3 Recurrent Neural Networks

In Chapter 4, we use a recurrent neural network (RNN) as a building block in the attention module. In this section, we first introduce the formulation of the RNN briefly, then describe an advanced variant of RNNs used in this Thesis, *i.e.*, long-short term memory.

In contrast to feed-forward neural networks, whose output is determined by the current input, the output of RNNs is decided by both the current input and the previous input. This is achieved by the recurrent operation in RNNs as illustrated in Fig. 2.4, and the hidden state in RNNs can memorize the information from the previous input. To be specific, the current hidden state is obtained by:

$$\boldsymbol{h}_t = \tanh\left(\boldsymbol{W}_h^\top \boldsymbol{h}_{t-1} + \boldsymbol{W}_x^\top \boldsymbol{x}_t\right). \tag{2.6}$$

Here $\boldsymbol{W}_h$ and $\boldsymbol{W}_x$ are learnable parameters in RNNs. Note that we omit the bias for simplicity of formulation. Then the output at state $\boldsymbol{h}_t$ is:

$$\boldsymbol{y}_t = \boldsymbol{W}_y^\top \boldsymbol{h}_t, \tag{2.7}$$

where $\boldsymbol{W}_y$ is the parameter for the output state.

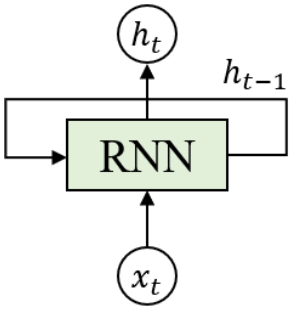


Figure 2.4: Illustration of a recurrent neural network.

Although superior performance in many sequence learning applications have been shown, RNNs have the issue of vanishing/exploding gradients when processing the long sequences of data. These issues are addressed by the long-short term memory (LSTM). In the LSTM, the memory allows the model to remember information for a long period of time, thereby building long-range interactions in the long data sequences. A standard LSTM unit are composed of three gates, namely an input gate, forget gate and output gate. Fig. 2.5 illustrates the architecture of an LSTM. The input gate can learn from the input data and discover what information can be stored in the memory. The forget gate can decide to discard the useless information by comparing the previous state (*i.e.*, $\boldsymbol{h}_{t-1}$) and current input (*i.e.*, $\boldsymbol{x}_t$). The output of the LSTM is produced by the output gate, which also considers both the previous feature and current feature. The formulation of the gates and states are given by:

- Forget gate: $\boldsymbol{f}_t = \text{Sigmoid}\left(\boldsymbol{W}_{fh}^\top \boldsymbol{h}_{t-1} + \boldsymbol{W}_{fx}^\top \boldsymbol{x}_t\right)$

- Input gate: $\boldsymbol{i}_t = \text{Sigmoid}\big(\boldsymbol{W}_{ih}^\top \boldsymbol{h}_{t-1} + \boldsymbol{W}_{ix}^\top \boldsymbol{x}_t\big)$
- Memory state: $\tilde{\boldsymbol{c}}_t = \tanh\big(\boldsymbol{W}_{ch}^\top \boldsymbol{h}_{t-1} + \boldsymbol{W}_{cx}^\top \boldsymbol{x}_t\big)$
- Cell state: $\boldsymbol{c}_t = (\boldsymbol{f}_t \otimes \boldsymbol{c}_{t-1}) \oplus (\boldsymbol{i}_t \otimes \tilde{\boldsymbol{c}}_t)$
- Output gate: $\boldsymbol{o}_t = \text{Sigmoid}\big(\boldsymbol{W}_{oh}^\top \boldsymbol{h}_{t-1} + \boldsymbol{W}_{ox}^\top \boldsymbol{x}_t\big)$
- Output state: $\boldsymbol{h}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{c}_t)$

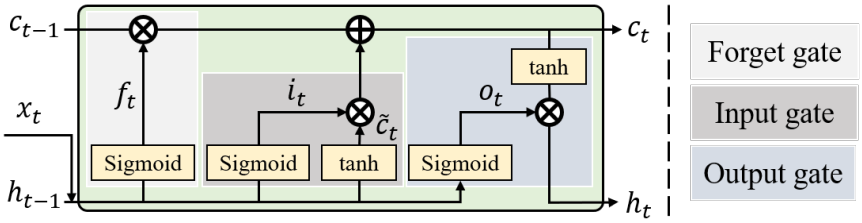


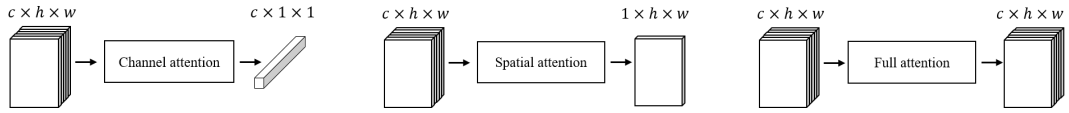
Figure 2.5: The architecture of a long-short term memory.

The output of the LSTM depends on the tasks at hand. For example, the work in [Liu et al., 2016] uses $\boldsymbol{h}_t$ as the final output. In contrast, our work in Chapter 4 utilizes all states (*i.e.*, $[\boldsymbol{h}_1, \ldots, \boldsymbol{h}_t]$) as the output of the LSTM.

2.4 Attention Mechanism

Attention mechanisms, inspired by the human sensing process, have been studied extensively in Natural Language Processing [Vaswani et al., 2017] and Computer Vision [Hu et al., 2018]. Self-attention is first proposed in [Vaswani et al., 2017] and achieves a breakthrough in machine translation, showing its superior performance over the RNN. Thereafter, several visual applications have incorporated this attention module in their formulation, *e.g.*, image classification [Wang et al., 2018b], scene segmentation [Fu et al., 2019a], image captioning [Huang et al., 2019] as well as video person re-ID [Li et al., 2019a]. In this thesis, we are particularly in soft attention since the attention network can adjust the weight to the target. Here we introduce some background knowledge of soft attention modules.

Generally speaking, the soft attention mechanism can be grouped into three categories, *i.e.*, channel attention (see Fig. 2.6(a)), spatial attention (see Fig. 2.6(b)) and full attention (see Fig. 2.6(c)), according to the size of the generated attention maps. The channel attention learns the non-linear transformation of the channel descriptor as attention weights and re-weights each slice of the feature map. In contrast, spatial attention can build the interaction spatially and weigh each patch descriptor of the feature map. Full attention, which produces attention weights per every element of the feature map, not only learns the channel pattern, but also preserves the structural information of the feature map. We then introduce some seminal instantiations of the above three categories.



(a) Channel attention (b) Spatial attention (c) Full attention

Figure 2.6: Categories of attention mechanisms.

In the squeeze and excitation (SE) block, the channel descriptor is obtained by the global average pooling. Then two fully connected layers are used to fully capture channel dependencies [Hu et al., 2018]. As illustrated in Fig. 2.7, the SE block can be formulated as:

$$\boldsymbol{X}^z = \text{Sigmoid}\Big(\sigma\big(\xi\big(\text{GAP}(\boldsymbol{M})\big)\big)\Big) \otimes \boldsymbol{X}. \tag{2.8}$$

Here, GAP indicates the global average pooling. σ and ξ are gating functions Hu et al. [2018].

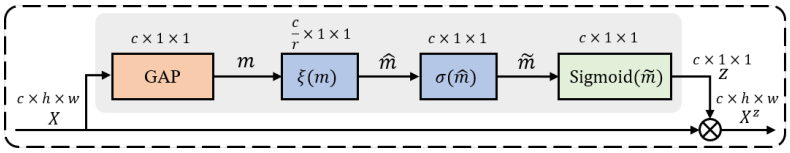


Figure 2.7: Squeeze and excitation block.

In [Wang et al., 2017], the spatial attention block (see Fig. 2.8) first aggregates all elements per patch feature in the feature map, then builds spatial interactions via convolution kernels in the attention map. Its formulation is given by:

$$\boldsymbol{X}^z = \text{Sigmoid}\Big(\delta\big(\text{Agg}(\boldsymbol{X})\big)\Big) \otimes \boldsymbol{X}, \tag{2.9}$$

where Agg and δ indicate the aggregation function and convolution function, respectively.

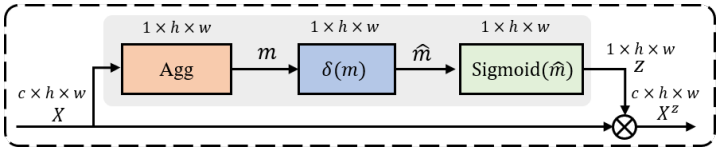


Figure 2.8: Spatial attention block.

In the full attention mechanism, the attention block can generate an attention map with the same size as its input feature map, thereby weighing every element of the input feature map. In [Wang et al., 2018a], the fully attentional block (FAB), as shown in Fig. 2.9, is the first proposed full attention, given by:

$$\boldsymbol{X}^z = \text{Sigmoid}\Big(\sigma\big(\xi\big(\boldsymbol{X}\big)\big)\Big). \tag{2.10}$$

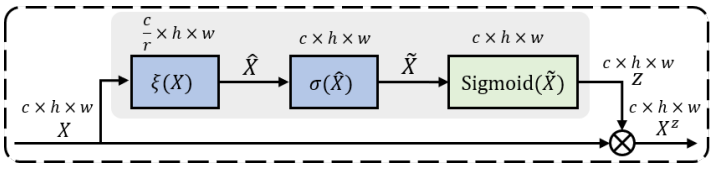


Figure 2.9: Fully attentional block.

As compared to the SE block (see Eq. (2.8)), the FAB (see Eq. (2.10)) only learns the pattern of patch descriptors, instead of the channel descriptor. However, such a pipeline helps the attention block preserve the spatial structure information.

Beyond the categories of attention mechanisms introduced above, self-attention or intra-attention can generate a feature at a position by relating features of different positions in a sequence. As a basic operation in self-attention, the non-local mean (see Fig. 2.10(a)) is defined by:

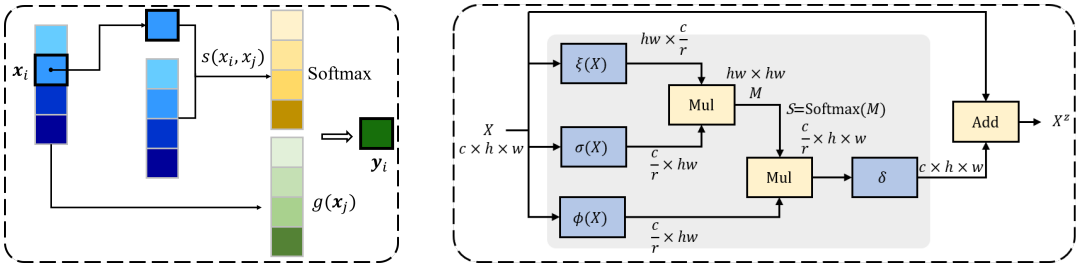
$$\boldsymbol{y}_i = \frac{1}{N} \sum_{j=1}^{N} s(\boldsymbol{x}_i, \boldsymbol{x}_j) g(\boldsymbol{x}_j). \tag{2.11}$$

Here $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the input feature and output feature at position i. $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ computes the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $g(\boldsymbol{x}_j)$ encodes the feature of $\boldsymbol{x}_j$. N is the number of features in the sequence. This operation can build long-range dependencies in the sequence.

Then the non-local (NL) block [Wang et al., 2018b] is further developed and formulated as follows:

$$\boldsymbol{X}^z = \delta\Big(\text{Softmax}\big(\xi(\boldsymbol{X})^\top \sigma(\boldsymbol{X})\big)\phi(\boldsymbol{X})\Big) \oplus \boldsymbol{X}, \tag{2.12}$$

where δ, ξ, σ and ϕ are non-linear transformations. $\oplus$ $\boldsymbol{X}$ indicates a residual connection.



(a) Non-local mean (b) Non-local block

Figure 2.10: Self-attention mechanism.

The existing attention mechanism has its limitations. This thesis develops two neural attentions to efficiently utilise the information hidden in the feature maps, thereby creating rich embeddings for the visual data.

2.5 Set Theory and Metrics

A function $f(\cdot)$, which maps its domain $\mathcal{X}$ to its range $\mathcal{Y}$, is considered a function of sets if it is permutation invariant to the order of elements in the input. In other words, given a set (*i.e.*, $\boldsymbol{X}$) as input, the function $f(\cdot)$ holds that $f(\boldsymbol{X}) = f(\boldsymbol{P}\boldsymbol{X})$ for any permutation matrix $\boldsymbol{P}$. In this case, the domain of $f(\cdot)$ is the power set of $\boldsymbol{X}$, *i.e.*, $\mathcal{X} = \wp(\boldsymbol{X})$.

Let $(\boldsymbol{X}, d)$ be a metric space. The distance between two nonempty sets $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\wp(\boldsymbol{X})$ (*i.e.*, $D : \wp(\boldsymbol{X}) \setminus \emptyset \times \wp(\boldsymbol{X}) \setminus \emptyset \to \mathbb{R}$) measures the similarity of two sets. The ordinary distance between sets (see Fig. 2.11(a)) is defined as:

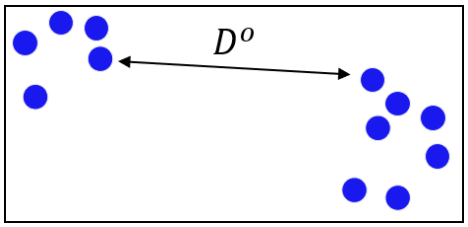
$$D^o(\boldsymbol{A}, \boldsymbol{B}) = \inf_{\boldsymbol{a}\in\boldsymbol{A}, \boldsymbol{b}\in\boldsymbol{B}} d(\boldsymbol{a}, \boldsymbol{b}), \tag{2.13}$$

where inf denotes the infimum function. The ordinary distance metric could be interpreted as the minimum pair-wise distance between two sets.

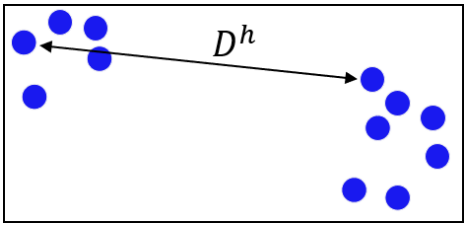
Another well-known set distance metric is the Hausdorff distance, which is defined as:

$$\begin{aligned} D^h(\boldsymbol{A}, \boldsymbol{B}) &= \max\left\{ \sup_{\boldsymbol{a}\in\boldsymbol{A}} d(\boldsymbol{a}, \boldsymbol{B}), \sup_{\boldsymbol{b}\in\boldsymbol{B}} d(\boldsymbol{b}, \boldsymbol{A}) \right\} \\ &= \max\left\{ \sup_{\boldsymbol{a}\in\boldsymbol{A}} \inf_{\boldsymbol{b}\in\boldsymbol{B}} d(\boldsymbol{a}, \boldsymbol{b}), \sup_{\boldsymbol{b}\in\boldsymbol{B}} \inf_{\boldsymbol{a}\in\boldsymbol{A}} d(\boldsymbol{a}, \boldsymbol{b}) \right\}, \end{aligned} \tag{2.14}$$

where sup represents the supremum function. As shown in Fig. 2.11(b), the geometrical interpretation of the Hausdorff distance can be understood as the greatest of all the distances from an element in one set to the closest element in the other set.



(a) Ordinary distance metric.



(b) Hausdorff distance metric.

Figure 2.11: Geometry interpretation of the set distance. (a) and (b) represent the ordinary distance metric and Hausdorff distance metric, respectively.

In Chapter 5 of this thesis, the video clip can be modelled as set since the video embedding is compact and invariant to the frame orders. On top of the well-defined metrics of sets, we also proposed a new metrics to optimise the hard frames in the video clip.

2.6 Hyperbolic Geometry

An n-dimensional hyperbolic space $\mathbb{H}^n$ is a Riemannian manifold with a constant negative curvature [Absil et al., 2007]. The Poincaré ball is a model of n-dimensional hyperbolic geometry in which all points are embedded within an n-dimensional sphere (or inside a circle in the 2D case which is called the Poincaré disk model). Formally, the Poincaré ball model, with curvature $-c, (c > 0)$, is defined as a manifold $\mathbb{D}_c^n = \{\boldsymbol{z} \in \mathbb{R}^n : c\|\boldsymbol{z}\| < 1\}$, with the Riemannian metric $g_c^{\mathbb{D}}(\boldsymbol{z}) = \lambda_c^2(\boldsymbol{z}) \cdot g^E$, in which $\lambda_c(\boldsymbol{z})$ is the conformal factor, defined as $\frac{2}{1-c\|\boldsymbol{z}\|^2}$, and $g^E = \mathbf{I}_n$ is the Euclidean metric tensor. Furthermore and to facilitate vector operations, the Möbius gyrovector space may come in handy. The Möbius addition for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$ is defined as:

$$\boldsymbol{z}_i \oplus_c \boldsymbol{z}_j = \frac{(1 + 2c\langle \boldsymbol{z}_i, \boldsymbol{z}_j\rangle + c\|\boldsymbol{z}_j\|^2)\boldsymbol{z}_i + (1 - c\|\boldsymbol{z}_i\|^2)\boldsymbol{z}_j}{1 + 2c\langle \boldsymbol{z}_i, \boldsymbol{z}_j\rangle + c^2\|\boldsymbol{z}_i\|^2\|\boldsymbol{z}_j\|^2}. \tag{2.15}$$

The geodesic distance on $\mathbb{D}_c^n$ is:

$$d_c(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{2}{\sqrt{c}}\tanh^{-1}(\sqrt{c}\|-\boldsymbol{z}_i \oplus_c \boldsymbol{z}_j\|). \tag{2.16}$$

For a point $\boldsymbol{z} \in \mathbb{D}_c^n$, the tangent space at $\boldsymbol{z}$, denoted by $T_{\boldsymbol{z}}\mathbb{D}_c^n$, is an inner product space, which contains the tangent vector with all possible directions at $\boldsymbol{z}$. The Riemannian metric $g_c^{\mathbb{D}}$ at point $\boldsymbol{z}$ is a positive definite symmetric bilinear function on $T_{\boldsymbol{z}}\mathbb{D}_c^n$ as $g_c^{\mathbb{D}}(\boldsymbol{z}) : (T_{\boldsymbol{z}}\mathbb{D}_c^n \times T_{\boldsymbol{z}}\mathbb{D}_c^n) \to \mathbb{R}$. The exponential map provides a way to project a point $\boldsymbol{p} \in T_{\boldsymbol{z}}\mathbb{D}_c^n$ to the Poincaré ball $\mathbb{D}_c^n$, as follows:

$$\Gamma_{\boldsymbol{z}}(\boldsymbol{p}) = \boldsymbol{z} \oplus_c \big(\tanh(\sqrt{c}\frac{\lambda_c(\boldsymbol{z}) \cdot \|\boldsymbol{p}\|}{2})\frac{\boldsymbol{p}}{\sqrt{c}\|\boldsymbol{p}\|}\big). \tag{2.17}$$

The inverse process is termed logarithm map, which projects a point $\boldsymbol{q} \in \mathbb{D}_c^n$, to the tangent plane of $\boldsymbol{z}$, given as:

$$\mathrm{Y}_{\boldsymbol{z}}(\boldsymbol{q}) = \frac{2}{\sqrt{c}\lambda_c(\boldsymbol{z})}\tanh^{-1}(\sqrt{c}\|-\boldsymbol{z} \oplus_c \boldsymbol{q}\|)\frac{-\boldsymbol{z} \oplus_c \boldsymbol{q}}{\|-\boldsymbol{z} \oplus_c \boldsymbol{q}\|}. \tag{2.18}$$

Note that $\mathrm{Y}_{\boldsymbol{z}}\big(\Gamma_{\boldsymbol{z}}(\boldsymbol{p})\big) = \boldsymbol{p} \in T_{\boldsymbol{z}}\mathbb{D}_c^n$. Both the exponential and the logarithm maps are injective functions. In the Chapter 6, we leverage the Euclidean space in the identity tangent plane to define the kernels for hyperbolic spaces.

2.7 Person Re-Identification

Person retrieval, also known as person re-identification (re-ID) [1] , has attracted an increasing amount of interest in the Computer Vision (CV) community due to its

[1]In this paper, we will use the terms "person retrieval", "person re-identification" and "person re-ID" interchangeably.

challenging nature and industrial prospects. The task of a person retrieval machine can be characterised as follows: given an image of a specific person, the machine should retrieve all images with the same identity, from a gallery.

Early works in the field of person re-ID relied mostly on designing hand-crafted feature representations [Gheissari et al., 2006] or learning latent spaces [Yi et al., 2014]. We refer interested readers to [Gong et al., 2014] for more details regarding traditional methods. Convolutional Neural Networks (CNN) are currently the method of choice for representation learning, delivering state-of-the-art results in person re-ID. In [Yi et al., 2014], Yi *et al*. proposed a unified framework for feature and similarity learning using Siamese networks [Roy et al., 2019]. Multi-level similarities are employed in [Wang et al., 2018c] to make more reliable decisions. Xiao *et al*. trained a model across multiple datasets [Xiao et al., 2016] and used domain guided dropouts to mute domain-irrelevant neurons to learn robust features. Structural constraints (*e.g*., orthogonality, geometry) on the embedding layer [Sun et al., 2017; Bai et al., 2017] have also been shown to learn robust person features and achieve superior results on the person re-ID task.

In deep metric learning, some works also concentrate on developing the ranking loss in formulation [Chen et al., 2017a] or mining strategies [Wang et al., 2018a]. Considering the camera distribution, the spatial and temporal signal is further adopted to eliminate the irrelevant, thereby improving the ranking results [Wang et al., 2019]. Besides the single image presentation, video data also introduces temporal cues to encode a compact and robust video presentation of a pedestrian [Yan et al., 2016; McLaughlin et al., 2016]. In the early work of [McLaughlin et al., 2016], the clip-level feature is fused by using a simple yet effective temporal pooling technique. A Recurrent Neural Network (RNN) is further employed to leverage the temporal information, and fuse the frame-level features [Yan et al., 2016]. Temporal attention mechanisms predict the importance of each frame feature and uses weighted sum to fuse them [Li et al., 2019a]

In person re-ID, the person misalignment [Suh et al., 2018] and background biases [Tian et al., 2018] obstruct learning of a robust feature representation. Visual attention mechanisms aim at emphasising informative regions for identification, while depreciating harmful ones (*e.g*., background and occluded regions). The spatial transformer network (STN) [Jaderberg et al., 2015], a binary hard attention, was used in [Li et al., 2017] to localise the latent body parts of a human. Liu *et al*. [Liu et al., 2016] proposed a Comparative Attention Network (CAN), which repeatedly localises discriminative parts and compares different local regions of person pairs. In Harmonious Attention Convolutional Neural Network (HA-CNN) [Li et al., 2018c], hard region-level attention and soft pixel-level attention are learned in a unified attention block. Wang *et al*. [Wang et al., 2018a] considered both the channel-wise and spatial-wise attention in a Fully Attentional Block (FAB), where the channel information is re-calibrated while the spatial structural information is also preserved. Besides aligning the feature maps, Dual ATtention Matching network (DuATM) [Si et al., 2018] also calibrates the features by matching the intra-feature sequence. In [Tay et al., 2019], the attention learning is driven by person attribute prediction. In the video person re-ID task,

attention mechanisms have also been employed in temporal modelling. For example, the attention weights for each frame is generated by temporal convolution [Gao and Nevatia, 2018]. The recent works continue to mine more spatial and temporal information via spatial-temporal attention [Fu et al., 2019b].

2.8 Summary

In summary, this chapter provides the notation, preliminary and background materials used in this Thesis. In the next chapters, we present how we improve the quality of the image embedding under the concern of visual attention and geometry constraint.

Part I

Visual Embedding Learning: Attention

Chapter 3

Attention in Attention Networks

In this chapter, we first propose a novel Attention in Attention (AiA) mechanism. The AiA mechanism models the capacity of building inter-dependencies among the local and global features by the interaction of inner and outer attention modules. Besides a vanilla AiA module, termed linear attention with AiA, two non-linear counterparts, namely, second-order polynomial attention and Gaussian attention, are also proposed to utilise the non-linear properties of the input features explicitly, via the second-order polynomial kernel and Gaussian kernel approximation. The deep convolutional neural network, equipped with the proposed AiA blocks, is referred to as Attention in Attention Network (AiA-Net). The AiA-Net learns to extract a discriminative pedestrian representation, which combines complementary person appearance and corresponding part features. Extensive ablation studies verify the effectiveness of the AiA mechanism and the use of non-linear features hidden in the feature map for attention design. Furthermore, our approach outperforms current state-of-the-art by a considerable margin across a number of benchmarks. In addition, state-of-the-art performance is also achieved in the video person retrieval task with the assistance of the proposed AiA blocks. This chapter is based on our published works [Fang et al., 2019, 2021c].

3.1 Introduction

In this chapter, we study the *Attention in Attention Networks* for encoding rich pedestrian representation on the person re-ID task.

In the real practice, there are quite a few factors that can lead to an unreliable person retrieval system, making the re-ID task daunting and challenging. For example, *misalignment* caused by spatial nuances in the person bounding box (*e.g.*, movements of body parts) can negatively affect a re-ID system [Su et al., 2017]. That is, the location of the person's body, and its parts, with respect to a reference frame, can be easily displaced due to the change in body orientation, pose, clothing, *etc*. This, in turn, causes mismatching of features during training and testing, leading to inaccurate re-identification. Much effort has been made into studying and addressing these difficulties [Suh et al., 2018; Sun et al., 2018; Chen et al., 2017b; Li et al., 2019b]. However, it still remains an open problem and calls for further study to learn a robust

and discriminative representation of the person(s).

In general, solutions to address the misalignment within a person bounding box can be broadly categorised into *human pose*-based, *human attributes*-based as well as *visual attention*-based methods. In recent years, several attempts that rely on human pose estimation have been undertaken to address this in [Su et al., 2017; Saquib Sarfraz et al., 2018; Li et al., 2019b]. These algorithms employ additional estimator networks that provide the baseline-network with complementary cues to learn a superior embedding space, thereby outperforming the baseline-network. Other solutions benefit from person attributes [Su et al., 2016; Tay et al., 2019; Zhao et al., 2019], that are invariant to variations in human pose, light illumination, background clutter, spatial misalignment, *etc*. Such solutions aim at learning a robust person representation as described by the human attributes. Recently, visual attention-based solutions have received an overwhelming interest in the re-ID task, since it outperforms the pose-based/attribute-based models without the need of any additional pose detector or attribute estimator network.

The attention-based models, inspired by the human visual and attentive sensing processes, aim to localise the discriminative regions within a person bounding box [Li et al., 2018c; Wang et al., 2018a; Qian et al., 2019]. The inherent attention module (*e.g.*, hard attention [Jaderberg et al., 2015; Li et al., 2018c] or soft attention [Hu et al., 2018; Wang et al., 2018a]) is designed to automatically select the informative parts of an image, and is trained in a weakly-supervised manner (*i.e.*, no explicit labelling information is given to identify the areas to attend). In our preliminary study [Fang et al., 2019], we proposed the Attention in Attention (AiA) mechanism to model the explicit interaction between global and local features of the feature map, and used a bilinear mapping [Lin et al., 2015] that benefits from the second-order statistics to generate the attention values. In this chapter, we aim to generalise the AiA mechanism by making use of higher-order statistics, explicitly encoded by non-linear kernel mappings within the AiA framework, to generate the attention map.

Designing non-linear embeddings (*e.g.*, feature space of kernel machines) by making use of the geometry of Reproducing Kernel Hilbert Spaces (RKHS) dates back to the celebrated work of Vapnik [Vapnik, 2000]. The machinery of RKHS is rich enough to even handle infinite-dimensional representations (through the use of the well-known kernel trick). Also, recent studies show that kernel methods along deep neural networks (DNNs) would help to attain rich models [Xu et al., 2020; Peng et al., 2019; Jayasumana et al., 2020; Cui et al., 2017]. This inspires us to benefit from the theory of RKHS and its approximations [Vedaldi and Zisserman, 2012; Rahimi and Recht, 2008] to design attention modules for DNNs. To the best of our knowledge, this is the first attempt where an attention mechanism is implemented from an RKHS perspective.

This chapter generalises the AiA framework by employing explicit non-linear mappings in RKHS to generate attention value(s). The AiA framework consists of an *outer attention* block that encompasses an *inner attention* block such that the inner block is tasked to determine the discriminative regions of the feature map where the

outer attention block should focus (See Fig. 3.1 and Fig. 3.2 for a conceptual diagram for AiA structure). Therefore, the AiA block models channel-wise inter-dependencies between the global and local features, while preserving the spatial structural information of the input feature map, in a unified block. Besides a vanilla AiA block, which only exploits linear features of its input feature map, we further propose and develop two non-linear versions of AiA, with each respectively using the second-order polynomial and Gaussian kernels of the feature map along the channels. The intuition behind adopting the features in RKHSs is that such features can benefit from the highly discriminative capacity of high- or infinite-dimensional spaces, thereby helping the attention block to focus on more discriminative areas within the feature maps. Even though functions in RKHS can approximate any function, the operational capacity is limited due to computationally expensive kernel operations on the whole training data [Rahimi and Recht, 2008; Vedaldi and Zisserman, 2012]. In this chapter, we further propose to alleviate these constraints by relying on advanced kernel estimation techniques. More specifically, the second-order polynomial kernel is modelled by a bilinear mapping [Lin et al., 2015], while the Gaussian kernel is estimated by random Fourier features [Rahimi and Recht, 2008]. By such transformations, learnable parameters are avoided in the non-linear transformation, leading to being optimised easily. We further propose a computationally efficient version of the attention block without the use of the inner attention block. Table 3.1 summarises the proposed attention modules.

The **contribution** of this chapter can be summarised as follows:

- We formulate a generalised Attention in Attention (AiA) mechanism, where the attention map is generated by the interaction between the inner and outer attention modules. This indeed results in modelling inter-dependencies between global and local features of its input feature map, while maintaining the spatial structural information.

- We further develop kernelized versions of the AiA block, namely, *second-order polynomial attention* (SoP-attention) and *Gaussian attention* (Gau-attention), by estimating the second-order polynomial and Gaussian features of the input feature map respectively. Furthermore, we employ advanced kernel estimation techniques to reduce the computational cost of the kernel matrix.

- We propose a novel deep architecture using the AiA block, creating our Attention in Attention Network (AiA-Net), for the task of person retrieval. This AiA-Net extracts complementary person appearance and part features for discriminative person representation learning.

- Extensive experiments performed on large scale standard benchmark datasets including **CUHK03** [Li et al., 2014], **Market-1501** [Zheng et al., 2015], **DukeMTMC-reID** [Ristani et al., 2016] and **MSMT17** [Wei et al., 2018], as well as a small scale benchmark dataset (*e.g.*, **CHUK01** [Li et al., 2012]), show that our approach outperforms the current state-of-the-art methods by a considerable margin in

terms of mAP and R-1 metrics. Meanwhile, we also conduct extensive ablation studies that verify the superiority of the AiA mechanism and the utility of the non-linear features.

- In the video person retrieval setting, our deep network (*e.g.*, AiA-Net-V) also achieves state-of-the-art results on the popular video benchmark dataset, *i.e.*, **MARS** [Zheng et al., 2016].

Additionally, we find an interesting observation that the Gau-attention mechanism is empirically superior to the SoP-attention, in terms of accuracy, computational cost as well as number of parameters. For instance, on the CUHK03 dataset, the mAP/R-1 of AiA-Net with Gau-attention is 77.6%/80.6% as compared to 77.0%/79.4% for SoP-attention, while the computational complexity/number of parameters of Gau-attention are three times smaller than that of SoP-attention (*e.g.*, $0.044 \times 10^9/0.58 \times 10^6$ vs. $0.117 \times 10^9/1.79 \times 10^6$).

3.2 Related Work

In this part, we review the kernel estimation techniques used in this Chapter, namely, the bilinear mapping and kernel approximation.

Bilinear Mapping. Bilinear mappings and models have been widely considered as a generalisation of their linear counterparts. Some prime examples are bilinear classifiers [Pirsiavash et al., 2009], bilinear pooling [Gao et al., 2016] and bilinear CNNs [Lin et al., 2015] with applications in visual question answering, fine-grained image recognition, texture classification to name a few. Related to our work, the bilinear pooling [Gao et al., 2016], is first introduced to model local pairwise feature interactions for fine-grained recognition applications and its representation power is also enhanced by normalising the higher order statistics [Lin and Maji, 2017]. Thereafter, Liu *et al*. [Liu et al., 2017a] proposed a compact form of the bilinear operation to pool a high-dimensional feature representation for the task of person re-ID. In [Ustinova et al., 2015], Ustinova *et al*. proposed a patch-based multi-regional bilinear pooling to account for the geometric misalignment problem between the person bounding boxes. Recently, Suh *et al*. [Suh et al., 2018] used a part-aligned representation to mitigate the misalignment problem by fusing the appearance and part feature maps in a bilinear pooling layer. To avoid a quadratic computational cost, the bilinear features are estimated by a compact representation, *e.g.*, the tensor sketch [Gao et al., 2016], or the Hadamard product of low-rank bilinear pooling [Kim et al., 2017].

Kernel Approximation. Feature embedding in RKHS has been commonly used in many machine learning methods, such as, non-linear SVM, kernel PCA and unsupervised learning [Hofmann et al., 2008]. Nonetheless, training such kernel machines is N times slower than the vanilla linear machine, where N is the size of the training data [Vedaldi and Zisserman, 2012]. This results in poor scalability of the non-linear kernel based algorithms as the feature learning operates on the kernel matrix, leading to the birth of accelerated kernel machines [Vedaldi and Zisserman, 2012; Rahimi and

Table 3.1: Summary of the proposed attention modules. Here, we use feature vectors (*e.g.*, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^c$) in the attention formulation instead of the tensor shaped feature map, for the purpose of simplicity. $\boldsymbol{z}$ denotes the attention mask, generated by the outer attention, $\boldsymbol{x}$ and $\varpi(\boldsymbol{m})$ denote the associated channel feature and inner attention mask. Refer to § 3.3 for more detail.

Kernel attention	Kernel formulation:$\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$	AiA formulation	Non-AiA formulation	Kernel approximation
Linear	$\boldsymbol{x}^\top \boldsymbol{y}$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\varphi(\boldsymbol{x})\big) \otimes \varpi(\boldsymbol{m})\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\varphi(\boldsymbol{x})\big)\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	Identity mapping
Second-order polynomial	$(\boldsymbol{x}^\top \boldsymbol{y} + c)^2$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\text{SoP}(\varphi(\boldsymbol{x}))\big) \otimes \varpi(\boldsymbol{m})\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\text{SoP}(\varphi(\boldsymbol{x}))\big)\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	$\text{SoP}(\cdot)$
Gaussian	$e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\text{Gau}(\varphi(\boldsymbol{x}))\big) \otimes \varpi(\boldsymbol{m})\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	$\boldsymbol{x}^z = \underbrace{\text{Sigmoid}\Big(\phi\big(\text{Gau}(\varphi(\boldsymbol{x}))\big)\Big)}_{\boldsymbol{z}} \otimes \boldsymbol{x}$	$\text{Gau}(\cdot)$

Recht, 2008]. One possible attempt is to approximate the high dimensional features by explicit mapping in RKHS, which is linearly scalable to the size of training samples [Joachims, 2006]. Maji *et al*. [Maji and Berg, 2009] approximated the intersection kernel by sparse projection. Shift-invariant kernels, *e.g*., Gaussian kernels, Cauchy kernels *etc*., are estimated by randomly mapping the feature in the Fourier domain of the associated kernel [Rahimi and Recht, 2008]. Approximation to a group of additive homogeneous by spectral analysis is studied by Vedaldi and Zisserman [Vedaldi and Zisserman, 2012], yielding closed-form solutions.

3.3 Attention in Attention Block

In this section, we will first describe the Attention in Attention (AiA) framework, which only uses the linear features of the input feature map in the attention block. This vanilla module is termed as *Linear attention with AiA*. Subsequently, its non-linear counterparts will be developed by making use of second-order polynomial and Gaussian kernels in the attention module. Each AiA module will be followed by a discussion of its simplified version (*i.e*., the attention w/o AiA). All proposed attention blocks are summarised in Table 3.1.

3.3.1 Linear Attention

Linear attention (Lin-attention) with AiA refers to the vanilla attention module under the AiA framework as it explicitly uses the linear features over the input feature map. This AiA mechanism models the inter-dependencies between the local and global features, whilst preserving the spatial structure of its input feature map. The architecture of Lin-attention with AiA is shown in Fig. 3.1.

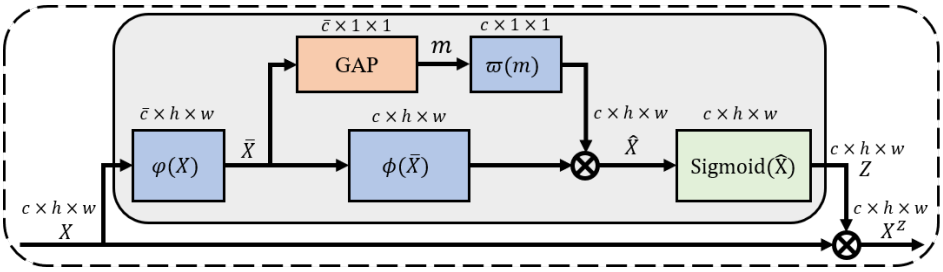


Figure 3.1: The structure of Linear attention with AiA. $\varphi(\cdot)$, $\phi(\cdot)$ and $\varpi(\cdot)$ are embedding functions. GAP indicates global average pooling. $\otimes$ indicates element-wise multiplication.

