Moving Towards Centers: Re-ranking with Attention and Memory for Re-identification

Zhou Yunhao

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

18 May 2021
.....
Date

Zhou Yunhao
.....
Zhou Yunhao

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

18 May 2021

.....

Date

.....

Lap-pui Chau

Authorship Attribution Statement

This thesis contains material from 1 paper under review in the following journal in which I am listed as an author.

Chapter 4 and 5 are submitted and under review at IEEE Transactions on Multimedia as Yunhao Zhou, Yi Wang, and Lap-Pui Chau, "Moving Towards Centers: Re-ranking with Attention and Memory for Re-identification," 2021.

The contributions of the co-authors are as follows:

- I proposed the method, designed the experiments and prepared the manuscript drafts.
- Dr Yi Wang, Prof Lap-Pui Chau and I discussed and revised the manuscripts.

18 May 2021

Date

Zhou Yunhao

Zhou Yunhao

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to my supervisor, Professor Chau Lap-Pui, who guides me patiently throughout my M.Eng study at Nanyang Technological University (NTU). The methodology that he taught me to carry out research is a great treasure for my entire life. I am also grateful to the wonderful friends that I met at NTU. Especially, I would like to thank Dr. Wang Yi for his support and cooperation. It is the company from these lovely friends that makes my colorful experience in Singapore.

Finally, I am extremely grateful to my family for the loves, cares, and sacrifices for educating and preparing me for my future. I am also very thankful to Ms. Duan Xu for the constant love, encouragement, and understanding.

"Don't judge each day by the harvest you reap but by the seeds that you plant."

—Robert Louis Stevenson

To my dear family

Abstract

Re-Identification (Re-ID) is a fundamental computer vision task, which refers to associating targets, such as humans or vehicles, captured from multiple non-overlapping cameras. After obtaining the initial re-ID result, re-ranking boosts the retrieval performance with contextual information in top-ranked samples. Current re-ranking approaches focus on hand-crafted rules, which generalize well on small re-ID benchmarks. However, they cannot handle complex relationships between the probe image and the retrieved samples. This inherent deficiency leads to unsatisfying results when dealing with massive data, which is unavoidable for real-world scenarios.

To eliminate the reliance on polishing hand-designed algorithms, this work proposed a deep learning-based re-ranking network to predict the correlations between images and their local neighbors. Specifically, all the feature embeddings of query and gallery images are expanded and enhanced by a linear combination of their neighbors, with the correlation prediction serves as discriminative combination weights. The combination process is equivalent to moving independent embeddings toward the identity centers, improving cluster compactness. For correlation prediction, we first aggregate the contextual information for probe's k-nearest neighbors via the Transformer encoder. Then, we distill and refine the probe-related features into the Contextual Memory cell via attention mechanism. Like humans that retrieve images by not only considering probe images but also memorizing the retrieved ones, the Contextual Memory produces multi-view descriptions for each instance. Finally, the neighbors are reconstructed with features fetched from the Contextual Memory, and a binary classifier predicts their correlations with the probe. Experiments on six widely-used person and vehicle re-ID benchmarks demonstrate the effectiveness of the proposed method. Especially, our method surpasses the state-of-the-art re-ranking approaches on large-scale datasets by a significant margin, i.e., with an average 3.08% CMC@1 and 7.46% mAP improvements on VERI-Wild, MSMT17, and VehicleID datasets.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\|\cdot\|_2$	the L2-norm of a vector
$(\cdot)^T$	Matrix or vector transpose
$Sigmoid(\cdot)$	Element-wise sigmoid activation
$\odot$	the Hadamard (eliminate-wise) product

Acronyms

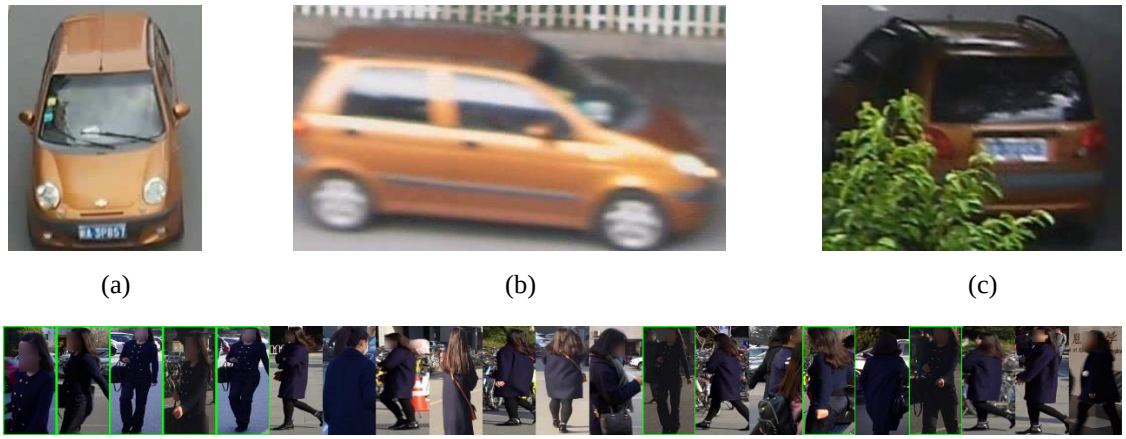
αQE	Alpha Query Expansion
ACP	Attention-based Correlation Predictor
AQE	Average Query Expansion
BN	Batch Normalization
CCTV	Closed Circuit Television
CMC	Cumulated Matching Characteristics
CNN	Convolution Neural Network
DBA	Database Side Augmentation
GNN	Graph Neural Network
LN	Layer Normalization
mAP	mean Average Precision
MHA	Multi-Head Attention
MLP	Multi-Layer Perceptron
MTL	Multi-Task Learning
QE	Query Expansion
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Background of Re-ID and Re-ranking

Recently, re-identification tasks have drawn increasing interest in the computer vision community. Given an image from the query set, re-ID aims at automatically finding all images containing the same instance as the query across a large image pool, namely, the gallery set. In general, query and gallery images are captured by different cameras in multiple scenes, such as Closed Circuit Television (CCTV). Thus, re-ID is often considered a sub-problem of image retrieval at the instance-specific level. Re-ID has a wide variety of applications according to the target instances. For example, person re-ID helps the criminal investigation [3] by looking for suspected persons with city surveillance cameras. Vehicle re-ID can analyze traffic [4], which is a crucial part of intelligent transportation systems in smart cities. With more and more cameras deployed in cities, re-ID plays a important role in automatically analyzing tons of recorded videos without requiring human intervention. The problem setting that retrieving instances from multiple non-overlapping cameras brings many challenges. For example, a vehicle captured from different cameras may encounter severe appearance variations. People dressed in the same color and style are hard to distinguish, even for humans experts. Besides, the illumination changes, occlusions, low-image resolutions, and cluttered backgrounds all harm the re-ID performance and hinder its real-world applications. Some challenging cases are shown in Fig. 1.1.



(a) (b) (c)

(d) Strong similar persons

FIGURE 1.1: Challenging cases. The first row of figures shows the appearance of a vehicle under different cameras. The second row of images is an initial retrieval result, where the true matches are marked with green borders. We can observe that falsely retrieved images have a similar appearance to true matches.

Therefore, the key to re-ID is building robust and discriminative feature embeddings to minimize intra-identity distance and maximize inter-identity discrepancy. As shown in Fig. 1.2, a re-ID baseline maps images to an embedding space. Early research mainly focused on designing hand-crafted image descriptors [5, 6]. As more and more training data becomes available, the ability of Convolution Neural Networks (CNN) to learn robust feature representations from data pushes the re-ID performance to a new level [7–11].

Basic re-ID model retrieves instances with pairwise distance measure, which only considers individual characters between two separate images at a time. Performance degrades quickly in challenging scenarios where images containing the same instance cannot be embedded into a small cluster and are not close enough to each other. To overcome the drawbacks of the pairwise matching rule, re-ranking leverages the contextual information in local neighbors to optimize the initial ranking list of re-ID, which conducts retrieval by integrating information from multiple images. Generally, re-ranking methods can be categorized into two groups: feature similarity and neighborhood similarity. Feature similarity refers to the similarity of images in the embedding space. For example, [12–14] aggregates the contextual information via directly averaging embeddings of k-nearest neighbors. It is equivalent to substitute the pairwise distance with the distance between centers of different groups of images. Neighborhood similarity resorts to more complex rules like common nearest neighbors [15] and k-reciprocal neighbors [16]. For instance,

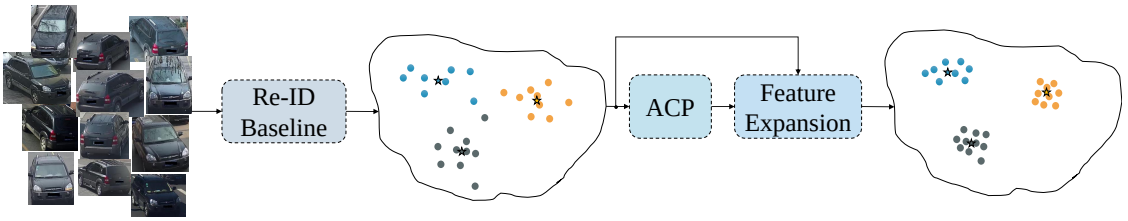


FIGURE 1.2: General pipeline of re-ID and our Attention-based Correlation Predictor (ACP). Embeddings and identities are shown with dots and colors. Images containing the same instances tend to form clusters in the embedding space. Our re-ranking method aggregates each embedding and the corresponding k-nearest neighbors with predicted correlations, which moves independent embeddings towards the identity centers marked with stars.

the common nearest neighbors use Jaccard index to measure the intersection of k-nearest neighbor set for different images. In general, the neighborhood similarity produces better results than the feature similarity-based methods because more images are considered at the semantic level. Re-ranking has been integrated into many re-ID methods [8, 17, 18] because of the performance boost, especially on mean Average Precision (mAP).

1.2 Motivations

Most existing re-ranking approaches rely on hand-designed algorithms for the utilization of contextual information, which lacks the ability to handle complex neighborhood relationships. Consequently, the results on large-scale datasets are far from satisfying. Take the query expansion as an example, k-nearest neighbors are combined to augment independent feature embeddings. However, Average Query Expansion [12], Alpha Query Expansion [13], and the ad-hoc variations [19, 20] adopt monotocially decreasing combination weights. This implies that weights assigned to bottom-ranked correct matches will not be larger than the top-ranked false matches. The contribution of non-relevant samples might dominate the expansion if we enlarge the neighborhood to include more contextual information. Although Discriminate Query Expansion [14] can allocate larger weights to bottom-ranked correct samples theoretically, its ability is limited because of the linear Support Vector Machine and training samples. The trained hyper-plane fails to separate the top-k samples for real-world scenarios where the probability of having strong similar-looking targets is much higher than the collected datasets.

Moreover, neighborhood similarity-based re-ranking methods are usually resource-hungry, making this line of work unappealing despite its performance improvement. For example, the k-reciprocal re-ranking [16] calculates reciprocal neighbors, which consists of extensive set comparisons, for each probe image. The requirement for system memory surges when the number of images grows larger. Besides, to the best of our knowledge, no straightforward GPU parallelization is implemented to speed up the whole process. Experiments on the largest test set of VERI-Wild show that it takes more than 900s and 150GB RAM to finish re-ranking.

1.3 Major Contributions

Given an initial ranking list from a re-ID baseline, humans achieve re-ranking by gradually verifying the retrieved images and memorizing the appearance information of the verified confident samples. Our brain can summarize the scattered contextual information to form an overall description of the target. Based on this observation, we propose a Contextual Memory cell to fulfill the information aggregation purpose powered by the attention mechanism. We later reconstruct top-ranked samples by comparing their features with the refined Contextual Memory. Because falsely retrieved samples share less common features with the memory, their reconstruction results will be different from those true matches, which can be used for correlation prediction, i.e., whether a retrieved sample is a true match for the probe image. With the predicted correlations, we can shrink the independent feature embeddings to the identity centers through a weighted combination of their local neighbors in the original embedding space. An overview of our re-ranking model is visualized in Fig. 1.2 where the improved cluster compactness can reduce false matches.

We reformulate re-ranking as a binary classification problem of the top-k retrieved samples and propose an Attention-based Correlation Predictor to predict the correlations. Specifically, the correlation prediction consists of three key steps. In the first step, each probe's k-nearest neighbors are fed into a Transformer encoder to aggregate the contextual information. Second, a Contextual Memory initialized by attention mechanism distills and stores the probe-related contextual features. We further refine the memory with a small group of highly confident samples for the purpose of eliminating feature pollutions brought by interfering false matches.

Third, we reconstruct neighbor embeddings with features fetched from the memory cell through Multi-Head Attention. A binary classifier with sigmoid activation predicts the correlation between each reconstructed neighbor and the probe.

The proposed method is evaluated on six widely used benchmarks including VERI-Wild [21], MSMT17 [22], VehicleID [23], VeRi [24], Market1501 [2] and DukeMTMC [25]. Experimental results show that our method surpasses state-of-the-art re-ranking methods by a large margin on large-scale benchmarks VERI-Wild, MSMT17, and VehicleID. For smaller benchmarks with limited training images, we ranked high amongst the competing methods. Ablation study verifies the effectiveness of our Contextual Memory and refinement process. In summary, the contribution of this paper is fourfold:

- This work proposes to use the Contextual Memory cell to mimic the remembering process that humans adopt for re-ID and re-ranking. Embeddings obtained from single-view images are expanded with attention mechanism that learns to focus on the most discriminative regions.
- We reformulate re-ranking as a binary classification problem, which explains the feature expansion from a new perspective. The independent embeddings are moved to the identity centers by combining local neighbors with correlation prediction produced by the classifier.
- The proposed method only requires finding the first-order nearest neighbors for each image. Besides, the whole model can be easily implemented with existing deep-learning frameworks with GPU acceleration available.
- Extensive experiments on six re-ID datasets demonstrate the superiority of the proposed method. Detailed model studies reflect how each module and parameter affect the performance.

Chapter 2

Literature Review

2.1 Deep feature representation of re-ID

Traditional re-ID methods use hand-crafted feature descriptors. Recently, CNN-based feature representation learning has dominated the solutions and achieved great success on re-ID tasks. In this section, we will briefly introduce two groups of re-ID approaches with deep learning techniques.

