Universiteit Leiden

Master Computer Science

Deep Learning for Visual Query Tasks Based on Spatio-Temporal Graph

Name:	[Ziwei Zhang]
Student ID:	[s2716240]
Date:	[12/03/2023]
Specialisation:	[Computer Science Data Science]
1st supervisor:	[Mitra Baratchi]
2nd supervisor:	[Bas van Stein]

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

In recent years, using deep learning methods for visual query tasks has garnered significant attention and has resulted in notable progress. Person re-identification represents a typical use case in visual query tasks. However, the visual query process still faces numerous challenges, particularly with regards to the issues of occlusion caused by obstacles and visual ambiguity arising from appearance-similar objects, and a joint solution to these challenges has been lacking. Furthermore, we observe that consecutive video frames can provide supplementary temporal information for each other. In contrast, different body parts within the same video frame can provide supplementary spatial information for each other. Therefore, modeling the temporal information across consecutive frames and the spatial information within each frame may offer a joint solution to occlusion and visual ambiguity challenges. In this thesis, we propose a novel approach that represents a video as a spatio-temporal graph (STG) by utilizing graph theory, integrating temporal relations and spatial information between video frames. Then, we apply graph convolutional networks (GCN) on the STG to extract comprehensive spatio-temporal features, which complement the appearance-based features and improve the model's discriminative ability. As shown in the experiments, our model performs better on multiple ReID video datasets than other state-of-the-art methods.

Ethic Statements

Our research method mainly focuses on video-based visual query tasks, however, visual query tasks can have various practical use cases such as video retrieval, autonomous vehicles, and sports analytics, etc. Due to the advantages of easy dataset acquisition and the maturity of previous applications, we have chosen person re-identification (i.e., person ReID) as a specific use case to experimentally validate our proposed method. The person re-identification technique can achieve cross-camera tracking of pedestrians between different cameras, therefore, our model will inevitably use real images and video information of pedestrians obtained from record equipment in real life, which may raise privacy and ethical concerns. Nevertheless, our research on person ReID technology can also have practical applications. For example, in retail, person ReID can be used to provide personalized shopping experiences by identifying customers and tracking their shopping behavior. In healthcare, person ReID can be used to track patients and staff within hospitals and other medical facilities. In event management, person ReID can be used to identify VIPs and authorized personnel and ensure their safety and security. Additionally, person ReID can also be used in smart cities to monitor traffic patterns, identify traffic violators, and manage traffic flow. Therefore, using person ReID technology correctly and in accordance with ethical standards can bring many benefits to people. To standardize our research process and potential future applications in ethical standards, we regulate our research from two aspects, data sources and data usage, in order to minimize the risk of indirect harm to others, and to comply with the international *ACM Code of Ethics and Professional Conduct*[1], aiming to be honest and trustworthy to the greatest extent possible.

Data Source: All ReID datasets used for our experiments already exist and are open source. Furthermore, these datasets have been widely used as benchmarks by many related studies in the ReID field over the past five years. Some datasets are directly obtained through open source interfaces of universities and institutions (e.g., Duke University[2]). And some datasets are from recently published well-known conference journals (e.g. ECCV, PAMI, Sprinter, etc.) whose authors or journals provide official open-source links for access. Although none of the data source authors have explicitly stated that their data is sourced through the ethic route, we assume that the datasets used in this research are collected ethically. In summary, all the data we use are open to the public, and all researchers have free access to these data. We never used self-recorded video or images from any surveillance equipment, nor did any volunteers participate in our dataset collection process.

Data Usage: On the other hand, we also insist on protecting personal privacy in the process of using data. Specifically, all pedestrians involved in the experimental data are anonymous, and their personal identities are only represented by ID numbers. All data does not provide accurate recording dates and locations. The time involved in the experiment is represented by the frame number of the video, and the shooting location is represented by the camera number of different positions. Through these measures, the personal privacy of the video characters is protected to the greatest extent possible.

[1]ACM Code of Ethics Conduct: https://www.acm.org/code-of-ethics

[2]Duke University Open Source Interface: https://exposing.ai/datasets/

Contents

1 Introduction **5**
- 1.1 Contribution . . . 7
- 1.2 Thesis Overview . . . 7

2 Related Work **8**
- 2.1 Image-based Person ReID. . . . 8
- 2.2 Video-based Person ReID. . . . 10
- 2.3 Background of Graph Convolutional Networks (GCN) . . . 12
  - 2.3.1 GCN Requirements . . . 13
  - 2.3.2 Graph Feature Aggregation for GCN . . . 13

3 Proposed Methods **16**
- 3.1 Overview of Our Model . . . 16
- 3.2 Local Feature Acquisition . . . 17
- 3.3 Feature Graph Structure . . . 17
- 3.4 Temporal Graph . . . 18
- 3.5 Spatial Graph . . . 19
- 3.6 Spatial-Temporal Graph & GCN Module . . . 20

4 Experiment **23**
- 4.1 Baseline . . . 23
- 4.2 Dataset . . . 23
- 4.3 Experiment Design . . . 24
- 4.4 Evaluation protocols . . . 25
- 4.5 Implementation Details . . . 28

5 Results and Analysis **29**
- 5.1 Comparison with the State-of-the-art Methods . . . 29

6 Conclusion **35**

7 Limitation and future work **36**

8 References **37**

Chapter 1

1 Introduction

Nowadays, in many vital locations, such as banks, schools, airports, factories, etc., for security reasons, a series of close-circuit television cameras (i.e., CCTV) need to be deployed for surveillance purposes, for example, in a busy airport, deploying multiple camera tracking systems with pedestrian re-identification can help police quickly target children in high-traffic public spaces if they get lost from their parents. However, due to the ever-expanding scale of deployed camera networks and the fact that multiple cameras usually provide video from different views simultaneously, this exceeds the ability of human operators to identify people's activities. Thus, this necessitates utilizing Automatic Tracking Systems (ATS) to tag people's behaviors [1]. However, ATS is usually limited by factors such as privacy protection, economic budget, geographical environment, etc., resulting in existing areas which cameras cannot cover. In this regard, people must be re-identified through multiple non-overlapping cameras, so-called person re-identification (ReID) [2]. Person re-identification is the task of detecting persons in a multi-camera system, where the system should identify persons who disappear from one camera view and reappear in another disjoint camera view.

Currently, the development of ReID is in two directions, namely image-based ReID and video-based ReID [3]. Image-based ReID relies on the appearance features of the image's content, such as the contour of the human body, the color of the cloth, etc., while ignoring the relations between images, for example, in successive images, a moving object contains continuous spatial changes over time. Thus, it usually underperforms when encountering occlusion or visual ambiguity cases. In contrast, video-based ReID is beneficial to exploit the richer complementary information between consecutive video frames to enhance model performance. However, existing video-based ReID research mainly focus on modeling the temporal relationship between frames without considering the spatial relationship of different body parts within or across frames, which may contain richer discriminative clues. Therefore, we focus on developing a method that simultaneously considers the temporal relationship across frames and the spatial relationship of corresponding parts within and across frames.

Figure 1 presents two common problems of ReID video, the goal is to identify the person in a series of video frames. Figure. 1.a shows a case of occlusion problem. In a ReID video, it is common to encounter pedestrians occluded by objects and reappearing in subsequent video frames, meaning different video frames can provide complementary information. Therefore, if a series of video frames can be modeled instead of each frame, the occlusion problem can be alleviated by exploiting the temporal relationship of the same body part between frames.

However, only considering temporal relationships across frames is insufficient to deal with the problem of visual ambiguity, which case is shown in Figure 1.b and Figure 1.c, two people have a similar visual appearance that would cause the ReID model not to distinguish them. Most image-based ReID models can easily distinguish different people with large differences in appearance, such as the case of Figure 1.a and Figure 1.b. Nevertheless, it is hard to distinguish different people with similar appearances, such as in the case of Figure 1.b and Figure 1.c, as both people have very similar visual appearances, resulting in the appearance difference may

insufficient for the model to make the judgment. However, it can also be observed that the structural information of their bodies is different (e.g., the body shape), so if the spatial relationship of different body parts can be used as a complementary clue for appearance features, it can help alleviate the problems caused by visual ambiguity.