Let $\boldsymbol{X} \in \mathbb{R}^{c \times h \times w}$ be the input feature map, where c, h and w stand for the number of channels, height and width respectively. We denote the local feature at spatial location (i, j) as $\boldsymbol{x}_{ij} \in \mathbb{R}^c, i \in \{1, 2, \ldots, h\},\ j \in \{1, 2, \ldots, w\}$. The embedding function,

$\varphi(\cdot)$, first compresses $\boldsymbol{x}$[1] from the original channel dimension c to $\bar{c}$ as follows:

$$\bar{\boldsymbol{x}} = \varphi(\boldsymbol{x}), \tag{3.1}$$

where $\bar{\boldsymbol{x}} \in \mathbb{R}^{\bar{c}}$ with $\bar{c} = c/r$. The hyper-parameter r is the dimensionality reduction factor and its effect is discussed in § 3.5.4.

We note that even though $\bar{\boldsymbol{x}}$ encodes the channel features (*i.e.*, $\boldsymbol{x}$), it doesn't change the spatial location of body parts in the feature map. As a result the misalignment issue within the feature map still persists, which hinders the performance gain by the attention module. To address this shortcoming, we introduce the concept of "Attention in Attention", which aims to adaptively re-weight the channel feature responses by modelling the inter-dependency between the global and local features [2] (see Fig. 3.1). We first model the global feature of the feature map using a Global Average Pooling (GAP) layer, as follows:

$$\boldsymbol{m} = \frac{1}{hw}\sum_{i=1}^{hw}\bar{\boldsymbol{x}}_i, \tag{3.2}$$

where $\boldsymbol{m} \in \mathbb{R}^{\bar{c}}$. The inter-dependency between the embedded global feature $\boldsymbol{m}$ and each embedded local features $\bar{\boldsymbol{x}}$ is calculated as follows:

$$\hat{\boldsymbol{x}} = \varpi(\boldsymbol{m}) \otimes \phi(\bar{\boldsymbol{x}}), \tag{3.3}$$

where $\otimes$ denotes the element-wise multiplication, and $\varpi(\boldsymbol{m}), \phi(\bar{\boldsymbol{x}}) \in \mathbb{R}^{c}$. The embedding functions, $\varpi(\boldsymbol{m})$ and $\phi(\bar{\boldsymbol{x}})$, not only process the channel feature responses, but also recover the dimension of the channel from $\bar{c}$ to c (*i.e.*, the channel size of the input $\boldsymbol{x}$). Refer to Fig. 3.2 for a detailed pictorial representation of the aforementioned steps. Intuitively, $\varpi(\boldsymbol{m})$ acts as an inner attention and emphasises the local feature $\phi(\bar{\boldsymbol{x}})$ which are more correlated to the global feature $\varpi(\boldsymbol{m})$ via Eq. (3.3). In § 3.5.1, we give the details of embedding functions (*i.e.*, $\varphi(\cdot)$, $\varpi(\cdot)$ and $\phi(\cdot)$).

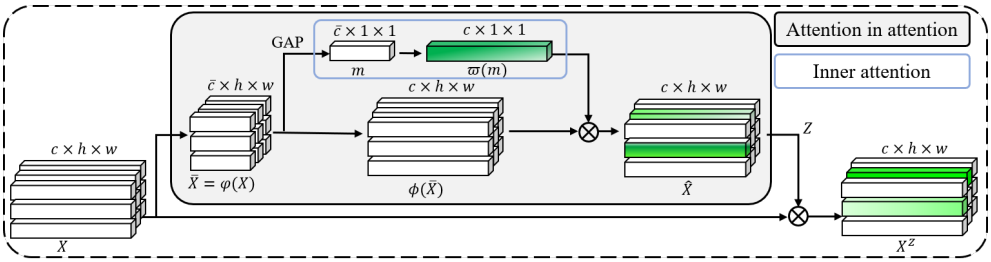


Figure 3.2: Details of the attention in attention (AiA) mechanism.

The final attention mask of input $\boldsymbol{x}$ is obtained by bounding $\hat{\boldsymbol{x}}$. In this chapter, we use Sigmoid$(\cdot)$ for this purpose (*i.e.*, $\boldsymbol{z} = \text{Sigmoid}(\hat{\boldsymbol{x}})$). This resulting vector will

[1]The subscripts have been omitted to avoid cluttering of notations.

[2]In this chapter, the physical meaning of the "global feature" and "local feature" indicates the "person's appearance feature" and "part feature".

act as an outer attention map, and emphasise/attenuate the significant/insignificant elements of its input feature vector $\boldsymbol{x}$ at the same spatial position as shown below:

$$\boldsymbol{x}^z = \boldsymbol{z} \otimes \boldsymbol{x}. \tag{3.4}$$

Remark 1 *The operations described by Eq.* (3.2) *and* (3.3) *resemble the Squeeze-and-Excitation (SE) Networks [Hu et al., 2018]. However, there is an essential difference. The SE Network first squeezes the information in each channel to a scalar which is then used to scale all the elements of a channel uniformly. In contrast, we use the channel attention as an inner attention module to perform significance weighting of the attention-dependent feature map (*e.g., *$\phi(\bar{\boldsymbol{x}})$*) in AiA and produce the output feature map (*e.g., *$\hat{\boldsymbol{X}}$*). Subsequently, our AiA module will further process $\hat{\boldsymbol{X}}$ to generate the final attention map (*e.g., *$\boldsymbol{Z}$*). In Fig. 3.1 and Fig. 3.3, we further illustrate the difference between the SE block and the proposed AiA block. Mathematically, for a given feature maps $\boldsymbol{X} \in \mathbb{R}^{c\times h\times w}$ as input, the output of SE block is given by*

$$\boldsymbol{X}^z = \text{Sigmoid}\Big(\sigma\big(\xi(\text{GAP}(\boldsymbol{X}))\big)\Big) \otimes \boldsymbol{X}, \tag{3.5}$$

where GAP *indicates Global Average Pooling and $\xi(\cdot)$, $\sigma(\cdot)$ are the gating functions, as that in [Hu et al., 2018]. In contrast, the output of our proposed AiA block is formulated as:*

$$\boldsymbol{X}^z = \text{Sigmoid}\Big(\phi\big(\varphi(\boldsymbol{X})\big) \otimes \varpi\big(\text{GAP}(\varphi(\boldsymbol{X}))\big)\Big) \otimes \boldsymbol{X}. \tag{3.6}$$

By comparing Eq. (3.5) *and Eq.* (3.6)*, one can observe that if $\varphi(\cdot)$ is the identity mapping, $\phi(\boldsymbol{X}) = \boldsymbol{I}$, and $\varpi(\cdot) = \sigma\big(\xi(\cdot)\big)$, then our AiA block realises the SE block. In other words, SE block is a special case in our AiA framework. It is noted $\boldsymbol{I} \in \mathbb{R}^{c\times h\times w}$ represents identity tensor here. Since our AiA block also encodes local features (*e.g., *$\phi(\bar{\boldsymbol{X}})$*), we believe our attention maintains the spatial structural information of the input feature map (*e.g., *$\boldsymbol{X}$*), which essentially improves the performance of the attention block (refer the study in § 3.5.4).*

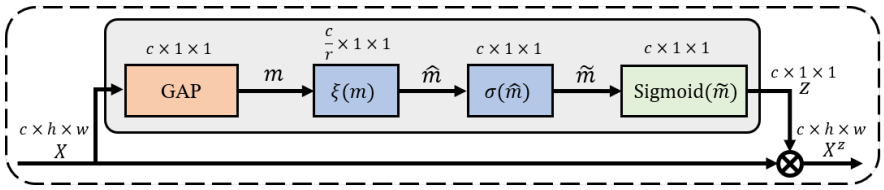


Figure 3.3: The structure of Squeeze and Excitation block.

3.3.1.1 Linear Attention without AiA

In case the number of parameters in the AiA module becomes a concern, one can resort to a simplified version which we denote as *Lin-attention without AiA* (see Fig. 3.4). This simplification reduces the number of parameters in the Lin-attention block while still obtaining competitive performance with respect to the current algorithms for

person re-ID tasks. (refer to § 3.5.4.2 for a comparison against various benchmarks). Formally, we have

$$\boldsymbol{x}^z = \text{Sigmoid}\Big(\phi\big(\varphi(\boldsymbol{x})\big)\Big) \otimes \boldsymbol{x}. \tag{3.7}$$

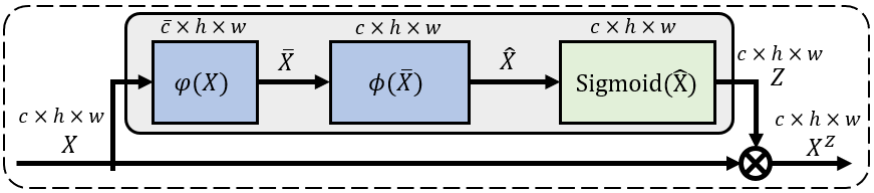


Figure 3.4: The structure of Linear attention without AiA.

In the Lin-attention module, the attention map is generated based on the linear property of the input feature map. To boost its discriminative capacity, we estimate second-order polynomial and Gaussian kernels to extract non-linear features from the input feature map so as to generate the attention map (or values). The two attention modules are called *second-order polynomial attention* and *Gaussian attention*, respectively (refer to Fig. 3.5 for a more detailed description).

3.3.2 Second-order Polynomial Attention

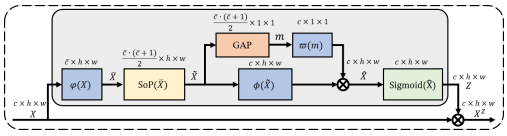
In the second-order polynomial attention (SoP-attention), we make use of the concept of polynomial kernels within AiA. The architecture of SoP-attention is shown in Fig. 3.5(a). In SoP-attention, we first obtain

$$\begin{aligned} \boldsymbol{Y} &= \varphi(\boldsymbol{x})\varphi(\boldsymbol{x})^\top = \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top \\ &= \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_{\bar{c}} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \dots \bar{x}_{\bar{c}} \end{bmatrix} \\ &= \begin{bmatrix} \bar{x}_1^2 & \dots & \bar{x}_1\bar{x}_{\bar{c}} \\ \vdots & \ddots & \vdots \\ \bar{x}_{\bar{c}}\bar{x}_1 & \dots & \bar{x}_{\bar{c}}^2 \end{bmatrix}. \end{aligned} \tag{3.8}$$

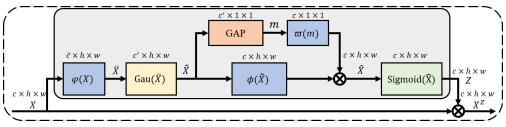
Since $\boldsymbol{Y}$ is a symmetric matrix, we only consider its upper triangular elements in the subsequent processing. This simple step reduces the feature dimensionality from $\bar{c}^2$ to $\bar{c} \cdot (\bar{c}+1)/2$, thereby resulting in faster and efficient processing in the subsequent modules (refer to Fig. 3.6). Specifically, we perform

$$\tilde{\boldsymbol{x}} = \text{Vec}\big(\text{UTri}(\boldsymbol{Y})\big), \tag{3.9}$$

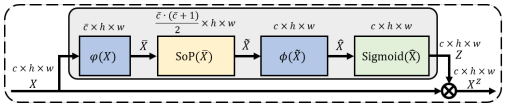
where $\text{Vec}(\cdot)$ and $\text{UTri}(\cdot)$ indicate vectorization and the operator that extracts the upper triangular elements of a matrix respectively. We summarise the bilinear pooling and feature rearrangement with: $\text{SoP}(\bar{\boldsymbol{x}}) = \text{Vec}\big(\text{UTri}(\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top)\big)$.



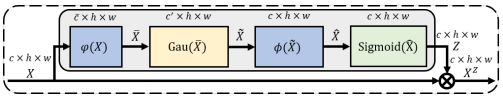
(a) Second-order polynomial attention with AiA



(b) Gaussian attention with AiA



(c) Second-order polynomial attention without AiA



(d) Gaussian attention without AiA

Figure 3.5: The structure of AiA modules employing non-linear features in the feature map. (a): Second-order polynomial attention with AiA, (b): Gaussian attention with AiA, (c): Second-order polynomial attention without AiA, (d): Gaussian attention without AiA. SoP$(\cdot)$ indicates the bilinear pooling and second order feature rearrangement function. Gau$(\cdot)$ indicates the random Fourier feature mapping function.

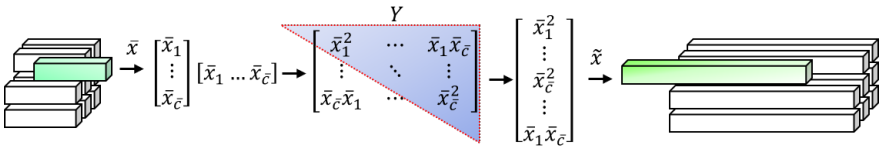


Figure 3.6: Processing of bilinear pooling and second order feature rearrangement, denoted by SoP$(\cdot)$. In this operation, we sample the elements in the upper triangle of $\boldsymbol{Y}$ and vectorize those elements to a new feature vector $\tilde{\boldsymbol{x}}$

Given the second order features (*e.g.*, $\tilde{\boldsymbol{x}}$) and following the similar aforementioned steps from Eq. (3.2) to Eq. (3.4), we propose

$$\text{Sigmoid}\Big(\varpi(\boldsymbol{m}) \otimes \phi(\tilde{\boldsymbol{x}})\Big), \tag{3.10}$$

as the attention map for $\boldsymbol{x}$, where $\boldsymbol{m} = \frac{1}{hw}(\sum_{i=1}^{hw} \tilde{\boldsymbol{x}}_i)$. It is worth mentioning that $\boldsymbol{m}$ contains the *second order statistical* information (*i.e.*, the vectorized version of the empirical auto-correlation matrix of $\bar{\boldsymbol{X}}$) of the input to AiA.

Remark 2 *The inner product between two vectors is widely used as a means of similarity matching. As an insight on the properties of SoP-attention, consider the inner product between* $\tilde{\boldsymbol{x}}_i$ *and* $\tilde{\boldsymbol{x}}_j$, (i.e., *the output of* SoP$(\cdot)$ *function):*

$$\begin{aligned}
\tilde{\boldsymbol{x}}_i^\top \tilde{\boldsymbol{x}}_j &= \text{SoP}(\bar{\boldsymbol{x}}_i)^\top \text{SoP}(\bar{\boldsymbol{x}}_j) \\
&= \text{Vec}\big(\text{UTri}(\bar{\boldsymbol{x}}_i \bar{\boldsymbol{x}}_i^\top)\big)^\top \text{Vec}\big(\text{UTri}(\bar{\boldsymbol{x}}_j \bar{\boldsymbol{x}}_j^\top)\big) \\
&= \sum_u (\bar{x}_{iu} \cdot \bar{x}_{ju})^2 + \sum_u \sum_{s \neq u} (\bar{x}_{iu}\bar{x}_{is} \cdot \bar{x}_{ju}\bar{x}_{js}).
\end{aligned} \tag{3.11}$$

Here, $\bar{x}_{iu}$ *is the u-th element in vector* $\bar{\boldsymbol{x}}_i$. *This shows that with second order pooling, one can*

introduce higher order statistics (e.g., *second term in Eq.* (3.11)*) into making decisions. This, as we will see empirically, boosts the accuracy of the model substantially.*

SoP-attention also has its simplified counterpart, shown in Fig. 3.5(c). This formulation approximately halves the number of parameters of the SoP-attention block, while still benefiting from second order information (using bilinear mapping). Here, the attended feature map is calculated as

$$\boldsymbol{x}^z = \text{Sigmoid}\Big(\phi\big(\text{SoP}(\varphi(\boldsymbol{x}))\big)\Big) \otimes \boldsymbol{x}. \tag{3.12}$$

3.3.3 Gaussian Attention

The SoP-attention module requires a large set of parameters if its input feature map is high-dimensional. To address this difficulty, we propose the Gaussian attention or Gau-attention for short (refer to Fig. 3.5(b) for a conceptual diagram). The Gau-attention makes use of the theory of random Fourier features to approximate the infinite dimensional feature space of a Gaussian kernel. This, as will be shown shortly, drastically reduces the number of parameters of the model and required FLOPs (see Table 3.12 in § 3.5).

Given the embedded feature $\bar{\boldsymbol{x}} = \varphi(\boldsymbol{x}) \in \mathbb{R}^{\bar{c}\times h\times w}$, the function $\text{Gau}(\bar{\boldsymbol{x}})$ estimates the Gaussian kernel along each channel, such that:

$$\mathcal{K}(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j) = e^{-\frac{\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j\|^2}{2\sigma^2}} \approx \kappa(\bar{\boldsymbol{x}}_i)^\top \kappa(\bar{\boldsymbol{x}}_j), \tag{3.13}$$

where $\kappa(\cdot)$ is a randomised embedding. The form of $\kappa(\cdot)$ for a Gaussian kernel [Rahimi and Recht, 2008] is shown below as

$$\kappa(\bar{\boldsymbol{x}}) = \sqrt{\frac{1}{c'}} \begin{bmatrix} \cos(\boldsymbol{\omega}_1^\top \bar{\boldsymbol{x}}) \\ \vdots \\ \cos(\boldsymbol{\omega}_{c'}^\top \bar{\boldsymbol{x}}) \\ \sin(\boldsymbol{\omega}_1^\top \bar{\boldsymbol{x}}) \\ \vdots \\ \sin(\boldsymbol{\omega}_{c'}^\top \bar{\boldsymbol{x}}) \end{bmatrix} \in \mathbb{R}^{2c'}, \tag{3.14}$$

where the weights (*i.e.*, $\boldsymbol{\omega}_i,\ i = 1, \ldots, c'$) are drawn from the scaled Fourier transformation of a Gaussian kernel. That is, we sample from

$$p(\boldsymbol{\omega}) = (2\pi)^{-\bar{c}/2} \exp(-\frac{\|\boldsymbol{\omega}\|^2}{2}) = \frac{1}{2\pi} \int e^{-\mathrm{j}\boldsymbol{\omega}^\top \boldsymbol{\delta}} e^{-\frac{\|\boldsymbol{\delta}\|^2}{2\sigma^2}}\, d\boldsymbol{\delta}. \tag{3.15}$$

The above processing is summarised as the $\text{Gau}(\cdot)$ function, with $\tilde{\boldsymbol{x}} = \text{Gau}(\bar{\boldsymbol{x}})$. Given the estimated random features (*i.e.*, $\tilde{\boldsymbol{x}}$ or $\kappa(\bar{\boldsymbol{x}})$), Gau-attention generates the attention map and attends to the input feature map, following Eq. (3.10) and (3.4) respectively.

Remark 3 *Here, we provide a brief analysis how the* $\mathrm{Gau}(\cdot)$ *function equips the input feature* $\bar{\boldsymbol{x}}$ *with the discriminative power of a Gaussian kernel. Given any two random feature vectors,* $\tilde{\boldsymbol{x}}_i$ *and* $\tilde{\boldsymbol{x}}_j$*, their similarity matching is calculated as follows:*

$$\begin{aligned}
&\mathbb{E}\big[\tilde{\boldsymbol{x}}_i^\top \tilde{\boldsymbol{x}}_j\big] = \mathbb{E}\big[\kappa(\bar{\boldsymbol{x}}_i)^\top \kappa(\bar{\boldsymbol{x}}_j)\big] \\
&= \frac{1}{c'}\mathbb{E}\Big[\sum_k \big(\cos(\boldsymbol{\omega}_k^\top \bar{\boldsymbol{x}}_i)\cos(\boldsymbol{\omega}_k^\top \bar{\boldsymbol{x}}_j) + \sin(\boldsymbol{\omega}_k^\top \bar{\boldsymbol{x}}_i)\sin(\boldsymbol{\omega}_k^\top \bar{\boldsymbol{x}}_j)\big)\Big] \\
&= \mathbb{E}\big[\cos(\boldsymbol{\omega}^\top(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j))\big] = \int_{\mathbb{R}^c} p(\boldsymbol{\omega}) e^{\mathrm{j}\boldsymbol{\omega}^\top(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j)} d\boldsymbol{\omega} \\
&= \boxed{\mathcal{K}(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j)},
\end{aligned} \tag{3.16}$$

where the last equality follows from the Bochner theorem [Rahimi and Recht, 2008]. In § 3.5, we also empirically verify the superior performance of Gaussian attention, which not only saves parameter numbers and computational overhead significantly, but also outperforms the other two in the majority of the experiments.

The simplified version of Gau-attention is shown in Fig. 3.5(d) and is denoted as Gaussian attention without AiA, and its formulation is shown as follows:

$$\boldsymbol{x}^z = \mathrm{Sigmoid}\Big(\phi\big(\mathrm{Gau}(\varphi(\boldsymbol{x}))\big)\Big) \otimes \boldsymbol{x}. \tag{3.17}$$

Remark 4 *Similar to the Fully Attentional Block (FAB) [Wang et al., 2018a], both SoP-attention and Gau-attention without AiA modules maintain the spatial structural information of the input feature map. However, unlike FAB that considers only the first order channel information, the aforementioned attention blocks additionally exploit the non-linear channel information in the second-order polynomial and Gaussian kernel spaces, so as to learn a superior discriminative embedding space for the re-ID task.*

It is worth mentioning that the proposed attention modules can be seamlessly placed in any existing convolutional neural network to enhance the representation learning similar to what most existing attention blocks do. In § 3.5, we will show the effectiveness of the proposed attention modules in the person re-ID application.

3.4 Attention in Attention Networks for Person Retrieval

In this section, we will first provide an overview of the problem formulation. Subsequently, it will be followed by a detailed description of the architecture of the proposed deep convolutional network, the Attention in Attention Network (AiA-Net).

3.4.1 Problem Formulation

Let $\boldsymbol{P}_i \in \mathbb{R}^{C\times H\times W}$ denote an input image, where C, H, and W represent the number of channels and its height and width, respectively. Each image $\boldsymbol{p}_i$ is labelled by its identity, denoted as $y_i \in \{1, \ldots, k\}$, where k represents the total number of distinct

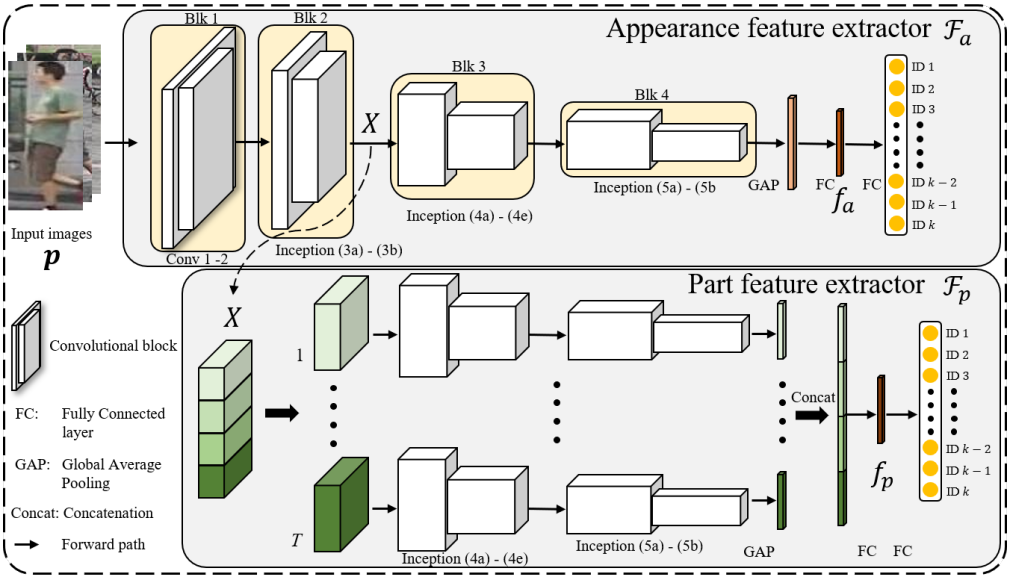


Figure 3.7: The deep architecture of the proposed feature extractor. AiA-Net has two feature extractors, *e.g.*, the person appearance feature extractor (*i.e.*, $\mathcal{F}_a$) and the part feature extractor (*i.e.*, $\mathcal{F}_p$). f_a and f_p are concatenated to give the final person representation as $f = [f_a^\top, f_a^\top]^\top$

identities of the training data. Thus, the training set with N_{train} images, can be represented as $\{\boldsymbol{P}_i, y_i\}_{i=1}^{N_{\text{train}}}$. The person retrieval system, $\mathcal{F}(\boldsymbol{P}, \theta)$, parameterised by θ, aims at encoding an image $\boldsymbol{P}$ to an embedding space, such that the intra-person variations are minimised while the inter-person variations are maximised. In our work, the final embedding space is obtained by concatenating the person-appearance embedding space, *i.e.*, $f_a = \mathcal{F}_a(\boldsymbol{P}, \theta_a)$, and the person-part embedding space, *i.e.*, $f_p = \mathcal{F}_p(\boldsymbol{P}, \theta_p)$, such that $\mathcal{F}(\boldsymbol{P}, \theta) = [f_a^\top, f_p^\top]^\top$.

3.4.2 Overview

The AiA-Net has two feature extractors, namely, (**1**) a person-appearance feature extractor (denoted by $\mathcal{F}_a$) and (**2**) a person part-feature extractor (denoted by $\mathcal{F}_p$). The overall architecture of the AiA-Net is shown in Fig. 3.7. The person holistic appearance is encoded by the appearance feature extractor; while the part feature extractor aims at encoding the different parts of the person.

The appearance feature extractor consists of 4 convolutional blocks. After each convolutional block, an AiA block is added to align the feature map and highlight its discriminative regions. The attended feature map encourages the network to learn a holistic representation (*i.e.*, f_a in Fig. 3.7) of the person.

Recent studies of the person re-identification task suggest that an independent modelling of the part regions can enhance the precision of the overall system [Suh et al., 2018; Sun et al., 2018; Li et al., 2018c]. We also equip the AiA-Net with a parts-based learning ability. More specifically, we use a simple sub-network as a

part feature extractor, which aims at learning distinct and discriminative parts in the input image. In the part feature extractor, the aligned feature map $\boldsymbol{X} \in \mathbb{R}^{c \times h \times w}$ is divided into T non-overlapping regions $\boldsymbol{X}_t$ s.t. $\boldsymbol{X}_t \in \mathbb{R}^{c \times \frac{h}{T} \times w}, t = 1, \ldots, T$. Each of the non-overlapped regions is resized to $c \times h \times w$ by bilinear interpolation and fed to the t-th stream of the part feature extractor network; which generates the part-feature embedding. Then, T part features are concatenated to represent the final person part representation (*i.e.*, $\boldsymbol{f}_p$ in Fig. 3.7).

Remark 5 *Our part feature extractor network is different from the current part-based solutions [Suh et al., 2018; Li et al., 2018c; Sun et al., 2018, 2019b; Zhao et al., 2017c]. For example, in [Suh et al., 2018], the part feature is extracted via a pose estimation network called OpenPose [Cao et al., 2017]. Zhao* et al. *uses an implicitly defined part detector to align the part features [Zhao et al., 2017c]. In [Li et al., 2018c], the parts are sampled through a hard attention network. In [Sun et al., 2018, 2019b], the parts are split evenly in the final feature map. In addition to the structural differences, each part model within the AiA-Net works independently from the others as no weights are shared between them. This, in turn, leads to an increased diversity of the learned parts, thereby learning a more generalised discriminative embedding space for retrieval purposes.*

3.4.3 Multi-Task Training

Multi-Task Training (MTT) has shown to be effective in modern person re-ID solutions. As the name suggests, MTT formulates the overall learning procedure as a combination of several sub-tasks; each having its own importance in the overall learning mechanism. Yu *et al*. uses cross-entropy loss for the classification task and triplet loss for the ranking task [Yu et al., 2018]. Mancs combines the triplet loss, focal loss and cross-entropy loss and learns a superior embedding space for person re-iD against the baseline algorithms [Wang et al., 2018a]. Recent works in [Ni et al., 2020; Zhu et al., 2019] also show person re-ID can benefit from various regularisation,*e.g.*, L2 regularisation, angular regularisation *etc*.. Following the protocol prescribed in [Yu et al., 2018], we train our network for the tasks of ranking and classification jointly, *e.g.*, $\mathcal{L} = \mathcal{L}_{\text{tri}} + \mathcal{L}_{\text{ce}}$. Each loss component is explained in the following.

Ranking Task. We use the well studied triplet loss for the ranking task. In a mini-batch, $\{\boldsymbol{P}_i\}_{i=1}^{N_{\text{batch}}}$, a possible triplet can be denoted as $\{\boldsymbol{P}_i, \boldsymbol{P}_i^+, \boldsymbol{P}_i^-\}$ such that the anchor $\boldsymbol{P}_i$ shares the same identity with the positive sample $\boldsymbol{P}_i^+$ and the negative sample $\boldsymbol{P}_i^-$ belongs to a different identity. In the embedding space $\mathcal{F}(\cdot)$, the triplet loss is formulated as follows:

$$\mathcal{L}_{\text{tri}} = \frac{1}{N_{\text{tri}}} \sum_{i=1}^{N_{\text{tri}}} \left[d_i^+ - d_i^- + \eta \right]_+, \tag{3.18}$$

where $[\cdot]_+ = \max(\cdot, 0)$, N_{tri} indicates the number of triplets within one batch, η is a margin. $d_i^+ = \|\mathcal{F}(\boldsymbol{P}_i) - \mathcal{F}(\boldsymbol{P}_i^+)\|$, and $d_i^- = \|\mathcal{F}(\boldsymbol{P}_i) - \mathcal{F}(\boldsymbol{P}_i^-)\|$. In the triplet mining, for each anchor, we mine one hard positive and 5 hard negatives, thus obtaining

5 triplets per anchor sample. This mining strategy is to avoid collapsing to local minima in the early stages of optimisation [Schroff et al., 2015].

Classification Task. The triplet loss only encodes the inter-person and intra-person information within a particular triplet, but does not fully take into account the identity specific information. To encode the class specific information, we augment the triplet loss with the cross-entropy based classification loss $\mathcal{J}_{\text{cls}}$, shown below

$$\mathcal{L}_{\text{ce}} = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} -\log\big(p(y_i|\mathcal{F}(P_i))\big), \tag{3.19}$$

where $p(y_i|\mathcal{F}(P_i))$ is the predicted probability that P_i belongs to identity y_i, and N_{batch} is the number of samples in one mini-batch.

3.5 Experiments on Image Person Retrieval

3.5.1 Implementation Details

Network Architecture. We implemented our AiA-Net model in the PyTorch [Paszke et al., 2017] deep learning framework. The backbone network is the GoogLeNet-V1 [Szegedy et al., 2015], pre-trained on ImageNet [Russakovsky et al., 2015] with Batch Normalisation [Ioffe and Szegedy, 2015]. The spatial size of the input image is fixed to 256×128. In the appearance feature extractor, the size of the feature after global average pooling (GAP) is 1024, which is followed by the 512-dimensional person appearance embedding layer $\boldsymbol{f}_a$. Another fully connected (FC) layer is connected to predict the person identity using the person appearance embedding. In the part feature extractor, we follow the work in [Li et al., 2018c], and fix $T = 4$ across all experiments. The output features of each of the T streams are concatenated, and is passed through a 512-dimensional part embedding f_p. A FC layer is further connected to predict the person identity using the person part embedding. During testing, $\boldsymbol{f}_a$ and $\boldsymbol{f}_p$ are concatenated to give the final person representation f, where $\boldsymbol{f} = [\boldsymbol{f}_a^\top, \boldsymbol{f}_a^\top]^\top \in \mathbb{R}^{1024}$.

In the AiA block, the embedding functions $\varphi(\cdot)$, $\phi(\cdot)$ and $\varpi(\cdot)$ are 1×1 convolutional layers, followed by a batch normalisation layer and a nonlinear layer. Here, the nonlinear layer uses the ReLU$(\cdot)$ function. In $\varphi(\cdot)$, the dimensionality reduction factor, r, is set to 8 for the CUHK03 [Li et al., 2014] and CUHK01 [Li et al., 2012] datasets, and to 4 for the other datasets. The dimension of the random feature (*i.e.*, c') in Eq. 3.14 is set to 960 for DukeMTMC-reID dataset and to 480 for the other datasets. The details of the datasets will be presented in § 3.5.2.

Network Training. We use the Adam [Kingma and Ba, 2014] optimiser with the default momentum values of (0.9, 0.999) for (β_1 and β_2). The weight decay is set to 0.0001. The learning rate is initialised to 1×10^{-3} for CUHK03 [Li et al., 2014] and CUHK01 [Li et al., 2012], and 5×10^{-4} for Market-1501 [Zheng et al., 2015], DukeMTMC-reID [Ristani et al., 2016] and MSMT17 [Wei et al., 2018]. The size of the mini-batch (*i.e.*, N_{batch} in Eq. (3.19)) is set to 64 for all experiments. The learning rate

is decayed by a factor of 0.1 at 150, 200, 250 epochs respectively for all the datasets. In the multi-task training, we pose the ranking task and classification task in both the appearance and part feature extractors separately; this is inspired by [Sun et al., 2018] where supervision on each respective feature extractor is vital for learning discriminative features. In the triplet loss, we set the margin (*i.e.*, τ in Eq. (3.18)) to 1.5 for the CUHK03 and CUHK01 datasets and 1 for the other datasets. We randomly apply horizontal flip to the input images. Similar to [Huang et al., 2018], we also apply random erasing [Zhong et al., 2017b] after 50 epochs of training in order to avoid any local optima within the embedding space. No such augmentations are used during the testing phase. We report the performance of the network after training it for 250 epochs. Moreover, it is worth noting that we do not apply any re-ranking algorithms to boost the ranking result in the testing phase.

3.5.2 Datasets and Evaluation Protocol

In this section, we evaluate our proposed algorithm across four large scale datasets, *i.e.*, **CUHK03** [Li et al., 2014], **Market-1501** [Zheng et al., 2015], **DukeMTMC-reID** [Ristani et al., 2016] and **MSMT17** [Wei et al., 2018], as well as one small scale dataset, *i.e.*, **CUHK01** [Li et al., 2012].

The **CUHK03** dataset consists of 13,164 person images of 1,467 identities, captured by six non-overlapping cameras. Each person is observed by two disjoint camera views. CUHK03 offers both hand-labelled and deformable part model (DPM)-detected [Felzenszwalb et al., 2010] bounding boxes, and we evaluate our model on both sets. In the CUHK03 dataset, there are two training/testing protocols. In the vanilla training protocol, the training set contains 1,367 identities, while the remaining 100 identities constitute the test set. However in the new protocol [Zhong et al., 2017a], the training set contains 767 identities and the testing set contains the remaining 700 identities. In this chapter, we adopt both the protocols to verify the effectiveness of the proposed attention blocks.

Market-1501 is one of the most popular re-ID dataset which consists of 32,668 person images of 1,501 identities observed under a maximum of 6 different cameras. The dataset is split into 12,936 training images of 751 identities and 19,732 testing images of the remaining 750 identities. Both the training and testing images are detected using a DPM [Felzenszwalb et al., 2010]. In this dataset, we use both the single query and the multi query setting to evaluate our algorithm.

DukeMTMC-reID dataset is collected using 8 different cameras and was originally proposed for video-based person tracking and re-identification. It has 1,404 identities and includes 16,522 training images of 702 identities, 2,228 query images of 702 identities and 17,661 gallery images. In this dataset, the person bounding boxes are manually labelled.

MSMT17 dataset is the largest person re-ID dataset, consisting of 126,441 person images from 4,101 different identities, which are detected using Faster R-CNN [Girshick, 2015]. This dataset is collected with using 15 different cameras. The training set consists of 32,621 images belonging to 1,041 identities, whereas the test set

contains 93,820 images of the remaining 3,060 identities. The test set is further randomly split into 11,659 query images and the remaining 82,161 are used as gallery images.

CHUK01 dataset, a small scale person re-ID dataset, contains 3,884 images of 971 identities. The person images are captured by two cameras with each person having two images in each camera view. The person bounding boxes are labelled manually. We adopt the 485/486 data split as the training protocol to evaluate our network.

We use both the mean Average Precision (mAP) and Cumulative Matching Characteristic (CMC) to evaluate the model performance. The CMC curve measures the correct matching rate for a given query image against the gallery images at various ranks, whereas the mAP measures the probability of all correct matches in the gallery images for a given query image, thereby measuring the overall ranking performance.

3.5.3 Comparison to the State-of-the-Art Methods

To show the superiority of the proposed deep architecture, we compare the performance of AiA-Nets with the current state-of-the-art methods across five datasets.

CUHK03. In the CUHK03 dataset, we evaluate our network under all data settings, that is, both labelled and detected data for the two training set protocols. Table 3.2 and Table 3.3 show the results for both training protocols. We observe that our methods outperform the current state-of-the-art results in vanilla setting and achieve competitive results in the new setting. In the vanilla training set protocol (Refer to Table 3.2), our AiA-Net with Gau-attention improves over the state-of-the-art result by 0.9%/7.6% on mAP for labelled and detected sets, respectively. With respect to the R-1 value, our network beats the current state-of-the-art result by 1.0%/0.4% across the labelled and detected sets. In the new training set protocol (Refer to Table 3.3), our AiA-Net improves the present state-of-the-art mAP value by 0.2%/0.3% and achieves competitive results on R-1 value. This validates the utility of our design choices in AiA-Net along with the importance of the various attention modules to obtain a superior discriminative embedding for person-retrieval.