2.1.1 Feature Representation Leaerning

Following image classification [1, 26–28], the first line of work directly learns a global feature embedding from the whole image. Zheng et al. [29] proposed an ID-discriminative Embedding (IDE) model, which considers re-ID a classification problem with identities treated as unique classes. PersonNet [30] uses small-sized convolutional filters in person re-ID to capture the detailed information. Sun et al. [31] proposed Singular Vector Decomposition Net (SVDNet), which uses Eigenlayer as the second last FC layer, to reduce the correlation among projection vectors and produce more discriminative feature embeddings. Qian et al. [32] present a multi-scale deep representation learning scheme, which adjusts the feature scale automatically for person matching. Song et al. [33] developed a Mask-Guided Contrastive Attention Model (MGCAM). MGCAM extract feature representations from body and background regions, separately. The binary segmentation masks eliminate the negative impact from cluttered backgrounds. To enhance representations, [34] fuses

features from multiple convolutional network layers. Low-level cues from shallower blocks are taken into consideration to augment the high-level semantic features.

However, a global feature often fails to distinguish two similar-looking instances like vehicles in the same model or persons dressed in the same color and style. To solve this problem, some methods extract supplementary features like strong discriminative regions [10], viewpoints [11, 35] or attribute characteristics [36] to provide auxiliary information for the global feature. For example, the Part-based Convolutional Baseline (PCB) [17] uniformly partitions the learned feature map into multiple horizontal stripes. Strip features provide part-level information from different body regions. [35] proposed a Pose-Sensitive Embedding for person re-ID with an additional view predictor. Zhou et al. [11] combined CNN with Long Short-Term Memory (LSTM) to learn the transformations across different viewpoints of vehicles. Multi-view vehicle representations can be inferred from a single view image input. Semantics-guided Part Attention Network (SPAN) [37] predicts part attention masks for different views of vehicles to extract discriminative partial features.

2.1.2 Deep Metric Learning

Instead of focusing on model architectures, another line of work adopts metric learning methods to improve the feature discriminability. Generally, deep metric learning focuses on designing loss functions to guide the feature learning process.

2.1.2.1 Identity Loss

Zheng et al. [29] treat the network training as an image classification problem. During testing, the classifier input feature embeddings are used for distance calculation, where the backbone network is adopted as a feature extractor. Denote an image as x_i with identity label as y_i, the network will predict the probability of x_i being labled as y_i, i.e., $p(y_i|x_i)$. Then, we can define the identity loss as,

$$\mathcal{L}_i = -\sum_{i}^{C} y_i log(p(y_i|x_i)) \tag{2.1}$$

where C represents the number of unique identities in the training set. In practice, identity loss is a popular choice because it is easy to optimize and converges fast. To enhace the intra-class compactness and inter-class discrepancy, many softmax cross-entropy loss variants have been proposed like ArcFace [38] and CosFace [39]. These variants are proven to be effective for re-ID tasks in [18].

2.1.2.2 Contrastive Loss

Contrastive Loss takes two images at a time, which measures the pairwise relationships. Varior et al. [40] proposed a Siamese Long Short-Term Memory Architecture for person re-ID supervised by contrastive loss:

$$\mathcal{L}_c = (1-\delta)\{max(0, \rho - d)\}^2 + \delta \cdot d^2 \tag{2.2}$$

where d refers to the Euclidean distance between the feature embeddings of two images, δ is the corresponding binary label, i.e., $\delta = 1$ if two input images belong to the same identity and vice versa. The ρ is a pre-defined margin. One problem of contrastive loss is that it suffers from slow convergence caused by the selection of training samples.

2.1.2.3 Triplet Loss

Triplet Loss measures the relationships between the triplets, i.e., an anchor image x_a, a positive image x_p, and a negative image x_n. It pulls the anchor and positive images together and pushes negative images away,

$$\mathcal{L}_t = max(\rho + d_{ap} - d_{an},\ 0) \tag{2.3}$$

where d represents the distance between two embeddings, and ρ is a pre-defined margin parameter. Based on this basic form, many variants of triplet loss are proposed to improve its performance and convergence speed. Hermans et al. [41] proposed an online batch hard sampling strategy, which is known as the PK sampler. Shi et al. [42] presented a Moderate Positive Mining Method to minimize the intra-identity variation while preserving the internal graphical structure. It has become one of the most common choices [8, 10, 37, 43–46] to train deep re-ID

networks supervised by the combination of triplet loss and cross-entropy identity loss.

2.2 Re-ranking for re-ID and Image Retrieval

Re-ranking plays a crucial role in improving retrieval performance at post-processing steps. Given an initial ranking list, re-ranking refines the result utilizing contextual information in top-ranked samples. Although feature representation learning ushers in the blossom of deep neural networks, most existing re-ranking methods still stagnate in hand-designed rules when analyzing the relations between neighboring embeddings. In this section, we first introduce two groups of traditional re-ranking approaches in Section 2.2.1 and Section 2.2.2. Then, some recent advancement in re-ranking with neural network is discussed in Section 2.2.3

2.2.1 Feature Similarity

Feature similarity-based methods utilize the k-nearest neighbors of each query image and the corresponding pairwise distance in the embedding space. Chum et al. [12] proposed Average Query Expansion (AQE) for image retrieval tasks, which replaces query feature embeddings with the mean of top-k retrieved gallery samples. However, its sensitivity to parameter selection makes it a non-trivial task to decide suitable parameters. Instead of taking the mean average, Radenović et al. [13] resort to weighted average named Alpha Query Expansion (αQE). The weights are formulated as taking the power of similarities between the query and top-k retrieved samples with the exponent as a hyper-parameter α. Because of its simplicity and robustness, αQE has been adopted as a common approach of boosting retrieval performance by a number of image retrieval and re-ID works. To improve the discriminability, Arandjelović and Zisserman [14] proposed Discriminate Query Expansion (DQE) which trains a linear Support Vector Machine (SVM) with top-ranked and bottom-ranked samples as positive and negative samples, respectively. The distance between samples and the decision boundary work as pseudo labels to aggregate the k-nearest neighbors.

2.2.2 Neighborhood Similarity

Direct utilization of top-ranked samples suffers from tackling noisy false matches. The neighborhood similarity-based method compares the neighbor sets of different images to restrain the negative effect of falsely retrieved samples. Bai and Bai [47] proposed Sparse Contextual Activation (SCA) that encodes the local distribution of an image in contextual space. Qin et al. [48] first proposed k-reciprocal nearest neighbors to eliminate outliers. Two images are called k-reciprocal nearest neighbors if they both ranked top-k when the other image serves as a probe. Zhong et al. [16] expand k-reciprocal nearest neighbor set to include more contextual information. The Jaccard distance of the expanded k-reciprocal sets between each query and gallery forms a new distance measure which is aggregated with the original distance via convex combination. Expanded Cross Neighbor (ECN) is introduced by [35] which sums the distance of images in expanded neighbors. Yu et al. [49] divide the extracted features into multiple sub-features, then the contextual information is iteratively encoded and fused into new feature vectors. Ye et al. [3] proposed to consider not only the similarity of top-k samples but also the dissimilarity of bottom-k samples from different baseline methods. Recently, Wang et al. [50] proposed reciprocal optimization that takes multiple queries into account. The verified historical queries will be accumulated for re-ranking. Unlike automatic re-ranking approaches, several other algorithms require human interaction [51, 52] to incorporate reliable supervision during ranking optimization.

2.2.3 Neural Re-ranking

The continuous progress of re-ranking pushes the performance forward by designing more and more sophisticated algorithms for exploiting the contextual information hiding in k-nearest neighbors. Hand-designed rules generalize well on small benchmarks but show difficulties in fitting large amounts of data. Recently, some works tried neural networks for re-ranking. Zhang et al. [53] re-think re-ranking from the perspective of graph neural network, which effectively accelerates the re-ranking with GPU. For each image, the k-nearest neighbors are encoded with an adjacent matrix A. To obtain the symmetric property, the transpose of A is added to itself, i.e., the new symmetric adjacent matrix A^* encodes the k-reciprocal nearest neighbors. Thus, the neighboring relationship for each image is represented with

a row vector from A^*, which is regarded as a new feature embedding. However, the core idea of [53] remains hand-crafted, without the process of learning to re-rank from data. Liu et al. [54] proposed to use graph convolutional network for link prediction and replace the original Euclidean distance with the predicted link probability. Instead, we consider our correlation prediction as combination weights to shrink independent embeddings toward their identity centers. Therefore, we are not suffered from predicting the entire pairwise correlations between queries and galleries. Besides, our method only requires finding the first-order neighbors for each image. The whole architecture can be easily implemented under existing deep learning frameworks with GPU acceleration available painlessly.

2.3 Attention and Transformer

Transformers were introduced by [55] and have achieved huge success on a wide range of natural language processing tasks [56, 57]. The core idea of Transformer is to update the sequence with information aggregated from the entire input sequence via attention mechanism. It can capture the complex relationships between different input tokens. Recently, Transformer starts to shine in the computer vision community. Here, we briefly introduce the implementation of Transformer on some classic vision tasks.

2.3.1 Image Recognition

Wang et al. [58] proposed Non-local Neural Networks that use self-attention to aggregate image feature representations from non-local regions. Given a feature map, the non-local operator calculates the response at a position with a weighted average of features from the entire input feature map. To reduce the memory consumption, Huang et al. [59] proposed criss-cross attention that restricts the attention only on the cross paths. Dosovitskiy et al. [60] present Vision Transformer (ViT) can achieve state-of-the-art performance on image recognition tasks. ViT relies on self-attention with flattened image patches as inputs discarding the convolutional architecture entirely.

2.3.2 Object Detection

Carion et al. [61] proposed Detection Transformer (DETR) with self-attention layers built on top of convolutional backbones. DETR predicts objects in a single forward pass, and it removes the reliance on the region proposal network and non-maximum suppression. Zhu et al. [62] proposed deformable DETR to resolve the slow convergence issue by using attention on a small set of key sampling points around a reference.

2.3.3 Image Generation

Parmar et al. [63] leverage the self-attention of Transformer for auto-regressive image generation. To reduce the compuation cost, Esser et al. [64] proposed to include inductive bias that prioritizes local interactions. This model efficiently learns a rich composition of visual patterns within high-resolution images.

2.3.4 Instance Re-Identification

For re-ID, TransReID proposed by [65] shows that the self-attention between image patches can produce more robust features than CNN because information loss on details is avoided by removing convolution and downsampling operators. In this work, the self-attention of Transformer is used to explore the contextual information in local neighborhoods.

2.4 Datasets

We evaluate the performance on six widely-used re-ID datasets including three person re-ID datasets, i.e., Marker1501[2], DukeMTMC-ReID[25] and MSMT17[22] as well as three vehicle re-ID datasets, i.e., VERI-Wild [21], VehicleID [23] and VeRi-776 [24]. An overall statistical comparison on identities, images, and cameras of the datasets is in Table 2.1.

TABLE 2.1: Re-ID datasets statistics. 'IDs' denotes the number of different identities. 'Imgs' represents the number of captured images. 'Cams' refers to the number of unique cameras. 'S', 'M' and 'L' indicate three different test set partitions namely small, medium and large for VERI-Wild and VehicleID.

		ReID Datasets									
Splits		VeRi-776	VERI-Wild			VehicleID			Market1501	DukeMTMC	MSMT17
Train	IDs	576	30671			13164			751	702	1041
	Imgs	37746	277797			113346			12936	16522	30248
	Cams	20	173			/			6	8	15
Query	IDs	200	S	M	L	S	M	L	750	702	3060
			3000	5000	10000	800	1600	2400			
	Imgs	1678	3000	5000	10000	800	1600	2400	3368	2228	11659
	Cams	19	105	113	126	/			6	8	15
Gallery	IDs	200	3000	5000	10000	800	1600	2400	751	1110	3060
	Imgs	11579	38861	64389	128517	5693	11777	17377	15913	17661	82161
	Cams	19	146	153	161	/			6	8	15

2.4.1 VeRi-776

VeRi-776 [24] contains 51,003 images of 776 vehicles from 20 traffic surveillance cameras. It also provides attribute annotations like vehicle types, colors and camera geo-locations as well as image timestamps. The training test is constituted of 37,781 images with 576 unique identities. The remaining 13,257 images of 200 vehicles are adopted as the testing set, including 1,678 query images and 11,579 galleries.

2.4.2 VehicleID

VehicleID [23] is a large-scale dataset which contains 26,267 vehicles and 221,763 images in total. It is collected from real-world surveillance cameras. Although timestamps and locations are provided for images inside VehicleID, specific camera identities for each image are not available. The test protocol of VehicleID randomly selects one image from one identity to form a gallery set. The remaining images are all utilized as queries. There are three subsets for testing in VehicleID based on the size of gallery sets: a small testing set with 800 vehicles, a medium testing set with 1600 vehicles and a large testing set with 2400 vehicles.

2.4.3 VERI-Wild

VERI-Wild [21] is another large-scale vehicle re-ID dataset. It is collected from a large CCTV system with 174 surveillance cameras. Besides, the images in VERI-Wild are captured under real-world unconstrained conditions over one month. Bounding boxes are detected by YOLO-v2 [66] and cleaned by human experts afterward. 416,314 vehicle images with 40,671 identities are collected in total after annotation. Due to the unconstrained conditions and the large number of cameras, VERI-Wild is the vehicle re-ID benchmark that most challenging and most close to real-world scenarios currently.

2.4.4 Market1501

Market1501 [2] is a person re-ID benchmark collected by six cameras. It contains 1501 identities with 32,668 pedestrian image bounding boxes in total. Bounding boxes are labled using DPM detecor [67]. Among the 32,668 images, 12,936 images with 751 identities are selected for training and the remaining 19,281 images are for testing For the testing set, 3368 images are used as queries and 15,913 as galleries. Market1501 provides multi-query labels to enable leveraging contextual information inside the query set. Here, all the methods are tested under single-query mode without using the provided query labels.

2.4.5 DukeMTMC-ReID

DukeMTMC-ReID [25] is a subset of the DukeMTMC for person re-ID which is captured under 8 different cameras. There are 36,411 images of 1,812 identities in total. Images inside DukeMTMC-ReID are labeled by hand-drawn bounding boxes. Among all the identifies, 1,404 identities are captured in more than two cameras and 408 identities only show up in one camera. The training set is composed of 16,522 images of 702 identities. The remaining 19,889 images with 702 identities are reserved for testing, with 2,228 as queries and 17,661 as galleries.

2.4.6 MSMT17

MSMT17 [22] is a large-scale Multi-Scene Multi-Time person re-ID dataset with 4,101 identities and 126,441 images in total. Raw videos in MSMT17 are taken by a 15-camera system on campus. It is composed of 12 outdoor cameras and 3 indoor cameras. The training set contains 32,621 bounding boxes of 1,041 identities, and the testing set includes 93,820 bounding boxes of 3,060 identities. The testing set uses 11,659 randomly selected bounding boxes as query images and the remaining 82,161 bounding boxes are galleries. Properties like a large number of identities, complex scenes and backgrounds, and severe lighting changes due to multiple time slots make MSMT17 a challenging benchmark.