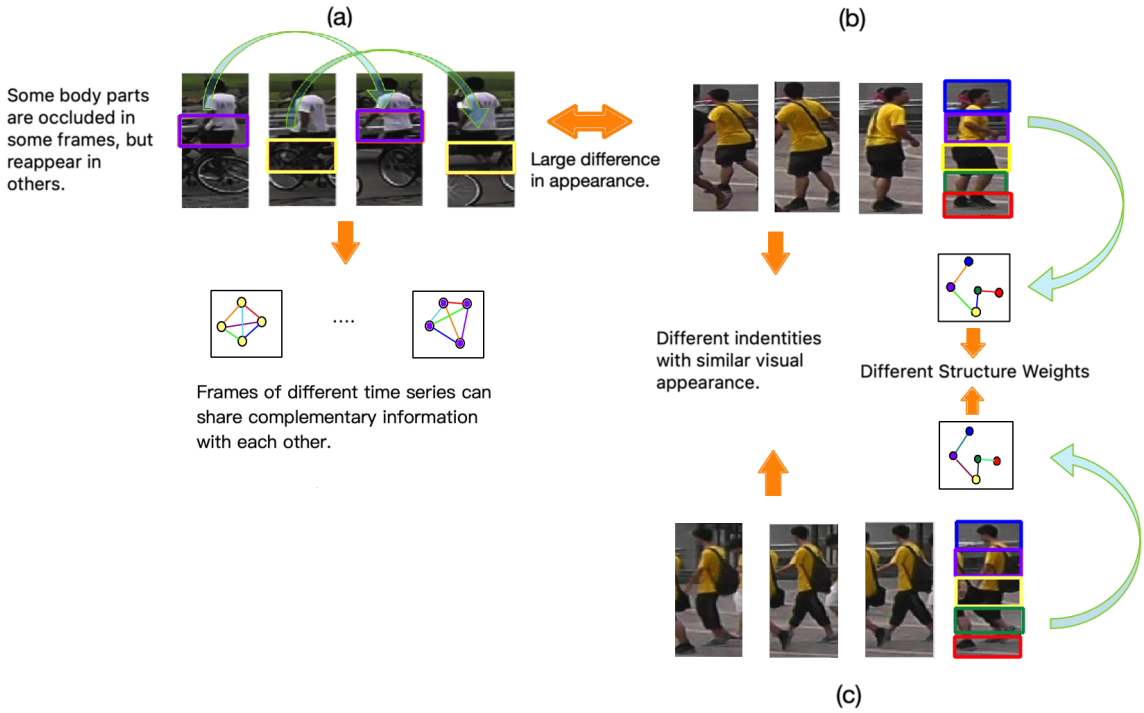


Figure 1: Three video sequences of three different identities on the MARS dataset. (a) shows the occlusion case that some body parts are occluded by obstacles in some frames but reappear in other frames. Using a graph to embed the body feature, the same body part in different frames can mutually provide complementary knowledge, thereby alleviating the occlusion. (b) and (c) show the visual ambiguity of two persons with similar visual appearances, making it difficult for most ReID models to distinguish. However, their body's structural information is different, so the weights (i.e., cosine similarity) of different body parts have a large gap that can be used as a complementary clue to alleviate the visual ambiguity.

In existing novel work, most methods address the problem of occlusion or visual ambiguity through modeling temporal or spatial relations independently, such as the *Temporal Knowledge Propagation (TKP)* [3] builds a shared network space to accumulate temporal complementary information between consecutive frames to alleviate information asymmetry caused by image occlusion. In addition, the *Relational Network* [2] adopts a video frame splitting method, which splits the video frame into multiple segments and models the interrelationships between local features to obtain additional spatial clues, which are then used to alleviate the visual ambiguity problem. These two works have achieved good results in solving either occlusion problems or visual ambiguity, our method can continue to follow their idea. However, it still lacks an approach to combine spatial and temporal clues to solve occlusion and visual ambiguity jointly. After exploration, we find that *Space-Time Region Graphs* [4] shows a successful application of GCN for video classification, which uses a space-time region graph to represent videos and GCN to capture the object's spatial state changes over time, this makes

it possible to use graphs to jointly represent temporal and spatial information. Inspired by the three works [2, 3, 4] stated above, we propose a new method to use a space-time graph (STG) to represent a video sequence and use GCN to model the relation of different body parts across frames. Specifically, given a video, we first extract the feature maps for each video frame and split each feature map into several segments. Then we build the STG by connecting the same body part across frames to obtain a temporal clue and connecting different body parts within a frame to obtain a spatial clue. Finally, GCN is used to extract overall spatiotemporal features on STG and use it to alleviate occlusion and visual ambiguity in ReID tasks.

1.1 Contribution

In summary, our contributions are the following. (1) We propose a new video-based person reidentification method that uses a space-time graph (STG) to represent a video sequence and models the spatial and temporal relations of different body parts intra-frame and inter-frames. (2) Our method considers temporal and spatial relations jointly, which dedicates to alleviating occlusion and visual ambiguity problems. (3) We validate our method on diverse datasets, i.e., two general video-based person ReID datasets and three partial/occluded ReID datasets. (4) We compare our method with seven baseline works in recent years, and use three types of metrics i.e., Rank-N accuracy, Cumulative Match Characteristic (CMC) curve and mean average precision (mAP) for evaluation. The experiments show that our proposed method outperforms existing state-of-the-art methods.

1.2 Thesis Overview

This thesis is divided into six chapters. Chapter 2 summarizes the methods used by different ReID types and gives the representative works through extensive literature reviews. Besides, we also conclude the advantages and disadvantages of different methods, thereby bringing the necessity of our method. Finally, we elaborate on the background mechanism of the graph convolutional networks (GCN), which is the core module used in our method. In Chapter 3, we elaborate on the method we propose in the paper and give detailed design ideas for Temporal Graph, Spatial Graph, and how to use GCN to capture the overall spatiotemporal features and then use them for the ReID task. In Chapter 4, we first introduce the baseline work, then gives the dataset information used in our experiments. Besides, we specify the experiments design and evaluation metrics and provide the detailed process of the experiments set up. In Chapter 5, we sorted out the experimental results and analyzed and compared the experimental results. In the final Chapter, we summarize our experimental conclusions and our contributions, then we also discuss possible directions for future research. as well as future work.

2 Related Work

This chapter summarizes two classes of person ReID methods, namely image-based ReID and video-based ReID, and presents representative works through an extensive literature review. In addition, the advantages and disadvantages of different methods are summarized, thus suggesting the necessity of our method. Finally, we elaborate on the underlying idea of the graph convolutional networks (GCN), which is the core module used in our model.

2.1 Image-based Person ReID.

Image-based ReID methods take several independent images into account. Image-based methods focus on using convolutional neural network (CNN) to obtain image features for re-identifying a person. The developing of image-based methods are divided into two categories, i.e., image classification and distance metric learning [5]. In Table 1 at the end of this section summarize the advantage and disadvantage for each category.

Image Classification methods [6, 7, 8, 9, 10] are inspired by image processing task, which treats person ReID problem as image classification problem, that is, the pedestrian's ID is used as a label to annotate the data, and the image classification method is used to classify pedestrians [5]. Specifically, similar to most deep learning-based image processing tasks, these ReID methods first use convolutional neural network (CNN) to extract features from image data, and then pass the obtained image features through a fully connected layer (FC) with softmax and cross-entropy loss for learning, and the final output is the prediction of a probability distribution of the image ID.