Table 3.2: Comparison with the SOTA methods on the CUHK03-vanilla dataset in both labelled and detected bounding box. The 1st best in **bold font**.

Model	@ Labelled		@ Detected	
	mAP	R-1	mAP	R-1
DKPM [Shen et al., 2018b]	89.2	91.1	-	-
IANet [Hou et al., 2019a]	-	92.4	-	90.1
MVP Loss [Sun et al., 2019a]	-	93.7	-	91.8
SGGNN [Shen et al., 2018a]	94.3	95.3	-	-
MuDeep [Qian et al., 2019]	-	95.8	-	93.7
AiA-Net w/ Lin-attention	94.8	96.1	91.5	93.6
AiA-Net w/ SoP-attention	**95.2**	**96.8**	92.1	94.0
AiA-Net w/ Gau-attention	94.9	96.6	**92.4**	**94.1**

Table 3.3: Comparison with the SOTA methods on the CUHK03-new dataset in both labelled and detected bounding box. The 1st best in **bold font**.

Model	@ Labelled		@ Detected	
	mAP	R-1	mAP	R-1
HPM [Fu et al., 2019c]	-	-	57.5	63.9
Mancs [Wang et al., 2018a]	63.9	69.0	60.5	65.5
OSNet [Zhou et al., 2019b]	-	-	67.8	72.3
Auto-ReID [Quan et al., 2019]	73.0	77.9	69.3	73.3
RGA [Zhang et al., 2020c]	77.4	**81.1**	74.5	**79.6**
AiA-Net w/ Lin-attention	76.4	79.1	72.8	75.8
AiA-Net w/ SoP-attention	77.0	79.4	74.2	76.9
AiA-Net w/ Gau-attention	**77.6**	80.6	**74.8**	77.8

Market-1501. We further evaluate our proposed AiA-Net against the recent state-of-the-art methods on the Market-1501 in both the single query and multi query settings. The results are shown in Table 3.4. In the single query setting, our method (*i.e.*, AiA-Net w/ Gau-attention) achieves very competitive results over the RGA and ABD-Net. Moreover, our AiA-Nets with Lin-attention, SoP-attention and Gau-attention outperform the present state-of-the-art Mancs by 3.7%, 4.0% and 4.3% on mAP, and by 0.4%, 0.7% and 1.2% on R-1, respectively in the multi query setting.

DukeMTMC-reID. The evaluation of our proposed algorithm on DukeMTMC-reID is shown in Table 3.4. It is obvious that our AiA-Nets obtain a competitive performance with respect to mAP and R-1 value. The AiA-Net with Gau-attention improves over DG-Net by 3.1% on mAP and 1.6% on Rank-1 accuracy. As for ABD-Net, AiA-Net with Gau-attention has competitive performance on the R-1 value (88.8% vs. 89.0%), while achieving the same performance on mAP value. It is worth mentioning that ABD-Net uses larger image sizes, which demands more computation resources.

MSMT17. Table 3.4 shows the result of our proposed network on the challenging MSMT17 dataset. As observed, our proposed networks outperform RGA by 1.3% on mAP value and. However, the present state-of-the-art method (*i.e.*, ABD-Net) beats our network considerably.

CUHK01. Besides learning a discriminative feature representation on large scale datasets, we also compare the performance of the AiA-Nets against the state-of-the art algorithms in the CUHK01 benchmark dataset, thereby demonstrating the generalisation ability of our proposed networks in learning discriminative representations on a small scale dataset. Table 3.5 compares our AiA-Net with current state-of-the-art methods. We observe that each of the AiA-Nets outperform the existing state-of-the-art approach (*i.e.*, PBR) by a large margin. In particular, our three AiA-Nets with Lin-/Sop-/Gau-attention improve the state-of-the-art accuracy by 2.1%, 2.8% and 3.2% on R-1. It is also noted that PBR is pre-trained on the CHUK03 dataset and further fine-tuned on the CUHK01 dataset to avoid over-fitting, while our network is solely trained on the CUHK01 dataset. This indeed shows that our network is able

Table 3.4: Comparison with the SOTA methods on the Market-1501, DukeMTMC-reID and MSMT17 datasets. In the Market-1501 dataset, we apply both single query and multi query to evaluate the model. The 1st best in **bold font**.

Model	Market-1501 @ SQ				Market-1501 @ MQ				DukeMTMC-reID @ SQ				MSMT17 @ SQ			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
MSCAN [Li et al., 2017]	57.5	80.3	-	-	66.7	86.8	-	-	-	-	-	-	-	-	-	-
SVDNet [Sun et al., 2017]	62.1	82.3	92.3	95.2	-	-	-	-	56.8	76.7	86.4	89.9	-	-	-	-
PDC [Su et al., 2017]	63.4	84.1	92.7	94.9	-	-	-	-	-	-	-	-	-	-	-	-
DaRe [Wang et al., 2018c]	69.9	86.0	-	-					56.3	74.5	-	-	-	-	-	-
AOS [Huang et al., 2018]	70.4	86.5	-	-	88.6	92.5	-	-	62.1	79.2	-	-	-	-	-	-
MLFN [Chang et al., 2018]	74.3	90.0	-	-	82.4	92.3	-	-	62.8	81.0	-	-	-	-	-	-
DKPM [Shen et al., 2018b]	75.3	90.1	96.7	97.9	-	-	-	-	63.2	80.3	89.5	91.9	-	-	-	-
HA-CNN [Li et al., 2018c]	75.7	91.2	-	-	82.8	93.8	-	-	63.8	80.5	-	-	-	-	-	-
PBR [Suh et al., 2018]	79.6	91.7	96.9	98.1	85.2	94.0	98.0	98.8	64.2	82.1	-	-	-	-	-	-
DuATM [Si et al., 2018]	76.6	91.4	97.1	-	-	-	-	-	64.6	81.8	90.2	-	-	-	-	-
PCB+RPP [Sun et al., 2018]	81.6	93.8	97.5	98.5	-	-	-	-	69.2	83.3	-	-	-	-	-	-
Mancs [Wang et al., 2018a]	82.3	93.1	-	-	87.5	95.4	-	-	71.8	84.9	-	-	-	-	-	-
SGGNN [Shen et al., 2018a]	82.8	92.3	96.1	97.4	-	-	-	-	68.2	81.1	88.4	91.2	-	-	-	-
HPM [Fu et al., 2019c]	82.7	94.2	97.5	98.5	-	-	-	-	74.3	86.6	-	-	-	-	-	-
IANet [Hou et al., 2019a]	83.1	94.4	-	-	-	-	-	-	73.4	87.1	-	-	46.8	75.5	85.5	88.7
AANet [Tay et al., 2019]	83.4	93.9	-	98.5	-	-	-	-	74.3	87.6	-	-	-	-	-	-
OSNet [Zhou et al., 2019b]	84.9	94.8	-	-	-	-	-	-	73.5	88.6	-	-	52.9	78.7	-	-
DG-Net [Zheng et al., 2019]	86.0	94.8	-	-	-	-	-	-	74.8	86.6	-	-	52.3	77.2	87.4	90.5
ABD-Net [Chen et al., 2019b]	88.3	95.6	-	-	-	-	-	-	**78.6**	**89.0**	-	-	**60.8**	**82.3**	-	-
RGA [Zhang et al., 2020c]	**88.4**	**96.1**	-	-	-	-	-	-	-	-	-	-	57.5	80.3	-	-
AiA-Net w/ Lin-attention	87.2	95.0	97.4	98.5	91.2	95.8	98.6	99.2	77.3	88.0	94.6	96.0	56.2	78.2	88.1	90.6
AiA-Net w/ SoP-attention	87.4	95.3	98.5	99.2	91.5	96.1	98.9	99.3	77.5	88.2	94.8	96.4	57.6	79.6	89.3	91.4
AiA-Net w/ Gau-attention	87.9	95.6	**98.5**	**99.1**	**91.8**	**96.6**	**99.0**	**99.6**	**78.6**	88.8	**94.9**	**96.7**	58.8	80.0	**89.7**	**92.0**

to generalise well while trained on a small dataset from scratch without the need of any such pre-training step.

Table 3.5: Comparison with the SOTA methods on the CUHK01 dataset. The 1st best in **bold font**.

Model	R-1	R-5	R-10	R-20
DGD [Xiao et al., 2016]	66.6	-	-	-
Zhao *et al*. [Zhao et al., 2017c]	75.0	93.5	95.7	97.7
Spindle Net [Zhao et al., 2017b]	79.9	94.4	97.1	98.6
PBR [Suh et al., 2018]	80.7	94.4	97.3	98.6
Baseline ($\mathcal{F}_a + \mathcal{F}_p$)	82.0	94.4	97.7	99.0
AiA-Net w/ Lin-attention	82.8	94.7	97.7	99.0
AiA-Net w/ SoP-attention	83.5	**95.6**	97.9	**99.3**
AiA-Net w/ Gau-attention	**83.9**	95.5	**98.0**	**99.3**

3.5.4 Ablation Study

We first perform experiments to verify the effectiveness of our proposed AiA mechanism and its variants on CUHK03, Market-1501, DukeMTMC-reID and MSMT17 under the single query setting (*i.e.*, SQ). For the CUHK03 dataset, we use the most difficult setting, *i.e.*, the new protocol with detected bounding boxes (*i.e.*, ND).

3.5.4.1 Effect of the Proposed Feature Extractor

In the field of person retrieval, ResNet-50 [He et al., 2016] and GoogLeNet [Szegedy et al., 2015] are the most commonly used backbones [Wang et al., 2018a; Zheng et al., 2019; Suh et al., 2018]. Since we also want the network to own the capacity of learning part features, the part feature extractor is further developed. We compare the performance of the ResNet-50 and GoogLeNet, with each equipped with the part feature extractor. As suggested in Table 3.6, we could observe that: **(1)** the retrieval accuracy increases when the GoogLeNet is equipped with the part feature extractor, thereby showing that our design is indeed effective in exploiting the complementary information between the two feature extractors. **(2)** GoogLeNet + part feature extractor is superior to the ResNet-50 counterpart in both the performance and the network size. Hence, Hence, we use the ImageNet pre-trained GoogLeNet against the ResNet-50 in our experiments. In the rest of this chapter, the GoogLeNet and part feature extractor are represented by $\mathcal{F}_a$ and $\mathcal{F}_p$, respectively.

3.5.4.2 Effect of the Attention in Attention Mechanism

We then evaluate the effectiveness of the proposed AiA mechanism and use the Linear attention for this study on the CUHK03 and Market-1501 datasets. In this study, we compare the Lin-attention without AiA and with AiA employed in the two feature extractors, *i.e.*, $\mathcal{F}_a$ and $\mathcal{F}_a + \mathcal{F}_p$. The attention block is added after the second

Table 3.6: Result of various backbone networks on the CUHK03 and Market-1501 datasets. PNs: parameter numbers. The 1st best in **bold font**.

Model	CUHK03 @ ND		Market @ SQ		PNs
	mAP	R-1	mAP	R-1	($\times 10^6$)
GoogLeNet	64.5	67.1	80.7	91.6	9.45
+ Part feature extractor	**67.8**	**71.1**	**85.1**	93.8	30.16
ResNet-50	64.0	67.6	85.0	**94.5**	25.61
+ Part feature extractor	65.3	68.2	84.1	94.2	46.84

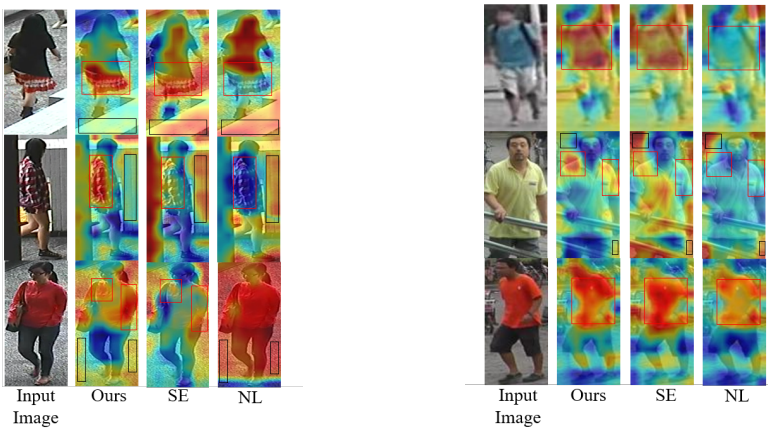
convolutional block (*i.e.*, Blk 2 in Fig. 3.7).In the attention block, we use the identical dimensionality reduction factor, *i.e.*, $r = 4$. The results are listed in Table 3.7. The table shows that: Addition of Lin-attention with and without AiA leads to an increase in the retrieval accuracy across either of the feature extractors, with the former outperforming the latter in terms of mAP and R-1 values respectively. This indeed verifies the design intuition of the AiA mechanism. Further, we replace the Lin-attention block by other popular attention blocks, *i.e.*, the Squeeze-and-Excitation (SE) block [Hu et al., 2018] and the Non-local (NL) block [Wang et al., 2018b], in the same position of the feature extractor (*i.e.*, $\mathcal{F}_a$ and $\mathcal{F}_a + \mathcal{F}_p$). We set the dimensionality reduction factor as 4 in both SE and NL blocks. In this study, we also compare the parameter numbers and inference time of attention networks. As suggested in Table 3.7, our attention outperforms the other two significantly without bringing any additional heavy computational cost[3], thereby verifying the effectiveness of our proposed AiA mechanism.

Table 3.7: Effect of the Attention in Attention mechanism on the CUHK03 and Market-1501 datasets. PNs: parameter numbers; Inf-time: inference time. The 1st best in **bold font**.

Model	CUHK03 @ ND		Market-1501 @ SQ		PNs	Inf-time
	mAP	R-1	mAP	R-1	($\times 10^6$)	(ms)
$\mathcal{F}_a$	64.5	67.1	80.7	91.6	9.45	3.2
Lin-attention w/o AiA	64.8	67.9	80.9	92.4	0.12	3.2
Lin-attention w/ AiA	**66.8**	**70.4**	**82.5**	**92.7**	0.18	3.4
SE block [Hu et al., 2018]	65.2	68.3	81.2	92.4	0.12	3.4
NL block [Wang et al., 2018b]	65.6	68.9	81.4	92.0	0.23	3.8
$\mathcal{F}_a + \mathcal{F}_p$	67.8	71.1	85.1	93.8	30.16	8.4
Lin-attention w/o AiA	68.5	71.4	85.3	94.0	0.12	8.4
Lin-attention w/ AiA	**72.2**	**75.1**	**87.1**	**94.7**	0.18	8.9
SE block [Hu et al., 2018]	70.7	73.0	86.3	94.4	0.12	8.6
NL block [Wang et al., 2018b]	70.9	72.9	86.8	94.5	0.23	9.2

[3]In the inference time, we calculate the averaging inference time per image on NVIDIA GeForce RTX TITAN V

To further verify the superiority of the AiA block, we compare the learned attention between Lin-attention and its alternatives (*i.e.*, SE block and NL block) in Fig. 3.8. We sample the person images in CUHK03 and Market-1501 datasets. Fig. 3.8 shows that our AiA block either highlights the informative foreground (denoted by red rectangles) or filters the non-informative background areas (denoted by black rectangles), thereby clearly demonstrating the benefits of the AiA mechanism.



(a) CUHK03 (b) Market-1501

Figure 3.8: Comparison of the learned attention on CUHK03 (a) and Market-1501 (b) datasets. In each dataset, we compare the the feature map from Lin-attention and its alternatives (*i.e.*, SE block and NL block). In the heat map, the response increases from blue to red. Best viewed in colour.

3.5.4.3 Effect of Employing Non-linear Features in Attention

Then, we study the effect of using non-linear features for attention design on the baseline network $\mathcal{F}_a + \mathcal{F}_p$. In this study, we first evaluate that the AiA framework benefits from the manual non-linear features in RKHSs (*i.e.*, SoP-attention w/ AiA and Gau-attention w/ AiA). We also verify that the manual non-linear features are superior to the learned non-linear features. We has two settings of learned non-linear feature: one is naive nonlinear activation and another one is a stack of nonlinear activation. They are denoted by non-linear attention V1 and non-linear attention V2, respectively. Note that both the two versions of the attention block are incorporated into the AiA framework.

The results on CUHK03 and Market-1501 datasets are shown in Table 3.8. It is observed that the non-linear features, modelled by bilinear mapping and random Fourier features, has superior performance compared to their linear counterpart, thereby highlighting the importance of using non-linear features to locate the highly discriminative regions in the input feature map. In addition, we also observe that AiA-Net with Gau-attention has superior performance over the other two attention variants across both the datasets, which reveals that Gau-attention can learn more

complicated non-linear functions than the other two attention blocks. Table 3.8 also reveals that both the versions of learned non-linearity in AiA achieve similar performance to the Lin-attention with AiA, while the manual non-linear features improves the performance over its linear counterpart, showing the advantage of manually designed non-linearity. It might be that the manual ones enjoy high discrimination power in RKHSs, and are easier to optimise, as compared to the learned non-linear features.

Table 3.8: Effect of the learned non-linearity in attention mechanism on the CUHK03 and Market-1501 datasets. PNs: parameter numbers.

Model	CUHK03 @ ND		Market-1501 @ SQ		PNs
	mAP	R-1	mAP	R-1	($\times 10^6$)
$\mathcal{F}_a + \mathcal{F}_p$	67.8	71.1	85.1	93.8	30.16
Lin-attention w/ AiA	72.2	75.1	87.1	94.7	0.18
SoP-attention w/ AiA	73.2	76.2	87.4	95.1	1.79
Gau-attention w/ AiA	74.0	76.8	87.5	95.2	0.58
Non-linear attention V1	72.7	75.0	87.0	94.6	0.69
Non-linear attention V2	72.4	74.9	86.9	95.0	0.60

3.5.4.4 Effect of the Dimensionality Reduction Factor

In the section, we study the effect of the reduction factor r in the embedding function $\varphi(\cdot)$ on CUHK03 and Market-1501 datasets. All the experiments for this study are conducted using the SoP-attention with AiA, as r is an important hyperparameter that directly affects the information pooled by the bilinear operation. The results and their comparisons, as shown in Table. 3.9, reveal that: **(1)** even though r is an important parameter, which influences the size of the attention model (*i.e.*, the learnable parameters within $\varpi(\cdot)$, $\phi(\cdot)$), our network has a weak dependency on r as changes in r lead to minuscule changes in the performance of our network across all datasets. **(2)** We further observe that while $r = 4$ obtains the best results in the large datasets (*i.e.*, Market-1501, DukeMTMC-reID and MSMT17), the best value of r is observed to be 8 when the network is trained on CUHK03. One plausible explanation is that the network trained on the large datasets is less prone to over-fitting due to its larger training set in comparison to CUHK03.

3.5.4.5 Effect of the Dimensionality in Random Features

In Gau$(\cdot)$, we approximate the channel features in the Gaussian kernel space via a random Fourier mapping. Therefore, we study the result of varying the dimensionality of the random feature (*i.e.*, c') in this section. Here, we have set r to 4 in the embedding function $\varphi(\cdot)$. The results are shown in Table 3.10. One can observe that: along any dimension value (*i.e.*, c'), the random Fourier feature helps to improve

Table 3.9: Effect of the dimensionality reduction factor r in the embedding function $\varphi(\cdot)$ on the CUHK03 and Market-1501 datasets. The 1st best in **bold font**.

Model	CUHK03 @ ND		Market-1501 @ SQ	
	mAP	R-1	mAP	R-1
$\mathcal{F}_a + \mathcal{F}_p$	67.8	71.1	85.1	93.8
$r = 2$	72.3	75.4	87.1	94.9
$r = 4$	72.6	74.9	**87.4**	**95.1**
$r = 8$	**73.2**	**76.2**	87.2	94.5
$r = 16$	72.5	75.6	86.9	94.4
$r = 32$	72.1	74.8	86.9	94.1

retrieval performance of the network. In addition, the network attains the best performance when $c' = 480$ for both datasets. Further, there is a negligible change in the performance of our proposed network with changes in c', thus clearly demonstrating the weak dependency of AiA-Net on c'.

Table 3.10: Effect of the dimensionality c' in random features on the CUHK03 and Market-1501 datasets. The 1st best in **bold font**.

Model	CUHK03 @ ND		Market-1501 @ SQ	
	mAP	R-1	mAP	R-1
$\mathcal{F}_a + \mathcal{F}_p$	67.8	71.1	85.1	93.8
$c' = 120$	72.7	75.3	86.7	94.7
$c' = 240$	73.1	75.9	87.1	94.8
$c' = 480$	**74.0**	**76.8**	**87.5**	**95.2**
$c' = 960$	72.9	75.6	87.3	95.0

3.5.4.6 Effect of the Position of the Attention Block

Table 3.11 shows the effect of adding the Lin-attention with the AiA block to different positions along the baseline network on the CUHK03 and Market-1501 datasets. p_1, p_2, p_3 and p_4 indicate the position of the output of Blk 1, Blk 2, Blk 3 and Blk 4 along the appearance feature extractor respectively (Refer to Fig. 3.7). Table 3.11 shows that: **(1)** using Lin-attention in the early stages, *i.e.*, p_1, p_2, is superior to using it in the later stages *i.e.*, p_3, p_4. A similar observation is also made in [Wang et al., 2018b], where the non-local block enhances the performance of ResNet [He et al., 2016] in its early stages. (**2**) Moreover, the performance of adding Lin-attention in p_2 surpasses the performance compared to when it is added in p_1. One reasonable explanation is that the feature maps at p_2 consist of richer channel, as well as spatial, structural information in comparison to the feature maps at p_1, thereby enabling the network

to emphasise more on the discriminative areas of the images. **(3)** In the CUHK03 dataset, which has a smaller training set, the performance of person retrieval degrades when Lin-attention is inserted at p_4. This is observed as the embedding layer of the Lin-attention module overfits on the training set due to the high dimensionality of the feature map at p_4. (**4**) It is also observed that the network with multiple attention blocks can further bring performance gain. In the rest of this chapter, AiA-Nets indicate plugging multiple attention blocks along with the baseline network (*i.e.*, $\mathcal{F}_a + \mathcal{F}_p$).

Table 3.11: Effect of the position of the AiA block on the CUHK03 and Market-1501 datasets. Here, we use Lin-attention in AiA-Net. The 1st best in **bold font**.

Model	CUHK03 @ ND		Market-1501 @ SQ	
	mAP	R-1	mAP	R-1
$\mathcal{F}_a + \mathcal{F}_p$	67.8	71.1	85.1	93.8
p_1	71.2	73.1	86.5	94.1
p_2	72.2	75.1	87.1	94.7
p_3	69.1	72.4	85.6	93.9
p_4	68.6	70.9	85.1	93.8
p_1 - p_4	**72.8**	**75.8**	**87.2**	**95.0**

3.5.4.7 Computational Complexity and Model Size

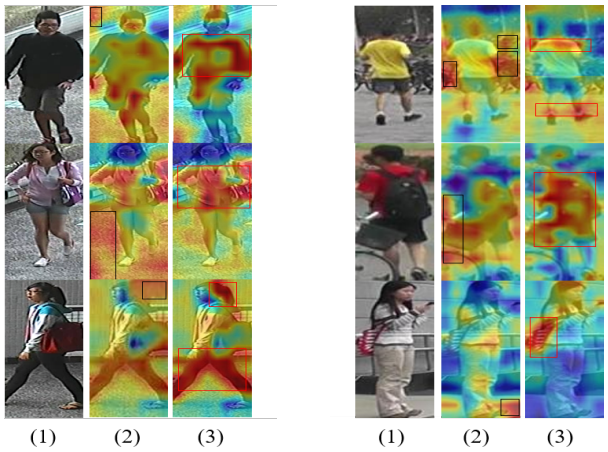
In § 3.5.4.3, we have studied the effect of non-linearity within the AiA module. In this part, we study the block properties (*i.e.*, computational complexity and module size) of each of the AiA blocks and the baseline network (*i.e.*, $\mathcal{F}_a + \mathcal{F}_p$). The computational complexity and model size are measured by the number of floating-point operations (FLOPs) and parameter numbers (PNs) respectively. This study is performed on the CUHK03 dataset and the results are shown in Table 3.12, along with the parameter settings of each attention block. The size of the input feature map to the attention block and input image to baseline network are set to $480 \times 16 \times 8$ and $3 \times 256 \times 128$ respectively. Table 3.12 depicts that: **(1)** compared against the baseline network, the computational complexity and model size of the attention blocks are insignificant, indicating that the performance gain significantly relies on the attention mechanism, rather than increasing the number of parameters. **(2)** Lin-attention and Gau-attention are light weight attention blocks, which can be used in other resource-constrained applications. **(3)** Taking into account the results obtained in Table 3.8, it is clearly observed that Gau-attention is superior to the SoP-attention as it results in a large performance gain (See Table 3.8), while using significantly fewer number parameters than the SoP-attention (*i.e.*, only 1/3 of the number of parameters of SoP-attention). This clearly indicates the hidden potential of the use of non-linear features in the Gaussian kernel space in attention design.

Table 3.12: Computational complexity and module size of proposed attention modules. FLOPs: the number of floating-point operations; PNs: number of parameters.

	Lin-attention	SoP-attention	Gau-attention	$\mathcal{F}_a + \mathcal{F}_p$
Hyper Parameter	$r = 4$	$r = 8$	$r = 4$, $c' = 480$	-
FLOPs ($\times 10^9$)	0.015	0.117	0.044	2.82
PNs ($\times 10^6$)	0.18	1.79	0.58	30.16

3.5.5 Visualisation of the Attention in Attention Module

We visualise the heat maps of the input (*i.e.*, $\boldsymbol{X}$) and output (*i.e.*, $\boldsymbol{X}^z$) of the Gau-attention block for person images in both the CUHK03 detected-set in Fig. 3.9(a) and Market-1501 dataset in Fig. 3.9(b). In each dataset, from left to right, (1): the input person image, (2): the input feature map to attention block, and (3): the masked feature map from attention block. In (2), we use black rectangles to bound the non-informative background clutters in images, which will be filtered by attention block. In (3), we use red rectangles to bound the discriminative parts of the person body parts, which are further emphasised by attention blocks. This visualisation indeed reveals the proposed AiA can focus on the discriminative areas of person images, thereby aligning the feature maps.



(a) CUHK03 (b) Market-1501

Figure 3.9: Visualisation of the attention mechanism in person images, sampled from the CUHK03 dataset (a) and the Market dataset (b). In each dataset, from left to right, (1) the input person image, (2) the input feature map to attention and (3) the masked feature map. The heat maps are generated in AiA-Net with Gau-attention. In the heat map, the response increases from blue to red. Best viewed in colour.

3.5.6 Discussion

Statistical significance of the proposed method. In § 3.5.3 and § 3.5.4, a thorough study has been studied to verify the superiority of proposed attention blocks. We

further study their statistical significance using t-test. We adopt the AiA-Net w/ Gau-attention and CUHK03 (See Table 3.2) in this study, and we obtain the p-value of 0.0026 / 0.0033, meaning that our results are significant ($p < 0.05$ is significant). Thus we believe that our AiA-Net is superior to the MuDeep [Qian et al., 2019]. We also plug the Gau-attention with AiA to the ResNet-50 backbone, The results of our AiA read 96.1 / 93.9 as compared to 95.8 / 93.7 of MuDeep, again showing the superiority of the AiA block. In this study, the p-values are 0.0003 / 0.0041, still showing that the results are significant.

Analysis of the "Attention in Attention" mechanism and "single attention" mechanism. In Table 3.7, we compared AiA against a simplified version, which still benefits from the use of an attention block without the use of any inner attention module (Fig. 3.4 vs. Fig. 3.1 in § 3.3). Empirically, we observe that by incorporating the inner attention module, improved results can be obtained in both baseline architectures(*i.e.*, $\mathcal{F}_a$ and $\mathcal{F}_a + \mathcal{F}_p$). To further verify this, we replace our AiA with the current state-of-the-art attention modules, namely the Squeeze-and-Excitation block [Hu et al., 2018] and the Non-local block [Wang et al., 2018b], and evaluate the resulting structure on the CUHK03 and Market-1501 datasets. The results on Table 3.7 and Fig. 3.8 clearly show the superiority of AiA over both the Squeeze-and-Excitation and Non-local blocks, even though only the linear kernel is used in this study.

Analysis of Failure Cases. In this section, we show some ranking lists of the failure cases (*i.e.*, the identity mismatch of R-1 retrieved images for certain query images) obtained by AiA-Net with Gau-attention across the person re-ID datasets. Fig. 3.10 shows that the AiA-Net may be affected by persons with similar distractors, such as similar clothing and stature (*i.e.*, the first and second ranking lists). Further, for the DukeMTMC-reID dataset, our network is also affected (*i.e.*, the third and forth ranking lists) by occlusions (*i.e.*, bike, car). Nonetheless, taking a closer look at those failure cases highlighted with red rectangles, they are in fact perceptually very similar to its respective query image (*i.e.*, colour of clothes, body orientation *etc*.). Having said that, these observations motivate us to further develop more robust person re-ID algorithms so as to differentiate such subtle changes successfully.

Generalisation of Attention Blocks. To verify the generalisation of proposed attention blocks, we employ other backbones to evaluate the effectiveness of AiA blocks, including ResNet-50 [He et al., 2016], GoogLeNet-V1 [Szegedy et al., 2015], DenseNet [Huang et al., 2017], MobileNet [Sandler et al., 2018] as well as ShuffleNet [Ma et al., 2018]. This study is conducted on CUHK03 dataset. Fig. 3.11 reveals that our AiA blocks can consistently bring performance gain across various backbones, clearly showing the generalisation and superiority of the AiA block.

3.6 Experiments on Video Person Retrieval

In this section, we further evaluate our AiA modules on the video person retrieval setting on the **MARS** [Zheng et al., 2016] benchmark dataset. This dataset contains 20,715 video sequences of 1,261 person identities. The identities for training and

Figure 3.10: Some failure cases on person re-ID datasets. In each ranking list, to the left is the query person and to the right is the corresponding ranked list in the gallery set. The correct and false matches are enclosed in green and red boxes. Best viewed in colour.

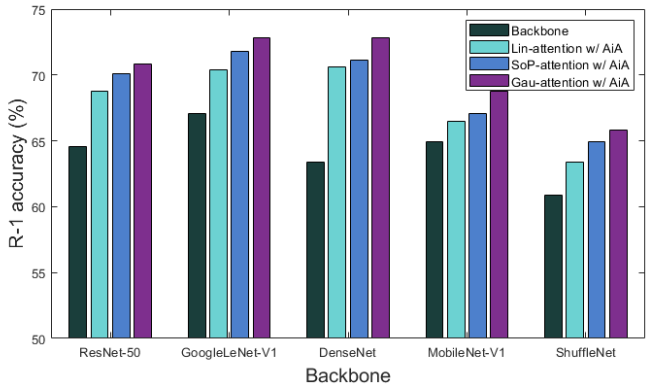


Figure 3.11: Evaluation for attention blocks on different backbone networks on the CUHK03 dataset.

testing are split into 631 and 630 respectively. The number of frames in each sequence varies from 2 to 920 and the average length of a video sequence is 59.5 frames. Each sequence is generated by the GMMCP tracker [Dehghan et al., 2015], and the bounding box for each frame is detected automatically by the DPM [Felzenszwalb et al., 2010].

Table 3.13 compares the result of our approach against the current state-of-the-art algorithms. It clearly shows that our AiA-Net-V improves the state-of-the-art mAP value by 1.0% and the R-1 value by 0.2%. It should be noted that AiA-Net-V only considers the spatial information in each frame to calculate the attention values and unlike [Gao and Nevatia, 2018; Fu et al., 2019b; Li et al., 2019a], it doesn't take into account the modelling of the temporal attention to fuse the frame features. This improvement clearly shows that our AiA-Net-V makes better use of spatial structure information and attends to the informative areas in each frame.

Table 3.13: Comparison with the SOTA methods on the MARS dataset in video person retrieval setting. The 1st best in **bold font**.

Model	mAP	R-1	R-5	R-10
PBR [Suh et al., 2018]	72.2	83.0	92.8	95.0
Zhao *et al*. [Zhao et al., 2019]	78.2	87.0	95.4	-
GLTR [Li et al., 2019a]	78.4	87.0	95.7	-
COSAM [Subramaniam et al., 2019]	79.9	84.9	95.5	-
STA [Fu et al., 2019b]	80.8	86.3	95.7	-
Baseline ($\mathcal{F}_a + \mathcal{F}_p$)	77.3	83.1	94.2	96.0
AiA-Net-V w/ Lin-attention	81.3	86.4	94.7	96.7
AiA-Net-V w/ SoP-attention	**81.8**	86.7	95.4	97.0
AiA-Net-V w/ Gau-attention	81.7	**87.2**	**95.6**	**97.2**

3.7 Summary

In this chapter, we generalise the Attention in Attention (AiA) mechanism for the person retrieval task. This AiA mechanism uses an inner attention, which encodes the global features of the input feature map, to re-weight the feature map. Thereafter, this feature map is further processed by an outer attention, to generate a well focused attention map. Besides the linear version of AiA, we propose and develop non-linear versions of AiA, where the features are approximated using the second-order polynomial and Gaussian kernel spaces respectively. We further propose simplified versions of the aforementioned attention blocks which exclude the inner attention (i.e., without AiA). With regards to the person retrieval task, we also propose an efficient feature extractor, which encodes both person appearance and part features. We incorporate the aforementioned AiA blocks in our network, termed AiA-Net, and empirically show that state-of-the-art performances can be achieved by incorporating the AiA modules in representation learning. This includes extensive evaluations on five standard person re-ID benchmarks along with the required ablation studies to understand the effect of various AiA blocks. Furthermore, our AiA-Net-V also achieves state-of-the-art result on the video person retrieval task, showing the generalisation to video data.

Chapter 4

Channel Recurrent Attention Networks

Following the previous chapter, we continue focusing on the attention mechanism for visual embedding. Full attention, which generates an attention value per element of the input feature maps, has been successfully demonstrated to be beneficial in visual tasks. In this chapter, we propose a fully attentional network, termed channel recurrent attention network, for the task of video pedestrian retrieval. The main attention unit, channel recurrent attention, identifies attention maps at the frame level by jointly leveraging spatial and channel patterns via a recurrent neural network. This channel recurrent attention is designed to build a global receptive field by recurrently receiving and learning the spatial vectors. Then, a set aggregation cell is employed to generate a compact video representation. Empirical experimental results demonstrate the superior performance of the proposed deep network, outperforming current state-of-the-art results across standard video person retrieval benchmarks, and a thorough ablation study shows the effectiveness of the proposed units. This chapter is based on our published work [Fang et al., 2020].

4.1 Introduction

This chapter proposes *Channel Recurrent Attention Networks* for the purpose of pedestrian retrieval, in challenging video data.

There are many challenges to the person re-ID task, with a majority stemming from a poor quality or large variation of the captured images. This often leads to difficulties in building a discriminative representation, which in turn results in a retrieval system to mismatch its queries. Video-, as opposed to single image-, person re-ID offers the possibility of a richer and more robust representation as temporal cues can be utilised to obtain a compact, discriminative and robust video representation for the re-ID task. In many practical situations, the retrieval performance suffers from spatial misalignment [Suh et al., 2018; Li et al., 2018c; Zhou et al., 2020], caused by the movement of body parts, which affects the retrieval machine negatively. Focusing on this issue, many efforts have been made to develop visual attention mechanisms [Li et al., 2018c; Wang et al., 2018a; Fang et al., 2019; Chen et al., 2019a; Subramaniam

et al., 2019], which makes the network attend to the discriminative areas within person bounding boxes, relaxing the constraints stemming from spatial nuances.

Attention mechanisms have been demonstrated to be successful in various visual tasks, such as image classification [Hu et al., 2018; Woo et al., 2018], object detection [Wang et al., 2018b], scene segmentation [Fu et al., 2019a; Li et al., 2018b] to name just a few. Generally speaking, attention mechanisms can be grouped into channel attention [Hu et al., 2018], spatial attention [Wang et al., 2017], and full attention [Wang et al., 2018a], according to the dimensions of the generated attention maps. The channel attention usually summarises the global spatial representation of the input feature maps, and learns a channel pattern that re-weights each slice of the feature maps. In contrast, the spatial attention learns the spatial relationships within the input feature maps and re-weights each spatial location of the feature maps. Lastly, full attention not only learns the channel patterns, but also preserves spatial information in the feature maps, which significantly improves the representation learning [Hjelm et al., 2019].

Various types of full attention mechanisms have been studied extensively for the task of pedestrian retrieval [Wang et al., 2018a; Fang et al., 2019; Chen et al., 2019a]. In [Wang et al., 2018a], the fully attentional block re-calibrates the channel patterns by a non-linear transformation. Thereafter, higher order channel patterns are exploited to attend to the channel features [Fang et al., 2019; Chen et al., 2019a]. However, the aforementioned attention fails to build *long-range* spatial relationships due to the use of a 1×1 convolution. The work in [Li et al., 2018c] learns spatial interactions via a convolutional layer with a larger kernel size (3×3), but the attention module therein still only has a small spatial receptive field. In visual attention, we want the network to have the capacity to view the feature maps globally and decide what to focus on for further processing [Wang et al., 2018b]. A global view can be achieved by applying fully connected (FC) layers, which, unfortunately, introduces a huge number of learnable parameters if implemented naively.