Chapter 3

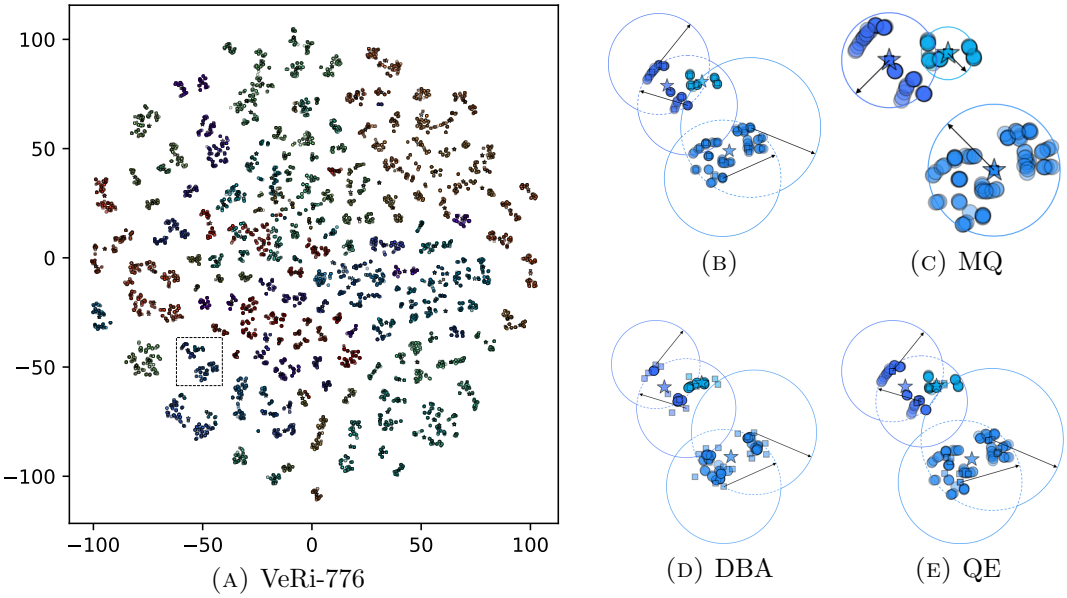
Feature Expansion

As we introduced in Section 2.2.1, feature similarity-based methods use the pairwise distance between queries and k-nearest neighbors for re-ranking. Usually, the original feature embeddings are expanded by combining k-nearest neighbors with some weights, i.e., feature expansion. Then, re-query with the expanded features boosts the retrieval performance because this simple aggregation manner can augment the pairwise matching rule by simultaneously considering appearance information from a group of similar images. Indeed, the idea of feature expansion can be used in other ways. For example, Market1501 [2] provides multi-query labels which can be used to aggregate feature representations from multiple query images captured under different cameras through average or max pooling. Multi-query is effective and efficient because of the reliable supervision from query labels. In image retrieval, Database Side Augmentation [14] (DBA) update every image in the database (or gallery set) by a combination of itself and its neighbors, which can be considered as an offline version of the query expansion method.

Generally, the performance improvement of expansion-based methods is explained as they can fuse multi-view appearance information, i.e., the contextual information. Instead of following this routine, we visualize and explain these feature expansion algorithms from a geometric view. Specifically, we reduce dimensions to two with t-SNE and plot distributions of the 2-dimensional embeddings of VeRi-776 [24] in Fig. 3.1a. In the scatter plot, queries and galleries are plotted with squares and circles, respectively. We mark their geometric centers with crosses and stars.

The identities are shown with different colors. The dimension reduction method t-SNE maps high dimensional embedding into a low dimensional space by preserving pairwise distance information. Therefore, it is appropriate for visualizing the embeddings of re-ID because it retrieves instances based on distance measure. From the scatter plot, we have the following observations:

- Feature embeddings of the same identity tend to form clusters. Different clusters keep a distance from each other. This pattern corresponds with the supervision from triplet loss.
- The query embeddings distribute evenly across the whole clusters and overlap with galleries.
- Clusters of some identities gather together, making them hard to distinguish.



(A) VeRi-776 (B) (C) MQ (D) DBA (E) QE

FIGURE 3.1: Visualization of feature expansion methods. (A) Feature embeddings of VeRi-776 after applying t-SNE where 2048-dimensional embeddings are mapped to 2-dimensional vectors. (B) Embedding distribution from a small area marked with the dashed line in (A). Circles reveal the difficulties of retrieving the hardest samples with queries as the circle centers. (C) - (E) Feature embeddings after expansion, where (C) refers to Multiple Queries, (D) represents DBA, and (E) shows Query Expansion.

3.1 Multiple Queries

The Multiple Queries (MQ) takes the intra-identity variation of queries into account. Each query embedding is replaced by the average of embeddings extracted from query images captured under multiple view angles. The averaging operation moves independent queries to their geometric centers of all the queries in the same identity. We can observe that the query centers (starts) are close to the center of galleries (crosses) in Fig. 3.1a because of the uniform distribution of the queries. Re-identification using the updated query embeddings produces better results since queries located at cluster borders move away from noisy gallery samples in different identities after expansion.

Multiple Queries requires the supervision of identity labels for expansion. However, the identity information is generally not available in real-world scenarios. Besides, multi-view feature expansion is restricted to queries while the gallery embeddings are kept unchanged. Denote the distance between a query and its farthest correct match as r. We can draw a circle with the radius as r to include all the correct matches inside. The area under this circle reflects the difficulty of retrieving the hardest sample. Ideally, we hope r to be small and no galleries from other identities are involved inside circles. Comparing Fig. 3.1b and Fig. 3.1c, Multiple Queries significantly reduces the union area under independent circles by moving queries to their geometric centers. We can find that some galleries are still far away from the expanded queries. They benefit nothing from the rich contextual information around them and keep at the original locations. The circle can be further shrunken by moving galleries towards their cluster centers.

3.2 Database Side Augmentation

Similar to multiple queries, Database Side Augmentation (DBA) aggregates cross-camera contextual information via feature expansion. DBA replaces each embedding with the average of itself and its k-nearest neighbors in the gallery set without accessing identity labels. The assumption behind DBA is that most of the nearest neighbors for a gallery embedding are in the same identity as this gallery.

We visualize the distribution of embeddings after applying DBA in Fig. 3.1d. It clearly shows the tendency for galleries to gather together after averaging their neighbors. However, the area under circles receives few contractions because of the following facts. First, query embeddings are kept unchanged. DBA only considers the gallery-to-gallery similarities with the query set excluded. It is more appropriate for large-scale offline feature expansion where the query set is not available. Second, DBA lacks the ability to fully take advantage of the contextual information due to its indiscriminate combination rule, i.e., not able to distinguish correct matches from falsely retrieved samples. In practice, k is set small to control the number of false matches in local neighbors.

3.3 Query Expansion

Query expansion (QE) makes use of top-k retrieved gallery samples. It exploits the query-to-gallery similarities to update query embeddings. Specifically, query expansion replaces query embeddings by averaging the query itself and its k-nearest neighbors in the gallery set. We can observe that independent queries aggregate together after query expansion in Fig. 3.1e.

There are two major problems with the query expansion method. First, similar to DBA, query expansion cannot reduce the circle size at a large scale because it only updates queries but leaves the galleries at their original positions. The query-to-query and gallery-to-gallery similarities are ignored. Second, the combination weights of most existing query expansion methods are either uniform or decreasing along with the distance monotonically. In other words, contextual information in bottom-ranked correct matches is combined with smaller weights compared to the top-ranked false matches. Therefore, k is usually small to restrain the negative effect from noisy false matches. Once k becomes large, the performance of query expansion drops quickly because averaging embeddings in other identities pulls queries away from their cluster centers.

3.4 Relationship

The methods mentioned above are designed to overcome the drawbacks of image-to-image retrieval rules with additional cues from local neighbors. The expansion process can be regarded as moving the original embedding towards cluster centers to improve cluster compactness and reduce false matches. Based on the previous visualizations and analyses, we can find that current feature expansion approaches suffer from the indiscriminative combination weights, along with the unsynchronized movement between queries and galleries.

To fix these two problems, we propose our deep learning-based feature expansion method. First, queries and galleries are combined into one image set and considered as independent embeddings surrounded by neighbors sharing common features. We replace each embedding with a weighted sum of itself and its k-nearest neighbors, which follows the basic feature expansion concept. Since the expansion is performed not only for queries but also for galleries, the movement of embeddings between two sets is synchronized. In other words, all the embeddings are moving towards their identity centers at the same time. Second, the combination weights are predicted from our ACP that is trained on existing data. A well-trained ACP can generate non-uniform discriminative weights for k-nearest neighbors by exploiting the similarities between sub-features of the original embeddings. In the next chapter, we will elaborate on how discriminate combination weights (correlations) are predicted.

Chapter 4

Attention-based Correlation Predictor

In this chapter, we first formulate the re-ID baseline and re-ranking with feature expansion in Section 4.1. Next, the detailed model architecture is described in Section 4.2. We visualize the pipeline of our re-ranking approach, named Attention-based Correlation Predictor (ACP), in Fig. 4.1.

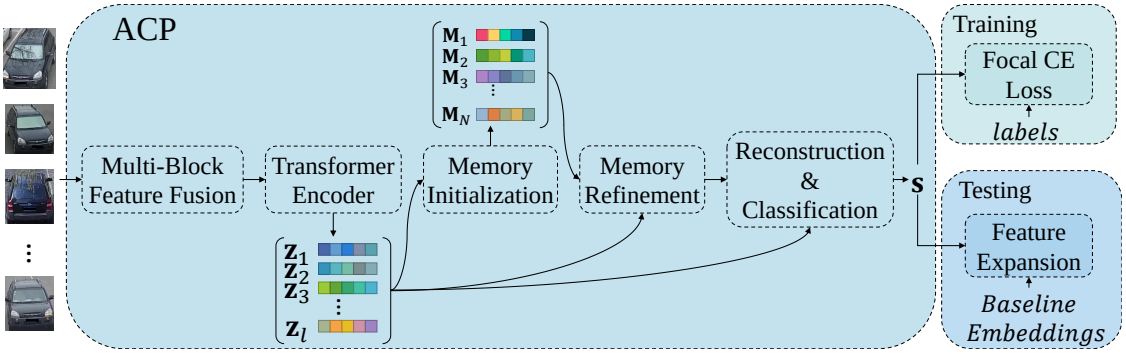


FIGURE 4.1: Architecture of our Attention-based Correlation Predictor.

4.1 Overview of re-ID and re-ranking

Given an image from the query set, re-ID aims at retrieving images containing the same instance in the gallery set. Re-ID first maps images to D-dimensional vectors through a feature extractor ϕ. We define the feature representation of images as $\mathbf{f} = \phi(\mathcal{I})$. The output feature vectors form two matrices: $\mathcal{Q} \in \mathcal{R}^{M \times D}$

and $\mathcal{G} \in \mathcal{R}^{N \times D}$ where M and N refer to the number of images in query and gallery sets, respectively. Then, pairwise distances between the feature representations in $\mathcal{Q}$ and $\mathcal{G}$ are calculated. The retrieval process of re-ID is achieved by sorting the distances in ascending order. It is expected that images holding the same instances will group together in the embedding space, so the correct matches in the gallery set will be ranked closer to the query.

Our re-ranking network exploits the contextual information in top-ranked samples to optimize the initial ranking list produced by the re-ID baseline. Given an image, we expand its embedding $\mathbf{p}$ with the linear combination of its k_1-nearest neighbors (including itself). The weights are predicted by our ACP as shown in Fig. 4.1. The expansion moves independent embeddings toward their identity centers, which boosts the performance because of the improved cluster compactness. Denote the embeddings of k_1-nearest neighbors of $\mathbf{p}$ as $\mathbf{N_f} = \{\mathbf{f}_i\}_{i=1}^{k_1}$, and the corresponding original images as $\mathbf{N_I} = \{\mathbf{I}_i\}_{i=1}^{k_1}$. The feature expansion of $\mathbf{p}$ is,

$$\mathbf{p}^{\star} = \sum_{j=1}^{k_1} \mathbf{N}_{\mathbf{f}}^{j} \odot \mathbf{s}_j \tag{4.1}$$

where $\mathbf{s} = \text{ACP}(\mathbf{N_I})$, $\mathbf{s} \in \mathcal{R}^{k_1}$ and $\odot$ is element-wise product. Note that we perform feature expansion for both the query and gallery sets. Finally, the pairwise distances between updated embeddings in two sets are re-calculated and sorted to accomplish the instance re-ID goal.

4.2 Model Architecture for Re-ranking

Given an image, we sort its k_1-nearest neighbors into a sequence. The attention is a powerful tool in aggregating contextual information over the sequence, while the Contextual Memory can distill the probe-related features for correlation prediction, which are integrated into the ACP. In this section, we first describe a multi-block feature fusion module. Then, we elaborate on how Contextual Memory is initialized and how the correlations are predicted via attention and the memory modules.

4.2.1 Multi-Block Feature Fusion

We visualize the feature fusion module in Fig. 4.2. The feature extractor of the re-ID baseline is ResNet-50 [1] which consists of 5 blocks, i.e., C_1, C_2, C_3, C_4, and C_5. Feature maps from successive two blocks have a stride difference of 2. The re-ID baseline model obtains image embeddings by feeding C_5 through a global average pooling (GAP) layer to reduce the spatial dimension. The features from shallower layers are discarded. Instead of following this routine, we propose a multi-block feature fusion module to fuse the features from different blocks because they provide complementary information to each other.

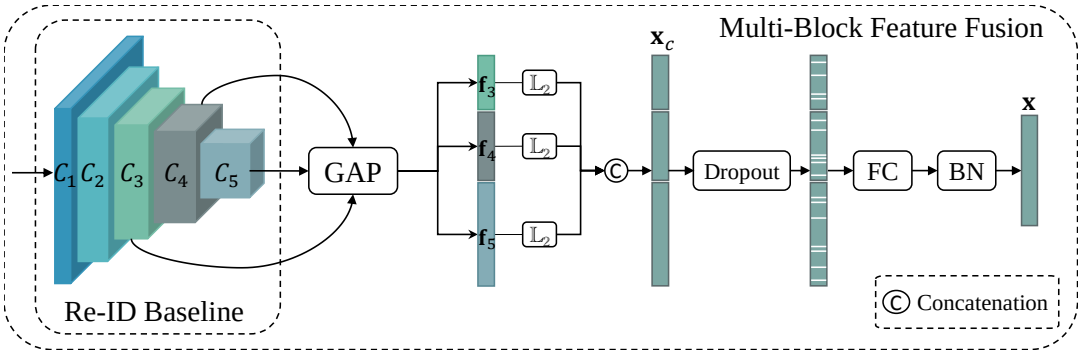


FIGURE 4.2: Multi-Block Feature Fusion. The colored cuboids represent features from different blocks of the backbone ResNet-50 [1]. GAP is a global average pooling layer. FC and BN are fully connected layer and Batch Normalization layer, respectively.