SDALF [6] is a person classification work that uses CNNs to extract local features on the chromatic content of images to predict person IDs. But some papers [7, 8, 9] reckon that learning appearance features only is insufficient to obtain a generalized model. Hence, in addition to appearance features, they marked some auxiliary attributes for the person in images, such as gender, hairstyle, clothing, etc. So, the classification loss is defined as $L = \lambda L_{ID} + \frac{1}{m}\sum L_{mi}$, where L_{ID} is the prediction loss of person IDs and $\frac{1}{m}\sum L_{mi}$ is the average loss of all attributes. By introducing these auxiliary attributes, the network can predict not only the pedestrian's IDs but also various pedestrian attributes to improve the model's generalization.

In recent years, there are many person ReID works based on image classification. The advantage of image classification is that the method is robust, the training is stable, and the results are easier to reproduce. However, a shortcoming is that the feature learning is easy to overfit on the dataset's domain when the training ID increases to a certain extent because the FC layer dimension of ID loss is consistent with the number of IDs [5]. Hence, when the training set is too large, the network is huge, and the training is difficult to converge.

Metric Learning methods focus on constructing distance metrics from raw image data, and the learned distance metric is then used to verify if the given images belong to the same identity. Specifically, if providing the neural network two images I_1 and I_2, after feature learning by the CNN, the normalized feature vectors obtained are represented as fI_1 and fI_2. Then, the distance of the feature vectors of the two images is defined as $d(I_1, I_2) = ||fI_1 - fI_2||_2$. Metric learning aims to minimize the distance between the same pedestrian images (positive

sample pairs)and maximize the distance between different pedestrian images (negative sample pairs). The commonly used metric learning methods on image-based ReID tasks are Contrastive loss [11], Triplet loss [12, 13, 14] and Quadruplet loss [15] etc.

Siamese Network [11] is a work with contrastive loss, and the method consists of two subnetworks. The input of Siamese network is a pair (two) images I_a and I_b, these two images can be the same person or different person. Each pair of training image has a label y, where $y = 1$ means two image belong to the same identity (positive sample pair), otherwise $y = 0$ means they belong to different identities (negative sample pair). Then, the contrastive loss function of the Siamese network is written as $L_c = yd^2_{I_aI_b} + (1-y)(\alpha - d_{I_aI_b})^2_+$, where $d_{I_aI_b}$ is the distance of the feature vectors of the two images, $(z)_+$ means $max(0, z)$, and α is a threshold value that based on actual needs. To minimize the loss function, when the network inputs a pair of positive samples, the $d(I_aI_b)$ will gradually decrease, that is, pedestrian images with the same ID gradually cluster in the feature space. Conversely, when the network inputs a pair of negative samples, the $d(I_aI_b)$ gradually increases until it exceeds the threshold α. By minimizing the loss function L_c, the distance between the positive sample pairs becomes smaller, and the distance between the negative sample pairs becomes larger to meet the verification needs of person ReID tasks.

The original triplet loss function [12, 13] requires three input images, i.e., a fixed image (Anchor) a, a positive sample (Positive) p, and a negative sample (Negative) n. Image a and image p are a pair of positive samples, and image a and image n are a pair of negative samples. So the triplet loss function is expressed as $L_t = (d_{a,p} - d_{a,n} + \alpha)_+$, where $(z)_+$ means $max(0, z)$, and α is a threshold setting indicates the smallest difference between $d(a, p)$ and $d(a, n)$. However, the work of Triplet CNN [14] thought the original triplet loss has the disadvantage that it only considers the relative distance between positive and negative samples without considering the absolute distance between them, which causes the model confuse positive and negative clusters with close distances. Therefore they proposed an improved triplet loss as $L_it = d_{a,p} + (d_{a,p} - d_{a,n} + \alpha)_+$. The formula adds a $d_{a,p}$ term to ensure that the network can pull the instances of the same person closer and, simultaneously, push the instances belonging to different persons farther from each other in the learned feature space.

Both image classification and metric learning use features learned on images to compute a similarity loss. The main difference between image classification and metric learning is in the loss function. For image classification methods, the learned features are passed to a fully connected layer with softmax and cross-entropy for predicting categories, but for metric learning methods, the learned features are used for distance-based loss function to calculate similarity, not a fully connected layer is required. Compared with image classification, the advantage of metric learning is that its network structure is simpler. After using CNN to capture image features, these features are used for distance metric learning instead of linking to an FC layer. Since there is no FC layer, the metric learning network size is independent of the training set size, so the performance is insensitive to increased pedestrian IDs. The disadvantage of the metric learning method is that the image features extraction by CNN still relies on single image without considering the connection between relevant images, which is hard to deal with occlusion or visual ambiguity.

Category	Image-based ReID	
Methods	**Image Classification (IC)**	**Metric Learning (ML)**
Summary	Use CNN to extract image features for person classification & verification task.	Establish distance metric to learn similarity between images.
Pros	1. Easy to reproduce. 2. Training is stable.	1. Simple network structure. 2. Insensitive to increasing number of IDs.
Cons	1. Easy to overfit when IDs increase. 2. Rely on the appearance features of single image without considering continuity between images.	Rely on the appearance features of single image without considering continuity between images.

Table 1: Comparison of Two Types of Image-Based ReID Methods.

2.2 Video-based Person ReID.

Video-based ReID methods take a sequence of video frames as input, different video frames are temporally continuous with each other. In addition to appearance features, video-based ReID methods exploit diverse relations between consecutive video frames to provide supplemental clues for re-identifying a person. The development of video-based methods is divided into three categories,i.e., optical flow, temporal pooling, and graph representation [5]. In Table 2 at the end of this section, summarize the advantages and disadvantages for each category.

Optical flow. In computer vision domain, optical flow is a term to describe the instantaneous velocity of pixel motion for a moving object on the viewing plane. In video-based ReID domain, the optical flow methods use the changes of pixels in the image sequence and the correlation between adjacent frames to find the motion clue of pedestrians between context frames [5].

AMOC [16] is a video-based ReID work using optical flow, it consists of a spatial network (Spat-Net) and a motion information network (Moti-Net). The Spat-Net is to obtain global content features on each frame of a given video sequence. The Moti-Net is to extract optical flow features for every two adjacent frames. Then the spatial and optical flow features are fused and input to a recurrent neural network (RNN) to extract temporal features. Through the AMOC network, each video sequence can extract a feature that combines both image content information and motion information. Finally, by employing classification loss and contrastive loss to train the model to improve person ReID performance.

The merit of the optical flow method is that both the content and the motion information of the people in the video sequence are considered. In addition, the optical flow method is sensitive to pixel movement and has an excellent performance in posture changes and action recognition [5]. However, optical flow information is computationally expensive because all pixels in a video sequence are traversed to extract optical information. Besides, the optical flow method requires that the pedestrian movement process cannot be interrupted. Once an object is occluded, the optical flow information will be lost, so the optical flow method is invalid when facing the occlusion problem.

RNN Temporal Pooling methods [17, 18, 3, 19, 20] focus on exploit temporal relation between consecutive video frames to provide supplemental clues for re-identifying a moving

person.

RNN-reID [17] is a representative video-based ReID work, which uses CNN incorporated with RNN to obtain the temporal information between video frames to improve model performance. First, a CNN extracts a time-step feature for each video frame, then all time-step features of a video sequence are connected to a recurrent layer to share the information over time. Before output, all time-steps features are summarized into a single feature vector by temporal pooling. The overall single feature vector obtains both the appearance features of a single frame and temporal features of frame-to-frame. Hence the overall feature is used to train the network to improve model performance in identifying moving pedestrians.

However, as mentioned in [18], the recurrent neural network has the limitation of extracting temporal information for the video sequence due to its' complex structure. Hence, instead of using RNN to obtain temporal relation, the work of TKP [3] is accumulating complementary temporal information between consecutive frames by constructing a shared network space. Besides, different from using RNN to assign the same weights to all video frames, some state-of-art works of TCRL [19] and TVCAN [20] use spatial and temporal attention to learn the weights of different frames to obtain a weighted temporal clue for re-identifying a moving person.