In this work, we propose a full attention mechanism, termed *channel recurrent attention*, to boost the video pedestrian retrieval performance. The channel recurrent attention module aims at creating a global view of the input feature maps. Here, the channel recurrent attention module benefits from the recurrent operation and the FC layer in the recurrent neural network. We feed the vectorized spatial map to the Long Short Term Memory (LSTM) sequentially, such that the recurrent operation of the LSTM captures channel patterns while the FC layer in the LSTM has a global receptive field of each spatial slice. To handle video data, we continue to develop a *set aggregation* cell, which aggregates the frame features into a discriminative clip representation. In the set aggregation cell, we re-weight each element of the corresponding frame features, in order to selectively emphasise useful features and suppress less informative features, with the aid of the associated clip features.

The **contributions** of this chapter include:

- The proposal of a novel channel recurrent attention module to jointly learn spatial and channel patterns of each frame feature map, capturing the global

view of the feature maps. To the best of the authors' knowledge, this is the first attempt to consider the global spatial and channel information of feature maps in a full attention design for video person re-ID.

- The development of a simple yet effective set aggregation cell, which aggregates a set of frame features into a discriminative clip representation.

- State-of-the-art performance across standard video re-ID benchmarks by the proposed network. The generalisation of the attention module is also verified by the competitive performance on the single image re-ID task.

4.2 Related Work

This section summarises the related work of relevant attention mechanisms.

Recent work has shown that person re-ID benefits significantly from attention mechanisms highlighting the discriminative areas inside the person bounding boxes when learning an embedding space [Liu et al., 2016, 2017c; Li et al., 2018c; Wang et al., 2018a; Fang et al., 2019; Chen et al., 2019a]. In [Liu et al., 2016, 2017c], the spatial attention mask is designed to attend one target feature map or various feature maps along the deep network. In [Wang et al., 2018a], a fully attentional block is developed to re-calibrate the channel features. Second or higher order statistical information is also employed in full attention frameworks [Fang et al., 2019; Chen et al., 2019a]. The full attention shape map is also generated in the harmonious attention module [Li et al., 2018c], by integrating channel attention and spatial attention. The aforementioned attention mechanism either fails to build spatial-wise relationships, or receives a limited spatial receptive field. Unlike the above methodology of full attention, we intend to develop an attention mechanism which preserves the advantage of the common full attention, while also perceiving a global spatial receptive field of the feature maps.

In contrast to the existing works, we aim to develop a full attention mechanism that can capture the global receptive field of the feature maps, improving the understanding of networks to images.

4.3 Channel Recurrent Attention Networks for Pedestrian Retrieval

This section details the proposed deep network in a top-down fashion: starting with the problem formulation of the application, followed by the network architecture and the main attention module in the network, namely, the channel recurrent attention module. Thereafter, we also introduce a set aggregation cell, to encode a compact clip representation.

4.3.1 Problem Formulation

Let a fourth-order tensor, $\mathcal{T}_i = [\boldsymbol{T}_i^1, \boldsymbol{T}_i^2, \ldots, \boldsymbol{T}_i^N] \in \mathbb{R}^{N \times C \times H \times W}$, denote the i-th video sequence of a pedestrian, where N, C, H, and W are the number of frames, channels, height and width, respectively. Each video sequence $\mathcal{T}_i$ is labelled by its identity, denoted by $y_i \in \{1, \ldots, k\}$. The training set with M video sequences is described by $\mathbb{T} = \{\mathcal{T}_i, y_i\}_{i=1}^M$. The video person re-ID model, $\mathcal{F}(\cdot, \theta) : \mathcal{T} \rightarrow \mathbb{R}^n$, describes a non-linear embedding from the video space, $\mathcal{T}$, to an embedding space, $\mathbb{R}^n$, in which the intra-class/person distance is minimised and the inter-class/person distance is maximised. The target of training a deep neural network is to learn a set of parameters, $\theta^\star$, with minimum loss value (*e.g.*, $\mathcal{L}$), satisfying: $\theta^\star = \arg\min_\theta \sum_{i=1}^M \mathcal{L}(\mathcal{F}(\mathcal{T}_i, \theta), y_i)$. In the training stage, we randomly sample batches of video clips, where each video clip has only t frames (randomly chosen). Such frames are order-less and hence, we are interested in set-matching for video re-ID.

4.3.2 Overview

We begin by providing a sketch of our design first. In video person re-ID, one would ideally like to make use of a deep network to extract the features of the frames and fuse them into a compact and discriminative clip-level representation. In the lower layers of our design, we have five convolutional blocks along with channel recurrent attention modules at positions P_1, P_2 and P_3 (see Fig. 4.1). Once the deep network extracts a set of frame features (*i.e.*, $[\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t]$ in Fig. 4.1), a set aggregation cell is utilised to fuse frame features into a compact clip-level feature representation (*i.e.*, $\boldsymbol{g}$). The final clip representation is $\boldsymbol{f} = \mathrm{ReLU}\big(\mathrm{BN}(\boldsymbol{W}_1^\top \boldsymbol{g})\big)$, followed by another FC layer to perform identity prediction (*i.e.*, $p = \boldsymbol{W}_2^\top \boldsymbol{f}$), where $\boldsymbol{W}_1, \boldsymbol{W}_2$ are the learnable parameters in the FC layers. We note that the output of the middle convolutional layers captures rich spatial and channel information [Wang et al., 2018b; Fang et al., 2019], such that the attention modules can make better use of this available information.

The network training benefits from multi-task learning, which formulates the network training as several sub-tasks. Our work follows [Gao and Nevatia, 2018], and trains the network using a triplet loss and a cross-entropy loss.

Triplet Loss. To take into account the between-class variance, we use the triplet loss Schroff et al. [2015], denoted $\mathcal{L}_{\text{tri}}$, to encode the relative similarity information in a triplet. In a mini-batch, a triplet is formed as $\{\boldsymbol{f}_i, \boldsymbol{f}_i^+, \boldsymbol{f}_i^-\}$, such that the anchor clip $\mathcal{T}_i$ and the positive clip $\mathcal{T}_i^+$ have the same identity, while the negative clip $\mathcal{T}_i^-$ has a different identity. With the clip feature embedding, the triplet loss is formulated as: $\mathcal{L}_{\text{tri}} = \frac{1}{PK} \sum_{i=1}^{PK} \Big[\|\boldsymbol{f}_i - \boldsymbol{f}_i^+\| - \|\boldsymbol{f}_i - \boldsymbol{f}_i^-\| + \eta \Big]_+$, where η is a margin and $[\cdot]_+ = \max(\cdot, 0)$. A mini-batch is constructed by randomly sampling P identities and K video clips for each identity. We employ a hard mining strategy Hermans et al. [2017] for triplet mining.

Cross-entropy Loss. The cross-entropy loss realises the classification task in training a deep network. It is expressed as: $\mathcal{L}_{\text{ce}} = \frac{1}{PK} \sum_{i=1}^{PK} -\log\big(p(y_i|\boldsymbol{f}_i)\big)$, where p is the predicted probability that $\boldsymbol{f}_i$ belongs to identity y_i. The classification loss encodes

the class specific information, which minimises the within-class variance. The total loss function is formulated as: $\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{tri}}$.

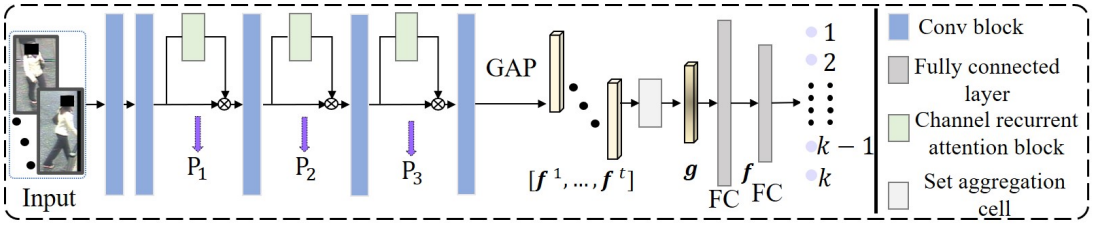


Figure 4.1: The architecture of the proposed deep neural network with channel recurrent attention modules and a set aggregation cell.

4.3.3 Channel Recurrent Attention

We propose the channel recurrent attention module (see Fig. 4.2), which learns the spatial and channel patterns globally in a collaborative manner with the assistance of an LSTM, over the feature maps of each frame. To be specific, we model the input feature maps as a sequence of spatial feature vectors, and feed it to an LSTM to capture global channel patterns by its recurrent operation. In our design, the hidden layer (*e.g.*, FC) of the LSTM unit, can be understood as having a global receptive field, acting on each spatial vector while sharing weights with other spatial vectors, addressing the limitation of a small receptive field in CNNs. In § 4.4, our claim is empirically evaluated in an ablation study.

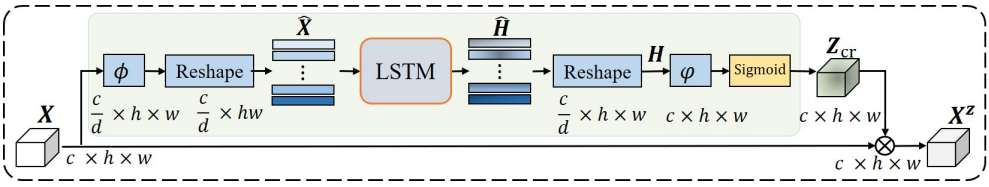


Figure 4.2: The structure of the proposed channel recurrent attention module.

Let $\boldsymbol{X} \in \mathbb{R}^{c \times h \times w}$ be the input of the channel recurrent attention module. In our implementation, we project $\boldsymbol{X}$ to $\phi(\boldsymbol{X})$, reducing the channel dimension by a ratio of $1/d$, and reshape the embedded tensor $\phi(\boldsymbol{X})$ to a matrix $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{\frac{c}{d}}]^\top \in \mathbb{R}^{\frac{c}{d} \times hw}$, where a row of $\hat{\boldsymbol{X}}$ (*e.g.*, $\hat{\boldsymbol{x}}_i \in \mathbb{R}^{hw}, i = 1, \ldots, \frac{c}{d}$) denotes the spatial vector of a slice. The effect of the ratio $1/d$ is studied in § 4.4.4. A sequence of spatial vectors is then fed to an LSTM unit and the LSTM generates a sequence of hidden states, in matrix form:

$$\hat{\boldsymbol{H}} = \text{LSTM}(\hat{\boldsymbol{X}}) = [\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_{\frac{c}{d}}]^\top, \tag{4.1}$$

where $\hat{\boldsymbol{h}}_i \in \mathbb{R}^{hw}, i = 1, \ldots, c/d$ is a sequence of hidden states and $\text{LSTM}(\cdot)$ represents

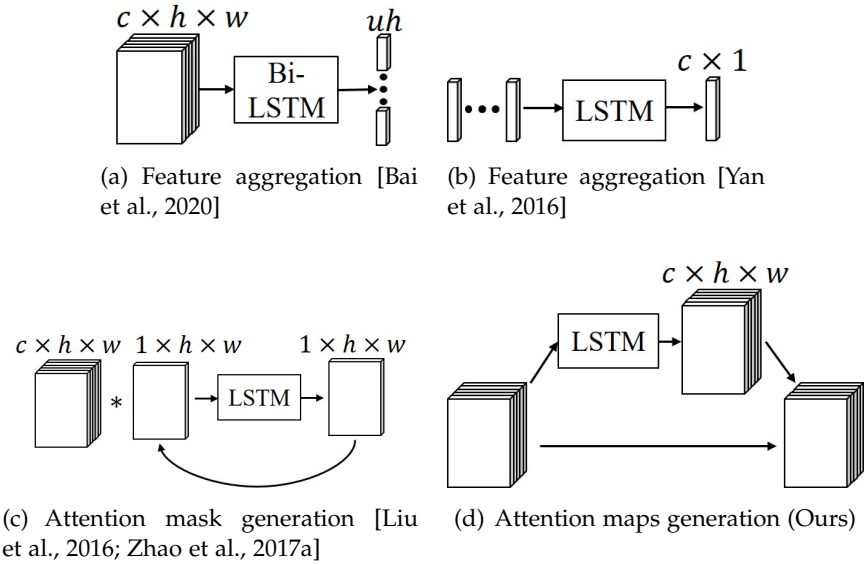
the recurrent operation in an LSTM. The insight is illustrated by the unrolled LSTM, shown in Fig. 4.5(a). $\hat{\boldsymbol{H}}$ is further reshaped to the same size as the input tensor $\phi(\boldsymbol{X})$ (*i.e.*, $\boldsymbol{H} = \text{Reshape}(\hat{\boldsymbol{H}}), \boldsymbol{H} \in \mathbb{R}^{\frac{c}{d} \times h \times w}$). The final attention value is obtained by normalising the embedded $\boldsymbol{H}$, written as:

$$\boldsymbol{Z}_{\text{cr}} = \text{Sigmoid}(\varphi(\boldsymbol{H})). \tag{4.2}$$

Here, $\varphi(\boldsymbol{H}), \boldsymbol{Z}_{\text{cr}} \in \mathbb{R}^{c \times h \times w}$. This normalised tensor acts as a full attention map and re-weighs the elements of the associated frame feature map (see Fig. 4.2), by element-wise multiplication:

$$\boldsymbol{X}^{\boldsymbol{Z}} = \boldsymbol{Z}_{\text{cr}} \otimes \boldsymbol{X}. \tag{4.3}$$

Remark 6 *There are several studies that use LSTMs to aggregate features [Bai et al., 2020; Yan et al., 2016] (see Fig. 4.3(a) and 4.3(b)), or generate attention masks [Liu et al., 2016; Zhao et al., 2017a] (see Fig. 4.3(c)). Our channel recurrent attention module (see Fig. 4.3(d)) is significantly different from existing works as shown in Fig. 4.3. The designs in [Bai et al., 2020] and [Yan et al., 2016] employ an LSTM to aggregate features either from input feature maps [Bai et al., 2020], or a sequence of frame features in a video [Yan et al., 2016]. In [Liu et al., 2016; Zhao et al., 2017a], an attention value for each spatial position of the feature maps* (i.e., *spatial attention) is constructed recursively, while ignoring the relation in the channel dimension. In contrast, our channel recurrent attention generates an attention value per element of the feature maps* (i.e., *full attention), thereby enabling the ability to learn richer spatial and channel features.*



(a) Feature aggregation [Bai et al., 2020]

(b) Feature aggregation [Yan et al., 2016]

(c) Attention mask generation [Liu et al., 2016; Zhao et al., 2017a]

(d) Attention maps generation (Ours)

Figure 4.3: Schematic comparison of our attention mechanism and existing LSTM-based works. In (c), the notation $*$ denotes a weighted sum operation.

4.3.4 Set Aggregation

To encode a compact clip representation, we further develop a set aggregation cell to fuse the per frame features (see Fig. 4.4 for a block diagram). The set aggregation cell highlights the frame feature, with the aid of the clip feature, firstly, and then aggregates them by average pooling.

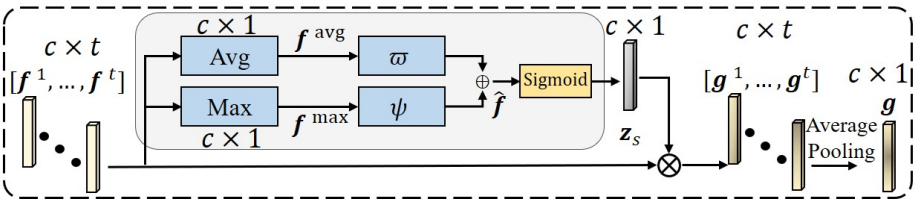


Figure 4.4: The structure of the proposed set aggregation cell.

Let $[\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t], \boldsymbol{f}^j \in \mathbb{R}^c$ be a set of frame feature vectors, encoded by a deep network (see Fig. 4.1). The set aggregation cell first re-weights the frame features. In our implementation, we combine average pooling and max pooling to aggregate frame features. This is due to the fact that both pooling schemes encode different statistical information and their combination is expected to increase the representation capacity. More specifically, each element in $\boldsymbol{f}^{\text{avg}}$ and $\boldsymbol{f}^{\text{max}}$ are defined as $\boldsymbol{f}_i^{\text{avg}} = \text{avg}(\boldsymbol{f}_i^1, \ldots, \boldsymbol{f}_i^t) = \frac{1}{t}\sum_{j=1}^{t}(\boldsymbol{f}_i^j)$ and $\boldsymbol{f}_i^{\text{max}} = \max(\boldsymbol{f}_i^1, \ldots, \boldsymbol{f}_i^t)$, respectively. Each aggregation is followed by self-gating layers (*i.e.*, $\varpi(\cdot)$ and $\psi(\cdot)$ in Fig. 4.4) to generate per-element modulation weights, and fused by element-wise summation as:

$$\hat{\boldsymbol{f}} = \varpi(\boldsymbol{f}^{\text{avg}}) \oplus \psi(\boldsymbol{f}^{\text{max}}). \tag{4.4}$$

This is then followed by normalising the fused weights to produce the final mask (*e.g.*, $\boldsymbol{z}_{\text{s}} = \text{Sigmoid}(\hat{\boldsymbol{f}})$) which is applied as follows:

$$\boldsymbol{g}^j = \boldsymbol{z}_{\text{s}} \otimes \boldsymbol{f}^j, \; j = 1, \ldots, t. \tag{4.5}$$

Finally, we use average pooling to obtain the clip feature, $\boldsymbol{g} = 1/t \sum_{j=1}^{t} \boldsymbol{g}^j$. We note that in our network the parameters in the two self-gating layers are not shared. This is to increase the diversity of features which is beneficial, and we evaluate it in § 4.4.

Remark 7 *The set aggregation cell is inspired by the Squeeze-and-Excitation (SE) block [Hu et al., 2018], in the sense that frame features will be emphasised under the context of the global clip-level features, but with a number of simple yet important differences:* ***(i)*** *The SE receives a feature map as input, while the input of our set aggregation is a set of frame features.* ***(ii)*** *The SE only uses global average pooling to encode the global feature of the feature maps, while the set aggregation employs both average and max pooling to encode hybrid clip features, exploiting more diverse information present in the frame features.*

4.4 Experiments on Video Person Retrieval

4.4.1 Implementation Details

Network Architecture. We implemented our approach in the PyTorch [Paszke et al., 2017] deep learning framework. We chose ResNet-50 [He et al., 2016] as the backbone network, pre-trained on ImageNet [Russakovsky et al., 2015]. In a video clip with t frames, each frame-level feature map, produced by the last convolutional layer, is squeezed to a feature vector $\boldsymbol{f}^j \in \mathbb{R}^{2048}, j = 1, \ldots, t$ by global average pooling (GAP). Subsequently, the set aggregation cell fuses the frame features to a compact clip feature vector $\boldsymbol{g}$. Following $\boldsymbol{g}$, the final clip-level person representation $\boldsymbol{F}$ is embedded by a fully connected (FC) layer with the dimension 1024. Thereafter, another FC layer is added for the purpose of final classification during training. In the channel recurrent attention module, the ratio d is set to 16 for the PRID-2011 and iLIDS-VID datasets, and 8 for the MARS and DukeMTMC-VideoReID datasets, and the LSTM unit has one hidden layer. In the set aggregation cell, the self-gating layer is a bottleneck network to reduce the number of parameters, the dimension of the hidden vector is $2048/r$, and we choose $r = 16$ as in [Hu et al., 2018], across all datasets. The ReLU and batch normalisation are applied to each embedding layer and self-gating layer. The details of the datasets is described in § 4.4.2.

Network Training. We use the Adam [Kingma and Ba, 2014] optimiser with default momentum. The initial learning rate is set to 3×10^{-4} for PRID-2011 and iLIDS0-VID, and 4×10^{-4} for MARS and DukeMTMC-VideoReID. The mini-batch size is set to 16 for the PRID-2011 and iLIDS-VID datasets and 32 for the MARS and DukeMTMC-VideoReID datasets, respectively. In a mini-batch, both P and K are set to 4 for the PRID-2011 and iLIDS-VID, whereas $P = 8$, $K = 4$ for the MARS and DukeMTMC-VideoReID. The margin in the triplet loss, *i.e.*, ξ, is set to 0.3 for all datasets. The spatial size of the input frame is fixed to 256×128. Following [Gao and Nevatia, 2018], t is chosen as 4 in all experiments and 4 frames are *randomly* sampled in each video clip [Zhao et al., 2019; Gao and Nevatia, 2018]. Our training images are randomly flipped in the horizontal direction, followed by random erasing (RE) [Zhong et al., 2017b]. We train the network for 800 epochs. The learning rate decay is set to 0.1, applied at the 200-th, 400-th epoch for the PRID-2011 and iLIDS-VID , and the 100-th, 200-th, 500-th epoch for the MARS and DukeMTMC-VideoReID, respectively. Moreover, it is worth noting that we do not apply re-ranking to boost the ranking result in the testing phase.

4.4.2 Datasets and Evaluation Protocol

In this section, we perform experiments on four standard video benchmark datasets, *i.e.*, **PRID-2011** [Hirzer et al., 2011], **iLIDS-VID** [Wang et al., 2016], **MARS** [Zheng et al., 2016] and **DukeMTMC-VideoReID** [Wu et al., 2018a] to verify the effectiveness of the proposed attentional network. The **PRID-2011** has 400 video sequences, showing 200 different people where each person has 2 video sequences, captured by two separate cameras. The person bounding box is manually labelled. **iLIDS-VID**

contains 600 image sequences of 300 pedestrians, captured by two non-overlapping cameras in an airport. Each of the training and test sets has 150 person identities. In this dataset, the target person is heavily occluded by other pedestrians or objects (*e.g.*, baggage). **MARS** is one of the largest video person re-ID datasets which contains 1,261 different identities and 20,715 video sequences captured by 6 separate cameras. The video sequences are generated by the GMMCP tracker [Dehghan et al., 2015], and for each frame, the bounding box is detected by DPM [Felzenszwalb et al., 2010]. The dataset is split into training and testing sets that contain 631 and 630 person identities, respectively. **DukeMTMC-VideoReID** is another large video person re-ID dataset. This dataset contains 702 pedestrians for training, 702 pedestrians for testing as well as 408 pedestrians as distractors. The training set and testing set has 2,196 video sequences and 2,636 video sequences, respectively. The person bounding boxes are annotated manually.

Following existing works, we use both the cumulative matching characteristic (CMC) curve and mean average precision (mAP) to evaluate the performance of the trained re-ID system.

4.4.3 Comparison to the State-of-the-Art Methods

To evaluate the superiority of our deep attentional network, we continue to compare our results with the current state-of-the-art approaches, shown in Table 4.1 and Table 4.2.

PRID-2011. On the PRID-2011 dataset, our network improves the state-of-the-art accuracy by 1.1% in R-1, compared to GLTR [Li et al., 2019a]. As for the mAP, our approach outperforms [Chen et al., 2018a] by 2.4%. When compared to SCAN [Zhang et al., 2018], which uses optical flow, our approach outperforms it by 1.3% in R-1.

iLIDS-VID. On the iLIDS-VID dataset, our approach improves the state-of-the-art mAP value by 5.2%, compared to [Chen et al., 2018a]. As for the R-1 accuracy, our approach also achieves a new state-of-the-art, outperforming [Zhao et al., 2019] by a comfortable 2.4%. In addition, our approach continues to outperform SCAN + optical flow [Zhang et al., 2018] by 0.7% in R-1.

MARS. On the MARS dataset, our approach achieves state-of-the-art performances on mAP and competitive performance on the CMC curve. In particular, our approach outperforms VRSTC [Hou et al., 2019b] on mAP, R-5 and R-10. It is worth mentioning that VRSTC uses a generator for data augmentation. Furthermore, when compared to other methods, we observe that our approach outperforms GLTR [Li et al., 2019a] by 1.3%/4.6% in R-1/mAP.

DukeMTMC-VideoReID. As for this new dataset, our network continues to show its superior performance (see Table 4.2). Our approach is superior to GLTR by 1.8% on mAP, and outperform the state-of-the-art mAP value of STA by 0.6%, and our network also achieves competitive performance on the CMC metric, outperforming the state-of the-art on R-5, R-10 and R-20.

Table 4.1: Comparison with the SOTA methods on PRID-2011, iLIDS-VID and MARS datasets. The 1st best in **bold font**.

Method	PRID-2011					iLIDS-VID					MARS				
	R-1	R-5	R-10	R-20	mAP	R-1	R-5	R-10	R-20	mAP	R-1	R-5	R-10	R-20	mAP
RFA-Net [Yan et al., 2016]	58.2	85.8	93.4	97.9	-	49.3	76.8	85.3	90.0	-	-	-	-	-	-
McLaughlin *et al.* [McLaughlin et al., 2016]	70.0	90.0	95.0	97.0	-	58.0	84.0	91.0	96.0	-	-	-	-	-	-
MSCAN [Li et al., 2017]	-	-	-	-	-	-	-	-	-	-	71.8	86.6	-	93.1	56.1
Zhou *et al.* [Zhou et al., 2017]	79.4	94.4	-	99.3	-	55.2	86.5	-	97.0	-	70.6	90.0	-	97.6	50.7
Chen *et al.* [Chen et al., 2018a]	88.6	99.1	-	-	90.9	79.8	91.8	-	-	82.6	81.2	92.1	-	-	69.4
+ Optcal flow	93.0	99.3	100.0	100.0	94.5	85.4	96.7	98.8	99.5	87.8	86.3	94.7	-	98.2	76.1
QAN [Liu et al., 2017d]	90.3	98.2	99.3	100.0	-	68.0	86.8	-	97.4	-	73.7	84.9	-	91.6	51.7
Li *et al.* [Li et al., 2018a]	93.2	-	-	-	-	80.2	-	-	-	-	82.3	-	-	-	65.8
Gao *et al.* [Gao and Nevatia, 2018]	-	-	-	-	-	-	-	-	-	-	83.3	93.8	96.0	97.4	76.7
SCAN [Zhang et al., 2018]	92.0	98.0	100.0	100.0	-	81.3	93.3	96.0	98.0	-	86.6	94.8	-	98.1	76.7
+ Optical flow	95.3	99.0	100.0	100.0	-	88.0	96.7	98.0	100.0	-	87.2	95.2	-	98.1	77.2
STIM-RRU [Liu et al., 2019b]	92.7	98.8	-	99.8	-	84.3	96.8	-	100.0	-	84.4	93.2	-	96.3	72.7
COSAM [Subramaniam et al., 2019]	-	-	-	-	-	79.6	95.3	-	-	-	84.9	95.5	-	97.9	79.9
STAR+Optical flow [Wu et al., 2019]	93.4	98.3	100.0	100.0	-	85.9	97.1	98.9	99.7	-	85.4	95.4	96.2	97.3	76.0
STA [Fu et al., 2019b]	-	-	-	-	-	-	-	-	-	-	86.3	95.7	-	98.1	80.8
VRSTC [Hou et al., 2019b]	-	-	-	-	-	83.4	95.5	97.7	99.5	-	**88.5**	96.5	97.4	-	82.3
Zhao *et al.* [Zhao et al., 2019]	93.9	99.5	-	100.0	-	86.3	97.4	-	99.7	-	87.0	95.4	-	98.7	78.2
GLTR [Li et al., 2019a]	95.5	**100.0**	-	-	-	86.0	98.0	-	-	-	87.0	95.7	-	98.2	78.4
Baseline	85.4	98.9	98.9	98.9	91.0	80.0	95.3	98.7	99.3	87.1	82.3	93.9	95.8	97.2	76.2
Ours	**96.6**	98.9	**100.0**	**100.0**	**96.9**	**88.7**	**97.3**	**99.3**	**100.0**	**93.0**	87.9	**96.6**	**97.5**	**98.8**	**83.0**

Table 4.2: Comparison with the SOTA methods on DukeMTMC-VideoReID dataset. The 1st best in **bold font**.

Method	DukeMTMC-VideoReID				
	R-1	R-5	R-10	R-20	mAP
ETAP-Net [Wu et al., 2018a]	83.6	94.6	-	97.6	78.3
STAR+Optical flow [Wu et al., 2019]	94.0	99.0	99.3	99.7	93.4
VRSTC [Hou et al., 2019b]	95.0	99.1	99.4	-	93.5
STA [Fu et al., 2019b]	96.2	99.3	-	99.7	94.9
GLTR [Li et al., 2019a]	**96.3**	99.3	-	99.7	93.7
Baseline	87.5	96.5	97.2	98.3	86.2
Ours	**96.3**	**99.4**	**99.7**	**99.9**	**95.5**

4.4.4 Ablation Study

This section demonstrates the effectiveness of the proposed blocks and the selection of appropriate hyper parameters via a thorough battery of experiments.

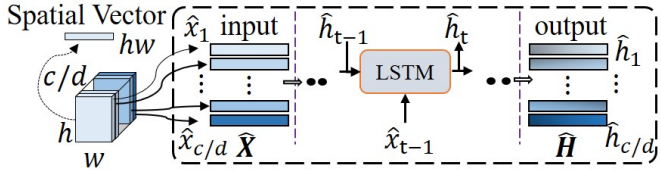
4.4.4.1 Effect of Channel Recurrent Attention

Here, we evaluate the effectiveness of the proposed channel recurrent attention, and verify our claim that our channel recurrent attention is able to capture more structure information as we sequentially feed the spatial vector to the LSTM. To show the design is reasonable, we compare our channel recurrent attention with two variations, namely, the spatial recurrent attention and the conv attention.

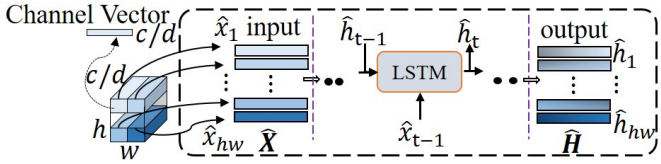
In the spatial recurrent attention, the LSTM receives a sequence of channel features from feature maps as input, with the recurrent operator along the spatial domain. In more detail, in channel recurrent attention (see Fig. 4.2), the input is a sequence of spatial vectors, (*e.g.*, $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{\frac{c}{d}}]^\top \in \mathbb{R}^{\frac{c}{d} \times hw}$). In the spatial recurrent attention, the input is a sequence of channel vectors, (*e.g.*, $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{hw}]^\top \in \mathbb{R}^{hw \times \frac{c}{d}}$). Though the recurrent operation along the spatial domain is also able to learn the pattern spatially, the spatial recurrent attention lacks explicit modelling in the spatial domain. Fig. 4.5 shows the schematic difference between channel recurrent attention (see Fig. 4.5(a)) and spatial recurrent attention (see Fig. 4.5(b)).

In addition, to verify the necessity of a global receptive field in our channel recurrent attention, we further replace the LSTM with a convolutional layer with a similar parameter size, which is called a conv attention. The architecture of the conv attention is shown in Fig. 4.6. In the Conv block, the kernel size is 3×3 and the sliding step is 1, and it produces a tensor with the shape of $\frac{c}{d} \times h \times w$. The generated attention mask can be formulated as $\boldsymbol{Z}_{\text{conv}} = \text{Sigmoid}\Big(\varphi\big(\text{Conv}(\phi(\boldsymbol{X}))\big)\Big)$, where $\text{Conv}(\cdot)$ indicates the convolutional operation.

Table 4.3 compares the effectiveness of three attention variations. It is shown that our channel recurrent attention has a superior performance over the other two variations. As can be observed, the channel recurrent attention cell improves the accuracy



(a) Channel recurrent attention.



(b) Spatial recurrent attention.

Figure 4.5: Schematic comparison between channel recurrent attention and spatial recurrent attention.

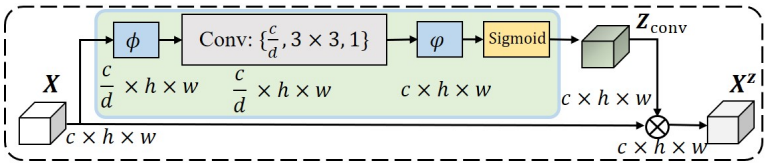


Figure 4.6: The architecture of the proposed conv attention module.

Table 4.3: Comparison of three attention variations across four datasets. CRA: Channel Recurrent Attention; SRA: Spatial Recurrent Attention; CA: Conv Attention. The 1st best in **bold font**.

	Model	PRID-2011		iLIDS-VID		MARS		DukeMTMC-VideoReID	
		R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
(i)	No Attention	85.4	91.0	80.0	87.1	82.3	76.2	87.5	86.2
(ii)	+ CRA	**92.1**	**94.6**	**87.0**	**90.6**	**86.8**	**81.6**	**94.7**	**94.1**
(iii)	+ SRA	87.9	92.1	83.3	87.4	84.6	78.4	89.4	87.8
(iv)	+ CA	89.6	92.8	84.2	88.2	85.2	79.7	91.2	90.1

significantly across all four datasets. This observation supports our assumption that the attention receives a performance gain from explicit modelling of the global receptive field in each slice of the feature maps.

4.4.4.2 Effect of the Position of Channel Recurrent Attention

The position of the channel recurrent attention block affects the information in the spatial or the channel dimensions. We want to explore the rich spatial and channel information; thus, we only consider the feature maps from the middle of the deep

network as input to channel recurrent attention (*i.e.*, P_1, P_2, and P_3 in Fig. 4.1). The comparison is illustrated in Table 4.4. It shows that the system receives a better gain when adding the channel recurrent attention module at position P_2, which aligns with our motivation that more spatial information is utilised in the feature maps. The works [Wang et al., 2018b; Fang et al., 2019] also present a similar observation. When applying the attention in P_1, P_2 and P_3, the network performs at its best.

Table 4.4: Effect of the position of channel recurrent attention across four datasets. CRA: Channel Recurrent Attention. The 1st best in **bold font**.

Model		PRID-2011		iLIDS-VID		MARS		DukeMTMC-VideoReID	
		R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
(i)	No Attention	85.4	91.0	80.0	87.1	82.3	76.2	87.5	86.2
(ii)	+ CRA in P_1	89.6	92.2	85.3	88.2	85.0	80.6	92.7	92.2
(iii)	+ CRA in P_2	91.0	94.4	86.7	90.2	86.4	81.2	94.2	93.4
(iv)	+ CRA in P_3	90.3	92.6	86.0	88.4	86.1	80.8	93.5	92.7
(v)	+ CRA in P_1&P_2&P_3	**92.1**	**94.6**	**87.0**	**90.6**	**86.8**	**81.6**	**94.7**	**94.1**

4.4.4.3 Effect of Reduction Ratio in Channel Recurrent Attention

The ratio $1/d$ in the embedding function $\phi(\cdot)$ (see Fig. 4.2) is to reduce the channel dimensionality of the input feature maps, consequently, reducing the sequence length input to the LSTM; thus, it is an important hyper-parameter in the channel recurrent attention. Table 4.5 reveals that the best performance is obtained when $d = 16$ for small-scale datasets and $d = 8$ for large-scale datasets. This could be due to the fact that training a network with a large amount of training samples is less prone to overfitting. Furthermore, this table also shows the fact that the LSTM has difficulties in modelling very long sequences (*e.g.*, smaller d in Table 4.5). However, when the sequences are too short (*e.g.*, $d = 32$), the channel features are compressed, such that some pattern information is lost.

Table 4.5: Effect of reduction ratio $1/d$ in channel recurrent attention across four datasets. The 1st best in **bold font**.

Reduction Ratio		PRID-2011		iLIDS-VID		MARS		DukeMTMC-VideoReID	
		R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
(i)	No Attention	85.4	91.0	80.0	87.1	82.3	76.2	87.5	86.2
(ii)	$d = 2$	88.7	92.1	84.0	88.7	84.8	80.2	93.4	92.8
(iii)	$d = 4$	89.8	92.6	85.6	89.1	85.2	80.3	93.9	93.4
(iv)	$d = 8$	91.0	93.2	86.3	89.4	**86.8**	**81.6**	**94.7**	94.1
(v)	$d = 16$	**92.1**	**94.6**	**87.0**	**90.6**	85.5	80.7	94.3	**94.3**
(vi)	$d = 32$	91.0	93.8	82.7	88.9	84.3	79.8	93.2	93.4

4.4.4.4 Why using LSTM in the Channel Recurrent Attention?

In our channel recurrent attention, we use the LSTM to perform the recurrent operation for the spatial vector. We observed that once the order of the spatial vectors is fixed, the recurrent operation in the LSTM is able to learn useful information along the channel dimension. We further investigated using Bi-LSTM to replace the LSTM in the attention and evaluate its performance. Compared with LSTM, the Bi-LSTM only brings a marginal/no performance gain across different datasets, whereas it almost doubles the number of parameters and FLOPs in the attention model. Please refer to §1 of the supplementary material for details of those experiments. These empirical experimental results support the use of a regular LSTM in our attention module.

4.4.4.5 Effect of Set Aggregation

Table 4.6 shows the effectiveness of set aggregation and the effectiveness of different pooling schemes in the set aggregation block. It is clear that the individual set aggregation improves the network performance and the combination of attention modules continues to increase the performance gain; showing that two attention modules mine complementary information in the network. Furthermore, all pooling schemes improve the results of the network, showing that the network receives gains from set aggregation. The combination of the average pooling and the max pooling scheme with non-sharing weights further shows its superiority over the individual average or max pooling schemes. This observation can be interpreted as the average and max pooled features have complementary information when encoding clip-level representations.

4.4.4.6 Visualisation of Channel Recurrent Attention

We visualise the feature maps from the baseline network and our channel recurrent attention network, trained on the MARS dataset. The feature maps are obtained in P_2 (see Fig. 4.1). In Fig. 4.7, we observed that compared to the baseline network, our attention network highlights more areas of human bodies, which verifies the effectiveness of our network qualitatively. Please refer to the supplementary material for further visualisations.