Suppose the multi-block features for an image $\mathcal{I}$ as $\phi(\mathcal{I}) = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_5\}$, where $\mathbf{f}_j$ is j-th feature vector generated by performing global average pooling on CNN feature maps. For ResNet-50, we fuse the last three pooled feature vectors, i.e., $\{\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5\}$. The feature vectors are first normalized by L_2 normalization and rescaled with a learnable scale parameter. The scaled L_2 normalization layer is formulated as,

$$\hat{\mathbf{f}}_j = \frac{\mathbf{f}_j}{\|\mathbf{f}_j\|_2} \odot \gamma \tag{4.2}$$

where $\|\cdot\|_2$ is the L_2 norm of a vector. The normalized feature vectors $\{\hat{\mathbf{f}}_3, \hat{\mathbf{f}}_4, \hat{\mathbf{f}}_5\}$ are concatenated into $\mathbf{x_c} \in \mathcal{R}^{d_c}$. To prevent over-fitting and improve generalization ability, a Dropout layer with dropout rate as p_d is placed after the normalization. Feature fusion is done with a fully connected layer followed by Batch Normalization (BN).

$$\mathbf{x} = \text{BN}(\text{Dropout}(\mathbf{x_c})\mathbf{W} + b),\ \mathbf{x} \in \mathcal{R}^d \tag{4.3}$$

where $\mathbf{x}$ is the fused feature vector for image $\mathcal{I}$, and d is the embedding dimension of $\mathbf{x}$.

4.2.2 Context Aggregation via Transformer

Although the multi-block feature fusion fuses information from different layers, the feature vector $\mathbf{x}$ only captures the appearance of an independent image. To aggregate contextual cues in multiple images, we feed the sequence (sorted k_1 nearest neighbors) into a Transformer encoder composed of Multi-Head Attention (MHA) and feed-forward networks (FFN). The Transformer encoder is visualized in Fig. 4.3.

In MHA, each sequence element is updated with a weighted average of all the other elements based on scaled dot-product similarity. Let's denote the output sequence $\{\mathbf{x}_i\}_{i=1}^{k_1}$ from the multi-block feature fusion module as $\mathbf{X} \in \mathcal{R}^{k_1 \times d}$. The scaled dot-product attention first maps embeddings to Queries ($\mathbf{Q} \in \mathcal{R}^{k_1 \times d_s}$), Keys ($\mathbf{K} \in \mathcal{R}^{k_1 \times d_s}$) and Values ($\mathbf{V} \in \mathcal{R}^{k_1 \times d_s}$) with three learnable projection matrices. After that, the similarity between Queries and Keys aggregates Values together,

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_s}})\mathbf{V} \tag{4.4}$$

The multi-head structure refers to concatenate the outputs from multiple scaled dot-product attention modules and fuse them with a learnable projection $\mathbf{W}^O$. It encapsulates complex relationships amongst different elements in the sequence by forcing each head to focus on some specific parts of the input embeddings. The output is added to $\mathbf{X}$ and finally normalized via Layer Normalization (LN),

$$\mathbf{Y} = \mathrm{LN}(\mathbf{X} + \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \tag{4.5}$$

Besides MHA, the Transformer encoder layer contains a position-wise feed-forward network for non-linearity. It consists of two linear transformations with a ReLU activation,

$$\mathrm{FFN}(\mathbf{Y}) = \mathrm{ReLU}(\mathbf{Y}\mathbf{W}_1 + \mathbf{b_1})\mathbf{W}_2 + \mathbf{b_2} \tag{4.6}$$

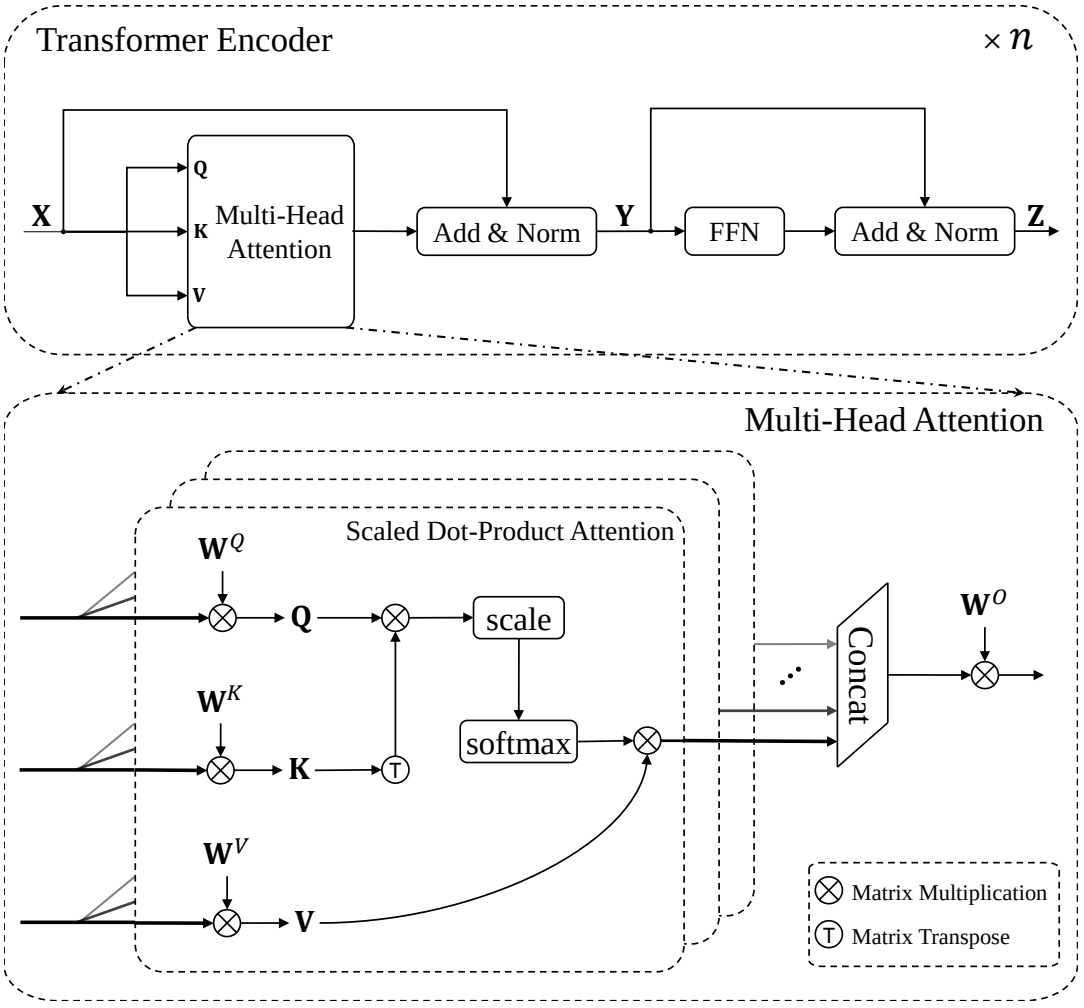


FIGURE 4.3: Architecture of Transformer encoder and Multi-Head Attention. The Transformer encoder layer consists of a Multi-Head Attention layer and a feed-forward network. Stacking multiple encoder layers builds a Transformer encoder. The Multi-Head Attention concatenates and fuses the outputs from multiple scaled dot-product attention sub-modules.

The final output of one complete Transformer encoder layer is

$$\mathbf{Z} = \mathrm{LN}(\mathbf{Y} + \mathrm{FFN}(\mathbf{Y})) \tag{4.7}$$

Following the original structure, we stack n Transformer encoder layers for the context aggregation purpose. The number of heads in MHA is controlled by a hyper-parameter h. We name this encoder as BaseEncoder.

4.2.3 Memory Initialization

The Contextual Memory collects different aspects of the probe image, like appearance in different angles, under different illuminations, etc. In other words, the memory cell is an augmented probe embedding with information from multiple images. It mimics the summarization ability of humans for re-ID and re-ranking. To achieve this, we aggregate the probe-related contextual features from local neighbors with attention, which is illustrated in Fig. 4.4.

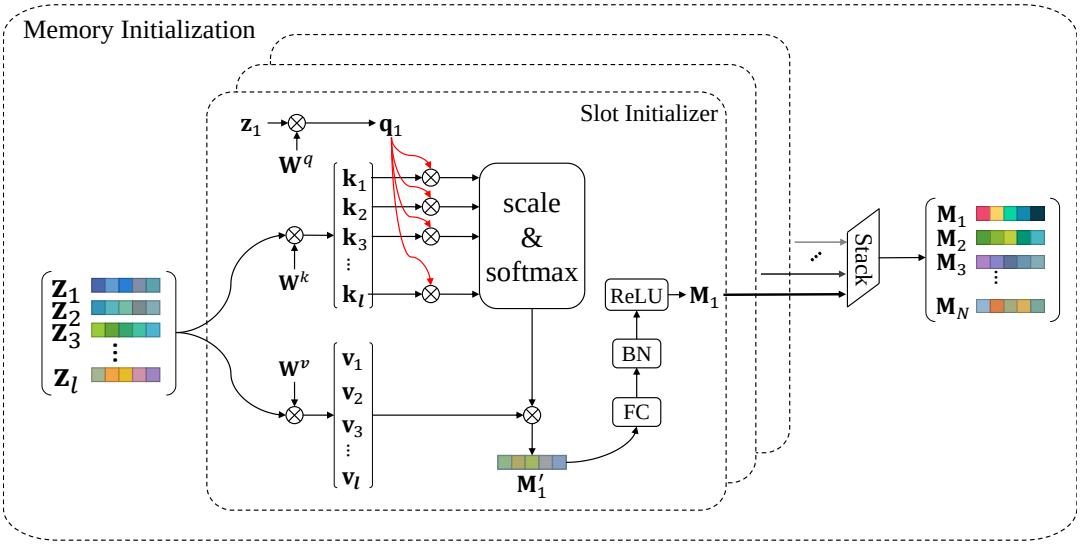


FIGURE 4.4: Memroy initialization. $\{\mathbf{z}_i\}_{i=1}^{k_1}$ denote the output embeddings from BaseEncoder. Multiple memory slots are stacked to form a complete memory cell.

The attention first maps $\mathbf{Z}$ to Queries, Keys, and Values. Instead of producing query vectors for all the elements, we only consider the probe embedding $\mathbf{z}_1$. This restriction guides our model to collect the probe-specific features from the local neighbors. Denote three transformation matrices as $\mathbf{W}^q \in \mathcal{R}^{d \times d_m}$, $\mathbf{W}^k \in \mathcal{R}^{d \times d_m}$ and $\mathbf{W}^v \in \mathcal{R}^{d \times d_m}$ where d_m is the sub-feature dimension and d_m is smaller than d. The transformation can be formulated as,

$$\begin{aligned} \mathbf{q} &= \mathbf{z}_1 \mathbf{W}^q \\ \mathbf{K} &= \mathbf{Z}\mathbf{W}^k \\ \mathbf{V} &= \mathbf{Z}\mathbf{W}^v \end{aligned} \tag{4.8}$$

We choose dot-product for similarity calculation and normalize them with softmax,

$$\begin{aligned}\mathbf{M}_i^{'} &= softmax(\frac{\mathbf{q}\mathbf{K}^T}{\mu})\mathbf{V} \\ &= softmax(\frac{\mathbf{z}_1\mathbf{W}^q(\mathbf{Z}\mathbf{W}^k)^T}{\mu})\mathbf{Z}\mathbf{W}^v\end{aligned} \tag{4.9}$$

Here, μ is a learnable scale parameter which controls the non-linearity of softmax. Finally, we insert a fully connected layer together with batch normalization and ReLU activation after attention. Thus, a memory slot $\mathbf{M}_i$ is generated by,

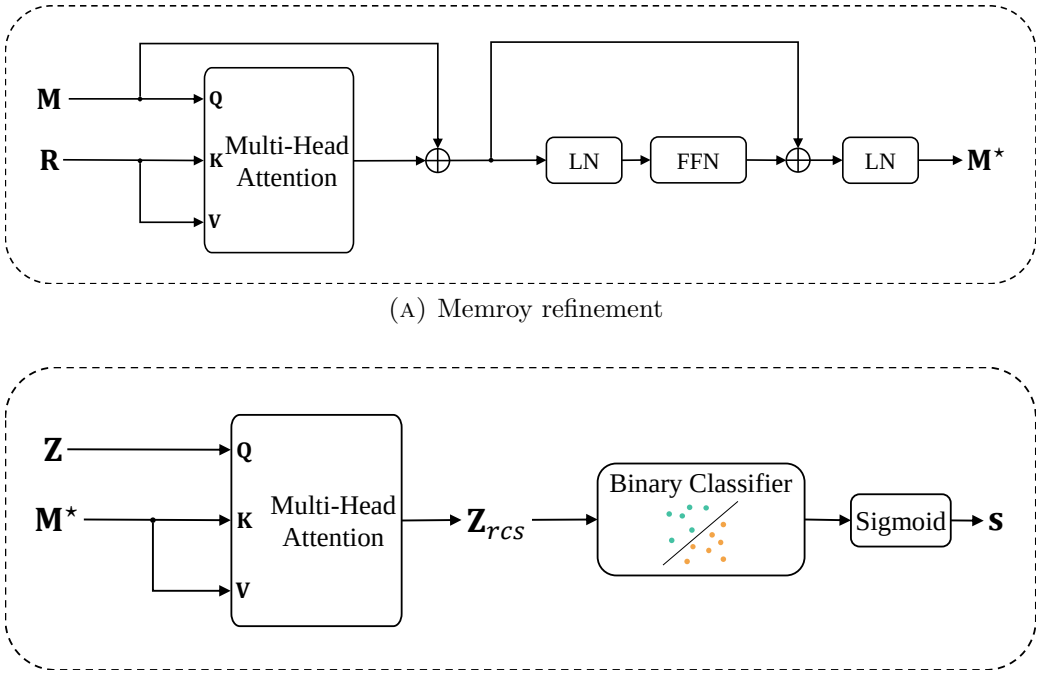
$$\mathbf{M}_i = \text{ReLU}(\text{BN}(\mathbf{M}_i^{'}\mathbf{W} + \mathbf{b})) \tag{4.10}$$

The fully connected layer expands $\mathbf{M}_i^{'}$ back to the original embedding dimension. As shown in Fig. 4.4, the memory cell comprises N independent memory slots.

4.2.4 Memory Refinement

The memory refinement scheme is proposed to eliminate the feature pollution from noisy false matches. Different from other neural memory networks [68, 69] whose memory writing or updating loops samples in a sequence one by one, we update $\mathbf{M}$ through the attention between the memory cell and top-ranked highly-confident samples accelerated with parallelization.