The advantage of the temporal pooling method is that in addition to appearance features, temporal pooling utilizes the temporal relationship between video frames to obtain pedestrian motion clues, thus enhancing the performance of video-based ReID for a moving person or occlusion problem. However, as mentioned in the introduction section, existing temporal pooling methods only consider obtaining temporal clues across frames, which are beneficial for solving occlusion problems, but it's hard to deal with visual ambiguity cases because two people with similar appearances cannot be distinguished only from time clues, key information needs to be obtained from the spatial structure of the human body.

Graph Learning methods [21, 22, 23] focuses on using a structural graph to represent the key points (skeleton) of the human body, and then uses a graph neural network (GNN) to obtain a template relation of the skeleton graph to predict the human shape for those occluded video frames.

ADGC-CGEA [22] is video-based ReID work using graph learning. It addresses the occlusion problem by learning the topology relation of the human body. In detail, given two video sequences, the framework uses three modules, i.e., semantic module S, relational module R, and topology module T, to verify if the people in these two videos are the same. Firstly, the semantic module S learns the local features of the human skeleton point area from each video frame, then the local features are seen as nodes to form a directed graph. After that, the relational module R uses an adaptive direction graph convolutional (ADGC) layer to propagate the relation in the graph. Finally, the T module uses a cross-graph embedding-alignment (CGEA) layer to learn the topological relation of two graphs obtained in two video frames and do similarity matching to verify whether they belong to the same identity.

The advantage of the graph learning methods of video-based ReID is that in addition to the temporal relation across video frames, it also considers using the structure relation of the

human body to provide the discriminative clue for identifying a person, especially for the occlusion problem. However, the skeleton-based graph comprises key points representing human body structures. The extraction of key points only considers the spatial location of different human body parts, ignoring that two identities of different body shapes can have similar skeleton graphs, thus generally underperforming in visual ambiguity cases.

Category	**Video-based ReID**		
Methods	**RNN Temporal Pooling**	**Optical Flow**	**Graph Learning**
Summary	CNN for appearance feature + RNN for temporal feature.	Capture motion clues based on pixel instantaneous changes between consecutive frames.	Temporal relation across frames + Global & local spatial relation across a body graph.
Pros	Temporal features across frames provide complementary clues to appearance features.	1. Both appearance feature and motion feature are taken into account; 2. Good performance in people posture change and action recognition;	1. Both the spatial and temporal features provide complementary clues to appearance features. 2. Good performance in occlusion.
Cons	1. RNN limited in extracting temporal feature; 2. Hard for visual ambiguity.	1. Sensitive to occlusion; 2. Optical flow extraction is computationally intensive;	1. Skeleton graph has limitations in representing human body shape. 2. Hard for visual ambiguity.

Table 2: Comparison of Three Types of Video-Based ReID Methods.

In summary, the image-based ReID methods rely on the appearance features of images but do not consider the continuous information between related images, and the performance is limited when encountering the occlusion problem. In addition to the appearance feature of the video frames, Video-Based ReID methods also consider the use of temporal information between video frames to provide supplementary clues, which is beneficial for dealing with occlusion problems. However, existing video-based methods do not consider the relationship between different body parts across frames, which may effectively alleviate the problems of both occlusion and visual ambiguity.

Compared to these methods, we propose a new video-based person re-identification method that uses a space-time graph (STG) to represent a video sequence and jointly models the spatial and temporal relations of different body parts intra-frame and inter-frames. Moreover, the spatio-temporal relation learned by our method contains robust and discriminative clues for the whole video sequence, effectively alleviating the problem of occlusion and visual ambiguity.

2.3 Background of Graph Convolutional Networks (GCN)

In our proposed method (see Chapter 3), since we need use GCN to capture spatiotemporal information on the built spatiotemporal graph, which used for re-identify person. Hence, it is necessary to give a background introduction about GCN in this section.

A graph structure or topology is irregular data because the shape structure of a graph is irregular and can be regarded as infinite-dimensional data [24]. The structure around each node is unique within the graph which makes the traditional CNNs and RNNs lose performance in capturing features on a graph. Currently, an effective graph feature extractor is

Graph Convolutional Networks (GCN) which was proposed by Kipf & Welling in 2017 [24]. GCN is an improved version of Graph Neural Networks (GNN). Compared to GNN, GCN lead to attention and convolution mechanisms, which can capture more targeted graph features.

2.3.1 GCN Requirements

Formally, given a graph $G = (V, E)$, the GCN model requires 3 inputs :

- a $N * N$ adjacency matrix A (N is the number of nodes);
- a $N * D$ feature matrix X (D is the number of features at each node);
- a $N * E$ binary label matrix B (E is the number of classes);

The output of the GCN is :

- a $N * F$ node-level feature matrix Z (F is the number of output features per node);

In the GCN, a hidden layer is expressed as $H^{(i+1)} = f(H^i, A))$, where $H^0 = X$, $H^L = Z$, i denotes current layer, L is the number of layers, and f denotes the propagation rule. Each hidden layer H^i outputs a feature matrix Z^i used as input to the next layer $H^{(i+1)}$. As the number of learning layers deepens, the initial feature matrix X becomes increasingly abstract and eventually becomes a high-level abstract feature matrix Z, which can be used for the goal tasks.

2.3.2 Graph Feature Aggregation for GCN

In this subsection, we provide a brief background introduction about the graph feature aggregation that used by GCN. We define a simple $f(H^i, A)$ function as the base network layer, where H^i is the i^{th} hidden layer, A is the adjacency matrix of the graph, which mentioned in section 2.3.1. Then, we adopt a simple layer-wise propagation rule:

$$f(H^{(i)}, A) = \sigma(AH^{(i)}W^{(i)}) \tag{1}$$

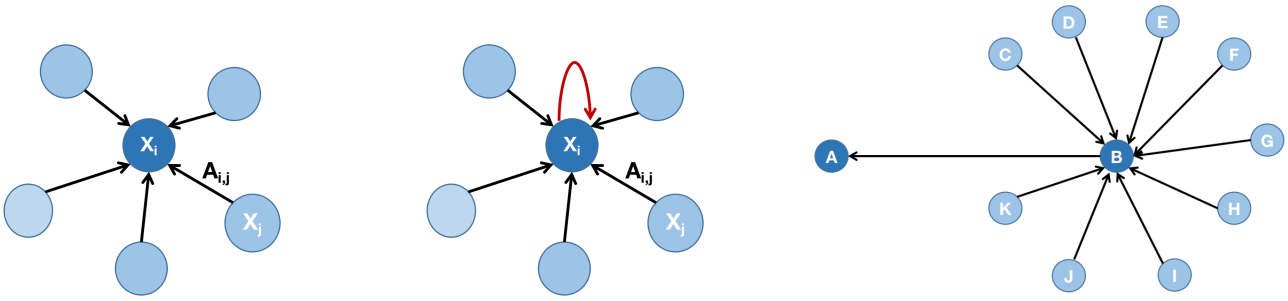
where $W^{(i)}$ is the weight matrix for layer i and σ is a non-linear activation function such as the ReLU function. Based on above the first layer, i.e., $i = 0$ can be written as:

$$f(H^{(0)}, A) = \sigma(AH^{(0)}W^{(0)}) = \sigma(AXW^{(0)}) \tag{2}$$

In Equation (2), the multiplication of the adjacency matrix A and the initial feature matrix $X = H^{(0)}$ represents that each node in the graph aggregates the features from its neighbors. More specifically, the feature aggregation process can be denoted as follows:

$$aggregate(X_i) = \sum_{j \in neighbor(i)} A_{i,j}X_j \tag{3}$$

The process of Equation (3) is shown in Figure. 2a, which demonstrate the classic feature aggregation method. In the classic aggregation method, the features of each neighbor X_j are all aggregated to the target node X_i, and $A_{i,j}$ is the corresponding element of the adjacent matrix A.
However, according to [24], there are two defects in this classical aggregation method:



(a) Mean-rule Aggregation (b) Aggregate with Self-loop (c) Aggregating Distortion

Figure 2: Feature Aggregation

1. The target node only aggregates the feature of its neighbors but ignores aggregating its own feature.

2. Different nodes have different degrees (the number of neighbors is different), as shown in Figure. 2c. Because the neural network is sensitive to the scale of the data input, it causes the gradient explosion or vanishing.