4.4.5 Further Analysis

In this part, extensive experiments are performed to choose a proper setting for the baseline network, including the number frames to use from a video clip, dimensionality of the video feature embedding and the training strategies (*e.g.*, pre-training and random erasing [Zhong et al., 2017b]). This ablation studies are performed on the iLIDS-VID and the MARS datasets.

Table 4.6: Effect of set aggregation across four datasets. CRA: Channel Recurrent Attention, SA: Set Aggregation, †: Sharing weights, ‡: Non-sharing weights. The 1st best in **bold font**.

	Method	PRID-2011		iLIDS-VID		MARS		DukeMTMC-VideoReID	
		R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
(i)	No Attention	85.4	91.0	80.0	87.1	82.3	76.2	87.5	86.2
(ii)	+ CRA	92.1	94.6	87.0	90.6	86.8	81.6	94.7	94.1
(iii)	+ SA (Average & Max Pooling)	87.6	92.3	84.7	89.1	85.2	80.5	91.2	88.9
(iv)	+ CRA & SA (Avg Pooling)	94.4	95.2	87.9	91.2	87.2	82.2	95.6	95.0
(v)	+ CRA & SA (Max Pooling)	93.3	94.8	87.3	90.8	86.9	81.2	95.2	94.6
(vi)	+ CRA & SA† (Avg & Max Pooling)	95.5	96.1	88.2	92.4	87.7	82.6	95.9	95.3
(vii)	+ CRA & SA‡ (Avg & Max Pooling)	**96.6**	**96.9**	**88.7**	**93.0**	**87.9**	**83.0**	**96.3**	**95.5**

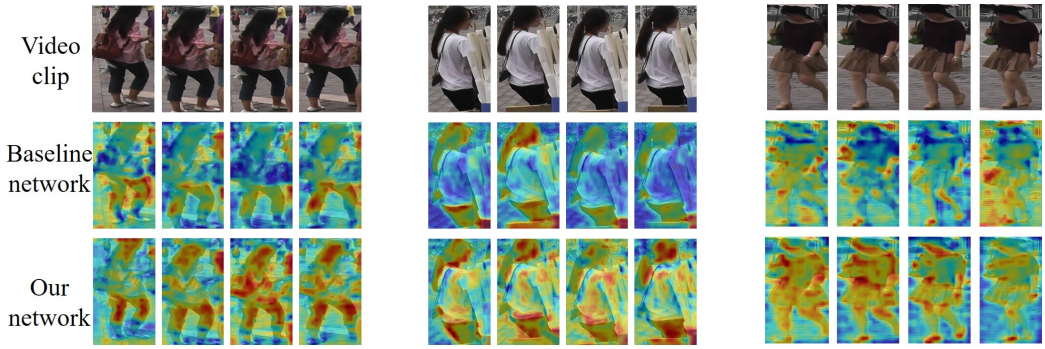


Figure 4.7: Visualisation of our channel recurrent attention in video clips, sampled from MARS dataset. We sample three video clips from different pedestrians and visualise the feature maps. In the heat map, the response increases from blue to red. Best Viewed in colour.

4.4.5.1 Number of Frames in Video Clip

First, we perform experiments with a different number of frames (*i.e.*, t) in a video clip. When $t = 1$, it is reduced to the single image-based model. From Table 4.7, we observe that $t = 4$ achieves the highest accuracy in both R-1 and mAP values. Thus we use $t = 4$ in our work.

Table 4.7: Effect of the number of frames in a video clip on the iLIDS-VID and the MARS datasets. The 1st best in **bold font**.

Num of Frames		iLIDS-VID		MARS	
		R-1	mAP	R-1	mAP
(i)	$t = 1$	76.3	84.2	79.2	74.3
(ii)	$t = 2$	79.3	86.1	81.5	75.6
(iii)	$t = 4$	**80.0**	**87.1**	**82.3**	**76.2**
(iv)	$t = 8$	79.6	86.4	82.1	76.0

4.4.5.2 Dimensionality of Video Feature Embedding

The dimension, *i.e.*, D_v, of the video feature embedding is evaluated and illustrated in Table 4.8 on both the iLIDS-VID and the MARSdatasets. On iLIDS-VID, it is clear that the video feature embedding with $D_v = 1024$ performs better for both R-1 and mAP accuracy. Therefore, we choose $D_v = 1024$ as the dimension of the feature embedding across all datasets. On the MARS dataset, we observe that R-1 has the peak value when $D_v = 512$, while mAP achieves the peak value when $D_v = 1024$. However, the mAP value in $D_v = 512$ is much lower than that in $D_v = 1024$. Thus we also choose $D_v = 1024$ for MARS.

Table 4.8: Effect of the dimensionality of video feature embedding on the iLIDS-VID and the MARS datasets. The 1st best in **bold font**.

Dim of Embedding		iLIDS-VID		MARS	
		R-1	mAP	R-1	mAP
(i)	$D_v = 128$	72.0	81.0	82.0	75.1
(ii)	$D_v = 256$	73.3	82.5	82.4	76.3
(iii)	$D_v = 512$	76.6	85.5	**82.6**	75.2
(iv)	$D_v = 1024$	**80.0**	**87.1**	82.3	**76.2**
(v)	$D_v = 2048$	79.6	86.5	82.0	75.6

4.4.5.3 Training Strategies

We further analyse the effect of different training strategies of the deep network (*e.g.*, random erasing, pre-training model) in Table 4.9 on both the iLIDS-VID and the MARS datasets. Here, $\mathcal{F}$ denotes the backbone network (see Fig. 4.1). PRE and RE denote pre-training on imageNet Russakovsky et al. [2015] and random erasing data augmentation, respectively. This table reveals that both training components of pre-training (*i.e.*, Num (ii)) and random erasing (*i.e.*, Num (iii)) improve the R-1 and mAP values, compared to the baseline (*i.e.*, Num (i)). In addition, the network continues to improve its performance when both training strategies are employed, showing that those two training strategies work in a complementary fashion. Thus we choose the network with the pre-trained model and random erasing as our baseline network.

Table 4.9: Effect of the different training strategies on the iLIDS-VID and the MARS datasets. $\mathcal{F}$, PRE and RE denote backbone network, pre-training and random erasing, respectively. The 1st best in **bold font**.

Model		iLIDS-VID		MARS	
		R-1	mAP	R-1	mAP
(i)	$\mathcal{F}$	60.8	67.6	76.4	71.8
(ii)	$\mathcal{F}$ + PRE	70.8	81.6	81.1	75.4
(iii)	$\mathcal{F}$ + RE	65.3	74.6	78.8	74.5
(iv)	$\mathcal{F}$ + PRE + RE	**80.0**	**87.1**	**82.3**	**76.2**

4.5 Experiments on Image Person Retrieval

To show the generalisation of the proposed channel recurrent attention, we employ it in a single image pedestrian retrieval task. We select a strong baseline network from [Fang et al., 2019], and insert the channel recurrent attention after each convolutional block. The deep network is fine-tuned from ImageNet pre-training [Rus-

sakovsky et al., 2015] and trained with the same hyper-parameter setting as in [Fang et al., 2019]. We use **CUHK01** [Li et al., 2012] and **DukeMTMC-reID** [Ristani et al., 2016] to evaluate the performance of the network. CHUK01 contains 3,884 images of 971 identities. The person images are collected by two cameras with each person having two images per camera view (i.e., , four images per person in total). The person bounding boxes are labelled manually. We adopt the 485/486 training/test data split protocol to evaluate our network. The DukeMTMC-reID is the image version of DukeMTMC-VideoReID dataset for the re-ID purpose. It has 1,404 identities and includes 16,522 training images of 702 identities, 2,228 query and 17,661 gallery images of 702 identities. The pedestrian bounding boxes are labelled manually. We use mAP and the CMC curve to evaluate the performance. Table 4.10 and Table 4.11 illustrate that our approach achieves competitive results to existing state-of-the-art approaches, showing the effectiveness and generalisation of our channel recurrent attention module.

Table 4.10: Comparison with the SOTA on CUHK01 dataset. The 1st best in **bold font**.

Method	CUHK01			
	R-1	R-5	R-10	R-20
Zhao *et al.* [Zhao et al., 2017c]	75.0	93.5	95.7	97.7
Spindle Net [Zhao et al., 2017b]	79.9	94.4	97.1	98.6
PBR [Suh et al., 2018]	80.7	94.4	97.3	98.6
Baseline	79.3	92.7	95.8	98.2
Ours	**83.3**	**96.3**	**98.4**	**98.9**

Table 4.11: Comparison with the SOTA on DukeMTMC-reID dataset. The 1st best in **bold font**.

Method	DukeMTMC-reID			
	R-1	R-5	R-10	mAP
OS-Net [Zhou et al., 2019a]	88.6	-	-	73.5
BAT-net [Fang et al., 2019]	87.7	94.7	96.3	77.3
ABD-Net [Chen et al., 2019b]	89.0	-	-	**78.6**
Baseline	85.4	93.8	95.5	75.0
Ours	**89.2**	**95.6**	**96.9**	78.3

4.6 Summary

This chapter proposes a novel deep attentional network for task of video pedestrian retrieval. This network benefits from the developed channel recurrent attention and set aggregation modules. The channel recurrent attention module is employed for a global view to feature maps, to learn the channel and spatial pattern jointly, given a frame feature maps as input. Then the set aggregation cell continues to re-weight each frame feature and fuses them to get a compact clip representation. Thorough evaluation shows that the proposed deep network achieves state-of-the-art results across four standard video-based person re-ID datasets, and the effectiveness of each attention is further evaluated by extensive ablation studies.

In this part, we develop some attention mechanisms for embedding learning and show that a properly designed attention module can significantly improve the embedding quality. However, this is not the only way to improve visual embeddings. In the next part, we will investigate how the embedding space benefits from geometry constraints.

Part II

Visual Embedding Learning: Geometry

Chapter 5

Set Augmented Triplet Loss

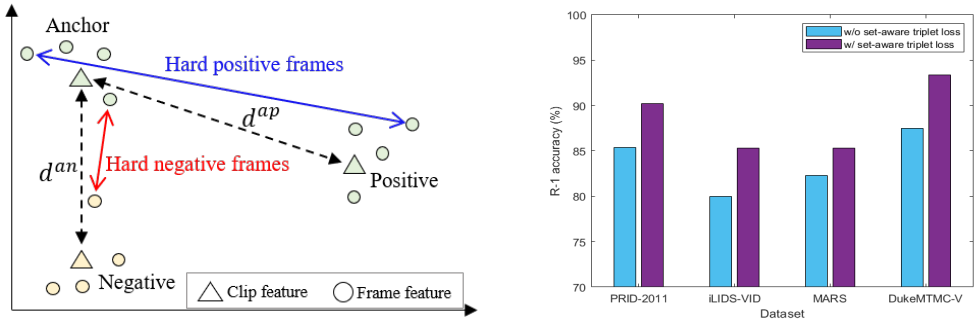
The previous part of this Thesis illustrates the effectiveness of the attention mechanism for embedding learning on visual data. Starting from this chapter, we will study how to improve visual embeddings using geometry constraints.

Modern video person re-identification (re-ID) machines are often trained using a metric learning approach, supervised by a triplet loss. The triplet loss used in video re-ID is usually based on so-called clip features, each aggregated from a few frame features. In this chapter, we propose to model the video clip as a set and instead study the distance between sets in the corresponding triplet loss. In contrast to the distance between clip representations, the distance between clip sets considers the pair-wise similarity of each element (*i.e.*, frame representation) between two sets. This allows the network to directly optimise the feature representation at a frame level. Apart from the commonly-used set distance metrics (*e.g.*, ordinary distance and Hausdorff distance), we further propose a hybrid distance metric, tailored for the set-aware triplet loss. Also, we propose a hard positive set construction strategy using the learned class prototypes in a batch. Our proposed method achieves state-of-the-art results across several standard benchmarks, demonstrating the advantages of the proposed method. This chapter is based on our published work [Fang et al., 2021b].

5.1 Introduction

In this chapter, we aim to create compact yet discriminative features from videos for accurate video re-ID. This is realised by learning the embedding of the video clip, modelled by a set, which is optimised by the proposed *set augmented triple loss*.

The pipeline of training a typical video re-ID machine consists of first extracting the frame-level features with the help of a deep network backbone and then aggregating them to a clip-level feature. In video re-ID, the ranking task (*i.e.*, triplet loss) is a popular choice to supervise the network to learn an embedding space, w.r.t. the clip-level features. This, however, could lead to sub-optimal learning of the video embedding space, as the aggregation operation to frame features will result in loss of information of the original frame features. Specifically, in the video-based applications, the triplet loss considers the distance between the clip representations (*i.e.*,



(a) (b)

Figure 5.1: (a): Geometry interpretation of the distance metrics for clip representation and frame representation. The colour represents the class of samples. d^{ap} and d^{an} denote the distance from positive pair and negative pair in a clip level. However, those two distances cannot reveal the original distribution of frame features, thereby ignoring the distance between hard frames (*i.e.*, $\leftrightarrow$ for hard negative pair and $\leftrightarrow$ for hard positive pair). (b): The comparison of R-1 accuracy from the networks trained without set-aware triplet loss and with set-aware triplet loss, across four datasets. The backbone network is ResNet-50, pre-trained on ImageNet. In the set-aware triplet loss, we use the proposed hybrid set distance metric to calculate the distance of anchor-positive pair and anchor-negative pair.

d^{an} and d^{ap} in Fig. 5.1(a)), which only indirectly penalises the hard frames between the clips (*i.e.*, hard positive frames and hard negative frames in Fig. 5.1(a)). This observation motivates us to directly leverage the frame features, to decrease the hard positive distance (*i.e.*, $\leftrightarrow$ in Fig. 5.1(a)) and increase the hard negative distance (*i.e.*, $\leftrightarrow$ in Fig. 5.1(a)) for frame features.

In video re-ID, we often aggregate the frame features (*i.e.*, $\{\boldsymbol{f}_i^1, \ldots, \boldsymbol{f}_i^t\}, \boldsymbol{f}_i^j \in \mathbb{R}^c, j = 1, \ldots, t$) to a clip-level representation (*i.e.*, $\hat{\boldsymbol{f}}_i \in \mathbb{R}^c$) using an aggregation function (*i.e.*, $\mathrm{Agg}(\cdot)$). This processing can be summarised as: F

$$\hat{\boldsymbol{f}}_i = \mathrm{Agg}(\{\boldsymbol{f}_i^1, \ldots, \boldsymbol{f}_i^t\}) = \phi\Big(\sum_{j=1}^{t}(\omega^j \boldsymbol{f}_i^j)\Big), \tag{5.1}$$

where $\phi(\cdot)$ and $\{\omega^1, \ldots, \omega^t\} \in \mathbb{R}^t$ denote non-linear mapping and aggregation weights, respectively. Due to the summation operator in Eq. (5.1), the clip feature (*i.e.*, $\hat{\boldsymbol{f}}_i$) is invariant to the order of frame features, indicating that the aggregation function is temporally invariant. In other words, the aggregation function acts on sets, in the sense that the response of the aggregation function is "insensitive" to the ordering of elements in the input [Zaheer et al., 2017]. With this intuition, we aim to use the theory of sets to make better use of the frame features within each video clip.

In this chapter, we propose to model the frame features within a clip as a set and propose to use the distance between sets in the triplet loss. Different from the L_2 dis-

tance between the aggregated clip features (see Fig. 5.2(a)), the distance between sets considers every pair-wise distance in two sets and explores more information of the frame features. In set theory, the distance between sets is usually measured by ordinary distance (see Fig. 5.2(b)) or Hausdorff distance (see Fig. 5.2(c)). However, these set distance measures cannot fully utilise hard frames (*i.e.*, hard positive and hard negative) in a triplet. To construct an effective set triplet loss, we further propose a hybrid distance metric (see Fig. 5.2(d)), where the hard frames for anchor-positive and anchor-negative sets are considered explicitly. In essence, our hybrid distance metric aims at penalising the hard frames between sets (*i.e.*, $\leftrightarrow$ and $\leftrightarrow$ in Fig. 5.1(a)). Fig. 5.1(b) shows the comparison of retrieval accuracies from video re-ID models, trained *without* our set-aware triplet loss, and *with* our set-aware triplet loss, across four video re-ID datasets. We further apply the class prototypes to frame-level features to construct hard sets by comparing the similarity between the class prototype and frame feature with the same instance. Then the constructed set acts as a hard positive set.

The **contributions** of this chapter are summarised as follows:

- We model the video clip as a set[1], and employ the distance metric between sets to construct the triplet loss. Furthermore, we propose a new hybrid set distance metric, which is tailored for the set triplet loss.
- We further model the weights in the last classification layer as class prototypes, to construct a hard positive set, w.r.t. each anchor set with the same identity.
- Our algorithm achieves state-of-the-art performance across four standard video person re-ID datasets (*i.e.*, PRID-2011 [Hirzer et al., 2011], iLIDS-VID [Wang et al., 2016], MARS [Zheng et al., 2016] as well as DukeMTMC-VideoReID [Wu et al., 2018a]), showing the effectiveness of the proposed set augmented triplet loss.

5.2 Related Work

In this section, we review the related work on set learning and metric learning.

Sets. The concept of modelling the training data as a set has appeared in many applications, *e.g.*, point cloud classification [Zaheer et al., 2017], image tagging [Zaheer et al., 2017], object localisation [Ribera et al., 2019] *et al.*. In general, the response of set functions is insensitive to the order of the elements in the set and the work in [Zaheer et al., 2017] studies the structure of such functions. The most popular function is the pooling operation (*i.e.*, max pooling, average pooling) across the elements of its input. For example, deep Convolutional Neural Networks (CNNs) use pooling layers to summarise the features in a patch [He et al., 2016]. In the point cloud classification task [Qi et al., 2016], a non-linear function extracts the latent representation of point coordination and the pooling function further summarises the

[1]In the remainder of this chapter, we will use "clip" and "set" interchangeably

features of objects. Attention using non-local connections also acts as a set function as the attention weights are produced by pairwise similarities of pixel features [Wang et al., 2018b]. In [Ribera et al., 2019], the locations of objects are estimated by training a detector which minimises the set distance between the prediction and ground truth of objects.

Metric Learning. Deep metric learning aims to project images to a low dimensional embedding space, in which the images with similar semantics are clustered together [Suh et al., 2019; Roy et al., 2019; Fang et al., 2019]. The most popular paradigm is to employ the triplet loss to penalise the positive pair or negative pair or both of them within a triplet [Schroff et al., 2015]. However, the possible number of triplets is exponential to the number of samples in a mini-batch, leading to a prohibitive computational cost. Much effort has gone into mining the triplets efficiently [Hermans et al., 2017; Fang et al., 2019; Suh et al., 2019]. For example, the hard mining strategy only selects the hard positive and hard negative for an anchor sample [Hermans et al., 2017]. However, a hard mining strategy often leads to getting caught in local minima during optimisation [Hermans et al., 2017]; thus the semi-hard mining method is further proposed to make use of more negative pairs [Fang et al., 2019]. Beyond mining the triplets in a mini-batch, the work in [Suh et al., 2019] employs the class signatures to mine hard negative classes for an anchor class in the whole dataset.

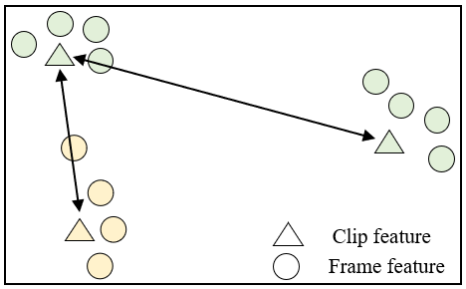
5.3 Set Augmented Triplet Loss

5.3.1 Triplet Loss

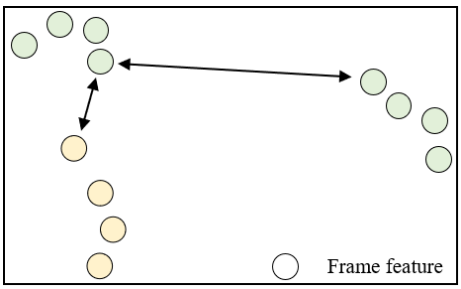
When training a deep video feature extractor, we first sample a mini-batch, which contains P different classes and K video clips for each class, with each video clip having T frames. The network first extracts the frame features, denoted by $\boldsymbol{A}_i = \{\boldsymbol{a}_i^1, \ldots, \boldsymbol{a}_i^T\}, i = 1, \ldots, PK$. Then the network aggregates the frame features to a clip feature as $\hat{\boldsymbol{a}}_i = \text{Agg}(\boldsymbol{A}_i)$. Given an anchor clip representation $\hat{\boldsymbol{a}}_i$, one possible triplet is formed as $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_i^+, \hat{\boldsymbol{a}}_i^-\}$, where the positive pair (*i.e.*, $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_i^+\}$) shares the same label, while the negative pair (*i.e.*, $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_i^-\}$) does not. The triplet loss aims to penalise the triplet in which the distance between the positive pair is not sufficiently smaller than that between the negative pair. The triplet loss with hard triplet mining is given by

$$\mathcal{L}_{\text{ctri}}^{\text{hm}} = \frac{1}{PK} \sum_{i=1}^{PK} \left[d_i(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_i^+) - d_i(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_i^-) + \eta \right]_+, \tag{5.2}$$

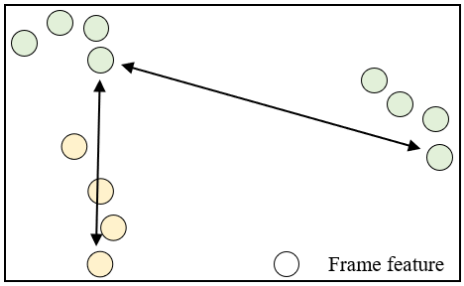
where $[\cdot]+ = \max(\cdot, 0)$, η is a task-specific margin, and d_i indicates the distance. Existing video re-ID machines [Fu et al., 2019b; Gao and Nevatia, 2018] only optimise the clip representation (see Fig. 5.2(a)) and it has never been considered to optimise the frame features within each video clip.



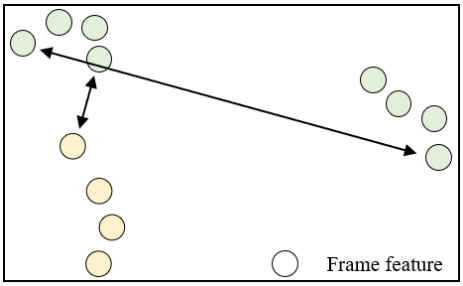
(a) Clip distance metric



(b) Ordinary distance metric



(c) Hausdorff distance metric



(d) Hybrid distance metric

Figure 5.2: Geometry interpretation of different distance metrics in a triplet. (a), (b), (c), and (d) denote L_2 distance metric between clip representation, ordinary distance metric, Hausdorff distance metric, and hybrid distance metric between sets. The color represents the class of samples.

5.3.2 Set-aware Triplet Loss

The nature of the triplet loss is to penalise the positive pairs with a large distance and negative pairs with a small distance. It works well in image re-ID where the triplets are constructed from the image features. However, in video re-ID, the distance measure is hampered by the aggregation operation, as shown in Fig. 5.1(a). To overcome this issue, we directly enforce the constraint of the triplet loss on the frame features. We first model the frame features within a video clip as a set and employ set theory to calculate the distance between sets. Eq. (2.13) and Eq. (2.14) formulate the commonly used set distance metrics. However, the geometry interpretation of Eq. (2.13) and Eq. (2.14) (see Fig. 5.2(b) and Fig. 5.2(c)) indicates that those two distance metrics cannot distinguish the distances from the hard positive frames ($\leftrightarrow$ in Fig. 5.1(a)) and hard negative frames ($\leftrightarrow$ in Fig. 5.1(a)) simultaneously. Thus, we further propose a hybrid distance metric tailored to the nature of the triplet loss.

Given a triplet, *i.e.*, $\{\boldsymbol{A}, \boldsymbol{A}^+, \boldsymbol{A}^-\}$, the hybrid distance metric is defined using the anchor-positive distance and anchor-negative distance individually, as follows:

$$D^{hd+}(\boldsymbol{A}, \boldsymbol{A}^+) = \sup_{\boldsymbol{a}\in\boldsymbol{A}, \boldsymbol{a}^+\in\boldsymbol{A}^+} d(\boldsymbol{a}, \boldsymbol{a}^+), \tag{5.3}$$

and

$$D^{hd-}(\boldsymbol{A}, \boldsymbol{A}^-) = \inf_{\boldsymbol{a}\in\boldsymbol{A}, \boldsymbol{a}^-\in\boldsymbol{A}^-} d(\boldsymbol{a}, \boldsymbol{a}^-), \tag{5.4}$$

where D^{hd+} and D^{hd-} denote the positive pair distance and negative pair distance, respectively. Fig. 5.2(d) shows the geometrical interpretation of the hybrid distance metric. This formulation allows the loss to penalize the hard frames in each set with the set-aware triplet loss:

$$\mathcal{L}_{\text{stri}}^{\text{hm}} = \frac{1}{PK}\sum_{i=1}^{PK}\left[0, D_i^{hd+} - D_i^{hd-} + \eta\right]_+. \tag{5.5}$$

5.3.3 Hard Positive Set Construction

The network is also supervised by a cross-entropy loss to minimise the within-class variance. Once the network aggregates the frame features to a clip feature as $\hat{\boldsymbol{a}}_i = \text{Agg}(\boldsymbol{A}_i)$. A following fully connected (FC) layer, parameterised by $\boldsymbol{W}$, is used to predict the identity of the video, normalised by the softmax function, as $\boldsymbol{p} = \text{softmax}(\boldsymbol{W}^\top \hat{\boldsymbol{a}}_i)$. A cross-entropy loss is employed to maximise the log likelihood of $\hat{\boldsymbol{a}}_i$ with respect to its label c as follows:

$$\mathcal{L}_{\text{ce}} = \frac{1}{PK}\sum_{i=1}^{PK} -\log\big(p(y_i = c|\hat{\boldsymbol{a}}_i)\big). \tag{5.6}$$

In Eq. (5.6), it holds that $p(y_i = c|\hat{\boldsymbol{a}}_i) \propto \boldsymbol{w}_c^\top \hat{\boldsymbol{a}}_i$. The optimisation will maximise $p(y_i = c|\hat{\boldsymbol{a}}_i)$, thereby maximising the similarity between $\boldsymbol{w}_c$ and $\hat{\boldsymbol{a}}_i$. Thus $\boldsymbol{w}_c$ can be understood as a prototype feature for the class c. Given K sets containing the same class c in one mini-batch, we can further approximate the probability of each frame feature belonging to its label as: $p(y_j = c|\boldsymbol{a}_j), j = 1, \ldots, KT$. For each class, we continue to mine T frame features $\hat{\boldsymbol{A}} = \{\boldsymbol{a}_r : r \in \boldsymbol{i}'\}$, where $\boldsymbol{i}'$ satisfies

$$\boldsymbol{i}' = \{r : \underset{r=1,\cdots,KT}{\arg\min}\, p_r;\ \text{s.t.}|\boldsymbol{i}'| = T\}, \tag{5.7}$$

and this set is summarised to a set representation (*i.e.*, $\hat{\boldsymbol{a}} = \text{Agg}(\hat{\boldsymbol{A}})$), acting as a hard positive with respect to the original set features $\{\hat{\boldsymbol{a}}_1, \ldots \hat{\boldsymbol{a}}_K\}$ in the batch, where $\hat{\boldsymbol{a}}_i = \text{Agg}(\boldsymbol{A}_i)$. Finally, we could form hard positive pairs as $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}\}, i = 1, \ldots, K$. The hard positive pairs are also minimised by the triplet loss. Besides the hard positive set, we mine a hard negative clip representation to form a valid triple loss, denoted by $\mathcal{L}_{\text{ctri}}^{\text{hpsc}}$. Algorithm 1 summarises the process of constructing hard positives.

5.3.4 Network and Optimisation

Fig. 5.3 shows the architecture of the deep network. The network receives a batch of video clips as input and produces frame representations. The original frame features

Algorithm 1 Hard Positive Set Construction

Require: K: Number of sets; T: Number of frame features in each set with same class; $\boldsymbol{A}_i = \{\boldsymbol{a}_{i1}, \ldots, \boldsymbol{a}_{iT}\}$: A set of frame features; $\hat{\boldsymbol{a}}_i$: Set feature; $\boldsymbol{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$: Class prototypes; c: Class of sets

Ensure: $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}\}, i = 1, \ldots, K$: Hard positive pairs

1: Merging all sets with the same class: $\boldsymbol{A} = \{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_T\} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{TK}\}$

2: Calculate the probability of predicting class c for each frame:

$$p(y_j = c|\boldsymbol{a}_j) = \frac{\exp(\boldsymbol{w}_c^\top \boldsymbol{a}_j)}{\sum_{m=1}^{n} \exp(\boldsymbol{w}_m^\top \boldsymbol{a}_j)}, \quad j = 1 \ldots TK$$

3: Pick T frame features with the lowest probability, satisfying

$$\boldsymbol{i}' = \{r : \underset{r=1,\cdots,KT}{\arg\min}\, p_r;\ \text{s.t.}|\boldsymbol{i}'| = T\}$$

4: Construct a hard positive set: $\hat{\boldsymbol{A}} = \{\boldsymbol{a}_r : r \in \boldsymbol{i}'\}$

5: Summarize to hard positive set feature: $\hat{\boldsymbol{a}} = \text{Agg}(\hat{\boldsymbol{A}})$

6: Form hard positive pairs: $\{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}\}, i = 1, \ldots, K$

are used to model the set and supervised by the set-aware triplet loss. We further use our proposed hard positive set construction to form hard positive pairs. Then average pooling is used to summarise the clip features. A vanilla triplet loss with hard mining and a triplet loss with hard positive set construction are utilised to supervise the clip features. An additional classifier is further used to train the network. The network is trained to update the parameters by jointly minimising the multiple triplet losses and cross-entropy loss. The total loss function is formally formulated as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{ce}} + \lambda_2 \mathcal{L}_{\text{ctri}}^{\text{hm}} + \lambda_3 \mathcal{L}_{\text{ctri}}^{\text{hpsc}} + \lambda_4 \mathcal{L}_{\text{stri}}^{\text{hm}}, \tag{5.8}$$

where $\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{ctri}}^{\text{hm}}$, $\mathcal{L}_{\text{ctri}}^{\text{hpsc}}$ and $\mathcal{L}_{\text{stri}}^{\text{hm}}$ denote cross entropy loss, clip-feature triplet loss with hard mining, clip-feature triplet loss with hard positive set construction, and set-aware triplet loss with hard mining. The loss terms are weighted by the factors $[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$.

5.4 Experiments on Video Person Retrieval

5.4.1 Implementation Details

Network and Data Organisation. We implement all experiments using the PyTorch [Paszke et al., 2017] machine learning package. We use ResNet-50 [He et al., 2016], SE-ResNet-50 [Hu et al., 2018] and GLTR [Li et al., 2019a] as baseline networks to evaluate our approach. Noted that the GLTR is self implemented version. All baselines are pre-trained on ImageNet [Russakovsky et al., 2015]. The baseline network extracts each frame feature to the dimension of 2048 and we further project them to

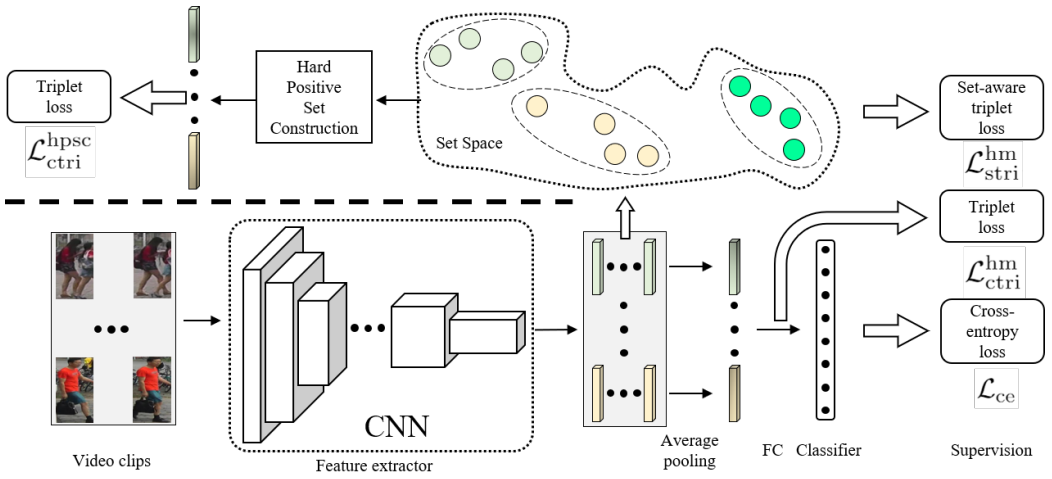


Figure 5.3: The architecture of the network, supervised by the proposed loss functions. The network receives frame images as input and produces the frame features. Then the network is trained by four losses, *i.e.*, $\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{ctri}}^{\text{hm}}$, $\mathcal{L}_{\text{stri}}^{\text{hm}}$ and $\mathcal{L}_{\text{ctri}}^{\text{hpsc}}$

a lower dimensional space of dimension 1024. Thereafter, a set of frame features are fused to a clip-level video representation and a linear-transformation layer is further utilised to predict the class of the video representation. In each video clip, T is chosen as 4 in all experiments and 4 frames are *randomly* sampled from a video sequence. The frames are first resized to 288×144, and then randomly cropped to 256×128. The data augmentations used in our experiments include randomly flipping in the horizontal direction and random erasing (RE) [Zhong et al., 2017b] during training. In the test phase, no data augmentation and re-ranking are used.

Optimisation Details. We train the network using the Adam [Kingma and Ba, 2014] optimiser with default momentum (*i.e.*, $[\beta_1, \beta_2] = [0.9, 0.999]$). The learning rate is initialized to 3e-4 for PRID-2011 and iLIDS-VID datasets, and 4e-4 for MARS and DukeMTMC-VideoReID datasets. During training, the learning rate is decayed by a fixed factor of 1e-1 at the 200^{th} and 400^{th} epoch for the PRID-2011 and iLIDS-VID , and the 100^{th}, 200^{th} and 500^{th} epoch for the MARS and DukeMTMC-VideoReID, respectively. The batch size is set to 16 for the PRID-2011 and iLIDS-VID datasets and 32 for the MARS and DukeMTMC-VideoReID datasets, respectively. In a mini-batch, both P and K are set to 4 for the PRID-2011 and iLIDS-VID, whereas $P = 8$, $K = 4$ for the MARS and DukeMTMC-VideoReID. The margin in Eq. (5.2) and Eq. (5.5), *i.e.*, η, is set to 0.3 for all datasets. $[\lambda_1, \lambda_2, \lambda_3, \lambda_4] = [1, 0.5, 0.5, 0.5]$. In § 5.4.4, we will verify each loss component in the total loss function. We report the results of the network at its 800^{th} epoch without any post processing tricks to boost the accuracy, *i.e.*, re-ranking.

5.4.2 Datasets and Evaluation Protocol

We evaluate our method on four popular video person re-identification benchmarks, including **PRID-2011** [Hirzer et al., 2011], **iLIDS-VID** [Wang et al., 2016], **MARS** [Zheng et al., 2016] and **DukeMTMC-VideoReID** [Wu et al., 2018a]. The **PRID-2011** consists of 200 identities, each with 2 video sequences, amounting to 400 video sequences in total. Both the train and test sets contain 100 person identities. The person trajectories are captured by two disjoint, static cameras. In each frame/image, the person bounding box is manually annotated. Similar to PRID-2011, **iLIDS-VID** is also a small scale dataset, which contains 600 video sequences of 300 identities, recorded by two cameras in an airport. Each of the train and test sets has 150 person identities. The main challenge of this dataset is the occlusion of the target person. **MARS** is one of the large-scale video datasets. It has 1,261 identities and 20,715 video sequences captured by 6 separate cameras. In this dataset, each video sequence is generated by the GMMCP tracker [Dehghan et al., 2015], and the bounding box of each frame is automatically detected by DPM [Felzenszwalb et al., 2010]. In this dataset, the train and test sets contain 631 and 630 person identities, respectively. The **DukeMTMC-VideoReID** is another large video re-ID dataset. This manually labelled dataset contains 702 pedestrians for training, 702 pedestrians for testing. Additionally, this dataset further employs 408 extra pedestrians as distractors. Those 1812 identities have 4832 video sequences.

Mean average precision (mAP) and cumulative matching characteristic (CMC) metrics are used to evaluate the proposed method. We report R-1, R-5, R-10 and R-20 values in the CMC metric.

5.4.3 Comparison to the State-of-the-Art Methods

We first compare our method to existing state-of-the art algorithms, as shown in Table 5.1 and Table 5.2.

PRID-2011. PRID-2011 is an old video re-ID dataset; thus only a few methods report the mAP value. To show the superiority of our method, we report both metrics for comparison in Table 5.1. Our method outperforms MG-RAFA [Zhang et al., 2020c] by 0.7% on the R-1 value. Our approach also outperforms the state-of-the-art mAP value in [Chen et al., 2018a] by 2.7%.

iLIDS-VID. Same as for the PRID-2011 dataset, we report the CMC accuracy and mAP value in Table 5.1. On the iLIDS-VID dataset, our method also achieves state-of-the-art performance. In particular, our network has the same R-1 value with MG-RAFA [Zhang et al., 2020c] and outperforms the state-of-the art mAP values by 5.1% in [Chen et al., 2018a].