As mentioned above, $\mathbf{M}$ aggregates multiple aspects information of a probe from $\mathbf{Z}$. The discriminability relies on the dot-product similarity between $\mathbf{q}$ and $\mathbf{K}$ in Eq. (4.9). False matches might be transferred into $\mathbf{M}$ with low-quality attention weights, which affects the subsequent feature reconstruction and correlation prediction. Hence, we add a memory refinement module after initialization as shown in Fig. 4.5a. Given the BaseEncoder output sequence $\mathbf{Z}$, we chuck it into two parts with the upper part as $\mathbf{R} = \{\mathbf{z}_i\}_{i=1}^{k_2}$. The sequence $\mathbf{R}$ is named as refinement sequence which includes the first k_2 samples of $\mathbf{Z}$ (k_2 nearest neighbors of the probe). Our memory refinement combines features through attention which converts $\mathbf{M}$ to Queries, and $\mathbf{R}$ to Keys and Values. One thing worth mentioning is that the softmax here in MHA is applied on the k_2 dimension of $\mathbf{R}$. In other words, each memory slot refines itself by extracting information of interest from



(A) Memroy refinement

(B) Feature reconstruction and Correlation Prediction

FIGURE 4.5: (a) Memory refinement updates the memory cell by aggregating relevant information from the refinement sequence. (b) Feature reconstruction reversely builds each feature embedding through a combination of multiple memory slots. The binary classifier separates the reconstructed embeddings predicting the correlations.

the refinement sequence. Memory initialization and refinement cooperate to exclude interfering features of falsely retrieved samples from entering the memory cell. We use this discriminative Contextual Memory $\mathbf{M}^{\star}$ for feature reconstruct and correlation prediction afterward.

4.2.5 Feature Reconstruction and Correlation Prediction

The goal of ACP is to accurately predict the correlations between an image and its local neighbors. Since features stored in the memory cell are probe-specific, neighbors reconstructed from the memory will exhibit distinct patterns relevant to their identities. In other words, falsely retrieved samples after reconstruction will be different from those correct matches because they share less common features with the memory. A binary classifier can leverage the feature divergence to distinguish correct matches from strong distractors, predicting the correlations.

The implementation of feature reconstruction is shown in Fig. 4.5b. The MHA compares the similarity between each neighbor and the Contextual Memory, which can be regarded as the reverse process of the memory initialization (see Fig. 4.4). Likewise, a learnable projection maps the $\mathbf{Z}$ to Queries, and the other two projections convert $\mathbf{M}^{\star}$ to Keys and Values. Each feature embedding is reconstructed by finding a combination of multiple memory slots. Let us denote the reconstruction output as $\mathbf{Z}_{rcs}$. Unlike Transformer encoder layers where attention outputs are added to the input sequence as residuals, we keep the attention outputs as the final reconstruction results because we reconstruct features for correlation prediction rather than updating the input sequence.

Finally, a binary classifier separates the reconstructed embeddings. Denote the weight matrix of the classifier as $\mathbf{W}_c \in \mathcal{R}^{d\times 1}$, and the bias as b_c. The correlations prediction is,

$$\mathbf{s} = Sigmoid(\mathbf{Z}_{rcs}\mathbf{W}_c + b_c),\ \mathcal{R}^{k_1\times d} \rightarrow \mathcal{R}^{k_1} \tag{4.11}$$

where $Sigmoid(\cdot)$ is the element-wise sigmoid activation. We choose Focal Loss [70] to supervise the network training.

Chapter 5

Experiments

5.1 Evaluation Metrics

Following [2], we choose two metrics to evaluate the re-ID performance.

Cumulated Matching Characteristics (CMC) measures the probability that the ground-truth appears in the top-k of the rank list. Here, we report CMC@1 that is generally considered as a reflection of the ability to retrieve the easiest samples.

Mean average precision (mAP) calculates the averaged area under the Precision-Recall curve of all the query images. It measures the ability to retrieve all related images. We compare the difference between CMC and average precision (AP) with a tiny example in Fig. 5.1. The CMC@1 equals one because the top-1 is a correct match for three rank lists. However, AP for the second rank list is higher than the third list if we consider the ranking positions of all correct matches.

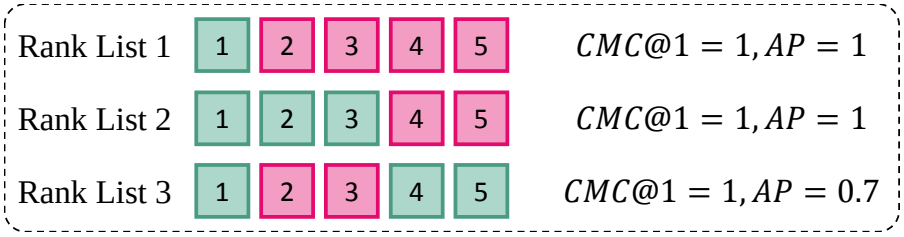


FIGURE 5.1: A tiny example shows the difference between CMC and average precision. The true and false matches are green and red, respectively. The CMC@1 equals 1 for all three rank lists, but the average precision differs.

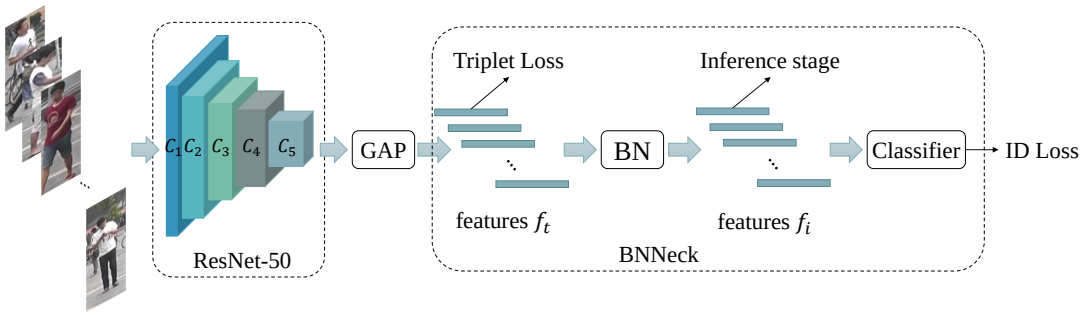


FIGURE 5.2: Architecture of the strong re-ID baseline with Batch Normalization Neck. The convolutional feature C_5 is pooled to a vector with a global average pooling layer.

5.2 Implementation Details

5.2.1 Re-ID Baseline

The baseline network is proposed by [8] which adopts a single-branch ResNet-50 as the backbone feature extractor. It maps the input image to a low dimension space while preserving features that can distinguish different instances. The overall pipeline is visualized in Fig. 5.2. In general, this strong baseline uses several following techniques to stabilize training and improve performance.

- **Batch Normalization Neck (BNNeck)** relieves the inconsistency between cross-entropy identity loss and triplet loss. The ID loss builds hyperplanes to separate the embeddings, which encourages fan-shaped distributions. Therefore, the embeddings trained with ID loss are more appropriate for cosine distance. However, the triplet loss uses Euclidean distance to minimize the inter-class compactness. If the ID loss and triplet loss are applied on the feature embedding simultaneously, they affect each other and results in performance degeneration. The Batch Normalization layer balances values of different dimensions of the pooled feature vector. The normalized feature f_i will be mapped to the surface of a hypersphere where a classifier separates each identity.

- **Learning Rate Warm-Up** increases the learning rate linearly in the initial training stage to slow down over-fitting and maintain the stability of deeper layers.

FIGURE 5.3: Examples of REA. Images in the first row are the original images in Market1501 [2]. The second row shows the effect of REA which masks out rectangle regions.

- **Random Erasing Augmentation (REA)** randomly masks out a rectangle region of the training image with a pre-defined probability. This prevents the backbone feature extractor from depending on some specific features for decision making, which reduces over-fitting and encourages robustness. The output from REA is shown in Fig. 5.3
- **Label Smoothing** can help to reduce the over-fitting for the feature extractor. Denote an input image as x_i with identity label as y_i. We smooth the label by,

$$y_i^* = \begin{cases} 1 - \frac{N-1}{N}\alpha & if \quad i = y_i \\ \frac{\alpha}{N} & otherwise \end{cases} \tag{5.1}$$

  where α is a constant that reduces the model's confidence for the prediction.

The implementation comes from FastReID[1][18] which is a powerful open-source toolbox designed for general instance re-ID.

5.2.2 Training Sequence Formulation

To train our ACP for correlation prediction, we need to construct a proper training set. Different from training the re-ID backbone whose training samples are triplets,

[1]https://github.com/JDAI-CV/fast-reid

the ACP is trained with image sequences. Specifically, we randomly choose one image from the image set with replacement each time. Denote it as a probe image with the feature embedding as $\mathbf{f}_p = \phi(\mathcal{I}_p)$. We can obtain its K-nearest neighbors, $\mathbf{N} = \{\mathbf{f}^j\}_{j=1}^{K}$. Next, we shuffle and select the top-K neighbors to generate a training sequence as shown in Fig. 5.4.

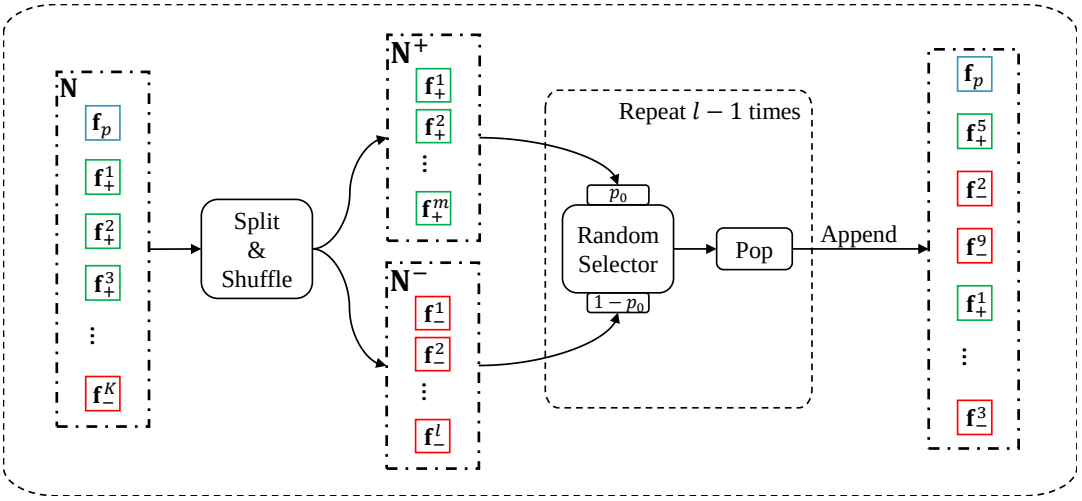


FIGURE 5.4: Training sequence formulation. K-nearest neighbors are divided into a positive set $\mathbf{N}^+$ and a negative set $\mathbf{N}^-$. The sequence for training is generated by randomly selecting samples from two sets with probability as p_0 and $1 - p_0$, respectively.

According to the identities of $\mathbf{f}_p$ and $\mathbf{f}^j$, $\mathbf{N}$ can be divided into two sets. The positive set $\mathbf{N}^+$ consists of samples in the same identity as $\mathbf{f}_p$, while the negative set $\mathbf{N}^-$ contains samples labeled with different identities. Suppose there are m samples in $\mathbf{N}^+$, then we have $K - m$ negative samples. We only keep the first l samples of $\mathbf{N}^-$ before random shuffling because $\mathbf{N}$ includes more negative samples than the positive ones. This prevents bottom-ranked easy negative samples from entering the training sequence. Then, we select the positive and negative sets with probability as p_0 and $1-p_0$, respectively. A sample is popped out from the selected set and appended to the training sequence. We repeat this operation $l - 1$ times with $\mathbf{f}_p$ placed on top of the training sequence afterward. In other words, $l - 1$ out of $m + l$ samples are selected for constructing a training sequence, which can be regarded as randomly masking out some neighbors for a given probe. We found this training strategy helps to alleviate the over-fitting problem.

5.2.3 Attention-based Correlation Predictor

The model architecture and the training process are controlled by several hyper-parameters as shown in Table 5.1. To construct a training sequence, we find the $K = 1000$ nearest neighbors for each image. We randomly select l_1 samples from the union set of the correct matches and the first several false matches. The refinement sequence length is set to l_2. The model training adopts Adam optimizer with weight decay as 5×10^{-4}. The initial learning rate is set to 2×10^{-4}. Similar to [8], we linearly warm up the learning process for the first ten epochs with warm-up factor equals 0.1. The parameter γ in Focal loss balances easy and hard samples. For MSMT17, VehicleID, and VERI-Wild, we choose $\gamma = 1$, making the Focal loss degrade to cross-entropy loss. We find this setting slightly improves the performance on three larger datasets. The testing parameters k_1 and k_2 are determined based on the test set size, empirically. All the experiments are conducted on a platform with 256GB RAM and 2 × Intel Xeon Silver 4214R CPU @ 2.40GHz. The GPU we use is a single GeForce RTX 2080Ti with 11GB VRAM.

TABLE 5.1: Hyper-parameters for model architectures, training and testing. The γ belongs to Focal loss.

	Model Architeture						Training			Testing	
	d	h	d_m	p_d	n	N	l_1	l_2	γ	k_1	k_2
Market1501	256	8	64	0.5	3	8	80	32	2	20	6
DukeMTMC	256	8	64	0.5	3	8	80	32	2	20	6
MSMT17	256	8	64	0.3	3	8	80	32	1	25	6
VeRi-776	256	8	64	0.5	3	8	160	64	2	55	10
VehicleID	256	8	64	0	3	8	24	8	1	25	4
VERI-Wild	1024	16	128	0	3	8	32	8	1	25	6

5.3 Performance Comparison

In this section, we compare the performance of our re-ranking approach with several widely-used re-ranking methods on the datasets introduced in Section 2.4. Results are reported and analyzed in Section 5.3.1. We choose parameters based on the recommendations tested in their original papers. Because re-ranking methods are generally sensitive to parameter selection, we also studied the influence of hyper-parameters in Section 5.3.2.

5.3.1 Comparisons with State-of-the-art Methods

The experimental results for each method are listed in Table 5.2. We tested AQE [12], αQE [13], k-reciprocal [16], SCA [47], GNN [53] and ECN [35]. The row named 'αQE + k-reciprocal' refers to perform k-reciprocal re-ranking after αQE. For GNN, ECN and SCA re-ranking, some results are not available because of insufficient memory. Note that the implementation of AQE and αQE here expand all the feature embeddings by considering the query and gallery sets as a whole.

TABLE 5.2: Performance (%) comparison. The best and second-best results are makred in red and blue, respectively.