For defect 1, Kipf Welling proposed the self-loop solution, as shown in Figure. 2b, by adding an identity matrix I to the adjacency matrix A to aggregate the feature of the target node itself, i.e., $\widetilde{A} = (A + I)$. Thus, the aggregation Equation (3) can be transcribed as:

$$\begin{aligned} aggregate(X) &= \widetilde{A}X = (A + I)X = AX + X \\ aggregate(X_i) &= \sum_{j \in neighbor(i)} A_{ij}X_j + X_i \end{aligned} \tag{4}$$

In essence, Equation (4) can be regarded as using the Sum Rule, each neighbor contributes its feature to the target node, hence the sum feature of the target node is exclusively determined by the neighbors.
For defect 2, to make the neural network insensitive to the input data scale, the input data (i.e., feature representation) are normalized by multiplying the adjacency matrix A with the inverse degree matrix D^{-1} [24]. Thereby the feature aggregation process can be expressed as:

$$\begin{aligned} aggregate(X) &= D^{-1}\widetilde{A}X \\ aggregate(X_i) &= \sum_{j=1}^{N} \frac{\widetilde{A}_{ij}}{D_{ii}} X_j \end{aligned} \tag{5}$$

In essence, Equation (5) can be regarded as using the Mean Rule after Sum Rule, the final aggregation features of the target node need to be averaged according to the degree of its nodes.

As the method of classic graph feature aggregation, Equation (5) solves the gradient explosion but brings the feature fusion distortion. Since the contribution of each node to feature aggregation is determined by its degree, as shown in Figure. 2c, the degree of node A is 1 (only one neighbor node B), but the degree of node B is 10, therefore when adapting the feature aggregation, the feature contribution of A only relies on B, but A only contribute a small part to B, which resulting in feature distortion. To tackle the distortion problem, Kipf & Welling

proposed a spectral rule-based method that takes into account both the degree of the target node and its neighbors, the spectral rule-based equation is written as:

$$\begin{aligned} aggregate(A, X)_i &= D^{-0.5}\widetilde{A}_i D^{-0.5} X \\ &= \sum_{j=1}^{N} \frac{1}{\sqrt{D_{ii} D_{jj}}} \widetilde{A}_{ij} X_j \end{aligned} \tag{6}$$

In Equation (6), we take Figure. 2c as an example, it can be seen from the deduction result that the degree value of D_{ii} for Node A is equal to 1, and the degree value of D_{jj} for Node B is equal to 9. The -0.5 power of both can effectively reduce the aggregation effect of B on A, to solve the problem of distortion of feature aggregation in unbalanced graphs.

Finally, retrospect to the neural network expression as Equation (2) and replace the first layer input (feature matrix X) in Equation (7) with a multi-layer network $H(i)$, the formula of Spectral Rule-based GCN can finally be expressed as:

$$H^{(i+1)} = f(A, H^{(i)}) = \sigma(D^{-0.5}\widetilde{A}_i D^{-0.5} H^{(i)} W^{(i)}) \tag{7}$$

where $w^{(i)}$ is the weight matrix for layer i and σ is a non-linear activation function such as the ReLU function. In the method part of Chapter 3, we use the Spectral Rule-based GCN, i.e., Equation (6) to capture features from spatiotemporal graphs for person re-identification task.

3 Proposed Methods

In brief, our method aims to model the temporal and spatial information in a video jointly. The temporal features are utilized to alleviate occlusion problems, while the spatial features are employed to alleviate visual ambiguity problems. To achieve this, we first adopt the idea proposed in literature [2] to locally segment the appearance features of each video frame into patches. Then, inspired by literature [4], we construct a spatiotemporal graph (STG) by treating these patches as nodes, to integrate the temporal and spatial information. Specifically, a spatiotemporal graph is composed of temporal graphs and spatial graphs. The temporal graph is constructed by connecting patches of the same part across consecutive frames, while the spatial graph is formed by connecting the patches in a top-down order within each frame. Finally, we employ GCN to extract a holistic spatiotemporal feature from the constructed STG, which is utilized to train the recognition model.

In the following sections of this chapter, we will first provide an overview of our model in Section 3.1. Subsequently, we will gradually introduce the details of our method in the following sections. Specifically, Section 3.2 will elaborate on the process of acquiring appearance local features of video frames, Section 3.3 will describe the process of constructing feature map structure based on local features, Section 3.4 and Section 3.5 will respectively provide details on the construction of temporal and spatial graphs. Finally, in Section 3.6, we will present the combination of temporal and spatial graphs, as well as the method of extracting spatiotemporal global features using GCN.

3.1 Overview of Our Model

The overview of our proposed STG-GCN model is shown in Figure 3. In summary, the whole model is divided into two main modules: Module 1 aims to extract the spatiotemporal features of the video; Module 2 uses the spatiotemporal feature obtained from Module 1 to train the neural network and perform classification task predictions on the testing set video.

Module 1: Module 1 aims to extract spatiotemporal features of videos. Specifically, The input to module 1 is a video consisting of different people in different camera views for training purposes. First, the STG-GCN model uses the CNN backbone to extract a feature map for each video frame, divides each feature map horizontally into several segments, and performs an average pooling operation on each feature segment to obtain each feature vertex. Next, after obtaining all the feature vertices, we leverage the Euclidean similarity to calculate their mutual relationship based on specific spatial and temporal clues, thereby forming an overall spatiotemporal graph (STG). Finally, we use GCN with max pooling on the STG to extract the comprehensive spatiotemporal feature f^{st}.

Module 2: Module 2 uses the spatiotemporal feature f^{st} obtained from Module 1 to train and fine tune the neural network and perform classification task predictions on the testing set video. Specifically ,the obtained spatiotemporal features f^{st} are first batch-normalized and input to the fully connected layer, and finally use the softmax as activation function and cross entropy as loss function to make classification predictions for the person on the test set video.