MARS. Compared with MG-RAFA [Zhang et al., 2020c], the state-of-the-art algorithm on the MARS dataset (see Table 5.1), our method improves the R-5 and R-20 by 0.2% and 0.4% and achieves competitive performance on the R-1 and mAP value.

DukeMTMC-VideoReID. We further evaluate our method on the DukeMTMC-VideoReID dataset. Table 5.2 compares the performance between our network and existing state-of-the-art algorithms and demonstrates that our method outperforms

Table 5.1: Comparison with the SOTA methods on PRID-2011, iLID-VID and MARS datasets. † indicates the self-implemented network. The 1st best in **bold font**.

Method	PRID-2011					iLIDS-VID					MARS				
	R-1	R-5	R-10	R-20	mAP	R-1	R-5	R-10	R-20	mAP	R-1	R-5	R-10	R-20	mAP
Chen *et al.* [Chen et al., 2018a]	88.6	99.1	-	-	90.9	79.8	91.8	-	-	82.6	81.2	92.1	-	-	69.4
+ Optcal flow	93.0	99.3	100.0	100.0	94.5	85.4	96.7	98.8	99.5	87.8	86.3	94.7	-	98.2	76.1
QAN [Liu et al., 2017d]	90.3	98.2	99.3	100.0	-	68.0	86.8	-	97.4	-	73.7	84.9	-	91.6	51.7
Li *et al.* [Li et al., 2018a]	93.2	-	-	-	-	80.2	-	-	-	-	82.3	-	-	-	65.8
PBR [Suh et al., 2018]	-	-	-	-	-	-	-	-	-	-	83.0	92.8	95.0	96.8	72.2
SCAN [Zhang et al., 2018]	92.0	98.0	100.0	100.0	-	81.3	93.3	96.0	98.0	-	86.6	94.8	-	98.1	76.7
+ Optical flow	95.3	99.0	100.0	100.0	-	88.0	96.7	98.0	100.0	-	87.2	95.2	-	98.1	77.2
STIM-RRU [Liu et al., 2019b]	92.7	98.8	-	99.8	-	84.3	96.8	-	100.0	-	84.4	93.2	-	96.3	72.7
COSAM [Subramaniam et al., 2019]	-	-	-	-	-	79.6	95.3	-	-	-	84.9	95.5	-	97.9	79.9
STAR+Optical flow [Wu et al., 2019]	93.4	98.3	100.0	100.0	-	85.9	97.1	98.9	99.7	-	85.4	95.4	96.2	97.3	76.0
STA [Fu et al., 2019b]	-	-	-	-	-	-	-	-	-	-	86.3	95.7	-	98.1	80.8
VRSTC [Hou et al., 2019b]	-	-	-	-	-	83.4	95.5	97.7	99.5	-	88.5	96.5	**97.4**	-	82.3
Zhao *et al.* [Zhao et al., 2019]	93.9	99.5	-	100.0	-	86.3	97.4	-	99.7	-	87.0	95.4	-	98.7	78.2
GLTR [Li et al., 2019a]	95.5	100.0	-	-	-	86.0	98.0	-	-	-	87.0	95.7	-	98.2	78.4
MG-RAFA [Zhang et al., 2020c]	95.9	99.7	-	100.0	-	**88.6**	98.0	-	99.7	-	88.8	97.0	-	98.5	**85.9**
STGCN [Yang et al., 2020]	-	-	-	-	-	-	-	-	-	-	**89.9**	96.4	-	98.3	83.7
ResNet-50	85.4	98.9	98.9	98.9	91.0	80.0	95.3	98.7	99.3	87.1	82.3	93.9	95.8	97.2	76.2
+ Set Triplet Loss (Ours)	90.2	99.6	100.0	100.0	93.6	85.3	96.0	98.6	99.4	90.4	85.3	95.4	97.1	98.2	81.8
SE-ResNet-50	89.9	98.9	100.0	100.0	94.3	84.0	96.0	98.7	99.3	89.5	85.2	95.3	97.0	97.8	80.0
+ Set Triplet Loss (Ours)	**96.6**	**100.0**	**100.0**	**100.0**	**97.2**	**88.6**	**98.6**	98.7	**100.0**	**92.9**	87.9	**97.2**	97.1	**98.9**	83.2
GLTR†	94.4	99.7	100.0	100.0	95.3	85.2	96.7	97.3	99.7	91.1	86.4	95.4	96.9	97.7	78.8
+ Set Triplet Loss (Ours)	**96.6**	**100.0**	**100.0**	**100.0**	96.9	88.0	98.0	**99.3**	100.0	92.5	87.8	95.5	97.0	97.9	82.2

the STGCN [Yang et al., 2020] by 0.2% in mAP. Our methods also outperform STA [Fu et al., 2019b] by 0.2%/0.8% and GLTR by 0.1%/2.0% in R-1/mAP respectively.

Table 5.2: Comparison with the SOTA methods on the DukeMTMC dataset. † indicates the self-implemented network. The 1st best in **bold font**.

Method	DukeMTMC-VideoReID				
	R-1	R-5	R-10	R-20	mAP
ETAP-Net [Wu et al., 2018a]	83.6	94.6	-	97.6	78.3
STAR+Optical flow [Wu et al., 2019]	94.0	99.0	99.3	99.7	93.4
VRSTC [Hou et al., 2019b]	95.0	99.1	99.4	-	93.5
STA [Fu et al., 2019b]	96.2	99.3	-	99.7	94.9
GLTR [Li et al., 2019a]	96.3	99.3	-	99.7	93.7
STGCN [Yang et al., 2020]	**97.3**	99.3	-	99.7	95.7
ResNet-50	87.5	96.5	97.2	98.3	86.2
+ Set Triplet Loss (Ours)	93.4	98.4	99.8	99.2	91.9
SE-ResNet-50	90.2	97.3	98.0	98.9	89.7
+ Set Triplet Loss (Ours)	96.8	**99.4**	**99.9**	**99.9**	**95.9**
GLTR†	96.0	99.2	99.3	99.5	93.5
+ Set Triplet Loss (Ours)	97.1	**99.4**	99.8	**99.9**	95.4

5.4.4 Ablation Study

In this section, we will conduct extensive experiments to evaluate the effectiveness of each component in this work.

5.4.4.1 Effect of Set-aware Triplet Loss

We first evaluate the effectiveness of set-aware triplet loss with different set distance metrics. In this study, we use the SE-ResNet-50 as the backbone network and employ all three distance metrics for the set-aware triplet loss. As shown in Table 5.3, the set-aware triplet loss indeed helps the network to learn a discriminative person description. Compared with the commonly-used set distance metrics (*i.e.*, ordinary distance, Hausdorff distance), the proposed hybrid distance metric brings the largest performance gain, showing that the optimisation to hard frames of anchor-positive pairs and anchor-negative leads the network to create a discriminative video representation.

5.4.4.2 Effect of Hard Positive Set Construction

We continue to verify the effectiveness of our hard positive set construction method. We still use the SE-ResNet-50 as the backbone network. Table 5.4 shows that our network benefits from the hard positive set construction method across two datasets. A reasonable explanation for this improvement is that the hard positive sample helps

Table 5.3: Effect of the set-aware triplet loss across the iLIDS-VID and DukeMTMC-VideoReID datasets. SATL: set-aware triplet loss, D^o: ordinary distance, D^h: Hausdorff distance, D^{hd}: Hybrid distance. The 1st best in **bold font**.

Model	iLIDS-VID		DukeMTMC-VideoReID	
	R-1	mAP	R-1	mAP
SE-ResNet-50	84.0	89.5	90.2	89.7
SATL w/ D^o	86.8	90.6	92.8	91.7
SATL w/ D^h	87.6	91.1	94.1	92.9
SATL w/ D^{hd}	**88.3**	**91.9**	**94.9**	**93.7**

the network minimise the intra-class variance, thereby improving the performance of the network.

Table 5.4: Effect of the hard positive set construction across the iLIDS-VID and the DukeMTMC-VideoReID datasets. HPSC: hard positive set construction. The 1st best in **bold font**.

Model	iLIDS-VID		DukeMTMC-VideoReID	
	R-1	mAP	R-1	mAP
SE-ResNet-50	84.0	89.5	90.2	89.7
HPSC	**86.2**	**91.4**	**92.4**	**91.9**

5.4.4.3 Effect of Each Loss Component

In the study above, we have shown that our network achieves a performance gain from the set-aware triplet loss and the hard positive set construction method. In this study, we will verify each component in the total loss function. SE-ResNet-50 is also used here as the backbone network. The total loss function has four components (*i.e.*, $\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{ctri}}^{\text{hm}}$, $\mathcal{L}_{\text{ctri}}^{\text{hpsc}}$ and $\mathcal{L}_{\text{stri}}^{\text{hm}}$). Table 5.4 shows the effectiveness of each loss term. In this study, the baseline model is trained by cross-entropy loss (*i.e.*, (i)). The rows in (ii), (iii), and (iv) show that each of the triple losses provides complementary cues to optimise the network. In addition, the terms $\mathcal{L}_{\text{ctri}}^{\text{hpsc}}$ and $\mathcal{L}_{\text{stri}}^{\text{hm}}$ will further improve the performance of the network. In summary, this study reveals that our method helps the network to learn complementary information when encoding the person representation.

5.4.4.4 Visualisation of Hard Positive Set Construction

We further visualise the hard positive set construction by Algorithm 1 on the iLIDS-VID dataset. The original and constructed video clips/sets are framed by black and

Table 5.5: Effect of each loss component across the iLIDS-VID and the DukeMTMC-VideoReID datasets. $[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$ denote the weights assigned to each loss term in Eq. (5.8). The 1st best in **bold font**.

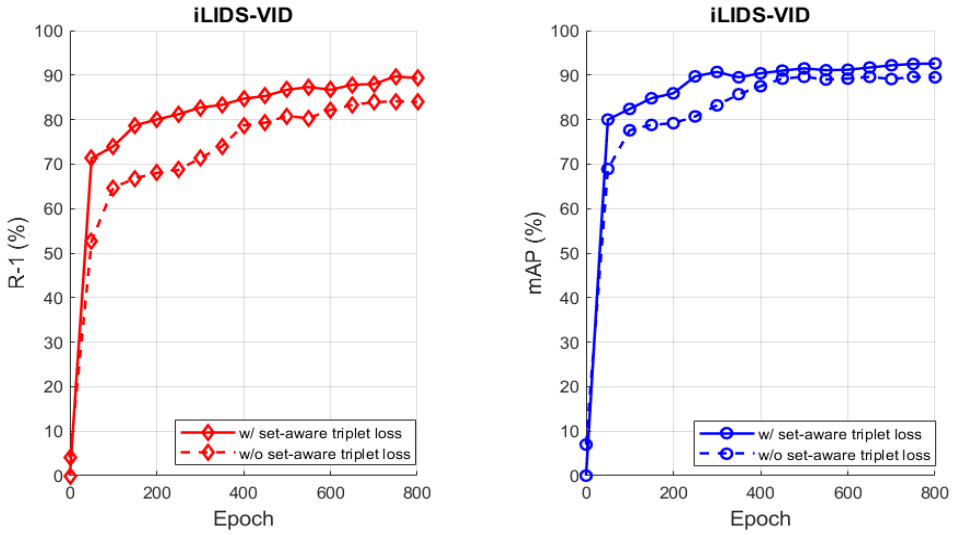
	$[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$	iLIDS-VID		DukeMTMC-VideoReID	
		R-1	mAP	R-1	mAP
(i)	$[1, 0, 0, 0]$	74.7	82.5	80.2	79.6
(ii)	$[1, 0.5, 0, 0]$	84.0	89.5	90.2	89.7
(iii)	$[1, 0, 0.5, 0]$	82.0	87.6	87.3	85.2
(iv)	$[1, 0, 0, 0.5]$	84.7	88.9	89.2	88.3
(v)	$[1, 0.5, 0.5, 0]$	85.2	90.4	91.4	90.9
(vi)	$[1, 0.5, 0.5, 0.5]$	**89.3**	**92.9**	**96.8**	**95.9**

red lines, respectively. As shown in Fig. 5.4, we can observe that the frames with occlusions or distractors will be easily selected as hard samples by our algorithm. This observation is also in line with our intuition that the hard set is constructed from the hard frames in a batch.

Figure 5.4: Example of hard positive set construction via Algorithm 1 on the iLIDS-VID dataset. The original and constructed video clips/sets are framed by black and red lines, respectively. The constructed clip indicates that the frames with occlusions or distractors will be easily selected as hard samples by our algorithm. Images are sampled from two video sequences from different pedestrians.

5.4.4.5 Training Convergence and Feature Embedding

In this part, we continue to demonstrate the superior performance of set-aware triplets by studying the training convergence and feature embedding of networks. In this study, we also use SE-ResNet-50 as the baseline network. Fig. 5.5(a) and Fig. 5.5(b) show the training curves of the network with our set-aware triplet loss and without our set-aware triplet loss w.r.t. the R-1 value and mAP value respectively. Fig. 5.6(a) and Fig. 5.6(b) visualise the features extracted by the network, trained without set-aware triplet loss, and with set-aware triplet loss. Both figures clearly show that the set-aware triplet loss indeed helps the network to learn a discriminative embedding space, in which the within-class variance is minimised and the between-class variance is maximised jointly.

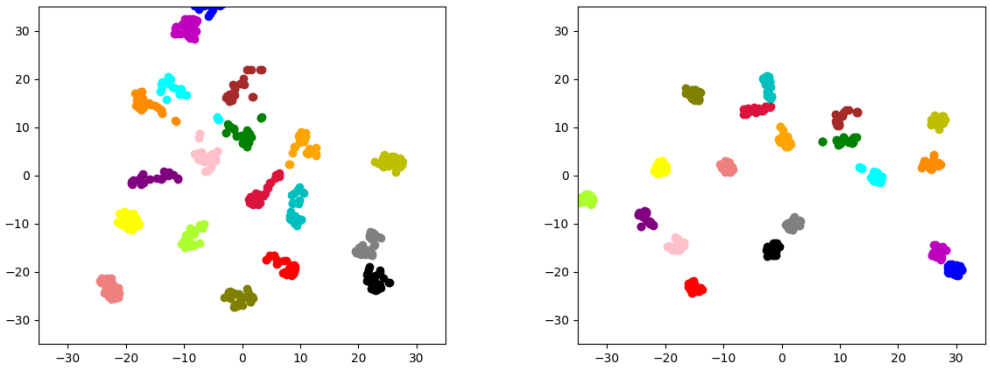


(a) (b)

Figure 5.5: The training process of the network without set-aware triplet loss and with set-aware triplet loss on the iLIDS-VID dataset. (a): The R-1 value along the training process. (b): The mAP value along the training process.

5.5 Summary

In this chapter, we construct a triplet loss to optimise the frame features of the video person re-ID task, by modelling the video clip as a set. We employ the commonly-used distance metric to measure the distance between sets, *i.e.*, ordinary distance and Hausdorff distance. Considering the hard pairs in the triplets, we further propose a new hybrid distance metric, which is defined for the anchor-positive pair and the anchor-negative pair separately. In addition, we also propose a hard positive set construction algorithm to decrease the within-class variance. Extensive experiments are conducted to verify the superior performance of the proposed method across the standard video person re-ID datasets.



(a) (b)

Figure 5.6: T-SNE visualisation [Laurens van der Maaten and Hinton, 2008] of learned features by the network (a) w/o set-aware triplet loss and (b) w/ set-aware triplet loss on the iLIDS-VID dataset. We select 20 people from the query set and visualise the frame features. Points with the same colour denote the features of the same person. Best viewed in colour.

Chapter 6

Kernel Methods in Hyperbolic Spaces

Following the previous chapter, we continue investigating the geometry constraint (*i.e.*, the curved space) as the embedding space for visual data. Embedding data in hyperbolic spaces has proven beneficial for many advanced machine learning applications such as image classification and word embeddings. However, working in hyperbolic spaces is not without difficulties as a result of its curved geometry (*e.g.*, computing the Fréchet mean of a set of points requires an iterative algorithm). Furthermore, in Euclidean spaces, one can resort to kernel machines that not only enjoy rich theoretical properties but that can also lead to superior representational power (*e.g.*, infinite-width neural networks). In this chapter, we introduce positive definite kernel functions for hyperbolic spaces. This brings in two major advantages, **1.** kernelization will pave the way to seamlessly benefit from kernel machines in conjunction with hyperbolic embeddings, and **2.** the rich structure of the Hilbert spaces associated with kernel machines enables us to simplify various operations involving hyperbolic data. That said, identifying valid kernel functions on curved spaces is not straightforward and is indeed considered an open problem in the learning community. Our work addresses this gap and develops several valid positive definite kernels in hyperbolic spaces, including the universal ones (*e.g.*, RBF). We comprehensively study the proposed kernels on a variety of challenging tasks including few-shot learning, zero-shot learning, person re-identification and knowledge distillation, showing the superiority of the kernelization for hyperbolic representations. This chapter is based on our work [Fang et al., 2021a].

6.1 Introduction

This chapter proposes a family of *positive definite (pd) kernels to map the representations in hyperbolic spaces into Reproducing Kernel Hilbert Spaces (RKHSs)*, which enables us to seamlessly benefit from kernel machines to analyse hyperbolic spaces.

In the machine learning community, the Euclidean space has been the "workhorse" for feature embeddings. This is mainly because the high-dimensional vector space is a natural generalisation from the familiar three-dimensional space we live in and per-

forming basic operations for comparison (*e.g.*, calculating distances and similarities) is straightforward. However, embedding in Euclidean spaces can harm and distort the encoding of structured data, thereby losing the complex geometric information inherently present in the data. For example, the Euclidean space fails to encode the hierarchical information in graph-structured data [Liu et al., 2019a].

Several recent studies in computer vision suggest that embedding images and video using hyperbolic geometry can be beneficial compared to the common practice of using Euclidean geometry. This includes tasks such as textual entailment [Ganea et al., 2018], image classification and retrieval [Khrulkov et al., 2020], and graph classification [Liu et al., 2019a] to name a few.

The hyperbolic space is characterised by a constant negative sectional curvature (in contrast to the flat structure of the Euclidean space), and does not satisfy Euclid's parallel postulate. One intriguing property of hyperbolic spaces is their capacity of encoding hierarchical data, as the volume of hyperbolic space expands exponentially [Hamann, 2011], thereby increasing their representation power. Although several studies have successfully employed the hyperbolic geometry for inference [Ganea et al., 2018; Khrulkov et al., 2020; Cho et al., 2019], the difficulties of working with such non-linear spaces still overwhelm their wider use. For example, while averaging in Euclidean geometry is straightforward, its counterpart in hyperbolic space is approximated by the Fréchet mean. Computing the Fréchet mean requires an iterative algorithm and could easily become costly [Karcher, 1977; Lou et al., 2020]. This motivates us to develop kernels to make it possible to seamlessly benefit and employ kernel machines towards analysing hyperbolic data.

To be able to make use of kernel machines, one needs to have a pd kernel function at its disposal. Loosely speaking, a kernel function is a measure of similarity. Many familiar kernels in the Euclidean space are defined as functions of the Euclidean distance (which is indeed the geodesic distance of the space). Take the RBF kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\xi d^2(\boldsymbol{x}, \boldsymbol{y}))$ as an example. This might imply that valid pd kernels in curved spaces, the hyperbolic space being one, can be constructed once the geodesic distance is known. Unfortunately, this is not the case as shown in [Jayasumana et al., 2015; Feragen et al., 2015] (*c.f.*, theorem 6.2 in [Jayasumana et al., 2015]), because such curved spaces are not isometric to flat Euclidean spaces. Interestingly, the difficulty of defining pd kernels on curved spaces is now considered an open problem in machine learning [Feragen and Hauberg, 2016].

In this chapter, we address the design challenge of pd kernels for hyperbolic representations using the Poincaré model. Here, we propose several valid pd hyperbolic kernels, including the powerful universal ones. To this end, we first make use of a lemma to construct a valid linear-like kernel. Leveraging this lemma, we further define valid RBF and Laplace kernels for the hyperbolic geometry. Finally, we propose the binomial kernel. Table 6.1 summarises the proposed kernels.

The **contributions** of this work include:

- We propose four pd kernels for the hyperbolic spaces, namely, the hyperbolic tangent kernel, the hyperbolic RBF kernel, the hyperbolic Laplace and the hy-

Table 6.1: Summary of the proposed positive definite kernels in hyperbolic spaces and their properties.

Kernel	Formulation: $k(z_i, z_j)$	Condition	Properties
$f_{\mathbb{D}}(z) = \tanh^{-1}(\sqrt{c}\|z\|)\frac{z}{\sqrt{c}\|z\|},\ c > 0$ and $z \in \mathbb{D}_c^n$			
Hyperbolic tangent kernel	$k^{\tan}(z_i, z_j) = \langle f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\rangle$	-	pd
Hyperbolic RBF kernel	$k^{\text{rbf}}(z_i, z_j) = \exp\big(-\xi\|f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\|^2\big)$	$\xi > 0$	pd, universal
Hyperbolic Laplace kernel	$k^{\text{lap}}(z_i, z_j) = \exp\big(-\xi\|f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\|\big)$	$\xi > 0$	pd, universal
Generalised Hyperbolic Laplace kernel	$k^{\text{glap}}(z_i, z_j) = \exp\big(-\xi\|f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\|^{2\alpha}\big)$	$\xi > 0, 0 < \alpha < 1$	pd, universal
Hyperbolic binomial kernel	$k^{\text{bin}}(z_i, z_j) = \big(1 - \langle f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\rangle\big)^{-\alpha}$	$\alpha > 0$	pd, universal

perbolic binomial kernel, in conjunction with their theoretical analysis. To the best of our knowledge, this is the first work to develop pd kernels in hyperbolic spaces.

- To evaluate the power of the proposed kernels, we conduct thorough experiments on various vision tasks including few-shot learning, zero-shot learning, person re-identification, and knowledge distillation, and employ the kernels along deep neural networks (DNNs) to attain rich models for inference. Empirically, we observed the superiority of the kernelization for the representation learning in hyperbolic spaces.

6.2 Related Work

In this section, we review the related work on geometric constraint learning and kernel methods on curved spaces.

Geometric Constraint Learning. Geometric constraints have been studied extensively in deep learning, which pushes the network to encode complex structures of the data. The representation power of a set is improved by fitting a subspace [Simon et al., 2020]. In SVDNet, the orthogonality constraint enforces the fully connected layer lying on the Grassmannian manifold, which de-correlates the features among entries [Sun et al., 2017]. The works in [Liu et al., 2017b; Meng et al., 2019] also show that embedding in a spherical space is particularly effective for similarity learning (*e.g.*, face verification, clustering) compared to using Euclidean spaces.

In recent years, hyperbolic geometry has gained substantial interest thanks to its tree-like nature, and the ability to encode hierarchical relationships in the data. Generalising the basic operations in Euclidean geometry, the work [Ganea et al., 2018] develops hyperbolic layers in neural networks. The following works further show the success of hyperbolic embeddings for graph-structured data, language data, visual data as well as 3D data [Liu et al., 2019a; Gulcehre et al., 2019; Khrulkov et al., 2020; Chen et al., 2020a]. More complex structures of data are also studied in [Gu et al., 2019; Skopek et al., 2020], which represents the data in a mixed-curvature geometry.

Kernel Methods. Kernel methods have been studied extensively and proven its success in a broad range of machine learning approaches, *e.g.*, SVM, PCA and clustering [Hofmann et al., 2008]. The main idea of kernel methods is to project the input samples, to a high-dimensional (or even infinite-dimensional) Reproducing Kernel Hilbert Space (RKHS), where the projected data can be analysed with linear models. To avoid explicit lifting to RKHS, the kernel trick provides a simple way to generate the similarity measure of pairs in RKHS.

As of late, attempts to boost the representational power of structured-data by generalising the kernel methods to non-linear geometries have gained increasing attention. The common strategy to define a valid pd kernel on non-Euclidean geometries is to adopt a proper distance metric. In [Jayasumana et al., 2013], the authors propose the main theoretical framework to design the Gaussian kernel on symmetric positive definite matrices. The proposed theory is further verified to develop the

Gaussian kernel on the Grassmannian manifold [Jayasumana et al., 2015]. Kernels for the Grassmannian manifold are studied in [Harandi et al., 2014]. The kernels using the Fisher information metric are developed for the persistence diagrams in [Le and Yamada, 2018]. The closest study to our work is the work of Cho *et al*. [Cho et al., 2019], which formulates the support vector machine (SVM) in hyperbolic spaces. To facilitate the nonlinear decision boundaries, the kernel SVM for the hyperbolic space is also introduced in [Cho et al., 2019]. However, the proposed indefinite kernel is not universal and hence violates the universal approximation property [Micchelli et al., 2006].

In contrast to existing works, this work develops the theoretical framework for positive definite kernels on the hyperbolic geometry. As a complementary concept to the indefinite kernel, our work kernelizes the hyperbolic space, and thus to embed hyperbolic data into a high, possibly infinite, dimensional Hilbert space. In the remainder of this chapter, we will present the developed theory and evaluate the algorithms across different challenging applications.

6.3 Kernel Methods in Hyperbolic Spaces

In this section, we propose positive definite (pd) kernels in hyperbolic spaces. Essentially, we are interested in identifying a bivariate function $k(\cdot,\cdot) : (\mathbb{D}_c^n \times \mathbb{D}_c^n) \to \mathbb{R}$, which represents an inner product in a Reproducing Kernel Hilbert Space (RKHS). Obviously, not all bivariate functions constitute valid kernels, meaning that they do not necessarily realise an RKHS. Also, popular kernels in Euclidean spaces cannot lead to meaningful solutions as they are not faithful to the geometry of the hyperbolic spaces. Embedding hyperbolic points into an RKHS is not only theoretically appealing but can also result in practical benefits due to the intriguing properties of RKHSs. This includes representational power of RKHS [Hofmann et al., 2008], kernel two-sample test [Gretton et al., 2012], neural tangent kernels [Jacot et al., 2018] to name a few.

In this chapter, we make use of the tangent space of the hyperbolic geometry to define a set of valid pd kernels. We start by formally defining a pd kernel.

Definition 1 *(Positive Definite Kernels [Berg et al., 1984]) Let $\mathcal{Z}$ be a non-empty set. A symmetric function $k(\cdot,\cdot) : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ is a positive definite kernel on $\mathcal{Z}$ if and only if $\sum_{i,j=1}^{m} c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \geq 0$ for any $m \in \mathbb{N}$, $\boldsymbol{z}_i \in \mathcal{Z}$ and $c_i \in \mathbb{R}$.*

Essential to our work is the following lemma;

Lemma 1 *Let $\mathcal{Z}$ be a non-empty set. Consider a function $f(\cdot) : \mathcal{Z} \to \mathbb{R}^n$, that maps each element of $\mathcal{Z}$ uniquely to $\mathbb{R}^n$. Then,*

$$k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \big\langle f(\boldsymbol{z}_i), f(\boldsymbol{z}_j) \big\rangle$$

is a pd kernel on $\mathcal{Z}$.

Proof 1 *The proof of this lemma follows immediately from Definition 1. To see this, define*

$$\boldsymbol{F}_{n\times m} \coloneqq \left[f(\boldsymbol{z}_1), f(\boldsymbol{z}_2), \cdots, f(\boldsymbol{z}_m)\right] \ .$$

Now, notice that

$$\sum_{i,j=1}^{m} c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{c}^\top \boldsymbol{K}\boldsymbol{c} = \boldsymbol{c}^\top \boldsymbol{F}^\top \boldsymbol{F}\boldsymbol{c} = \|\boldsymbol{F}\boldsymbol{c}\|^2 \geq 0 \ .$$

The $\left[\boldsymbol{K}_{m\times m}\right]_{i,j} = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$ *is called the gram matrix.*

Based on Lemma 1, we propose to make use of $f_{\mathbb{D}}(\cdot) : \mathbb{D}_c^n \to \mathbb{R}^n$ defined as,

$$f_{\mathbb{D}}(\boldsymbol{z}) \coloneqq \tanh^{-1}(\sqrt{c}\|\boldsymbol{z}\|)\frac{\boldsymbol{z}}{\sqrt{c}\|\boldsymbol{z}\|}, \tag{6.1}$$

to develop valid pd kernels on $\mathbb{D}_c^n$. The function $f_{\mathbb{D}}(\cdot)$ enjoys various unique properties. First note that the function is bijective and $f_{\mathbb{D}}(\boldsymbol{z}) = \mathrm{Y}_{\boldsymbol{0}}(\boldsymbol{z})$. The next theorem establishes an important property and justifies our choice here better.

Theorem 1 (Curve Length Equivalence) *A curve in* $\mathbb{D}_c^n$ *is a continuous function* $\gamma(\cdot) : [0,1] \to \mathbb{D}_c^n$*; joining the starting point* $\gamma(0)$ *to the end point* $\gamma(1)$*. Define the distance induced by* $f_{\mathbb{D}}$ *as*

$$d_e(\boldsymbol{z}_i, \boldsymbol{z}_j) \coloneqq \left\|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\right\|. \tag{6.2}$$

The length of any given curve γ *is the same under* d_e *and the geodesic distance* d_c *up to a scale of* $1/\tilde{\lambda}_c$*, where* $\tilde{\lambda}_c = 2$ *is the conformal factor at the origin.*

Before the proof of Theorem 1, we first formally define the curve length and intrinsic metric.

Definition 2 (Curve Length) *The length of a curve* γ *is the supremum of* $L(\gamma; \{t_i\}_{i=0}^n)$ *over all possible partitions* $\{t_i\}_{i=0}^n$*, where* $0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = 1$ *and* $L(\gamma; \{t_i\}_{i=0}^n) = \sum_{i=1}^n d(\gamma(t_{i-1}), \gamma(t_i))$*.*

Definition 3 (Intrinsic Metric) *The intrinsic metric* $\hat{\delta}(x, y)$ *on* $\mathcal{M}$ *is defined as the infimum of the lengths of all paths from* x *to* y*.*

Theorem 2 ([Hartley et al., 2012]) *If the intrinsic metrics induced by two metrics* d_1 *and* d_2 *are identical to a scale* ζ*, then the length of any given curve is the same under both metrics up to* ζ*.*

Theorem 3 ([Hartley et al., 2012]) *if* $d_1(x, y)$ *and* $d_2(x, y)$ *are two metrics defined on a space* $\mathcal{M}$ *such that*

$$\lim_{d_1(x,y)\to 0} \frac{d_2(x, y)}{d_1(x, y)} = 1, \tag{6.3}$$

uniformly (with respect to x *and* y*), then the length of any given curve is the same under both metrics. Consequently, the intrinsic metrics induced by* d_1 *and* d_2 *are identical.*

Therefore, we need to study the the behaviour of

$$\lim_{d_c(z_i,z_j)\to\mathbf{0}} \frac{\tilde{\lambda}_c \, d_e(z_i, z_j)}{d_c(z_i, z_j)}, \tag{6.4}$$

to prove our theorem. This is also equivalent to study

$$\lim_{\gamma\to\mathbf{0}} \frac{\tilde{\lambda}_c \, d_e(z_i, z_j)}{d_c(z_i, z_j)}, \tag{6.5}$$

where $z_j = z_i \oplus_c \gamma$.

Proof 2 *We first prove $f_{\mathbb{D}}(z_i \oplus_c \gamma) = f_{\mathbb{D}}(z_i) + f_{\mathbb{D}}(\gamma)$ for $\gamma \to \mathbf{0}$.*

$$\begin{aligned} z_i \oplus_c \gamma &= \frac{(1 + 2c\langle z_i, \gamma\rangle + c\|\gamma\|^2)z_i + (1 - c\|z_i\|^2)\gamma}{1 + 2c\langle z_i, \gamma\rangle + c^2\|z_i\|^2\|\gamma\|^2} \\ &\approx \frac{(1 + 2c\langle z_i, \gamma\rangle)z_i + (1 - c\|z_i\|^2)\gamma}{1 + 2c\langle z_i, \gamma\rangle} \\ &\approx z_i + \frac{1 - c\|z_i\|^2}{1 + 2c\langle z_i, \gamma\rangle}\gamma \\ &= z_i + \kappa\gamma. \end{aligned} \tag{6.6}$$

Then the first order approximation of $f_{\mathbb{D}}(z_i + \kappa\gamma)$ can be obtained:

$$\begin{aligned} f_{\mathbb{D}}(z_i + \kappa\gamma) &= \tanh^{-1}(\sqrt{c}\|z_i + \kappa\gamma\|)\frac{z_i + \kappa\gamma}{\sqrt{c}\|z_i + \kappa\gamma\|} \\ &\approx z_i + \kappa\gamma + \frac{c\|z_i + \kappa\gamma\|^2}{3}(z_i + \kappa\gamma) \\ &\approx z_i + \kappa\gamma + \frac{c\|z_i + \kappa\gamma\|^2}{3}z_i \\ &+ \frac{c\|z_i + \kappa\gamma\|^2}{3}\kappa\gamma \\ &\approx z_i + \kappa\gamma + \frac{c\|z_i\|^2}{3}z_i \\ &+ \frac{2c\langle z_i, \kappa\gamma\rangle}{3}z_i + \frac{c\|z_i + \kappa\gamma\|^2}{3}\kappa\gamma \end{aligned} \tag{6.7}$$

The first order approximation of $f_{\mathbb{D}}(z_i)$ and $f_{\mathbb{D}}(\gamma)$ can also been obtained:

$$f_{\mathbb{D}}(z_i) \approx z_i + \frac{c\|z_i\|^2}{3}z_i \tag{6.8}$$

and

$$f_{\mathbb{D}}(\gamma) \approx \gamma + \frac{c\|\gamma\|^2}{3}\gamma. \tag{6.9}$$

Then we can see:

$$\lim_{\gamma \to \mathbf{0}} \left(f_{\mathbb{D}}(\boldsymbol{z}_i + \kappa\gamma) - f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\gamma) \right) = 0. \tag{6.10}$$

Since it holds that $f_{\mathbb{D}}(\boldsymbol{z}_i \oplus_c \gamma) = f_{\mathbb{D}}(\boldsymbol{z}_i) + f_{\mathbb{D}}(\gamma)$ *for* $\gamma \to \mathbf{0}$*, then we have*

$$\begin{aligned}
\lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c d_e(\boldsymbol{z}_i, \boldsymbol{z}_j)}{d_c(\boldsymbol{z}_i, \boldsymbol{z}_j)} &= \lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c \| f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j) \|}{\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \| -\boldsymbol{z}_i \oplus_c \boldsymbol{z}_j \|)} \\
&= \lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c \| f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_i \oplus_c \gamma) \|}{\frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c} \| (-\boldsymbol{z}_i) \oplus_c (\boldsymbol{z}_i \oplus_c \gamma) \|\right)} \\
&= \lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c \| f_{\mathbb{D}}(\boldsymbol{z}_i) - (f_{\mathbb{D}}(\boldsymbol{z}_i) + f_{\mathbb{D}}(\gamma)) \|}{\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \|\gamma\|)} \\
&= \lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c \| f_{\mathbb{D}}(\gamma) \|}{\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \|\gamma\|)} \\
&= \lim_{\gamma \to \mathbf{0}} \frac{\tilde{\lambda}_c \| \tanh^{-1}(\sqrt{c} \|\gamma\|) \frac{\gamma}{\sqrt{c}\|\gamma\|} \|}{\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \|\gamma\|)} \\
&= \lim_{\gamma \to \mathbf{0}} \frac{\frac{\tilde{\lambda}_c}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \|\gamma\|) \| \frac{\gamma}{\|\gamma\|} \|}{\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \|\gamma\|)} \\
&= 1.
\end{aligned} \tag{6.11}$$

This ends the proof.

Having $f_{\mathbb{D}}$ at our disposal, we are now ready to define the kernels in hyperbolic spaces.

6.3.1 Hyperbolic Tangent Kernel

The simplest pd kernel resembles the linear kernel in Euclidean spaces and is defined as $k^{\tan}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j) \rangle$. We call this kernel hyperbolic tangent kernel as it can be understood as the linear kernel in the identity tangent space of the Poincaré ball. This kernel is attractive as it is parameter-less, making it ideal for fast prototyping. The proof of positive-definiteness of the hyperbolic tangent kernel follows directly from Lemma 1.

6.3.2 Hyperbolic RBF Kernel

The Gaussian RBF kernel is a popular universal kernel in Euclidean spaces. In $\mathbb{R}^n$, the RBF kernel can be written as $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\xi \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$, $\xi > 0$, where the metric is the squared Euclidean distance in $\mathbb{R}^n$. Taking into account the properties of the RBF kernel [Christmann and Steinwart, 2008], it is very desirable to extend this kernel to hyperbolic spaces. One may assume that replacing the Euclidean distance by the

geodesic distance (*i.e.*, Eq. (2.16)) can lead to a valid pd kernel. This, unfortunately, is not the case as shown by the toy example below.

Example 1 *Consider* $\mathbb{D}_{0.1}^3$ *and the following points:*

$$\boldsymbol{z}_1 = \begin{bmatrix} 0.1885 \\ 0.2330 \\ 0.9526 \end{bmatrix}, \boldsymbol{z}_2 = \begin{bmatrix} 0.6586 \\ 0.2053 \\ 0.0894 \end{bmatrix}, \boldsymbol{z}_3 = \begin{bmatrix} 0.3017 \\ 0.4155 \\ 0.5357 \end{bmatrix}, \boldsymbol{z}_4 = \begin{bmatrix} 0.2388 \\ 0.8290 \\ 0.3790 \end{bmatrix}.$$

*The gram matrix (*i.e., $\exp(-\xi d_c^2(\boldsymbol{z}_i, \boldsymbol{z}_j))$ *for* $\xi = 0.01$*) for these points has a negative eigenvalue of* -3.0605×10^{-5}.