Method	MSMT17		VeRi-776		VERI-Wild						VehicleID						Market1501		DukeMTMC	
					Samll		Medium		Large		Small		Medium		Large					
	CMC@1	mAP	CMC@1	mAP	CMC@1	mAP	CMC@1	mAP	CMC@1	mAP	CMC@1	CMC@5	CMC@1	CMC@5	CMC@1	CMC@5	CMC@1	mAP	CMC@1	mAP
Baseline	73.75	50.31	96.07	78.56	93.27	76.74	90.04	70.61	86.52	62.56	82.82	95.50	77.51	91.23	75.57	88.97	93.85	86.31	86.36	76.98
AQE	76.82	64.26	96.01	84.54	85.41	71.64	81.04	65.51	76.20	56.64	71.83	89.69	65.83	85.12	65.80	85.05	94.24	91.85	89.95	86.56
αQE	79.04	65.42	97.44	85.95	92.67	79.37	89.50	72.75	85.59	63.92	84.72	96.54	77.47	91.65	76.64	90.47	95.37	93.08	90.22	86.92
k-reciprocal	78.92	67.12	96.54	85.61	93.84	80.32	90.66	73.94	87.09	65.65	84.00	95.43	77.34	90.47	75.68	88.31	95.34	93.98	90.53	88.93
αQE + k-reciprocal	78.83	69.94	96.96	87.58	91.83	78.94	88.01	72.18	82.80	63.15	83.23	95.63	76.48	91.09	75.35	89.50	95.37	93.97	90.26	89.15
SCA	79.10	69.08	96.60	84.52	92.77	79.65	89.52	73.30	-	-	82.54	93.98	76.12	87.81	74.85	86.91	95.19	94.14	90.04	89.34
GNN	-	-	96.84	86.10	-	-	-	-	-	-	82.79	94.20	76.79	88.66	75.40	88.02	95.34	94.55	90.71	90.03
ECN	-	-	97.14	85.52	90.63	79.68	-	-	-	-	77.25	92.66	72.85	87.55	71.48	86.45	95.61	94.47	91.11	89.71
Ours	82.00	72.11	97.02	87.50	95.18	88.16	92.45	83.60	89.64	75.80	88.13	97.10	82.12	93.74	81.66	91.73	95.49	93.95	90.98	89.08

5.3.1.1 MSMT17

MSMT17 is the largest person re-ID dataset in our experiments, especially for the test set which adds up to almost 100k images in total. As shown in Table 5.2, we ranked first for both CMC@1 and mAP. The baseline CMC@1 improved from 73.75% to 82.00%, and baseline mAP increased to 72.11% from 50.31% with 21.80% improvement. The improvement proves that our method is better at handling complex relationships between the k-nearest neighbors. Besides, we only require the access of first-order nearest neighbors for each probe embedding.

5.3.1.2 VeRi-776

VeRi-776 has relatively more images per identity on average. Therefore, we increase the neighborhood size k_1 accordingly. Results show that we produce the second-best mAP, which equals 87.50%. Although our method does not rank first for CMC@1, there is only 0.42% difference compared to the best result obtained by αQE. Our method achieved the balance between CMC@1 and mAP.

5.3.1.3 VERI-Wild

We can significantly boost the baseline performance on the small, medium, and large subsets of VERI-Wild. By comparing the results across different subsets, we can observe that the k-reciprocal rule begins to fails as the testing set becomes larger. The mAP improvements for k-reciprocal re-ranking are 3.58%, 3.33%, 3.09% for small, medium, and large, respectively. However, our method is robust to the test set size achieving 11.42%, 12.99%, and 13.24%. Some methods run out of memory halfway.

5.3.1.4 VehicleID

The metric mAP is not provided because there is only one correct match in the gallery set for each query image. Instead, we report CMC@5 which measures the probability of having the correct match in top-5 candidates. As shown in the table, our method outperforms other re-ranking approaches for all three subsets by a large margin. The improvement of k-reciprocal re-ranking is limited, which might be caused by the insufficient contextual information in the gallery set (one image per identity). The αQE gains a slight improvement over the baseline by taking the query-to-query similarities into account. The comparison between αQE and our approach demonstrates the superiority of discriminative correlation prediction.

5.3.1.5 Market1501

Our method achieves the second-best CMC@1, which equals 95.49% with only 0.12% difference compared to the best obtained by ECN re-ranking. For mAP, the best result 94.55% is produced by GNN re-ranking, which is 0.6% higher than ours. We ascribe this unsatisfying result to the lack of training data. As shown in Table 2.1, Marker1501 is the smallest amongst all six datasets with only 13k images for training. The backbone itself has already been facing over-fitting problems, not even for building a re-ranking network without extra training data.

5.3.1.6 DukeMTMC

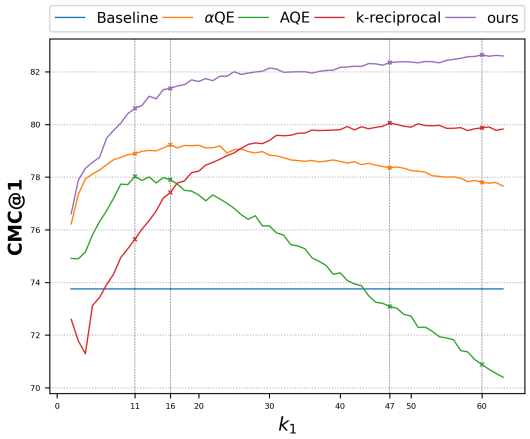
Similar to Market1501, our approach ranked second for CMC@1. We obtain 90.98% which is 0.13% lower than the best 91.11%. We consider the inferior results are mainly caused by three facts. First of all, DukemTMC has less than 17k images for training with only 3586 additional images compared to the Market1501. Second, the test sets for both benchmarks are relatively simpler than others. Images are taken under 6 and 8 cameras for the Market1501 and DukeMTMC, respectively. The limited cameras indicate the limited appearance variation within an identity. Last, unique identities are less than 800, which reduces the probability of having strong similar persons. Therefore, the neighborhood relationships of Market1501 and DukeMTMC are less complex than MSMT17 and VERI-Wild. This kind of simplicity can be well modeled by hand-designed algorithms but imposes overfitting risks on our deep learning-based approach. For mAP, all the methods bring considerable improvements, and our approach is 12.10% higher than the baseline. The highest mAP is produced by ECN re-ranking.

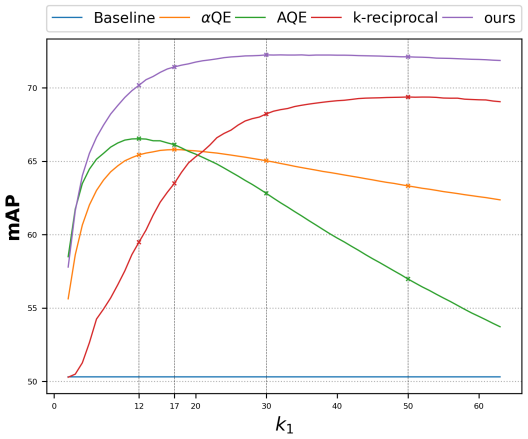
5.3.2 Parameter Sensitivity

Here, we study the influence of the testing parameters k_1 and k_2. Generally, re-ranking approaches should be robust to parameter selection.

5.3.2.1 Neighborhood size k_1.

We use k_1 to control the number of nearest neighbors. Generally, larger k_1 means richer contextual information but also involves more false matches. The previous section tests the performance with empirically decided parameters, which may not fully reveal the maximum capacity for each method. For example, αQE can leverage contextual cues in broader neighborhoods than AQE because of the power-normalized combination weight. To better study the influence of neighborhood size, we evaluate re-ranking methods via gradually increasing k_1. The performance variation is compared amongst AQE, αQE, k-reciprocal re-ranking, and our method. We fix the α in αQE as 3 and λ in k-reciprocal re-ranking as 0.3. Because of the space limitation, we only show the figures for MSMT17 in Fig. 5.5 and the small subset of VERI-Wild in Fig. 5.6. The k_2 is fixed as 6 for both datasets.





k_1	Baseline	αQE	AQE	k-reciprocal	ours
11	73.75	78.89	78.03	75.64	**80.61**
16	73.75	79.23	77.90	77.42	**81.37**
47	73.75	78.35	73.09	80.05	**82.35**
60	73.75	77.80	70.89	79.87	**82.64**

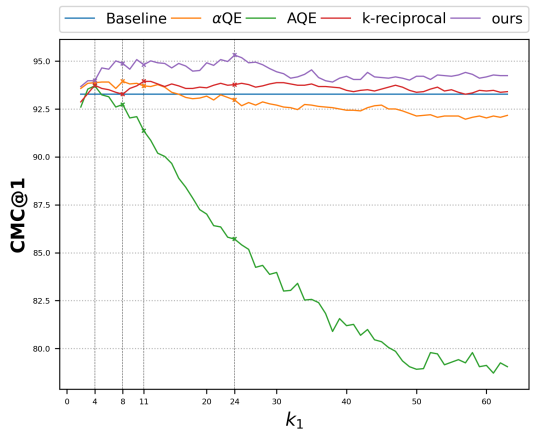
(A) CMC@1 (%) wrt. k_1

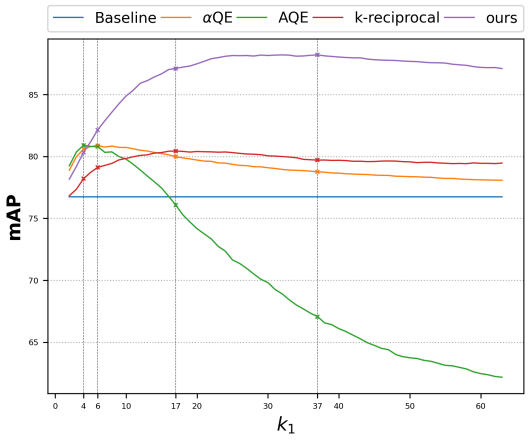
k_1	Baseline	αQE	AQE	k-reciprocal	ours
12	50.31	65.44	66.54	59.49	**70.18**
17	50.31	65.80	66.13	63.49	**71.43**
30	50.31	65.03	62.81	68.23	**72.25**
50	50.31	63.32	56.97	69.38	**72.12**

(B) mAP (%) wrt. k_1

FIGURE 5.5: Performance versus k_1 on MSMT17.

We list the peak performance below in a table with the best result in each row marked with bold type. From the curve plots, we have the following observations. First, the performance of AQE drops quickly as k_1 becomes larger because AQE assigns uniform weight to each neighbor. The aggregated feature embedding will be pulled towards the wrong directions if false matches dominate the neighborhood. The αQE uses power-normalized combination weights, which significantly relieves the false-match pollution issue. However, it is still not enough to entirely reject falsely retrieved samples resulting in the gradual performance degeneration. Second, our approach consistently outperforms αQE for CMC@1 and mAP with a large margin. Given a specific k_1, the performance gap proves that the correlation prediction provides a more accurate direction to shrink embeddings toward their identity centers. Third, compared to αQE, the k-reciprocal rule with backward verification is more robust to false matches. The turning point of k-reciprocal re-ranking where the performance begins to drop comes later than αQE.





k_1	Baseline	αQE	AQE	k-reciprocal	ours
4	93.27	93.88	93.71	93.74	**93.98**
8	93.27	93.94	92.74	93.27	**94.88**
11	93.27	93.71	91.37	93.94	**94.81**
24	93.27	92.97	85.71	93.78	**95.31**

(A) CMC@1 (%) wrt. k_1

k_1	Baseline	αQE	AQE	k-reciprocal	ours
4	76.74	80.58	**80.91**	78.20	80.33
6	76.74	80.88	80.80	79.12	**82.15**
17	76.74	79.99	76.08	80.44	**87.11**
37	76.74	78.76	67.06	79.71	**88.21**

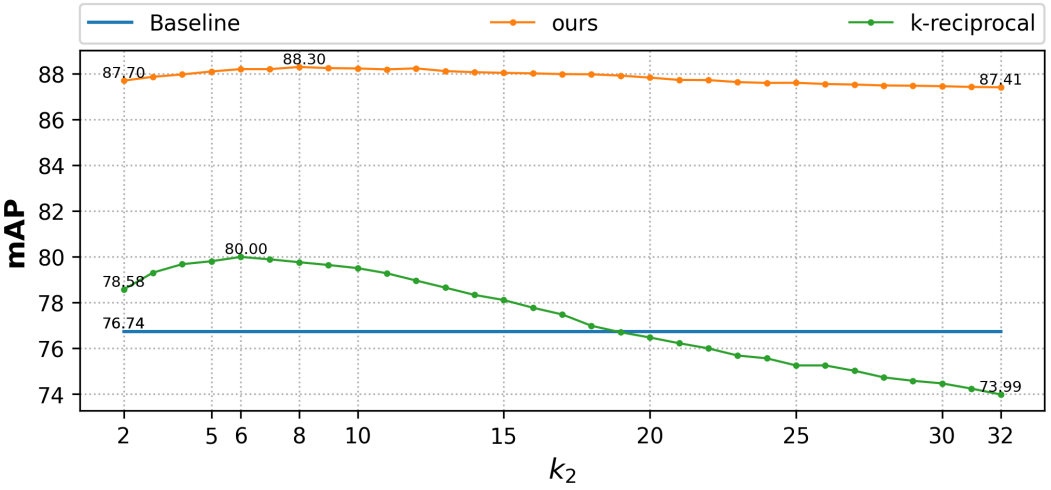
(B) mAP (%) wrt. k_1

FIGURE 5.6: Performance versus k_1 on the small subset of VERI-Wild.

5.3.2.2 Refinement sequence k_2.

As mentioned above, memory refinement reduces feature pollution from falsely aggregated neighbors using a small group of high-confident samples. To measure the sensitivity, we test the performance of our method on the small subset of VERI-Wild with different k_2. Fig. 5.7 shows the variation with k_1 fixed as 32.

The result shows that mAP increases as the refinement sequence becomes longer, starting from a small value. We reach the best mAP when refining on the 8-nearest neighbors. After that, mAP gradually drops if we continue to include more samples. Compared to the best reuslt 88.30%, mAP drops to 87.41% when $k_2 = 32$, which is even worse than $k_2 = 2$ whose mAP reaches 87.70%. This indicates that a small group of highly-confident neighboring embeddings are helpful in improving the discriminability of memory. In Fig. 5.7, we also plot the performance of k-reciprocal re-ranking versus k_2 which controls the local query expansion. The decrease of mAP shows that it is more sensitive to k_2 because of its indiscriminate combination.



(A) mAP (%) wrt. k_2

FIGURE 5.7: Performance variation versus k_2 for k-reciprocal re-ranking and our method.

5.4 Model Studies

In this section, we study the influence of model architectures. Specifically, we verify the effectiveness of the proposed modules in Section 5.4.1 by conducting ablation experiments. Besides, we test how the model architecture parameters, such as the number of heads in MHA, affect the final result in Section 5.4.2.