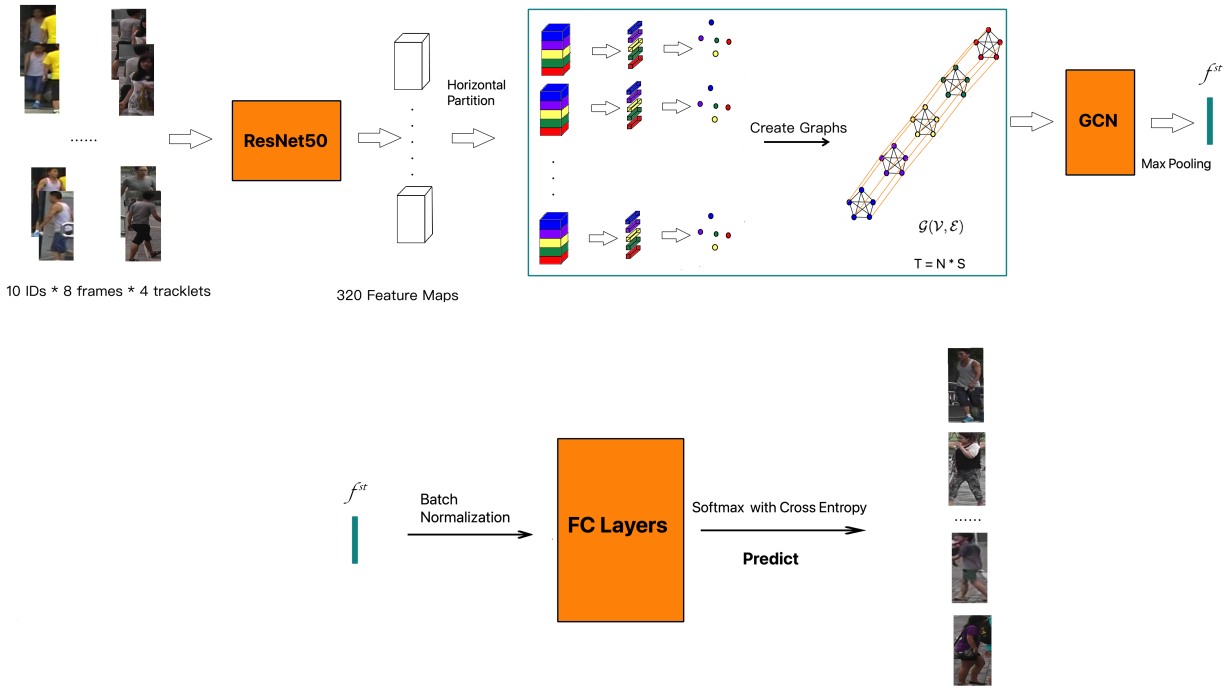


Figure 3: The overall model of the proposed methods

3.2 Local Feature Acquisition

Our model aims to learn the relation of different body parts across frames, so the first step is to split different body parts of each frame, then study the relationship between those body parts. To achieve this, we refer to a segmentation method based on local feature learning proposed in the literature [2], which divides each video frame into several segments according to body parts. Specifically, when given a video sequence, we denote it as $V = \{F_1, F_2, ..., F_N\}$, where F_i represents a specific frame and N is the number of video frames sampled. Then we use the CNN backbone (ResNet-50) to capture a feature map for each video frame, so as the feature maps of a video sequence are denoted as M,

$$M = \{M_1, M_2, ...M_N\} \tag{8}$$

where $M_i \in \mathbb{R}^{h \times w \times c}$ represents the feature map of the i-th frame in the video sequence, h, w, c denote the height, width and image channels of a frame, respectively. After that, each feature map M_i is horizontally split into J equal-sized segments that denoted as $S = \{S1, S2, ...S_J\}$. Then we use average pooling to obtain a local feature vector for each segment, denoted as $x_i \in \mathbb{R}^c, i = 1...T$. Accordingly, the number of local features of a video sequence is $T = N * J$.

3.3 Feature Graph Structure

After splitting the body parts to obtain the local features, the next step is to study the relation of those independent features within a frame or across frames. To achieve this, we refer to a method of Space-Time Region Graphs proposed in literature [4], which dedicates to connect the local features with a graph structure, then use a graph-based feature extractor i.e., Graph Neural Network (GCN) [24] to study the relation on the built graph. Specifically, we first

use the segmenting local features as vertices to establish a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the graph contains T number of vertices denoted as $v_i \in \mathcal{V}$ and the edge of two vertices are represented as $e_{ij} = (v_i, v_j) \in \mathcal{E}$. In the graph, each local feature is regarded as a vertex, the edges between vertices represent the relations between local features. Computationally, we calculate the cosine similarity of two local features to represent a edge of the graph:

$$e(x_i, x_j) = \cos(\mathbf{x_i}, \mathbf{x_j}) = \frac{\sum_{i=j=1}^{n} \mathbf{x}_i \mathbf{x}_j}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2} \sqrt{\sum_{j=1}^{n} (\mathbf{x}_j)^2}} = \frac{\mathbf{x_i} \cdot \mathbf{x_j}}{\|\mathbf{x_i}\| \times \|\mathbf{x_j}\|} = \frac{\phi(x_i)^T \phi(x_j)}{\|x_i\| \|x_j\|} \quad (9)$$

In Equation 9, ϕ represents a linear transformation for local feature x. Specifically, $\phi(x) = \mathbf{w}x$, the parameter $\mathbf{w}$ is a $d \times d$ dimensional weight matrix that can be trained by backpropagation. It enables the model to choose and learn the relevance between distinct local features inside a frame or across frames in adaptive manners (the details of these manners are described in Section 3.5 and Section 3.4).

In actual computation, we use an adjacency matrix $A \in \mathbb{R}^{T*T}$ associated with $\mathcal{G}$ to store the pairwise relationships of the local features. In matrix A, and each element A_{ij} is numerically equal to $e(x_i, x_j)$, which reflect the correlation between vertices x_i and x_j. Motivated by [25, 26], to keep adjacency matrix values at the same scale, we normalize the A on the basis of the following two rules: 1) In matrix A, the sum of all elements (i.e., edge values) for each row should be 1; 2) The edge values (i.e., cosine similarity between vertices) are all non-negative values in a range of (0,1); Thereby we obtain matrix A by:

$$A(i, j) = \frac{e^2(x_i, x_j)}{\sum_{j=1}^{T} e^2(x_i, x_j)} \quad (10)$$

After the feature graph is established, GCN is used to learn the relational features of the graph structure. However, as mentioned in [24], when traditional GCNs learn features for the graph with unbalanced vertices (e.g., some vertices have many connection neighbors but some vertices only have one neighbor), it has a distortion problem of feature aggregation. To release this distortion problem of unbalanced graph nodes, a spectral-based GCN is used in stead of traditional GCN [24]. Therefore, we transform adjacency matrix A to match spectral-based GCN, i.e., convert A to $\widetilde{A} = A + I_n$ to represent a self-loop adjacency matrix, where $I_n \in \mathbb{R}^{N \times N}$ is the identity matrix. Then, leveraging the symmetric normalization trick to approximate the Laplace graph:

$$\widehat{A} = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}. \quad (11)$$

In Equation. (11), $\widetilde{D}$ is the diagonal node degree matrix of $\widetilde{A}$ i.e. $\widetilde{D}_(i, i) = \sum_j \widetilde{A}_(i, j)$. Eventually we create an adjacency matrix $\widehat{A}$ based on $\mathcal{G}(\mathcal{V}, \mathcal{E})$ containing the correlated information between local features.

3.4 Temporal Graph

As mentioned in Chapter 1, for a video sequence, temporal complementary information across frames can provide additional discriminative clues for each other. So, if temporal supplementary information across frames can be shared, the problem due to occlusions or background noise can be alleviated. To achieve this, in our method, after obtaining local features through feature segmentation (mentioned in Section 3.2), we share temporal information across frames

by building a Temporal Graph (TG) module.

Specifically, as shown in Figure. 4, a video is divided into N independent frames by time, each frame is divided into J segments, and the local features x from all segments in a video sequence are used to construct the temporal graph $\mathcal{G}^t(\mathcal{V}^t, \mathcal{E}^t)$, where $\mathcal{G}^t$ denotes the temporal graph, and $\mathcal{V}^t$ represents the vertices set of the temporal graph, i.e., $\mathcal{V}^t = \{x_1, x_2, ...x_T\}$, and $\mathcal{E}^t$ denotes the edge set of the temporal graph, i.e., $e_{ij} = (x_i, x_j) \in \mathcal{E}^t$, the value of e_{ij} is calculated by cosine similarity of two local features x_i and x_j which is mentioned in Section. 3.3. In order to capture the temporal information across frames, we use Equation (9) to calculate pairwise relational values of local features of the same segment position across frames, that is, to build edges e_{ij} between two vertices x_i and x_j. And the corresponding adjacency matrix $\widehat{A}^t$ is calculated by Equations (10) and (11).