Further to the counterexample above, the RBF kernel derived from the geodesic distance is shown to be pd iff the space is isometric to the Euclidean space per the following theorem.

Theorem 4 (Theorem 6.2 in [Jayasumana et al., 2015]) *Let* $\mathcal{M}$ *be a complete Riemannian manifold and* $d_{\mathcal{M}}$ *be the induced geodesic distance on the manifold. The Gaussian RBF kernel* $k(\cdot,\cdot) : (\mathcal{M} \times \mathcal{M}) \to \mathbb{R} : k(\boldsymbol{m}_i, \boldsymbol{m}_j) \coloneqq \exp(-\xi d_{\mathcal{M}}^2(\boldsymbol{m}_i, \boldsymbol{m}_j))$ *is positive definite for all* $\xi > 0$ *if and only if the Riemannian manifold* $\mathcal{M}$ *is isometric to some Euclidean space* $\mathbb{R}^n$.

According to Theorem 4, it is theoretically impossible to obtain a valid RBF kernel using geodesic distance on hyperbolic spaces [1]. Given the above, we propose to make use of $d_e(\cdot,\cdot)$ and define the hyperbolic RBF kernel as

$$k^{\text{rbf}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|^2\big) . \tag{6.12}$$

To show that the form in Eq. (6.12) is a valid pd kernel, we first define negative definite (nd) kernels.

Definition 4 (Negative Definite Kernels [Berg et al., 1984]) *Let* $\mathcal{Z}$ *be a non-empty set. A symmetric function* $k(\cdot,\cdot) : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ *is a negative definite kernel on* $\mathcal{Z}$ *if and only if* $\sum_{i,j=1}^m c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \leq 0$ *for any* $m \in \mathbb{N}$, $\boldsymbol{z}_i \in \mathcal{Z}$ *and* $c_i \in \mathbb{R}$ *with* $\sum_{i=0}^m c_i = 0$.

Note the difference between pd and nd kernels. For nd kernels, an additional condition (*i.e.*, $\sum_{i=0}^m c_i = 0$) is required. The following lemma shows that $d_e^2(\cdot,\cdot) = \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|^2$ is indeed nd.

Lemma 2 *Let* $\mathcal{Z}$ *be a non-empty set. An injective function* $f(\cdot) : \mathcal{Z} \to \mathbb{R}^n$*, maps each vector in* $\mathcal{Z}$ *onto an inner product space* $\mathbb{R}^n$*. Then* $k(\boldsymbol{z}_i, \boldsymbol{z}_j) \coloneqq \|f(\boldsymbol{z}_i) - f(\boldsymbol{z}_j)\|^2$ *is negative definite.*

[1] If a manifold $\mathcal{M}$ is isometric to some Euclidean spaces $\mathbb{R}^n$, then the geodesic distance on $\mathcal{M}$ is the Euclidean distance in $\mathbb{R}^n$. However, it is impossible to find an isometry between $\mathbb{D}_c^n$ and $\mathbb{R}^n$ because of the difference in the curvature of two geometries.

Proof 3 *Suppose* $\sum_{i=1}^{m} c_i = 0$, *then we have:*

$$\begin{aligned}
&\sum_{i,j=1}^{m} c_i c_j \|f(z_i) - f(z_j)\|^2 \\
&= \sum_{i,j=1}^{m} c_i c_j \big\{ \|f(z_i)\|^2 + \|f(z_j)\|^2 - \langle f(z_i), f(z_j)\rangle - \langle f(z_j), f(z_i)\rangle \big\} \\
&= \sum_{i=1}^{m} c_i \|f(z_i)\|^2 \sum_{j=1}^{m} c_j + \sum_{j=1}^{m} c_j \|f(z_j)\|^2 \sum_{i=1}^{m} c_i - \langle \sum_{i=1}^{n} c_i f(z_i), \sum_{j=1}^{n} c_j f(z_j)\rangle \qquad (6.13) \\
&- \langle \sum_{j=1}^{n} c_j f(z_j), \sum_{i=1}^{n} c_i f(z_i)\rangle \\
&= -2\| \sum_{i=1}^{n} c_i f(z_i)\|^2 \leq 0.
\end{aligned}$$

Thus $k(z_i, z_j) = \|f(z_i) - f(z_j)\|^2$ *is negative definite. This ends the proof.*

The following important theorem establishes the connection between positive definite kernels and negative definite kernels.

Theorem 5 *([Berg et al., 1984]) Let* $\mathcal{Z}$ *be a non-empty set and* $k : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ *be a kernel. The kernel* $k(z_i, z_j) = \exp(-\xi\Phi(z_i, z_j))$ *is positive definite for all* $\xi > 0$ *if and only if* $\Phi(\cdot, \cdot)$ *is negative definite.*

Stating the fact that $d_e^2(\cdot, \cdot)$ is nd along with Theorem 5 concludes our claim that the hyperbolic RBF kernel defined in Eq. (6.12) is pd.

6.3.3 Hyperbolic Laplace Kernel

The Laplace kernel is another widely used universal kernel in Euclidean spaces, formulated as $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\xi\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$, $\xi > 0$. When extending the Laplace kernel to hyperbolic spaces, we use the following theorem to build a nd kernel for hyperbolic spaces.

Theorem 6 *([Berg et al., 1984]) If* $k : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ *is negative definite and satisfies* $k(z_i, z_j) \geq 0$, *then* k^α *is also negative definite for* $0 < \alpha < 1$.

Combining Theorem 5 and Theorem 6, and choosing $\alpha = \frac{1}{2}$, we could obtain the hyperbolic Laplace kernel as $k^{\text{lap}}(z_i, z_j) = \exp\big(-\xi d_e(f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j))\big) = \exp\big(-\xi\|f_{\mathbb{D}}(z_i) - f_{\mathbb{D}}(z_j)\|\big)$. A more general form of the Laplace kernel (*i.e.*, generalised hyperbolic Laplace kernel) can be further derived as: $k^{\text{glap}}(z_i, z_j) = \exp\big(-\xi\|f_{\mathbb{D}}(z_i) - f_{\mathbb{D}}(z_j)\|^{2\alpha}\big)$, where $0 < \alpha < 1$.

6.3.4 Hyperbolic Binomial Kernel

In addition to the exponential type kernels, we further construct a hyperbolic binomial kernel. To obtain the hyperbolic binomial kernel, we make use of the following lemma.

Lemma 3 *Let $\mathcal{Z}$ be a non-empty set. An injective function $f : \mathcal{Z} \to \mathbb{R}^n$, maps each vector in $\mathcal{Z}$ onto an inner product space $\mathbb{R}^n$. Then $k(z_i, z_j) := \big(1 - \langle f(z_i), f(z_j)\rangle\big)^{-\alpha}$ defines a binomial kernel on $\mathcal{Z}$ when $\alpha > 0$ and $\|f(z)\| < 1$.*

Proof 4 *According to Lemma 4.8 of [Christmann and Steinwart, 2008], if the function $k(\cdot,\cdot)$ can be decomposed by a full Taylor series with each term being non-negative, then we can claim $k(\cdot,\cdot)$ is a valid pd kernel. Let $t = \langle f(z_i), f(z_j)\rangle$, the binomial series $k(z_i, z_j) = (1-t)^{-\alpha} = \sum_{n=0}^{\infty} \binom{-\alpha}{n}(-1)^n t^n$ holds for all $|t| < 1$, where the binomial coefficient $\binom{\beta}{n} := \prod_{i=1}^{n}(\beta - i + 1)/i$. It can be seen $\binom{-\alpha}{n}(-1)^n > 0$ when $\alpha > 0$, which indicates the binomial kernel has a non-negative and full Taylor series.*

According to the Lemma 3, we could obtain the hyperbolic binomial kernel as

$$k^{\text{bin}}(z_i, z_j) = \big(1 - \langle f_{\mathbb{D}}(z_i), f_{\mathbb{D}}(z_j)\rangle\big)^{-\alpha},\ \alpha > 0. \tag{6.14}$$

Also, given the non-negativeness and full Taylor series in the above proof, we can further claim that the hyperbolic binomial kernel satisfies the necessary and sufficient condition of being universal, shown in Corollary 4.57 of [Christmann and Steinwart, 2008].

Remark 8 *As alluded to earlier, we have made use of the identity tangent space of the Poincaré ball* (i.e., *$\mathbb{D}_c^n$) to define pd kernels for the hyperbolic spaces. This implies that the kernels are defined using the Lie algebra of $\mathbb{D}_c^n$. Such a construction has been used with success in other manifolds (*e.g., *SPD as in [Jayasumana et al., 2015]).*

In this chapter, we employ the kernels along with convolutional neural networks (CNNs) to attain rich models for computer vision tasks. The CNNs encode the input data to vectors, distributed in hyperbolic spaces. Then the proposed kernels are further used to train the network.

6.4 Experiments

We first explain the inference with cross entropy-like loss function using kernels. Specifically, for a training sample $\boldsymbol{f}_i$ with label l, the cross entropy loss is given by:

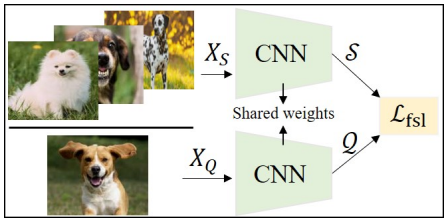
$$\mathcal{L} = -\log\Big(\frac{\exp(s(\boldsymbol{f}_i, \boldsymbol{w}_i))}{\sum_{j=1}^{N}\exp(s(\boldsymbol{f}_i, \boldsymbol{w}_j))}\Big), \tag{6.15}$$

where $\boldsymbol{w}_i$ indicates the weights or prototype for $\boldsymbol{f}_i$ and N is the number of classes in the dataset. Then we apply our kernels in Eq. (6.15) as:

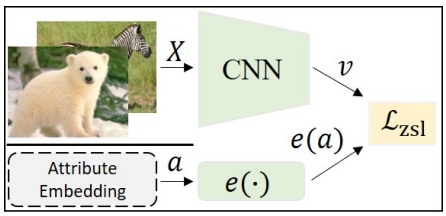
$$\mathcal{L}^K = -\log\left(\frac{g(k(\boldsymbol{f}_i, \boldsymbol{w}_i))}{\sum_{j=1}^{N} g(k(\boldsymbol{f}_i, \boldsymbol{w}_j))}\right). \tag{6.16}$$

Here, $g(\cdot)$ is exp mapping if $k(\cdot,\cdot)$ is non-exponential type kernels. Otherwise, $g(\cdot)$ is the identity mapping.

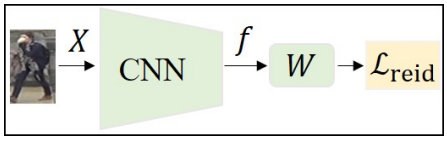
In the remainder of this chapter, we comprehensively evaluate the effectiveness of the proposed algorithms for a variety of challenging tasks, *i.e.*, few-shot learning, zero-shot learning, person re-identification and knowledge distillation.



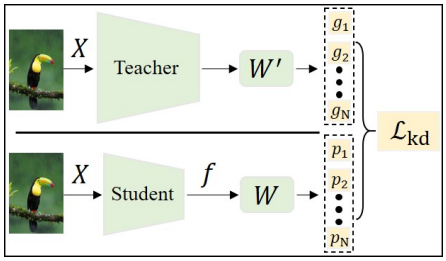
(a) The pipeline of the deep network for few-shot recognition. $\boldsymbol{X}_S$ and $\boldsymbol{X}_Q$ denote the input images in the support set and query set.



(b) The pipeline of the deep network for zero-shot learning. $\boldsymbol{X}$ and $\boldsymbol{a}$ denotes input images and attribute descriptors.



(c) The pipeline of the deep network for person re-identification. $\boldsymbol{X}$ denotes the input pedestrian images.



(d) The pipeline of teacher-student network for knowledge distillation. $\boldsymbol{X}$ denotes the input images.

Figure 6.1: The pipeline of three applications we consider: (a) few-shot learning, (b) zero-shot learning , (c) person re-identification and (d) knowledge distillation.

6.4.1 Few-shot Learning

Few-shot learning (FSL) is required to learn an embedding space, which should be adapted to recognise unseen classes at test time, given only a few samples of each new class. In our experiments, we follow the general practice (*i.e.*, 5-way 1-shot and 5-way 5-shot and 15 query images) to evaluate the model. We employ the pipeline in the prototypical network (ProtoNet) [Snell et al., 2017] along with the proposed kernels to train the feature extractor (see Fig. 6.1(a)).

Four popular benchmarks, *i.e.*, ***mini*ImageNet** [Deng et al., 2009], **CUB** [Wah et al., 2011], ***tiered*-ImageNet** [Ren et al., 2018] and **Few-shot-CIFAR100** (FC100) [Oreshkin et al., 2018] are adopted to assess our algorithms. The ***mini*ImageNet** is a subset of the ImageNet dataset, and contains 60,000 images in total. It has 100 classes and each class has 600 images. We also follow the standard evaluation protocol, which splits the 100 classes into 64 for training, 16 for validation and 20 for testing. The **CUB** dataset is a fine-grained image recognition dataset and we also use it to evaluate our few-shot learning algorithms. The CUB dataset consists of 200 different species of birds and 11,788 images in total. We also follow the standard setting to split the dataset into 100 base classes, 50 validation classes and 50 test classes. Similar to *mini*ImageNet, ***tiered*-ImageNet** is also a subset of ImageNet with broader classes (*i.e.*,, 608 classes in total). The *tiered*-ImageNet contains 351 classes for training, 97 classes for validation and 160 classes for testing. **FC100**, which is based on the CIFAR-100, is proposed for the FSL task. It also contains three data splits, *i.e.*, training split, validation split and test split, with each having 60, 20, 20 classes.

In terms of the feature extractor, we use both Conv-4 [Snell et al., 2017] and ResNet-18 [He et al., 2016] CNN backbones in our experiments. We use the Conv-4 and ResNet-18 backbones to evaluate the *mini*ImageNet and CUB datasets and the Conv-4 backbone to evaluate the *tiered*-ImageNet and FC100 datasets.

Tables 6.2, 6.3, 6.4 illustrate the results on four datasets. We observe that our algorithms improve the few-shot recognition performance as compared to their hyperbolic counterpart and other advanced methods. In addition, the results from the hyperbolic RBF kernel in general exceed the results from other kernels. For example, in 5-way 5-shot setting, the hyperbolic RBF kernel outperforms the Hyperbolic ProtoNet [Khrulkov et al., 2020] by 3.42, 2.68, 4.52 and 2.64 for *mini*ImageNet, CUB, *tiered*-ImageNet and FC100, respectively, clearly showing the potential and superiority of universal kernels.

6.4.2 Zero-shot Learning

Zero-shot learning (ZSL) aims to identify objects that are unseen during the training phase [Akata et al., 2015]. We first build a baseline network for the scenario of zero-shot recognition. In the training phase, we randomly sample N_b seen visual features as $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N_b}\}$. All the semantic features are projected to the visual space, denoted by $\boldsymbol{E} = \{e(\boldsymbol{a}_1), \ldots, e(\boldsymbol{a}_{|L^s|})\}$, where $|L^s|$ denotes the number of seen classes in the training set. In our implementation, the embedding function (*i.e.*, $e(\cdot)$) is a simple two layer MLP, with each layer stacking the linear transformation, ReLU activation and batch normalisation. Then the network is trained by the following cross-entropy type loss:

$$\mathcal{L}_{\text{zsl}} = -\frac{1}{N_b}\sum_{i=1}^{N_b}\log\Big(\frac{\exp\big(-\|(e(\boldsymbol{a}^*) - \boldsymbol{v}_i\|\big)}{\sum_{j=1}^{|L^s|}\exp\big(-\|e(\boldsymbol{a}_j) - \boldsymbol{v}_i\|\big)}\Big),$$

Table 6.2: Few-shot classification results on the *mini*ImageNet dataset with 95% confidence interval. The 1st best in **bold font**.

Model	Backbone	1-shot	5-shot
MatchingNet [Vinyals et al., 2016]	Conv-4	43.56 ± 0.84	55.31 ± 0.73
ProtoNet [Snell et al., 2017]	Conv-4	44.53 ± 0.76	65.77 ± 0.66
MAML [Finn et al., 2017]	Conv-4	48.70 ± 1.84	63.11 ± 0.92
RelationNet [Sung et al., 2018]	Conv-4	50.44 ± 0.82	65.32 ± 0.70
DN4 [Li et al., 2019d]	Conv-4	51.24 ± 0.74	71.02 ± 0.64
DSN [Simon et al., 2020]	Conv-4	51.78 ± 0.96	68.99 ± 0.69
Hyper ProtoNet [Khrulkov et al., 2020]	Conv-4	54.43 ± 0.20	72.67 ± 0.15
Hyperbolic tangent kernel	Conv-4	55.61 ± 0.21	74.81 ± 0.16
Hyperbolic RBF kernel	Conv-4	56.48 ± 0.20	$\mathbf{76.09 \pm 0.16}$
Hyperbolic Laplace kernel	Conv-4	56.26 ± 0.20	75.35 ± 0.15
Hyperbolic binomial kernel	Conv-4	$\mathbf{56.82 \pm 0.20}$	75.27 ± 0.15
Baseline [Chen et al., 2019c]	ResNet-18	51.75 ± 0.80	74.27 ± 0.63
Baseline++ [Chen et al., 2019c]	ResNet-18	51.87 ± 0.77	75.68 ± 0.63
MatchingNet [Vinyals et al., 2016]	ResNet-18	52.91 ± 0.88	68.88 ± 0.69
ProtoNet [Snell et al., 2017]	ResNet-18	54.16 ± 0.82	73.68 ± 0.65
SNCA [Wu et al., 2018b]	ResNet-18	57.80 ± 0.80	72.80 ± 0.70
Hyper ProtoNet [Khrulkov et al., 2020]	ResNet-18	59.47 ± 0.20	76.84 ± 0.14
Hyperbolic tangent kernel	ResNet-18	59.91 ± 0.21	76.65 ± 0.16
Hyperbolic RBF kernel	ResNet-18	60.91 ± 0.21	77.12 ± 0.15
Hyperbolic Laplace kernel	ResNet-18	60.52 ± 0.21	$\mathbf{77.33 \pm 0.15}$
Hyperbolic binomial kernel	ResNet-18	$\mathbf{61.04 \pm 0.21}$	77.01 ± 0.15

where $\boldsymbol{a}^*$ shares the same label with $\boldsymbol{v}_i$. The baseline network is conducted on Euclidean spaces and the pipeline of the network for ZSL is illustrated in Fig. 6.1(b).

Four datasets, *i.e.*, **SUN** [Patterson and Hays, 2012], **CUB** [Wah et al., 2011], **AWA1** [Lampert et al., 2013] and **AWA2** [Akata et al., 2015] are adopted to evaluate our algorithms in the generalised ZSL (GZSL) setting. The visual features of all datasets are extracted from the ImageNet pre-trained ResNet-101 and the dimension are 2048. The dimensions of semantic features are 102, 312, 85, and 85 for SUN, CUB, AWA1 and AWA2, respectively. **SUN** is a fine-grained dataset and contains 717 classes with 14,340 images in total. Those 717 classes are annotated with 102 attributes. **CUB**, another fine-grained dataset, contains 11,788 images of 200 different species of birds, annotated with 312 attributes. The **AWA1** is a coarse-grained dataset with animal images. It has 30,475 images with 50 classes, which are annotated by 85 attributes. Similar to AWA1, **AWA2** consists of 37,322 images with the same animal classes and attributes as AWA1.

We report the top-1 mean class accuracy (MCA) for both the unseen classes (U) and the seen classes (S) and also calculate the harmonic mean (HM) score, *i.e.*, $\mathrm{HM} = 2 \times \mathrm{U} \times \mathrm{S}/(\mathrm{U} + \mathrm{S})$.

We first evaluate the effectiveness of our methods by comparing them against the baseline. As shown in Table 6.5, each hyperbolic kernel brings a significant improvement to the baseline network. For example, the simplest hyperbolic tangent kernel improves the HM value over the baseline by 6.1, 21.6, 21.9 and 14.1 for SUN, CUB,

Table 6.3: Few-shot classification results on the CUB dataset with 95% confidence interval. † indicates the network was self-implemented. The 1st best in **bold font**.

Model	Backbone	1-shot	5-shot
MatchingNet [Vinyals et al., 2016]	Conv-4	61.16 ± 0.89	72.86 ± 0.70
ProtoNet [Snell et al., 2017]	Conv-4	51.31 ± 0.91	70.77 ± 0.69
MAML [Finn et al., 2017]	Conv-4	55.92 ± 0.95	72.09 ± 0.76
RelationNet [Sung et al., 2018]	Conv-4	62.45 ± 0.98	76.11 ± 0.69
DN4 [Li et al., 2019d]	Conv-4	53.15 ± 0.84	81.90 ± 0.60
Hyper ProtoNet [Khrulkov et al., 2020]	Conv-4	64.02 ± 0.20	82.53 ± 0.14
Hyperbolic tangent kernel	Conv-4	66.14 ± 0.23	82.11 ± 0.15
Hyperbolic RBF kernel	Conv-4	$\mathbf{70.98 \pm 0.22}$	$\mathbf{85.21 \pm 0.13}$
Hyperbolic Laplace kernel	Conv-4	68.27 ± 0.23	84.64 ± 0.13
Hyperbolic binomial kernel	Conv-4	69.05 ± 0.23	83.00 ± 0.14
Baseline [Chen et al., 2019c]	ResNet-18	65.51 ± 0.87	82.85 ± 0.55
Baseline++ [Chen et al., 2019c]	ResNet-18	67.02 ± 0.77	83.58 ± 0.54
RelationNet [Sung et al., 2018]	ResNet-18	67.59 ± 0.58	82.75 ± 0.58
MAML [Finn et al., 2017]	ResNet-18	69.96 ± 1.01	82.70 ± 0.65
ProtoNet [Snell et al., 2017]	ResNet-18	71.88 ± 0.91	86.64 ± 0.51
MatchingNet [Vinyals et al., 2016]	ResNet-18	72.36 ± 0.90	83.64 ± 0.60
Hyper ProtoNet† [Khrulkov et al., 2020]	ResNet-18	72.86 ± 0.22	85.69 ± 0.13
Hyperbolic tangent kernel	ResNet-18	73.52 ± 0.22	88.75 ± 0.11
Hyperbolic RBF kernel	ResNet-18	$\mathbf{75.79 \pm 0.21}$	$\mathbf{89.98 \pm 0.11}$
Hyperbolic Laplace kernel	ResNet-18	74.37 ± 0.21	89.08 ± 0.12
Hyperbolic binomial kernel	ResNet-18	74.46 ± 0.22	89.28 ± 0.11

AWA1 and AWA2, respectively. In addition, the powerful hyperbolic RBF kernel or hyperbolic Laplace kernel continues to improve the representation capacity, again showing the superiority of the kernel design for embedding learning.

To further verify the effectiveness of our approach, we continue to compare our methods to a couple of popular ZSL algorithms, including the state-of-the-art non-generative methods [Zhang and Shi, 2019; Li et al., 2019c]. We observe that our hyperbolic RBF kernel and hyperbolic Laplace kernel achieve competitive results to the state-of-the-art methods across four datasets. ZSL is a very challenging task, and while none of the methods in Table 6.5 achieved the best performance across all four datasets, it is very competitive. Thus, to establish this objectively, we employ the Friedman test[2] [Demšar, 2006] to compare the algorithms. As shown in the last column of Table 6.5, the ranking list clearly shows that our methods with the hyperbolic Laplace kernel and the hyperbolic RBF kernel are the best two options in general for the ZSL task.

6.4.3 Person Re-Identification

Person retrieval or person re-identification (re-ID) is an important application in the video/multi-camera surveillance task [Su et al., 2017]. Following the work [Khrulkov et al., 2020], ResNet-50, pre-trained on ImageNet, is employed as a backbone network

[2]The Friedman test is a non-parametric measure for multiple datasets. It ranks the algorithms for each dataset separately and calculates the average ranks for each dataset as a ranking score.

Table 6.4: Few-shot classification results on the *tiered*-ImageNet and the FC100 datasets with 95% confidence interval. † indicates the network was self-implemented. The 1st best in **bold font**.

Model	*tiered*-ImageNet		FC100	
	1-shot	5-shot	1-shot	5-shot
Hyper ProtoNet† [Khrulkov et al., 2020]	54.44 ± 0.23	71.96 ± 0.20	37.59 ± 0.19	51.76 ± 0.19
Hyperbolic tangent kernel	54.73 ± 0.22	74.37 ± 0.18	37.66 ± 0.17	52.29 ± 0.18
Hyperbolic RBF kernel	$\mathbf{57.78 \pm 0.23}$	76.11 ± 0.18	$\mathbf{38.93 \pm 0.18}$	$\mathbf{54.40 \pm 0.18}$
Hyperbolic Laplace kernel	57.33 ± 0.22	$\mathbf{76.48 \pm 0.18}$	37.99 ± 0.17	53.54 ± 0.18
Hyperbolic binomial kernel	56.72 ± 0.22	75.87 ± 0.18	38.32 ± 0.18	53.50 ± 0.18

and we also perform experiments across three dimensions, *i.e.*, 32, 64, 128, for the feature representation. The pipeline of the deep network for person re-ID is shown in Fig. 6.1(c). Both **Market-1501** [Zheng et al., 2015] and **DukeMTMC-reID** [Ristani et al., 2016] pedestrian datasets are used to evaluate our approaches. The **Market-1501** dataset consists of 32,668 pedestrian images, captured by 6 disjoint cameras. The person bounding boxes are detected automatically by DPM Felzenszwalb et al. [2010]. This dataset is split into 12,936 images of 751 identities for training and 19,732 of 750 identities for testing. **DukeMTMC-reID** is collected by 8 non-overlapped cameras and the person bounding boxes are manually annotated. Following the standard training protocol, this dataset is divided into 16,522 and 19,889 images for training and testing, respectively.

We use both mean average precision (mAP) and rank-1 accuracy of cumulative matching characteristic (CMC) to evaluate our algorithms. Different from FSL and ZSL, we use the generalised hyperbolic Laplace kernel in the re-ID experiment, as we observe that the generalised hyperbolic Laplace kernel achieves fairly good performance compared to the hyperbolic Laplace one.

We compare the proposed algorithms to the methods in [Khrulkov et al., 2020]. As shown in Table 6.6, we observe that our algorithms bring positive effects to the retrieval performance on both datasets, especially for the mAP value. In the market-1501 dataset, most of our methods achieve competitive performance compared to [Khrulkov et al., 2020]. However, we also observe that the binomial kernel cannot perform well in different embedding sizes. In the DukeMTMC-reID dataset, our method could outperform its hyperbolic counterpart on both R-1 and mAP values and the RBF kernel is the most powerful one, which is superior to the other kernels in every dimension. For example, the hyperbolic RBF kernel improves the R-1 / mAP values over the work [Khrulkov et al., 2020] by 5.1 / 6.6, 3.0 / 7.2 and 1.9 / 6.8 for the dimension of 32, 64 and 128, respectively.

6.4.4 Knowledge Distillation

Knowledge distillation (KD) is an efficient method to train a small student network, under the supervision of a pre-trained larger teacher network [Hinton et al., 2014]. In the teacher-student network (see Fig. 6.1(d)), the output of the teacher network acts

Table 6.5: Zero-shot recognition results on SUN, CUB, AWA1 and AWA2 datasets. U and S indicate the accuracy for unseen and seen classes, respectively. HM is the harmonic mean of U and S. The 1st best in **bold font**.

Model	SUN			CUB			AWA1			AWA2			Friedman test (rank)
	U	S	HM	U	S	HM	U	S	HM	U	S	HM	
LATEM [Xian et al., 2016]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	12.0 (12)
DEVISE [Frome et al., 2013]	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	10.0 (11)
DEM [Zhang et al., 2017]	20.5	34.3	25.6	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1	9.33 (9)
ALE [Akata et al., 2015]	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	9.33 (9)
SP-AEN [Chen et al., 2018b]	24.9	38.6	30.3	34.7	70.6	46.6	-	-	-	23.3	**90.9**	37.1	7.67 (7)
CRnet [Zhang and Shi, 2019]	34.1	36.5	35.3	45.5	56.8	50.5	58.1	74.7	65.4	52.6	78.8	63.1	3.00 (4)
Kai *et al.* [Li et al., 2019c]	36.3	42.8	39.3	**47.4**	47.6	47.5	**62.7**	77.0	69.1	**56.4**	81.4	**66.7**	2.83 (3)
Baseline	22.8	38.0	28.5	18.6	44.6	26.3	29.8	76.4	42.9	25.5	76.4	38.2	8.67 (8)
Hyperbolic tangent kernel	29.4	42.0	34.6	40.8	**58.1**	47.9	52.3	85.2	64.8	37.1	88.5	52.3	5.00 (5)
Hyperbolic RBF kernel	**37.0**	**43.3**	**39.9**	44.6	57.8	50.3	59.0	84.6	69.5	42.9	89.5	57.9	2.67 (2)
Hyperbolic Laplace kernel	35.1	44.2	39.1	46.2	56.1	**50.7**	60.7	83.5	**70.3**	54.1	87.1	**66.7**	1.83 (1)
Hyperbolic binomial kernel	26.9	43.8	33.3	39.8	56.9	46.8	43.7	**88.9**	58.6	39.8	90.5	55.4	5.67 (6)

Table 6.6: Person re-ID results on the Market-1501 and the DukeMTMC-reID datasets. The value in $\boxed{\cdot}$ denotes the result below the performance in [Khrulkov et al., 2020]. g-Hyperbolic Laplace kernel indicates the generalised hyperbolic Laplace kernel. The 1st best in **bold font**.

Model	Dim	Market-1501		DukeMTMC-reID	
		R-1	mAP	R-1	mAP
Euclidean [Khrulkov et al., 2020]	#32	68.0	43.4	57.2	35.7
Hyperbolic [Khrulkov et al., 2020]	#32	75.9	51.9	62.2	39.1
Hyperbolic tangent kernel	#32	$\boxed{75.4}$	53.3	63.9	42.5
Hyperbolic RBF kernel	#32	76.0	54.3	**67.3**	**46.3**
g-Hyperbolic Laplace kernel	#32	**78.7**	**56.3**	64.1	40.7
Hyperbolic binomial kernel	#32	$\boxed{75.2}$	55.0	63.7	44.7
Euclidean [Khrulkov et al., 2020]	#64	80.5	57.8	68.3	45.5
Hyperbolic [Khrulkov et al., 2020]	#64	84.4	62.7	70.8	48.6
Hyperbolic tangent kernel	#64	**85.8**	68.0	**73.9**	54.2
Hyperbolic RBF kernel	#64	85.2	65.7	73.8	**55.8**
g-Hyperbolic Laplace kernel	#64	85.4	**68.4**	73.3	50.6
Hyperbolic binomial kernel	#64	$\boxed{83.0}$	64.6	71.5	54.0
Euclidean [Khrulkov et al., 2020]	#128	86.0	67.3	74.1	53.3
Hyperbolic [Khrulkov et al., 2020]	#128	87.8	68.4	76.5	55.4
Hyperbolic tangent kernel	#128	**89.4**	**74.1**	**78.6**	60.9
Hyperbolic RBF kernel	#128	88.9	73.5	78.4	**62.2**
g-Hyperbolic Laplace kernel	#128	$\boxed{87.6}$	72.4	77.3	59.6
Hyperbolic binomial kernel	#128	$\boxed{87.6}$	72.0	$\boxed{75.4}$	59.2

as ground truth to train a student network. For a training image (*e.g.*, $\boldsymbol{X}$), the teacher network and student network generate the prediction scores $\boldsymbol{g} = [g_1, g_2, \ldots, g_N]$ and $\boldsymbol{p} = [p_1, p_2, \ldots, p_N]$, respectively. Noted that $\boldsymbol{g}$ and $\boldsymbol{p}$ are normalised by the softmax function. Then the KD loss is given by:

$$\mathcal{L}_{kd} = -\sum_{i=1}^{N} g_i \log(p_i). \tag{6.17}$$

We use the ResNet-20 as a teacher network and a simple 4-layer CNN as a student network. We report the results on **CIFAR-10** and **CIFAR-100** benchmarks [Krizhevsky, 2009]. Both CIFAR-10 and CIFAR-100 have 50,000 images for training and 10,000 images for evaluation. **CIFAR-10** contains 10 classes, with each containing 5,000 samples, while **CIFAR-100** contains 100 classes, and each class has 500 samples. The input size of CIFAR-10 and CIFAR-100 are fixed to 32×32. We use the top-1 mean accuracy to evaluate the networks. Please refer to the supplementary material for more details about the network training and corresponding hyper-parameters. As shown in Table 6.7, we can again find that our hyperbolic kernels improve the accuracy over the baseline, and the hyperbolic RBF kernel brings the maximum performance gain, 3.1 / 4.5 for CIFAR-10 / CIFAR-100, respectively.

Table 6.7: Knowledge distillation results on the CIFAR-10 / 100 datasets. g-Hyperbolic Laplace kernel indicates the generalised hyperbolic Laplace kernel. The 1^{st} best in **bold font**.

Model	CIFAR-10	CIFAR-100
Baseline	80.5	49.9
Hyperbolic tangent kernel	82.1	50.5
Hyperbolic RBF kernel	**83.6**	**54.4**
g-Hyperbolic Laplace kernel	83.2	53.9
Hyperbolic binomial kernel	81.6	51.8

6.4.5 Further Studies

To the best of our knowledge, our work is the first to develop pd kernels in hyperbolic spaces. That said, indefinite hyperbolic kernels are developed in [Cho et al., 2019]. We compare and contrast the two school of thoughts. In doing so, we consider the problem of few-shot learning and follow the setup of [Khrulkov et al., 2020]. As for the indefinite kernel, we use the Minkowski inner product kernel, presented in [Cho et al., 2019] (see supplementary material for details). We have evaluated the performance of our pd kernels and the indefinite kernel for the task of 5-way 5-shot learning across the *mini*ImageNet, CUB, *tired*-ImageNet and FC100 datasets. Fig. 6.2 shows that the performance attained by the indefinite kernel does not match that of pd kernels, clearly showing the potential of pd kernels for hyperbolic representations.

One may wonder how useful the hyperbolic spaces are and their kernels in comparison to simple Euclidean kernels. In the end, the Poincaré ball is embedded in n-dimensional Euclidean spaces and hence conventional kernels can be applied seamlessly. In Fig. 6.3, we compare the proposed kernels against their Euclidean counterparts again on the task of few-shot learning using the *mini*ImageNet dataset. We observe: **(1)** the kernel machines in both Euclidean spaces and hyperbolic spaces bring performance gain to the deep neural network. **(2)** The proposed hyperbolic kernels can outperform the vanilla Euclidean kernels significantly, again showing the reasonable design of the proposed kernels.

Remark 9 (Good Practice of Employing Hyperbolic Geometry) *Few works have studied the problem of learning an embedding in hyperbolic spaces [Chen et al., 2020a; Khrulkov et al., 2020]. However, the existing works generate the vectors in the tangent space at the origin and project to the hyperbolic spaces using $\Gamma_{\mathbf{0}}(\cdot)$ mapping. A drawback of this framework is that the hyperbolic geometry is not fully utilised as every representation is flattened at the identity. In other words, only the vectors very close to the origin represent hyperbolic distances. In contrast, and in our experiments, we generate hyperbolic representations directly in the Poincaré ball. Empirically, we observe that various applications can benefit from a high curvature (*i.e.*, c). For example, in the person re-identification task, the curvature of the Poincaré ball is 10^{-2} in our algorithms, while the work in [Khrulkov et al., 2020] sets it*

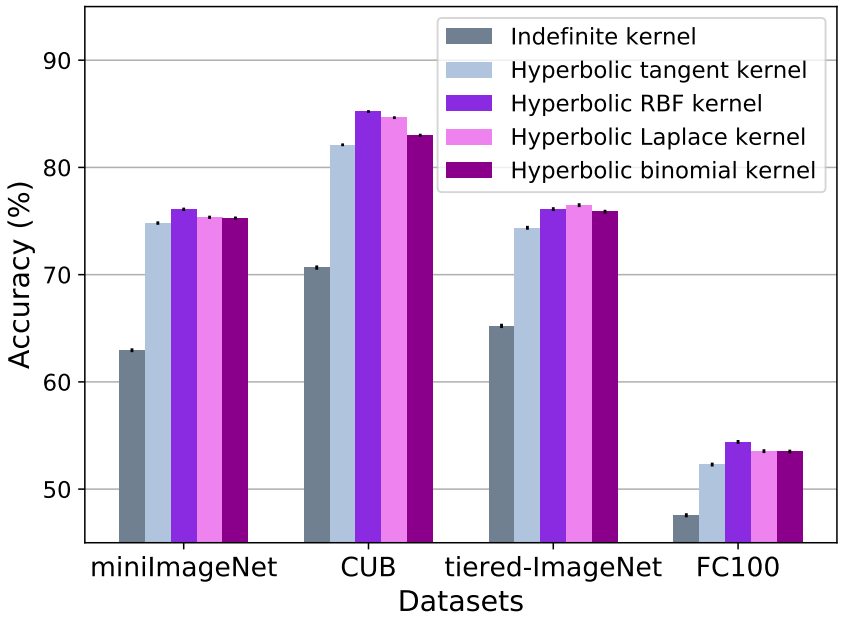


Figure 6.2: The performance comparison between the indefinite kernel and pd kernels for hyperbolic representations.