5.4.1 Ablation

The proposed model comprises multiple modules, i.e., multi-block feature fusion, BaseEncoder, Contextual Memory, and memory refinement. We train the model utile convergence with one or multiple modules removed. The re-ranking performance is tested on the small subset of VERI-Wild with $k_1 = 25, k_2 = 6$. Results are shown in Table 5.3. Note that the output from BaseEncoder will be fed into the final binary classifier if the memory cell is disabled.

5.4.1.1 Multi-Block Feature Fusion

We conducted two experiments (Exp-E and Exp-G) to study whether features from shallower layers contain discriminative information. Results show the CMC@1 and

TABLE 5.3: Ablation study. In the first row, experiments are tagged with Exp-X where X is a capital letter. The best and second best results are marked in red and blue respectively.

Modules	Baseline	Exp-A	Exp-B	Exp-C	Exp-D	Exp-E	Exp-F	Exp-G
Multi-Block Feature Fusion			✓	✓			✓	✓
BaseEncoder		✓	✓	✓		✓		✓
Memory cell				✓	✓	✓	✓	✓
Memory refinement					✓	✓	✓	✓
CMC@1	93.27%	86.18%	86.98%	92.87%	92.07%	94.38%	93.37%	95.18%
mAP	76.74%	77.93%	77.90%	86.31%	84.20%	87.09%	86.56%	88.16%

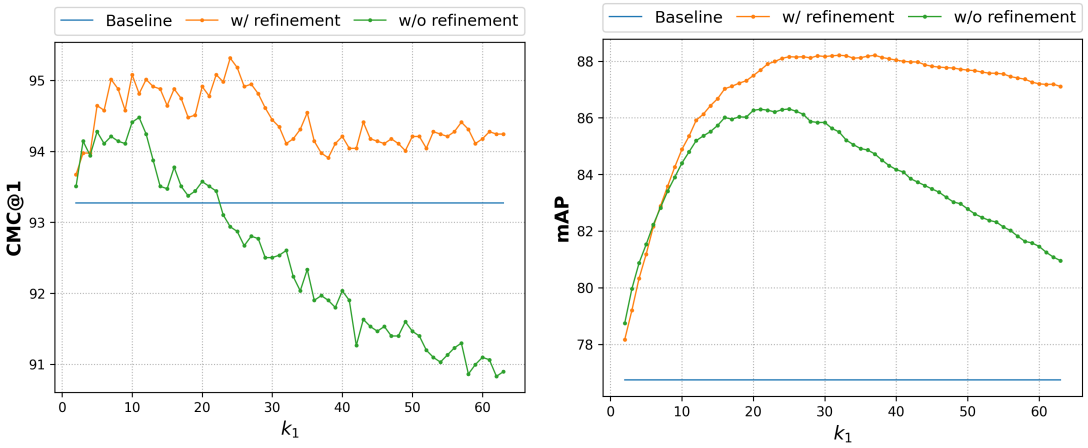
mAP improve from 94.38% to 95.18% and 87.09% to 88.16%, respectively. The improvement verifies our assumption that different blocks provide information in various granularities. Our re-ranking approach can make use of this information that is discarded in the baseline model.

5.4.1.2 BaseEncoder

Before memory initialization, multi-block features are preprocessed by the BaseEncoder which aggregates the contextual information for each embedding. Comparing Exp-F and Exp-G, we can observe that CMC@1 and mAP drop 1.81% and 1.60% respectively with BaseEncoder disabled. However, the decline is more severe if multi-block feature fusion is removed at the same time. From Exp-E to Exp-D, the CMC@1 and mAP decrease 2.31% and 2.89%, respectively. A possible explanation is that the multi-block feature vector provides richer information than the baseline. Once discriminative features from shallower blocks are cut off, the ability of BaseEncoder to aggregate homogeneous sub-features plays an important role.

5.4.1.3 Contextual Memroy

Contextual Memory is the most critical module in our architecture. As shown in Exp-B and Exp-C, directly predicting correlations for embeddings generated by BaseEncoder significantly harms the performance. The CMC@1 decreases from 92.87% to 86.98% with 5.89% decline. Similarly, mAP also drops 8.41%, reaching 77.90%. The performance drop suggests that a fixed hyperplane is not enough to separate embeddings from the Transformer encoder. Instead, the similarity between each embedding counts.



(A) CMC@1 (%) wrt. k_1

(B) mAP (%) wrt. k_1

FIGURE 5.8: Ablation study with memory refinement module removed.

5.4.1.4 Memory refinement

Memory refinement is a sub-module of the Contextual Memory, which aims at preventing feature pollutions. We conduct Exp-C and Exp-G to study the refinement process with k_1 fixed as 25. The CMC@1 increases from 92.87% to 95.18% with 2.31% improvement, and mAP also improved 1.85%. To better reveal the power of memory refinement, we evaluate the re-ranking performance under different k_1 for the consideration that refinement is more crucial if more false matches are included in the neighborhood. The metric curve versus k_1 is shown in Fig. 5.8. Comparing the results, we can observe that the model without refinement deteriorates rapidly when k_1 becomes larger. Memory refinement enables our model to leverage contextual information in larger neighborhoods without affected by false matches.

5.4.2 Model Architecture Parameters

The ablation studies verify the effectiveness of each proposed module. Here, we perform experiments to study the relationships between architecture parameters and the re-ranking performance. The basic model follows Exp-G in Table 5.3 with all the modules enabled. Each time, we set different values for an architecture parameter with others fixed. After the model converges, CMC@1 and mAP are tested on the small subset of VERI-Wild with $k_1 = 25, k_2 = 6$.

5.4.2.1 BaseEncoder Layers

We control the model capacity by adjusting the number of layers in the BaseEncoder. Results are shown in Fig. 5.9. We can observe that the two-layer BaseEncoder obtains the best results. If we stack more layers, the performance gradually decreases but is relatively stable. At the same time, we noticed that the model becomes harder to converge. The warm-up period is doubled to 20 epochs to stabilize the model at initial training stages for 4-layer, 5-layer, and 6-layer BaseEncoder. The model degeneration is likely to be caused by that deeper BaseEncoder affects the backflow of gradient.

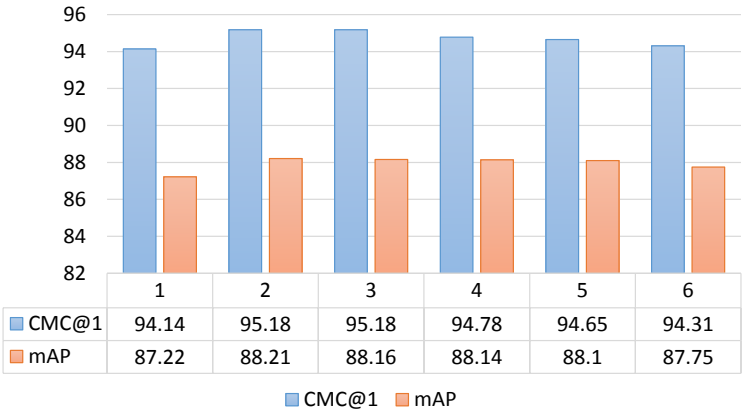


FIGURE 5.9: Performance variation wrt. number of encoder layers in the BaseEncoder.

5.4.2.2 Heads in Multi-Head Attention

MHA is one of the important building blocks in our model. The head refers to split original feature embeddings into multiple smaller ones for attention calculation independently. As shown in Fig. 5.11, results are unsatisfying for models with limited number of heads. For example, the CMC@1 drops from 95.18% to 92.03% when heads reduce from 16 to 1. The mAP also drops 3.88%. Intuitively, the multi-head structure captures the similarity between different sub-features. It can be considered an improved version of [49] where sub-features are selected either manually or randomly. The projection matrices in Multi-Head Attention are learned from data, which enable the transformed sub-features to focus on the most discriminative regions.

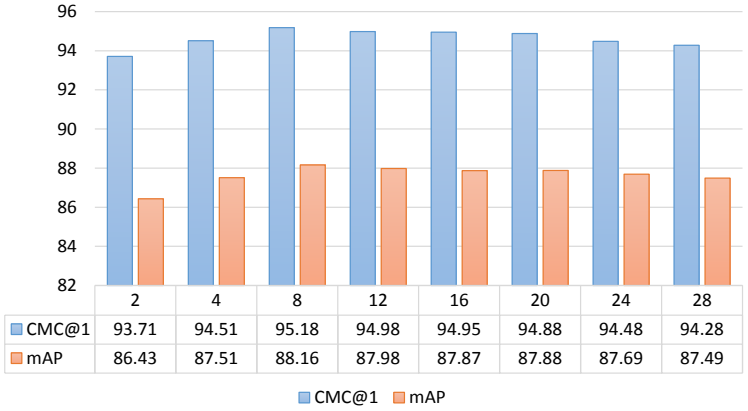


FIGURE 5.10: Performance variation wrt. the memory size.

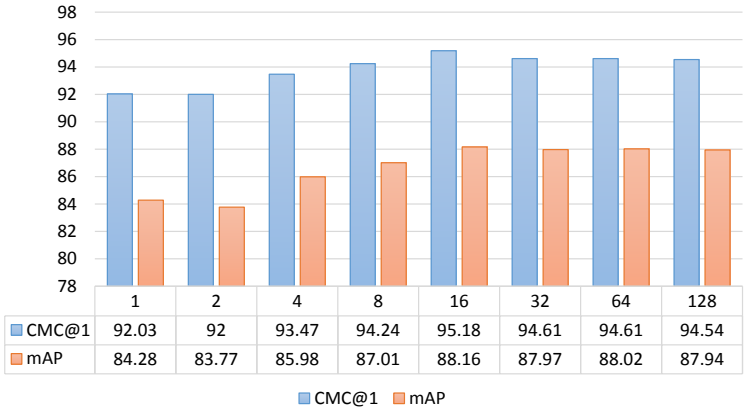


FIGURE 5.11: Performance variation wrt. number of heads in MHA.

5.4.2.3 Memory Size

Memory size refers to the number of slots in the Contextual Memory. Similar to the Multi-Head Attention, each memory slot focuses on some specific aspects. Therefore, the memory size relates to the distinctive aspects of a person or vehicle. We study the memory size in Fig. 5.10. From the table, the eight-slot memory outperforms others. An interesting finding is that the number 8 happens to be the same as the views in multi-view feature inference [11], which suggests the memory cell stores view-related information inside. Further increasing the memory size slightly harms the performance. However, the effect is more significant if we adopt a smaller memory. The CMC@1 decreases 1.47% and mAP drops 1.73% from eight-slot memory to two-slot memory because small Contextual Memory can not provide enough discriminative features for neighborhood reconstruction.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we have proposed a novel Contextual Memory cell to mimic the remembering process that humans adopt for re-ID and re-ranking. By comparing neighbors' features with the multi-view appearance information in the memory, we predict the correlations between each image and its k-nearest neighbors. The re-ranking is achieved by shrinking each embedding towards the identity centers with correlation prediction as discriminative combination weights. Compared to the state-of-the-art re-ranking approaches, our model has the following advantages: First, ACP does not rely on hand-designed rules. Instead, the ability of correlation prediction is learned from data with gradient descent algorithm. Second, we treat re-ranking as a binary classification problem of top-k retrieved images whose embeddings are utilized for the estimation of cluster centers. Third, the correlation prediction only requires the access of the first-order nearest neighbors, which significantly reduces memory consumption. Besides, our method can scale to different platforms by chucking the input into small batches. Last, the whole architecture can be easily implemented with many deep-learning frameworks. The acceleration with parallelization on GPU is available without extra efforts.

Extensive experiments on six widely-used re-ID datasets validate the effectiveness of the proposed method. Especially, the performance boost on large-scale datasets VERI-Wild, MSMT17, and VehicleID exhibits the ability of ACP in handling complex neighborhood relationships.

6.2 Future Works

The proposed ACP provides a solution of re-ranking in a data-driven manner. Although ACP surpasses many traditional re-ranking approaches on several datasets, the current design has certain limitations.

First and foremost, ACP is a feature similarity-based re-ranking method. The most crucial part is the Multi-Head Attention module, which disassembles the high-dimensional embeddings into smaller sub-features and compares their similarities with dot-product. In extreme cases where feature embeddings of two different persons or vehicles possess significant similarity, feature comparison might fail, and ACP will have trouble distinguishing one object from the other. In the meantime, we noticed that the precious graphical structure of each image's local neighbors is discarded during the comparison process. This kind of semantic neighboring relationship is essential for excluding strong outliers as proven in [16, 47, 48, 53]. Experiment results show that these methods ranked high on small datasets like Market1501 [2], DukeMTMC [25] even by solely considering the graphical structure. The performance will gain another boost if we can leverage this additional graphical information together with appearance features.

Another limitation of ACP is that it requires re-training on different datasets because of the domain gap. In other words, a well-trained ACP on one dataset cannot be ported into other datasets directly. Instead, the model must be trained from scratch to accommodate to the new data distribution. The model training and parameter fine-tuning for different datasets is not only time-consuming but also tedious. For real-world applications where conditionals are unconstrained, the unpredictable data distribution may lead to false predictions. Besides, the isolation between different datasets makes it hard to take advantage of the training data from larger datasets to relieve the over-fitting problem for small datasets.

Based on the discussion of the existing limitations, some possible and promising directions for future research are analyzed in 6.2.1 and 6.2.2.

6.2.1 Graphical Structure

Graph Neural Network (GNN). GNN is a common approach to dealing with structured data like relationships and interactions. It learns the interactions between different nodes from labeled data. Therefore, GNN has been widely used in representation learning such as node classification and link prediction [71]. An example of link prediction is shown in Fig. 6.1 As we mentioned in Section 2.2.3,

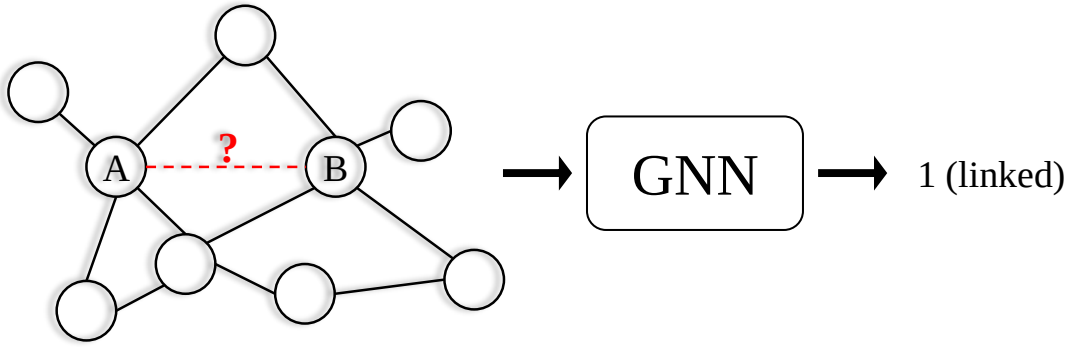


FIGURE 6.1: Illustration of link prediction with graph neural network. Given the feature embeddings and graphical structure of different nodes, GNN predicts the probability of forming links of for the selected node.