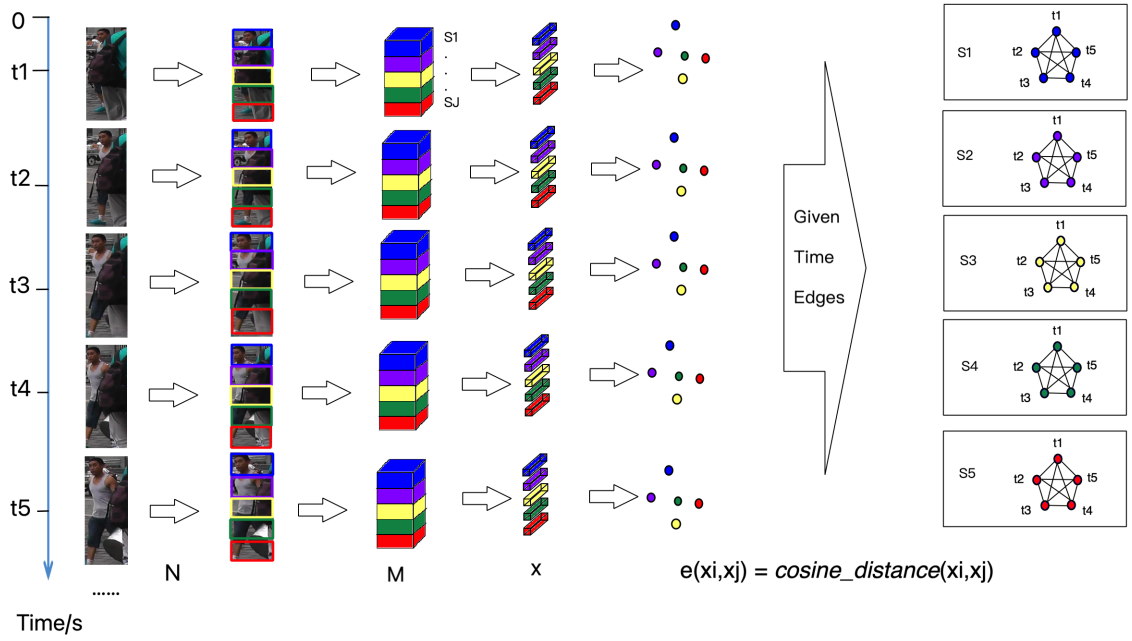


Figure 4: Temporal Graph (TG) construction

3.5 Spatial Graph

As mentioned in Chapter 1, only modeling temporal relations across frames is insufficient to deal with the problem of visual ambiguity (i.e., two identities with similar appearance). However, for two persons with similar visual appearance, the spatial structure relation of different body parts for each person is different. Therefore, if the body structure information can be shared as supplementary information for appearance, the problem of visual ambiguity can be alleviated. To achieve this, in our method, after obtaining local features through feature segmentation (mentioned in Section 3.2), we study structure information within a frame by building a Spatial Graph (SG) module.

As shown in Figure. 5, the structure of the Spatial Graph (SG) module is different from the structure of the Temporal Graph (TG) module. TG uses local features of the same body parts in all frames to construct one graph for studying supplementary temporal information across frames. In comparison, SG uses local features of adjacent body parts in the same frame to build multiple spatial graphs for studying supplementary structural information within a frame, therefore, each frame has one spatial graph.

Specifically, a given video is divided into N frames by time, then each frame is used to build a spatial graph $\mathcal{G}^s$ where s represents the meaning of spatial. So a spatial graph of i-th frame is denoted as $\mathcal{G}_i^s(V_i^s, \mathcal{E}_i^s)$, where $\mathcal{V}_i^s$ is the vertices set, i.e., $\mathcal{V}_i^s = \{x_{i,1}, x_{i,2}, x_{i,3} \dots x_{i,J}\}$, as mentioned in Section 3.2, J as subscript here represents the feature map M_i is horizontally split into J equal-sized segments so as after average pooling, a spatial graph contains j local features. In order to capture the spatial information of a person's body parts within a frame, we use cosine similarity to calculate the relational value of adjacent body parts according to the body structure of the person from head to foot, that is, to sequentially build edges $\mathcal{E}_i^s$ for two neighboring local features of $\mathcal{V}_i^s = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots x_{i,J}\}$ for each video frame, thus, finally obtained N number of spatial graphs. And the corresponding adjacency matrix $\widehat{A}^s$ for each $\mathcal{G}_i^s(V_i^s, \mathcal{E}_i^s)$ can obtained by Equations (10) and (11).

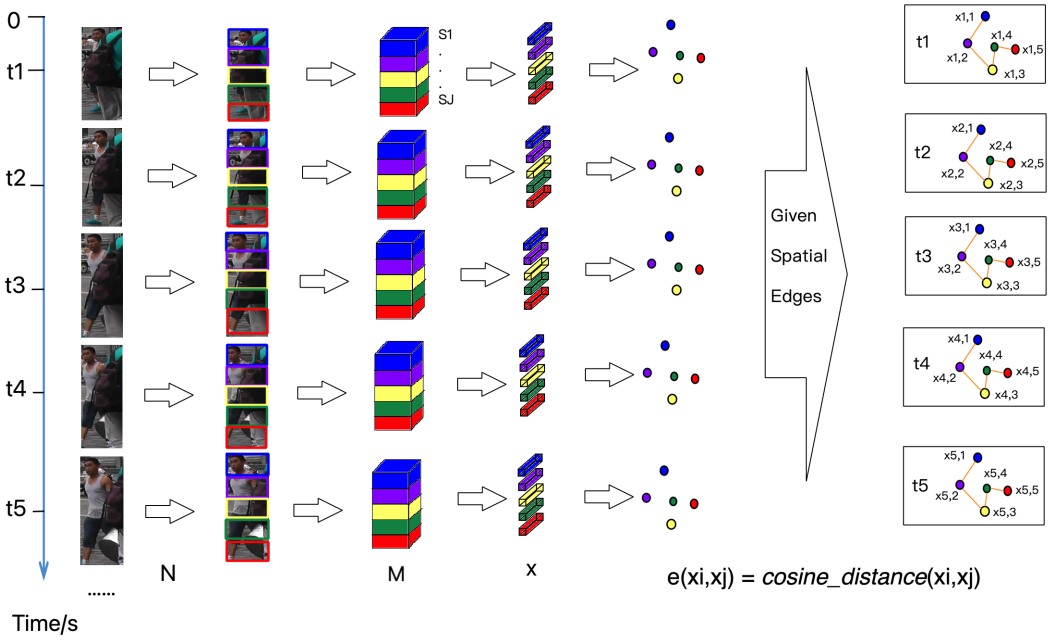


Figure 5: Spatial Graph (SG) Construction

3.6 Spatial-Temporal Graph & GCN Module

As mentioned in Sections 3.4 and 3.5, the temporal graph (TG) aims to capture complementary relations across video frames and use this temporal information to alleviate the occlusion

problem, while a spatial graph targets learning the complementary relations of different body parts within a video frame and use this spatial information to alleviate the visual ambiguity problem. However, TG and SG each focus on solving a single problem. Furthermore, our method uses spatial and temporal relations across frames to jointly alleviate occlusion and visual ambiguity problems. To achieve the goal, our method combines the temporal graph and spatial graphs to form a holistic spatial-temporal graph (STG) (see Figure. 6), then use GCN to learn the implicit spatial and temporal relation jointly and use it to re-identify a person.