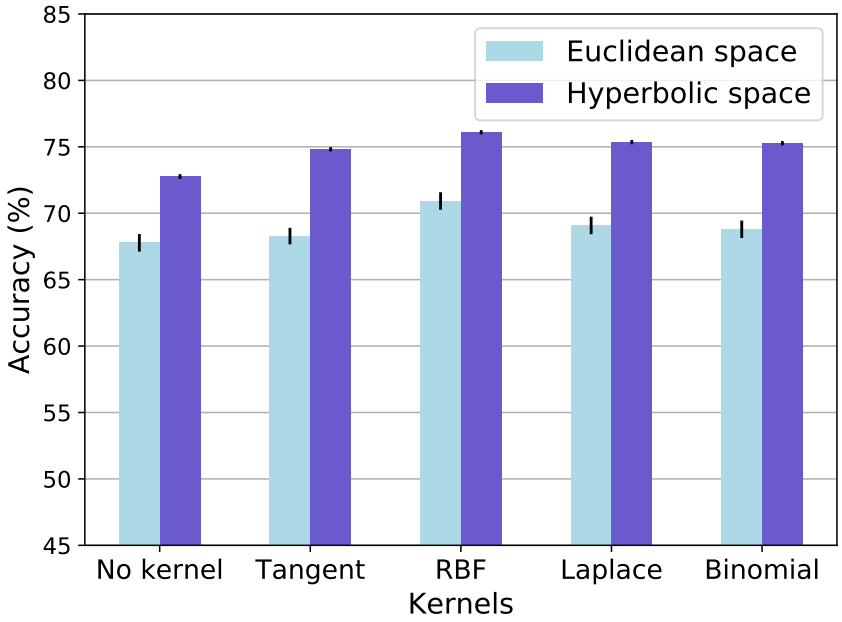


Figure 6.3: The performance comparison for kernels on Euclidean spaces and Hyperbolic spaces.

to 10^{-5}*, which makes the Poincaré ball very flat.*

6.5 Summary

This chapter proposes a family of positive definite kernels to embed hyperbolic representations in Hilbert spaces. In such kernels, we leverage the identity tangent space of the Poincaré ball and further define valid positive definite kernels in identity tangent spaces. The proposed kernels include powerful universal kernels (i.e., the hyperbolic RBF kernel, the hyperbolic Laplace kernel and the hyperbolic binomial kernel). We evaluate the effectiveness of the kernels in a variety of challenging applications, such as few-shot learning, zero-shot learning, person re-identification and knowledge distillation, and the empirical results have shown positive results for embedding learning via the kernels in hyperbolic spaces. Future works include exploiting the proposed kernels to other applications (*i.e.*, natural language processing and graph neural networks). In addition, we have found that the effectiveness of the kernel is data-dependent and we want to develop a rule for choosing the right kernel for a given data.

Chapter 7

Conclusion

This Thesis focuses on learning embeddings for visual data, *i.e.*, image/video. We contribute by providing two separate perspectives, visual attention and geometry constraint.

In the first part, we propose two attention modules, *i.e.*, Attention in Attention and Channel Recurrent Attention. The major contributions of this part are outlined below:

- We propose a novel attention mechanism, termed Attention in Attention, or AiA for short. In AiA, we explicitly model the interaction between the inner attention and the outer attention. Such interaction only helps the network to localise the information region of the feature map, but also preserves the spatial structural information of the feature map. We generalise the AiA mechanism by benefiting from the rich structure of Hilbert Spaces. To achieve this, we employ advanced kernel approximation techniques to map the feature to Reproducing Kernel Hilbert Spaces (RKHSs). The superiority of AiA and its generalisation is verified by extensive experiments on the person re-identification task.

- We then develop another attention mechanism, Channel Recurrent Attention (CRA), to make better use of information in the feature map. The existing attention mechanism cannot learn both spatial and channel features. Our work aims to build a global receptive field to its input feature map. The CRA first flattens each slice of the feature map to a spatial vector. Then an inbuilt LSTM unit receives the spatial vectors sequentially and produces a sequence of hidden states as an attention map. As a result, the fully-connected layers in the CRA have a global receptive field to the spatial vector, while the recurrent operation of the LSTM learns the channel pattern of the feature map. Extensive experiments on image and video applications verify the effectiveness of our approach.

In the second part, we investigate two geometry constraints for the embedding, including the set and hyperbolic geometry. The major contributions of the second part are outlined below:

- We develop a set-aware triplet loss to optimise the frame features of the video person re-identification task, by modelling the video clip as a set. We first

employ the well-known set distance metrics, including ordinary distance and Hausdorff distance. Considering the nature of the triplet loss (minimising the distance of positive pairs and maximising the distance of negative pairs jointly), we separately define the set distance for the anchor-positive pair and the anchor-negative pair as our hybrid distance metric. Extensive experiments are conducted to verify the superior performance of the proposed method across the standard video person re-identification datasets.

- We then study a powerful curved space, hyperbolic geometry as embedding spaces for visual data. We propose a family of positive definite kernels to embed hyperbolic representations in Hilbert spaces. In our work, we use the Poincaré ball to model the hyperbolic space and define the positive kernels by leveraging the identity tangent plane of the Poincaré ball. The proposed positive kernels include powerful universal ones, *i.e.*, the hyperbolic RBF kernel, the hyperbolic Laplace kernel and the hyperbolic binomial kernel. Extensive experiments on few-shot learning, zero-shot learning, person re-identification and knowledge distillation verify the power of the proposed kernels.

7.1 Future Work

This Thesis focuses on the embedding learning for visual data. Below, we list some potential future works based on insights from our research.

- **Mixed-curvature embeddings.** Existing visual embedding techniques use only a single geometry as an embedding space. However, such embeddings cannot fully encode the structured data, since the data is not distributed uniformly. This issue can be addressed by learning embeddings in a product manifold. This product manifold includes a mixture of Euclidean spaces, hyper-sphere spaces and hyperbolic spaces, thereby being able to encode a wide variety of structures of visual data.

- **Embedding learning to graph and language data.** In this Thesis, the proposed approaches are developed for visual data, *i.e.*, image or video. In contrast to visual data, graph and language data contain structure or sequential information, which brings difficulty to learn embeddings. Complex attention mechanisms (*i.e.*, self-attention) or embedding spaces can be investigated and applied to such problems.

- **Other advanced settings of person re-identification.** In this Thesis, most of the approaches are verified on a fully supervised person re-identification (re-ID) task. As our future research, we will investigate other settings, *i.e.*, unsupervised, semi-supervised, cross-domain, or cross-modality person re-ID tasks. The performance of those tasks also highly depends on the quality of the embedding space.

Bibliography

Absil, P.-A.; Mahony, R.; and Sepulchre, R., 2007. *Optimization algorithms on matrix manifolds*. Princeton University Press. (cited on page 17)

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C., 2015. Label-embedding for image classification. *TPAMI*, (2015). (cited on pages 101, 102, and 105)

Bahdanau, D.; Cho, K.; and Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*. (cited on page 3)

Bai, S.; Bai, X.; and Tian, Q., 2017. Scalable person re-identification on supervised smoothed manifold. In *CVPR*. (cited on page 18)

Bai, X.; Yang, M.; Huang, T.; Dou, Z.; Yu, R.; and Xu, Y., 2020. Deep-person: Learing discriminative deep features for person re-identification. *Pattern Recognition*, (2020). (cited on page 58)

Basri, R. and Jacobs, D. W., 2003. Lambertian reflectance and linear subspaces. *TPAMI*, (2003). (cited on page 4)

Berg, C.; Christensen, J. P. R.; and Ressel, P., 1984. *Harmonic Analysis on Semigroups*. Springer. (cited on pages 93, 97, and 98)

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y., 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*. (cited on page 36)

Chang, X.; Hospedales, T. M.; and Xiang, T., 2018. Multi-level factorisation net for person re-identification. In *CVPR*. (cited on page 41)

Chen, B.; Deng, W.; and Hu, J., 2019a. Mixed high-order attention network for person re-identification. In *ICCV*. (cited on pages 53, 54, and 55)

Chen, D.; Li, H.; Xiao, T.; Yi, S.; and Wang, X., 2018a. Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding. In *CVPR*. (cited on pages 61, 62, 81, and 82)

Chen, J.; Qin, J.; Shen, Y.; Liu, L.; Zhu, F.; and Shao, L., 2020a. Learning attentive and hierarchical representations for 3d shape recognition. In *ECCV*. (cited on pages 92 and 107)

Chen, L.; Zhang, H.; Xiao, J.; Liu, W.; and Chang, S.-F., 2018b. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*. (cited on page 105)

Chen, T.; Ding, S.; Xie, J.; Yuan, Y.; Chen, W.; Yang, Y.; Ren, Z.; and Wang, Z., 2019b. Abd-net: Attentive but diverse person re-identification. In *ICCV*. (cited on pages 3, 41, and 70)

Chen, W.; Chen, X.; Zhang, J.; and Huang, K., 2017a. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*. (cited on page 18)

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; and Huang, J.-B., 2019c. A closer look at few-shot classification. In *ICLR*. (cited on pages 102 and 103)

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z., 2020b. Dynamic convolution: Attention over convolution kernels. In *CVPR*. (cited on page 2)

Chen, Y.; Zhu, X.; Zheng, W.; and Lai, J., 2017b. Person re-identification by camera correlation aware feature augmentation. *TPAMI*, (2017). (cited on page 23)

Cho, H.; DeMeo, B.; Peng, J.; and Berger, B., 2019. Large-margin classification in hyperbolic space. In *ICML*. (cited on pages 5, 90, 93, and 107)

Christmann, A. and Steinwart, I., 2008. *Support Vector Machines*. Springer. (cited on pages 96 and 99)

Cui, Y.; Zhou, F.; Wang, J.; Liu, X.; Lin, Y.; and Belongie, S., 2017. Kernel pooling for convolutional neural networks. In *CVPR*. (cited on page 24)

Dehghan, A.; Assari, S. M.; and Shah, M., 2015. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*. (cited on pages 50, 61, and 81)

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. *JMLR*, (2006). (cited on page 103)

Deng, J.; Dong, W.; Li, R. S. L.-J.; Li, K.; and Li, F.-F., 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*. (cited on page 101)

Fang, P.; Harandi, M.; and Petersson, L., 2021a. Kernel methods in hyperbolic spaces. In *ICCV*. (cited on pages 7 and 89)

Fang, P.; Ji, P.; Petersson, L.; and Harandi, M., 2021b. Set augmented triplet loss for video person re-identification. In *WACV*. (cited on pages 7 and 73)

Fang, P.; Ji, P.; Zhou, J.; Petersson, L.; and Harandi, M., 2020. Channel recurrent attention networks for video pedestrian retrieval. In *ACCV*. (cited on pages 7 and 53)

Fang, P.; Zhou, J.; Roy, S. K.; Ji, P.; Petersson, L.; and Harandi, M., 2021c. Attention in at-tention networks for person retrieval. *TPAMI*, (2021). (cited on pages 6 and 23)

Fang, P.; Zhou, J.; Roy, S. K.; Petersson, L.; and Harandi, M., 2019. Bilinear attention networks for person retrieval. In *ICCV*. (cited on pages 7, 23, 24, 53, 54, 55, 56, 65, 69, 70, and 76)

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D., 2010. Object Detection with Discriminatively Trained Part-Based Models. *TPAMI*, (2010). (cited on pages 38, 50, 61, 81, and 104)

Feragen, A. and Hauberg, S., 2016. Open problem: Kernel methods on manifolds and metric spaces. what is the probability of a positive definite geodesic exponential kernel? In *CoLT*. (cited on page 90)

Feragen, A.; Lauze, F.; and Hauberg, S., 2015. Geodesic exponential kernels: when curvature and linearity conflict. In *CVPR*. (cited on page 90)

Finn, C.; Abbeel, P.; and Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. (cited on pages 102 and 103)

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M. A.; and Mikolov, T., 2013. Devise: A deep visual-semantic embedding model. In *NeurIPS*. (cited on page 105)

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H., 2019a. Dual attention network for scene segmentation. In *CVPR*. (cited on pages 10, 13, and 54)

Fu, Y.; Wang, X.; Wei, Y.; and Huang, T., 2019b. Sta: Spatial-temporal attention for large-scale video-based person re-identification. In *AAAI*. (cited on pages 19, 50, 51, 62, 63, 76, 82, and 83)

Fu, Y.; Wei, Y.; Zhou, Y.; Shi, H.; Huang, G.; Wang, X.; Yao, Z.; and Huang, T., 2019c. Horizontal pyramid matching for person re-identification. In *AAAI*. (cited on pages 40 and 41)

Ganea, O.-E.; Bécigneul, G.; and Hofmann, T., 2018. Hyperbolic neural networks. In *NeurIPS*. (cited on pages 4, 90, and 92)

Gao, J. and Nevatia, R., 2018. Revisiting temporal modeling for video-based person reid. *arXiv:1805.02104*, (2018). (cited on pages 19, 50, 56, 60, 62, and 76)

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T., 2016. Compact bilinear pooling. In *CVPR*. (cited on page 26)

Gheissari, N.; Sebastian, T. B.; Tu, P. H.; Rittscher, J.; and Hartley, R., 2006. Person reidentification using spatiotemporal appearance. In *CVPR*. (cited on page 18)

Girshick, R., 2015. Fast r-cnn. In *ICCV*. (cited on page 38)

Gong, S.; Cristani, M.; Yan, S.; and Loy, C. C., 2014. *Person Re-Identification*. Springer. (cited on page 18)

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Scholköpf, B.; and Smola, A., 2012. A kernel two-sample test. *JMLR*, (2012). (cited on page 93)

Gu, A.; Sala, F.; Gunel, B.; and Ré, C., 2019. Learning mixed-curvature representations in product spaces. In *ICLR*. (cited on page 92)

Gulcehre, C.; Denil, M.; Malinowski, M.; Razavi, A.; Pascanu, R.; Hermann, K. M.; Battaglia, P.; Bapst, V.; Raposo, D.; Santoro, A.; and de Freitas, N., 2019. Hyperbolic attention networks. In *ICLR*. (cited on pages 4, 5, and 92)

Hamann, M., 2011. On the tree-likeness of hyperbolic spaces. *arXiv:1105.3925*, (2011). (cited on page 90)

Harandi, M. T.; Salzmann, M.; Jayasumana, S.; Hartley, R.; and Li, H., 2014. Expanding the family of grassmannian kernels: An embedding perspective. In *ECCV*. (cited on page 93)

Hartley, R.; Trumpf, J.; Dai, Y.; and Li, H., 2012. Rotation average. *IJCV*, (2012). (cited on page 94)

He, K.; Girshick, R.; and Dollar, P., 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (cited on page 2)

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R., 2017. Mask r-cnn. In *ICCV*. (cited on page 10)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *CVPR*. (cited on pages 2, 10, 11, 42, 46, 49, 60, 75, 79, and 101)

Hermans, A.; Beyer, B.; and Leibe, B., 2017. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, (2017). (cited on pages 56 and 76)

Hinton, G.; Vinyals, O.; and Dean, J., 2014. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*. (cited on page 104)

Hirzer, M.; Beleznai, C.; Roth, P. M.; and Bischof, H., 2011. Person re-identification by descriptive and discriminative classification. In *Image Analysis*. (cited on pages 60, 75, and 81)

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y., 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*. (cited on page 54)

Hofmann, T.; Scholkopf, B.; and Smola, A. J., 2008. Kernel methods in machine learning. *The Annals of Statistics*, (2008). (cited on pages 26, 92, and 93)

Hou, R.; Ma, B.; Chang, H.; Gu, X.; Shan, S.; and Chen, X., 2019a. Interaction-and-aggregation network for person re-identification. In *CVPR*. (cited on pages 39 and 41)

Hou, R.; Ma, B.; Chang, H.; Gu, X.; Shan, S.; and Chen, X., 2019b. Vrstc: Occlusion-free video person re-identification. In *CVPR*. (cited on pages 61, 62, 63, 82, and 83)

Hu, J.; Shen, L.; and Sun, G., 2018. Squeeze-and-excitation networks. In *CVPR*. (cited on pages 3, 13, 14, 24, 30, 43, 49, 54, 59, 60, and 79)

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q., 2017. Densely connected convolutional networks. In *CVPR*. (cited on pages 2, 11, and 49)

Huang, H.; Li, D.; Zhang, Z.; Chen, X.; and Huang, K., 2018. Adversarially occluded samples for person re-identification. In *CVPR*. (cited on pages 38 and 41)

Huang, L.; Wang, W.; Chen, J.; and Wei, X.-Y., 2019. Attention on attention for image captioning. In *ICCV*. (cited on page 13)

Hubei, D. and Wiesel, T., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, (1962). (cited on page 2)

Ioffe, S. and Szegedy, C., 2015. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. (cited on page 37)

Jacot, A.; Gabriel, F.; and Hongler, C., 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*. (cited on page 93)

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and kavukcuoglu, k., 2015. Spatial transformer networks. In *NeurIPS*. (cited on pages 3, 18, and 24)

Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; and Harandi, M., 2013. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*. (cited on page 92)

Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; and Harandi, M., 2015. Kernel methods on riemannian manifolds with gaussian RBF kernels. *TPAMI*, (2015). (cited on pages 90, 93, 97, and 99)

Jayasumana, S.; Ramalingam, S.; and Kumar, S., 2020. Kernelized classification in deep networks. ArXiv:2012.09607 [cs.ML]. (cited on page 24)

Jetley, S.; Lord, N. A.; Lee, N.; and Torr, P. H., 2018. Learn to pay attention. In *ICLR*. (cited on page 3)

Joachims, T., 2006. Training linear svms in linear time. In *KDD*. (cited on page 28)

Karcher, H., 1977. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, (1977). (cited on page 90)

Khrulkov, V.; Mirvakhabova, L.; Ustinova, E.; Oseledets, I.; and Lempitsky, V., 2020. Hyperbolic image embeddings. In *CVPR*. (cited on pages xxiii, 4, 5, 90, 92, 101, 102, 103, 104, 106, and 107)

Kim, J.-H.; On, K.-W.; Lim, W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T., 2017. Hadamard product for low-rank bilinear pooling. In *ICLR*. (cited on page 26)

Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on pages 2, 37, 60, and 80)

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. In *Technical report*. (cited on page 106)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. (cited on pages 2 and 11)

Lampert, C. H.; Nickisch, H.; and Harmeling, S., 2013. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, (2013). (cited on page 102)

Laurens van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-sne. *JMLR*, (2008). (cited on pages xix and 87)

Le, T. and Yamada, M., 2018. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. In *NeurIPS*. (cited on page 93)

LeCun, Y.; Bengio, Y.; and Hinton, G., 2015. Deep learning. *Nature*, (2015). (cited on pages 1 and 10)

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P., 1998. Gradient-based learning applied to document recognition. *IEEE*, (1998). (cited on pages 1 and 9)

Li, D.; Chen, X.; Zhang, Z.; and Huang, K., 2017. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*. (cited on pages 18, 41, and 62)

Li, J.; Wang, J.; Tian, Q.; Gao, W.; and Zhang, S., 2019a. Global-local temporal representation for video person re-identification. In *ICCV*. (cited on pages 13, 18, 50, 51, 61, 62, 63, 79, 82, and 83)

Li, J.; Zhang, S.; Tian, Q.; Wang, M.; and Gao, W., 2019b. Pose-guided representation learning for person re-identification. *TPAMI*, (2019). (cited on pages 23 and 24)

Li, K.; Min, M. R.; and Fu, Y., 2019c. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*. (cited on pages 103 and 105)

Li, S.; Bak, S.; Carr, P.; and Wang, X., 2018a. Diversity regularized spatiotemporal attention for video-based person re-identification. In *CVPR*. (cited on pages 62 and 82)

Li, W.; Jafari, O. H.; and Rother, C., 2018b. Deep object co-segmentation. In *ACCV*. (cited on page 54)

Li, W.; Wang, L.; Xu, J.; Huo, J.; Yang, G.; and Luo, J., 2019d. Revisiting local descriptor based image-to-class measure for few-shot learning. In *CVPR*. (cited on pages 102 and 103)

Li, W.; Zhao, R.; and Wang, X., 2012. Human reidentification with transferred metric learning. In *ACCV*. (cited on pages 25, 37, 38, and 70)

Li, W.; Zhao, R.; Xiao, T.; and Wang, X., 2014. DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. In *CVPR*. (cited on pages 25, 37, and 38)

Li, W.; Zhu, X.; and Gong, S., 2018c. Harmonious attention network for person re-identification. In *CVPR*. (cited on pages 3, 18, 24, 35, 36, 37, 41, 53, 54, and 55)

Lin, T.-Y. and Maji, S., 2017. Improved Bilinear Pooling with CNNs. In *BMVC*. (cited on page 26)

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S., 2015. Bilinear cnn models for fine-grained visual recognition. In *ICCV*. (cited on pages 24, 25, and 26)

Liu, H.; Feng, J.; Jiang, J.; and Yan, S., 2016. End-to-end comparative attention networks for person re-identification. https://arxiv.org/abs/1606.04404. ArXiv:1606.04404 [cs.CV]. (cited on pages 13, 18, 55, and 58)

Liu, J.; Yang, Z.; Tao, Z.; and Huilin, X., 2017a. Multi-part compact bilinear cnn for person re-identification. In *ICIP*. (cited on page 26)

Liu, Q.; Nickel, M.; and Kiela, D., 2019a. Hyperbolic graph neural networks. In *NeurIPS*. (cited on pages 4, 90, and 92)

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L., 2017b. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*. (cited on pages 4 and 92)

Liu, X.; Zhao, H.; Tian, M.; Sheng, L.; Shao, J.; Yi, S.; Yan, J.; and Wang, X., 2017c. Hydraplus-net: Attentive deep features for pedestrian analysis. In *ICCV*. (cited on page 55)

Liu, Y.; Junjie, Y.; and Ouyang, W., 2017d. Quality aware network for set to set recognition. In *CVPR*. (cited on pages 62 and 82)

Liu, Y.; Yuan, Z.; Zhou, W.; and Li, H., 2019b. Spatial and temporal mutual promotion for video-based person re-identification. In *AAAI*. (cited on pages 62 and 82)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. (cited on page 10)

Lou, A.; Katsman, I.; Jiang, Q.; Belongie, S.; Lim, S.-N.; and Sa, C. D., 2020. Differentiating through the fréchet mean. In *ICML*. (cited on pages 5 and 90)

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2004), 91–110. (cited on page 1)

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J., 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*. (cited on pages 2 and 49)

Maji, S. and Berg, A. C., 2009. Max-margin additive classifiers for detection. In *ICCV*. (cited on page 28)

McLaughlin, N.; Martinez del Rincon, J.; and Miller, P., 2016. Recurrent convolutional network for video-based person re-identification. In *CVPR*. (cited on pages 18 and 62)

Meng, Y.; Huang, J.; Wang, G.; Zhang, C.; Zhuang, H.; Kaplan, L.; and Han, J., 2019. Spherical text embedding. In *NeurIPS*. (cited on page 92)

Micchelli, C. A.; Xu, Y.; and Zhang, H., 2006. Universal kernels. *JMLR*, (2006). (cited on page 93)

Mnih, V.; Heess, N.; Graves, A.; and Kavukcuoglu, K., 2014. Recurrent models of visual attention. In *NeurIPS*. (cited on page 3)

Ni, X.; Fang, L.; and Huttunen, H., 2020. Adaptive l2 regularization in person re-identification. In *ICPR*. (cited on page 36)

Oreshkin, B.; Rodríguez López, P.; and Lacoste, A., 2018. TADAM: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*. (cited on page 101)

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A., 2017. Automatic differentiation in pytorch. In *NIPS*. (cited on pages 37, 60, and 79)

Patterson, G. and Hays, J., 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*. (cited on page 102)

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z., 2019. Correlation congruence for knowledge distillation. In *ICCV*. (cited on page 24)

Pirsiavash, H.; Ramanan, D.; and Fowlkes, C. C., 2009. Bilinear classifiers for visual recognition. In *NeurIPS*. (cited on page 26)

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J., 2016. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv:1612.00593*, (2016). (cited on page 75)

Qian, X.; Fu, Y.; Xiang, T.; Jiang, Y.; and Xue, X., 2019. Leader-based multi-scale attention deep architecture for person re-identification. *TPAMI*, (2019). (cited on pages 24, 39, and 49)

Quan, R.; Dong, X.; Wu, Y.; Zhu, L.; and Yang, Y., 2019. Auto-reid: Searching for a part-aware convnet for person re-identification. In *ICCV*. (cited on page 40)

Rahimi, A. and Recht, B., 2008. Random features for large-scale kernel machines. In *NeurIPS*. (cited on pages 24, 25, 26, 28, 33, and 34)

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S., 2018. Meta-learning for semi-supervised few-shot classification. In *ICLR*. (cited on page 101)

Ribera, J.; Güera, D.; Chen, Y.; and Delp, E. J., 2019. Locating objects without bounding boxes. In *CVPR*. (cited on pages 4, 75, and 76)

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*. (cited on pages 25, 37, 38, 70, and 104)

Roy, S. K.; Harandi, M.; Nock, R.; and Hartley, R., 2019. Siamese networks: The tale of two manifolds. In *ICCV*. (cited on pages 18 and 76)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al., 2015. Imagenet large scale visual recognition challenge. *IJCV*, (2015). (cited on pages 37, 60, 69, and 79)

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*. (cited on pages 2 and 49)

Saquib Sarfraz, M.; Schumann, A.; Eberle, A.; and Stiefelhagen, R., 2018. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *CVPR*. (cited on page 24)

Schroff, F.; Kalenichenko, D.; and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. (cited on pages 2, 37, 56, and 76)

Shen, Y.; Li, H.; Yi, S.; Chen, D.; and Wang, X., 2018a. Person Re-identification with deep similarity-guided graph neural network. In *ECCV*. (cited on pages 39 and 41)

Shen, Y.; Xiao, T.; Li, H.; Yi, S.; and Wang, X., 2018b. End-to-end deep kronecker-product matching for person re-identification. In *CVPR*. (cited on pages 39 and 41)

Si, J.; Zhang, H.; Li, C.; Kuen, J.; Kong, X.; Alex, K. C.; and Gang, W., 2018. Dual attention matching network for context-aware feature sequence based person re-identification. In *CVPR*. (cited on pages 18 and 41)

Simon, C.; Koniusz, P.; Nock, R.; and Harandi, M., 2020. Adaptive subspaces for few-shot learning. In *CVPR*. (cited on pages 4, 92, and 102)

Simonyan, K. and Zisserman, A., 2015. Very deep convolutional network for large-scale image recognition. In *ICLR*. (cited on page 2)

Skopek, O.; Ganea, O.-E.; and Bécigneul, G., 2020. Mixed-curvature variational autoencoders. In *ICLR*. (cited on page 92)

Snell, J.; Swersky, K.; and Zemel, R., 2017. Prototypical networks for few-shot learning. In *NeurIPS*. (cited on pages 100, 101, 102, and 103)

Su, C.; Li, J.; Zhang, S.; Xing, J.; Gao, W.; and Tian, Q., 2017. Pose-driven deep convolutional model for person re-identification. In *ICCV*. (cited on pages 23, 24, 41, and 103)

Su, C.; Zhang, S.; Xing, J.; Gao, W.; and Tian, Q., 2016. Deep attributes driven multi-camera person re-identification. In *ECCV*. (cited on page 24)

Subramaniam, A.; Nambiar, A.; and Mittal, A., 2019. Co-segmentation inspired attention networks for video-based person re-identification. In *ICCV*. (cited on pages 51, 53, 62, and 82)

Suh, Y.; Han, B.; Kim, W.; and Lee, K. M., 2019. Stochastic class-based hard example mining for deep metric learning. In *CVPR*. (cited on page 76)

Suh, Y.; Wang, J.; Tang, S.; Mei, T.; and Mu Lee, K., 2018. Part-aligned bilinear representations for person re-identification. In *ECCV*. (cited on pages 18, 23, 26, 35, 36, 41, 42, 51, 53, 70, and 82)

Sun, H.; Chen, Z.; Yan, S.; and Xu, L., 2019a. Mvp matching: A maximum-value perfect matching for mining hard samples, with application to person re-identification. In *ICCV*. (cited on page 39)

Sun, Y.; Zheng, L.; Deng, W.; and Wang, S., 2017. Svdet for pedestrian retrieval. In *ICCV*. (cited on pages 18, 41, and 92)

Sun, Y.; Zheng, L.; Li, Y.; Yang, Y.; Tian, Q.; and Wang, S., 2019b. Learning part-based convolutional features for person re-identification. *TPAMI*, (2019). (cited on page 36)

Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; and Wang, S., 2018. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*. (cited on pages 23, 35, 36, 38, and 41)

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M., 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*. (cited on pages 102 and 103)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *CVPR*. (cited on pages 11, 37, 42, and 49)

Tay, C.-P.; Roy, S.; and Yap, K.-H., 2019. Aanet: Attribute attention network for person re-identifications. In *CVPR*. (cited on pages 18, 24, and 41)

Tian, M.; Yi, S.; Hongsheng, L.; Shihua, L.; Zhang, X.; Shi, J.; Yan, J.; and Wang, X., 2018. Eliminating background-bias for robust person re-identification. In *CVPR*. (cited on page 18)

Tifrea, A.; Becigneul, G.; and Ganea, O.-E., 2019. Poincare glove: Hyperbolic word embeddings. In *ICLR*. (cited on page 5)

Ustinova, E.; Ganin, Y.; and Lempitsky, V., 2015. Multi-region bilinear convolutional neural networks for person re-identification. https://arxiv.org/abs/1512.05300. ArXiv:1512.05300 [cs.CV]. (cited on page 26)

Vapnik, V., 2000. *The Nature of Statistical Learning Theory*. Springer. (cited on page 24)

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I., 2017. Attention is all you need. In *NeurIPS*. (cited on page 13)

Vedaldi, A. and Zisserman, A., 2012. Efficient additive kernels via explicit feature maps. *TPAMI*, (2012). (cited on pages 24, 25, 26, and 28)

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D., 2016. Matching networks for one shot learning. In *NeurIPS*. (cited on pages 102 and 103)

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S., 2011. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology. (cited on pages 101 and 102)

Wang, C.; Zhang, Q.; Huang, C.; Liu, W.; and Wang, X., 2018a. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *ECCV*. (cited on pages 3, 14, 18, 24, 34, 36, 40, 41, 42, 53, 54, and 55)

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X., 2017. Residual attention network for image classification. In *CVPR*. (cited on pages 14 and 54)

Wang, G.; Lai, J.; Huang, P.; and Xie, X., 2019. Spatial-temporal person re-identification. In *AAAI*. (cited on page 18)

Wang, T.; Gong, S.; Zhu, X.; and Wang, S., 2016. Person re-identification by discriminative selection in video ranking. *TPAMI*, (2016). (cited on pages 60, 75, and 81)

Wang, X.; Girshick, R.; Gupta, A.; and He, K., 2018b. Non-local neural networks. In *CVPR*. (cited on pages 3, 13, 15, 43, 46, 49, 54, 56, 65, and 76)

Wang, Y.; Wang, L.; You, Y.; Zou, X.; Chen, V.; Li, S.; Huang, G.; Hariharan, B.; and Weinberger, K. Q., 2018c. Resource aware person re-identification across multiple resolutions. In *CVPR*. (cited on pages 18 and 41)

Wei, L.; Zhang, S.; Gao, W.; and Tian, Q., 2018. Person trasfer gan to bridge domain gap for person re-identification. In *CVPR*. (cited on pages 25, 37, and 38)

Woo, S.; Park, J.; Lee, J.-Y.; and So Kweon, I., 2018. Cbam: Convolutional block attention module. In *ECCV*. (cited on pages 3 and 54)

Wu, G.; Zhu, X.; and Gong, S., 2019. Spatio-temporal associative representation for video person re-identification. In *BMVC*. (cited on pages 62, 63, 82, and 83)

Wu, Y.; Lin, Y.; Dong, X.; Yan, Y.; Ouyang, W.; and Yang, Y., 2018a. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *CVPR*. (cited on pages 60, 63, 75, 81, and 83)

Wu, Z.; Efros, A. A.; and Yu, S., 2018b. Improving generalization via scalable neighborhood component analysis. In *ECCV*. (cited on page 102)

Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; and Schiele, B., 2016. Latent embeddings for zero-shot classification. In *CVPR*. (cited on page 105)

Xiao, T.; Li, H.; Ouyang, W.; and Wang, X., 2016. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*. (cited on pages 18 and 42)

Xu, J.; Ton, J.-F.; Kim, H.; Kosiorek, A. R.; and Teh, Y. W., 2020. Metafun: Meta-learning with iterative functional updates. In *ICML*. (cited on page 24)

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2048–2057. (cited on page 3)

Yan, Y.; Ni, B.; Song, Z.; Ma, C.; Yan, Y.; and Yang, X., 2016. Person re-identification via recurrent feature aggregation. In *ECCV*. (cited on pages 18, 58, and 62)

Yang, J.; Zheng, W.-S.; Yang, Q.; Chen, Y.-C.; and Tian, Q., 2020. Spatial-temporal graph convolutional network for video-based person re-identification. In *CVPR*. (cited on pages 82 and 83)

Yi, D.; Lei, Z.; and Li, S. Z., 2014. Deep metric learning for person re-identification. In *ICPR*. (cited on page 18)

Yu, R.; Dou, Z.; Bai, S.; Zhang, Z.; Xu, Y.; and Bai, X., 2018. Hard-aware point-to-set deep metric for person re-identification. In *ECCV*. (cited on pages 4 and 36)

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J., 2017. Deep sets. In *NeurIPS*. (cited on pages 4, 74, and 75)

Zhang, C.; Fu, H.; Wang, J.; Li, W.; Cao, X.; and Hu, Q., 2020a. Tensorized multi-view subspace representation learning. *IJCV*, (2020). (cited on page 4)

Zhang, D.; Li, Y.; and Zhang, Z., 2020b. Deep metric learning with spherical embedding. *arXiv preprint arXiv:2011.02785*, (2020). (cited on page 4)

Zhang, F. and Shi, G., 2019. Co-representation network for generalized zero-shot learning. In *ICML*. (cited on pages 103 and 105)

Zhang, L.; Xiang, T.; and Gong, S., 2017. Learning a deep embedding model for zero-shot learning. In *CVPR*. (cited on page 105)

Zhang, R.; Sun, H.; Li, J.; Ge, Y.; Lin, L.; Luo, P.; and Wang, X., 2018. Scan: Self-and-collaborative attention network for video person re-identification. *arXiv:1807.05688*, (2018). (cited on pages 61, 62, and 82)

Zhang, Z.; Lan, C.; Zeng, W.; and Chen, Z., 2020c. Relation-aware global attention for person re-identification. In *CVPR*. (cited on pages 40, 41, 81, and 82)

Zhao, B.; Wu, X.; Feng, J.; Peng, Q.; and Yan, S., 2017a. Diversified visual attention networks for fine-grained object classification. *TMM*, (2017). (cited on page 58)

Zhao, H.; Tian, M.; Sun, S.; Shao, J.; Yan, J.; Yi, S.; Wang, X.; and Tang, X., 2017b. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*. (cited on pages 42 and 70)

Zhao, L.; Li, X.; Zhuang, Y.; and Wang, J., 2017c. Deeply-learned part-aligned representations for person re-identification. In *ICCV*. (cited on pages 36, 42, and 70)

Zhao, Y.; Shen, X.; Jin, Z.; Lu, H.; and Hua, X.-s., 2019. Attribute-driven feature disentangling and temporal aggregation for video person re-identification. In *CVPR*. (cited on pages 24, 51, 60, 61, 62, and 82)

Zheng, L.; Bie, Z.; Sun, Y.; Wang, J.; Su, C.; Wang, S.; and Tian, Q., 2016. Mars: A video benchmark for large-scale person re-identification. In *ECCV*. (cited on pages 26, 49, 60, 75, and 81)

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q., 2015. Scalable person re-identification: A benchmark. In *ICCV*. (cited on pages 25, 37, 38, and 104)

Zheng, Z.; Yang, X.; Yu, Z.; Zheng, L.; Yang, Y.; and Kautz, J., 2019. Joint discriminative and generative learning for person re-identification. In *CVPR*. (cited on pages 41 and 42)

Zhong, Z.; Zheng, L.; Cao, D.; and Li, S., 2017a. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*. (cited on page 38)

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y., 2017b. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, (2017). (cited on pages 2, 38, 60, 66, and 80)

Zhou, J.; Roy, S. K.; Fang, P.; Harandi, M.; and Petersson, L., 2020. Cross-correlated attention networks for person re-identification. *Image and Vision Computing*, (2020). (cited on page 53)

Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T., 2019a. Learning generalisable omni-scale representations for person re-identification. *arXiv:1910.06827v2*, (2019). (cited on page 70)

Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T., 2019b. Omni-scale feature learning for person re-identification. In *ICCV*. (cited on pages 40 and 41)

Zhou, Z.; Huang, Y.; Wang, W.; Wang, L.; and Tan, T., 2017. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *CVPR*. (cited on page 62)

Zhu, Z.; Jiang, X.; Zheng, F.; Guo, X.; Huang, F.; Sun, X.; and Zheng, W., 2019. Viewpoint-aware loss with angular regularization for person re-identification. In *AAAI*. (cited on page 36)