[54] adopts GNN to predict the link probability, i.e., whether the same identity as the query or not. They directly sort the predicted links by treating them as a new distance measure in the contextual space. Similarly, we can use link predictions for the center estimation that is more efficient than forming a new distance. To encapsulates more complex feature similarities, the Multi-Head Attention can be integrated too.

Graph Embedding. Another way of leveraging graphical structure is injecting the graphical information into the Transformer with graph embedding. The original transformer architecture [55] uses positional embedding to represent the relative order of the input elements. Inspired by this, we consider adding the graph embedding to the feature embeddings, which stores the connectivity of k-nearest neighbors.

Graph embedding is an efficient way of solving graph analytics problems, which suffers from the high computation problem and high space cost with adjacent matrices. To put it simple, graph embedding projects the graph to a low dimensional

space with graphical information preserved [72]. Besides, the converted graph embedding is more flexible for computation since its in Euclidean space. An schematic of graph embedding is shown in Fig. 6.2.

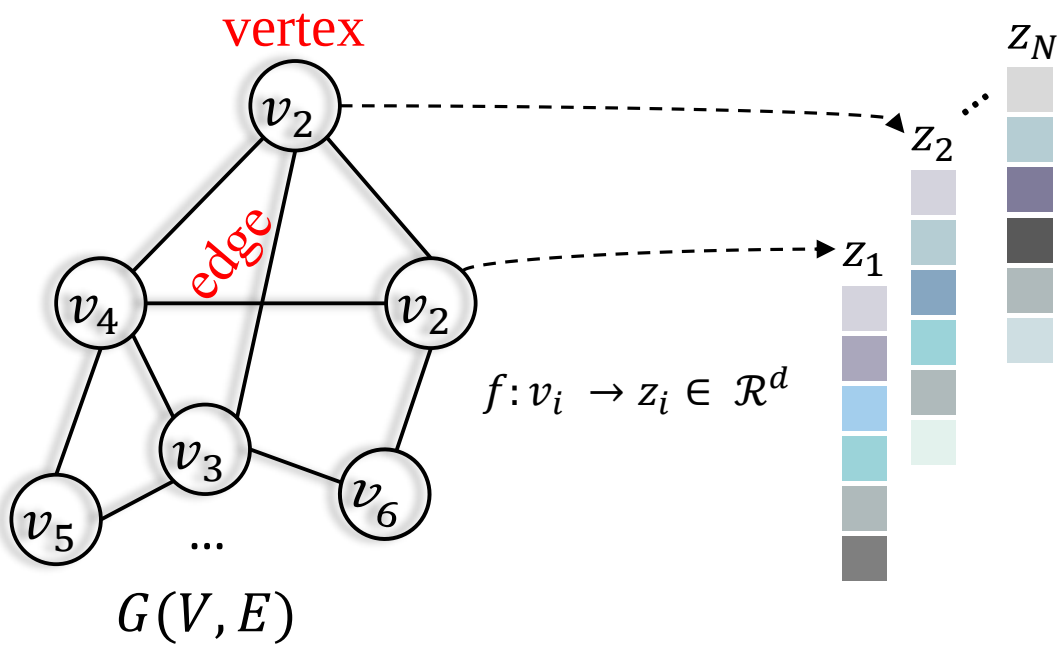


FIGURE 6.2: Illustration of graph embedding. The original graph $G(V, E)$ is sparse and in high-dimensional non-Euclidean space. The embedding function f projects the vertex v_i to low-dimensional dense vector z_i, where d refers the graph embedding dimension.

Many graph embedding methods have been proposed for different graphs, e.g., weighted or binary, directed or undirected, etc. The classic DeepWalk is proposed by [73], which learns graph representations with local information obtained from truncated random walks on the graph. Some methods apply deep learning techniques on graph embedding without random walks. Niepert et al. [74] select node sequence from a graph via graph labeling. A local neighborhood graph is assembled for part of nodes in the selected sequence, which is processed with existing CNN architecture.

With the converted graph embedding, Multi-Head Attention can capture the similarities from not only the appearance of an image but also its local connectivities to other images. The graph augmented Transformer might also provide some insights for the general graph analytics task, such as the link prediction.

6.2.2 Parameter Sharing

Parameter sharing is a common approach to relieving over-fitting in the deep learning community. For example, CNN shares the convolutional kernels across the entire feature map, which significantly reduces the parameters compared to Multi-Layer Perceptron (MLP). On the other hand, parameter sharing also exists on larger scales, e.g., sharing the deep neural network completely, which is also known as Multi-Task Learning (MTL). MTL has been used successfully for various kinds of applications, such as natural language processing [75], object dectection [76]. It can be considered as an implicit way of data augmentation.

The parameter sharing for re-ID has been studied by Zheng et al. [4] where multiple datasets are combined to encourage robust and discriminative features. A unique large-scale vehicle dataset (VehicleNet) is proposed by harnessing four public vehicle re-ID datasets. From the view of MTL, training model on VehicleNet is equivalent to performing re-ID on four different domains at the same time. As shown in Fig. 6.3, the existence of an auxiliary identity squeezes the embedding space for other identities. When the auxiliary identity is removed, it creates large margins between the embeddings and the new decision boundary.

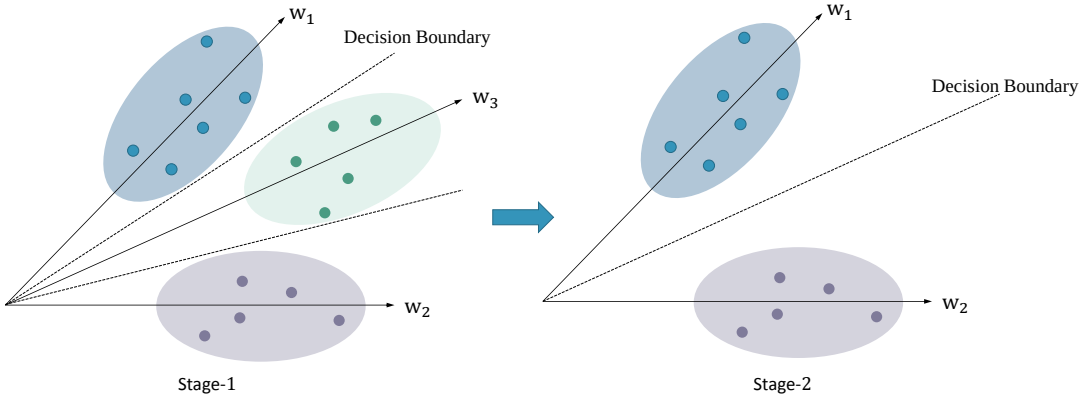


FIGURE 6.3: Geometric interpretation of VehicleNet. The w_i refers to the weight of the final classifier, where w_3 corresponds to an auxiliary identity that belongs to vehicles from other datasets that not exists in the target domain. The auxiliary identity forces the clusters of w_1 and w_2 to be compact. When w_3 is removed in stage-2, the new decision boundary has a large margin.

Inspired by [4], we consider sharing parameters of the feature extractor for re-ranking to break the data barriers between different re-ID datasets. The current model extracts features from different blocks of the ResNet-50 that are trained

under the standard re-ID framework. During the training of ACP, the backbone ResNet-50 is set frozen to save computation power. However, features of re-ID might not be optimal for the re-ranking purpose. To solve this problem, we argue sharing parameters by detaching ACP from the re-ID backbones. Specifically, the multi-block feature fusion block should take image inputs, and parameters of the base feature extractor will be updated with gradient descent algorithm. Next, the training sequence will be sampled from multiple datasets. The enriched training image variation due to dataset combination can facilitate learning robust features and reduce over-fitting, especially on small datasets.

Bibliography

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. xix, 7, 25

[2] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. xx, 5, 13, 15, 17, 33, 35, 50

[3] Mang Ye, Chao Liang, Yi Yu, Zheng Wang, Qingming Leng, Chunxia Xiao, Jun Chen, and Ruimin Hu. Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *IEEE Transactions on Multimedia*, 18(12):2553–2566, 2016. 1, 11

[4] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. Vehiclenet: learning robust visual representation for vehicle re-identification. *IEEE Transactions on Multimedia*, 2020. 1, 53

[5] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *European conference on computer vision*, pages 1–16. Springer, 2014. 2

[6] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1363–1372, 2016. 2

[7] Zhihui Zhu, Xinyang Jiang, Feng Zheng, Xiaowei Guo, Feiyue Huang, Xing Sun, and Weishi Zheng. Aware loss with angular regularization for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13114–13121, 2020. 2

[8] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 22(10):2597–2609, 2019. 3, 9, 34, 37

[9] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2018.

[10] Pirazh Khorramshahi, Neehar Peri, Jun-cheng Chen, and Rama Chellappa. The devil is in the details: Self-supervised attention for vehicle re-identification. In *European Conference on Computer Vision*, pages 369–386. Springer, 2020. 8, 9

[11] Yi Zhou, Li Liu, and Ling Shao. Vehicle re-identification by deep hidden multi-view inference. *IEEE Transactions on Image Processing*, 27(7):3275–3287, 2018. 2, 8, 47

[12] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2, 3, 10, 38

[13] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 3, 10, 38

[14] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012. 2, 3, 10, 17

[15] Mang Ye, Jun Chen, Qingming Leng, Chao Liang, Zheng Wang, and Kaimin Sun. Coupled-view based ranking optimization for person re-identification. In *International Conference on Multimedia Modeling*, pages 105–117. Springer, 2015. 2

[16] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017. 2, 4, 11, 38, 50

[17] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European conference on computer vision (ECCV)*, pages 480–496, 2018. 3, 8

[18] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: a pytorch toolbox for real-world person re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 3, 9, 35

[19] Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas. Total recall ii: Query expansion revisited. In *CVPR 2011*, pages 889–896. IEEE, 2011. 3

[20] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. 3

[21] Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Ling-Yu Duan. Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3235–3243, 2019. 5, 13, 15

[22] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer gan to bridge domain gap for person re-identification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. doi: 10.1109/CVPR.2018.00016. 5, 13, 16

[23] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2167–2175, 2016. doi: 10.1109/CVPR.2016.238. 5, 13, 14

[24] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, 2017. 5, 13, 14, 17

[25] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016. 5, 13, 15, 50

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 7

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 7

[29] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1367–1376, 2017. 7, 8

[30] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016. 7

[31] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3800–3808, 2017. 7

[32] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. Multi-scale deep learning architectures for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5399–5408, 2017. 7

[33] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1179–1188, 2018. 7

[34] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification across multiple resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8042–8051, 2018. 7

[35] M Saquib Sarfraz, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–429, 2018. 8, 11, 38

[36] Huibing Wang, Jinjia Peng, Dongyan Chen, Guangqi Jiang, Tongtong Zhao, and Xianping Fu. Attribute-guided feature learning network for vehicle reidentification. *IEEE MultiMedia*, 27(4):112–121, 2020. 8

[37] Tsai-Shien Chen, Chih-Ting Liu, Chih-Wei Wu, and Shao-Yi Chien. Orientation-aware vehicle re-identification with semantics-guided part attention network. In *European Conference on Computer Vision*, pages 330–346. Springer, 2020. 8, 9

[38] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 9

[39] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 9

[40] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European conference on computer vision*, pages 135–153. Springer, 2016. 9

[41] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 9

[42] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *European conference on computer vision*, pages 732–748. Springer, 2016. 9

[43] Yiluan Guo and Ngai-Man Cheung. Efficient and deep person re-identification using multi-level similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2335–2344, 2018. 9

[44] Guangyi Chen, Chunze Lin, Liangliang Ren, Jiwen Lu, and Jie Zhou. Self-critical attention learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9637–9646, 2019.

[45] Sanping Zhou, Fei Wang, Zeyi Huang, and Jinjun Wang. Discriminative feature learning with consistent attention regularization for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8040–8049, 2019.

[46] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Densely semantically aligned person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 667–676, 2019. 9

[47] Song Bai and Xiang Bai. Sparse contextual activation for efficient visual re-ranking. *IEEE Transactions on Image Processing*, 25(3):1056–1069, 2016. 11, 38, 50

[48] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR 2011*, pages 777–784. IEEE, 2011. 11, 50

[49] Rui Yu, Zhichao Zhou, Song Bai, and Xiang Bai. Divide and fuse: A re-ranking approach for person re-identification. *arXiv preprint arXiv:1708.04169*, 2017. 11, 46

[50] Zheng Wang, Junjun Jiang, Yi Yu, and Shin'ichi Satoh. Incremental re-identification by cross-direction and cross-ranking adaption. *IEEE Transactions on Multimedia*, 21(9):2376–2386, 2019. 11

[51] Chunxiao Liu, Chen Change Loy, Shaogang Gong, and Guijin Wang. Pop: Person re-identification post-rank optimisation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 441–448, 2013. 11

[52] Hanxiao Wang, Shaogang Gong, Xiatian Zhu, and Tao Xiang. Human-in-the-loop person re-identification. In *European conference on computer vision*, pages 405–422. Springer, 2016. 11

[53] Xuanmeng Zhang, Minyue Jiang, Zhedong Zheng, Xiao Tan, Errui Ding, and Yi Yang. Understanding image retrieval re-ranking: A graph neural network perspective. *arXiv preprint arXiv:2012.07620*, 2020. 11, 12, 38, 50

[54] Hongmin Liu, Zhenzhen Xiao, Bin Fan, Hui Zeng, Yifan Zhang, and Guoquan Jiang. Prgcn: Probability prediction with graph convolutional network for person re-identification. *Neurocomputing*, 423:57–70, 2021. 12, 51

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 12, 51

[56] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018. 12

[57] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 12

[58] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 12

[59] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 12

[60] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 12

[61] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 13

[62] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 13

[63] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 13

[64] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2020. 13

[65] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *arXiv preprint arXiv:2102.04378*, 2021. 13

[66] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 15

[67] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 15

[68] Yuan Yuan, Dong Wang, and Qi Wang. Memory-augmented temporal dynamic learning for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9167–9175, 2019. 29

[69] Tianyu Yang and Antoni B Chan. Visual tracking via dynamic memory networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(1): 360–374, 2019. 29

[70] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 31

[71] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *arXiv preprint arXiv:1802.09691*, 2018. 51

[72] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques and applications.(2017). *arXiv preprint arxiv:1709.07604*, 2017. 52

[73] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. 52

[74] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016. 52

[75] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008. 53

[76] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 53