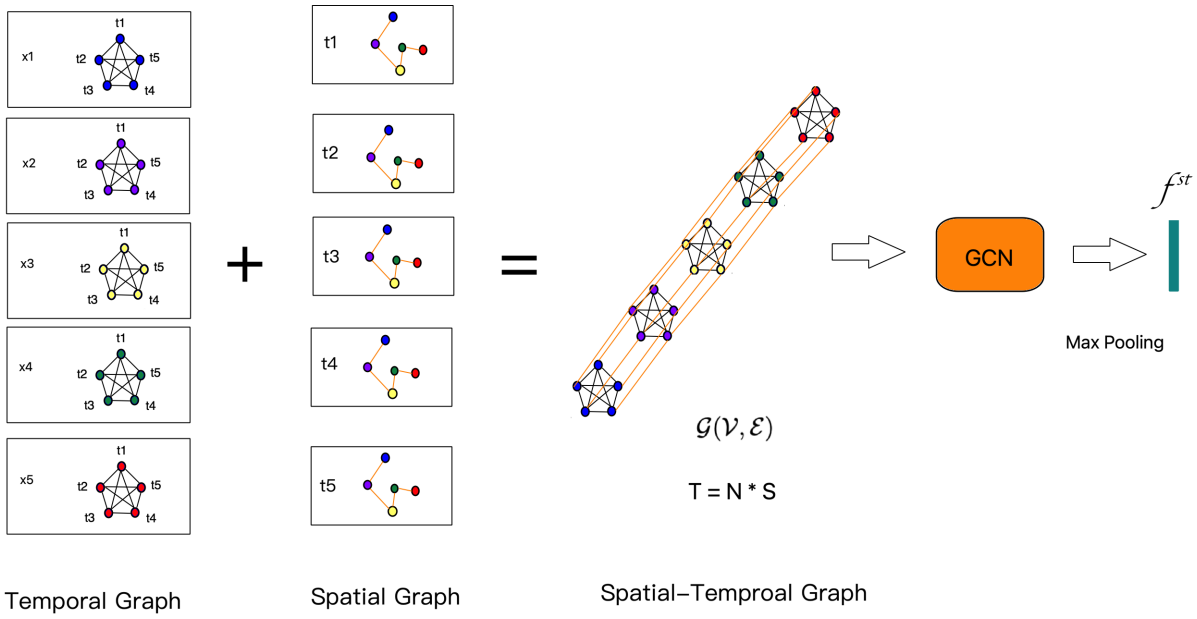


Figure 6: GCN operation on Spatial-Temporal Graph (STG).

Specifically, we combine the $\widehat{A}^t$ and $\widehat{A}^s$ to an overall STG adjacency matrix with the size of $T * T$, which is denoted as $\widehat{A}^{st}$, and use a padding value of 0 as the correlation coefficient to fill in the blank edge items in $\widehat{A}^{st}$. In our padding method, we choose 0 as the padding value, which is based on the property of cosine similarity, that is, the value of the angle between two vectors in the space is usually [-1, 1], and using 0 as the padding value aims to moderately weak associations are allocated between unconnected vertices. After that, we utilize GCN to extract the spatiotemporal relations of the whole video on STG. As mentioned in Equation (2), for the GCN network, we build a graph convolution of L layers, thus the l-th $(1 \leq l \leq L)$ hidden layer in GCN is represented as:

$$f(H^{(l)}, \widehat{A}^{st}) = \sigma(\widehat{A}^{st} H^{(l-1)} W^{(l)}) \tag{12}$$

where $W^{(l)}$ is the weight matrix for layer l and σ is a non-linear activation function. Namely, the convolutional operation between each layer can also be denoted as:

$$X^l = AX^{l-1}W^l \tag{13}$$

where $X^{(l)} \in \mathbb{R}^{T \times d_l}$ is the feature matrix of all segments (vertices) for the l-th hidden layer, it is also the input of next layer $H^{(l+1)}$; And d_l is the dimension of hidden features; Thus, the

input of the first layer $H^{(0)}$ is the feature matrix $X^{(0)}$ which is the initial segment features obtained from the CNN Backbone; $W(l) \in \mathbb{R}^{d_l \times d_l}$ is the weight matrix that needs to be learned for each layer. Inspired by [27], in each convolutional layer of GCN, the activation function of each layer in GCN is set to LeakyReLU accompanied by the negative input slope $\alpha = 0.2$. After the graph convolution, the output of the GCN model is an X^L feature matrix (graph embedding features), and finally we perform a max pooling on the X^L, so that obtained a spatiotemporal embedding features $f^{st} \in \mathbb{R}^{1 \times d_L}$.

4 Experiment

4.1 Baseline

As mentioned before, most of the current methods only considered the temporal relations across video frames but potentially miss the relations of different (body) parts within a frame or inter-frames, but our method believes such spatial information across temporal frames could provide more discriminative complementary clues for video-based ReID model, which can enhance model distinguishable ability, especially for occlusion and visual ambiguity scenario.

Based on the problem, we choose two categories of baselines for comparative analysis, the first category is the state-of-art work [2, 3, 19, 20] that uses temporal relations as discriminative cues for ReID task and those experiments are performed on the general ReID dataset, the purpose is to see whether spatial information across frames can provide more robust discriminative cues than only considering temporal information across frames; And another category of baseline is state-of-art work [22, 28, 23] specifically designed for solving occlusion problems, in which the experiments are conducted on the partial or occluded datasets, which in order to evaluate the performance of our method for solving occlusion problems.

4.2 Dataset

As mentioned in the baseline Section 4.1, in the experiments of this thesis, we evaluate our proposed method on two categories of datasets. The first category is 2 general purposes video-based ReID datasets i.e., MARS [29] and iLIDS-VID [30]. Another category of datasets is 2 video-based ReID datasets dedicated to partial or occlusion scenes, where each video frame is partially occluded, namely Partial-iLIDS [31], Partial-REID [32], Occluded-DukeMTMC [28]. The settings for the datasets used for our experiments are summarized in the Table. 3.

MARS [29] is a large-scale video pedestrian re-identification datase that derived from Market1501. The images of this data set are automatically cut by the detector, including the entire tracking sequence (Tracklet) of pedestrian images. MARS provides a total of 20,715 image sequences of 1,261 pedestrians, which come from 6 cameras on the Tsinghua University campus, and at least 2 cameras capture each identity. In addition, the MARS dataset contains 3,248 distractors identities, providing good visual ambiguity and occlusion cases.

iLIDS-VID [30] is a video-based ReID benchmark, it includes 600 images of 300 unique person identities that are recorded from 2 cameras, in the dataset each person has a pair of sequence video frames from 2 cameras. The average length of each video sequence is 72. The iLIDS-VID dataset is challenging because the data content contains similar clothing between people, lighting viewpoint changes, background noise, and random occlusions.

Partial-iLIDS [31] is a video-based ReID dataset derived from iLIDS-VID, which collects 119 identities and 476 images, and the data is collected from 4 disjoint cameras. 119 of these images are cropped from one camera view according to occlusions, the remaining 119 are full-body images. In experiments of this paper, we use MARS and Partial-iLIDS as the training and testing sets. For the testing set (i.e., Partial-iLIDS), the cropped images are regarded as the Query set, and full-body images are regarded as the Gallery set, and each identity in Query

can be found in the Gallery set.

Partial-REID [32] is a small-scale video-based ReID dataset that collects more than 600 video images of 60 identities using 6 synchronous cameras. Among them, each identity contains five full-body images and five cropped images. Similar to the Partial-iLIDS dataset, the partially cropped data is used as the Query Set, and the full-body image will be used as the Gallery set. In our experiments, the Partial-REID dataset is only used for testing, and we apply the model trained on MARS to this dataset for testing.

Occluded-DukeMTMC [28] is a video-based ReID dataset sampled from DukeMTMC-reID, dedicated to the occlusion ReID task. Occluded-DukeMTMC collects data on 1,221 identities from 8 disjoint cameras, among which, the training set contains 702 identities and 15,618 video images. The testing set contains the remaining 519 identities and 15,620 video images. Every video frame for every identity contains occluded elements. In the experiments in this paper, the occluded-DukeMTMC data is used for both training and testing on our method.

Datasets	Identity	Training		Query (Testing)		Gallery (Testing)		Cam.NO
		IDs	Img	IDs	Img	IDs	Img	
MARS	1,261	625	10,357	636	1,296	736	9,062	6
iLIDS-VID	300	150	300	150	50	180	250	2
Partial-iLIDS	119	-	-	119	119	119	119	2
Partial-REID	60	-	-	60	300	60	300	4
Occluded-DukeMTMC	1,221	702	15,618	519	2,603	620	13,017	8

Table 3: The inspection of the ReID datasets in performance comparison. IDs: The number of identities; Img: The number of video frames (images); Cam.No: the number of cameras used to collect data.

4.3 Experiment Design

As mentioned in Chapter 3, the goal of our method is to extract a spatial-temporal feature f^{st} of different human body parts within or across video frames and use it as complementary clues to alleviate collusion and visual ambiguity problems.

To validate the performance of our proposed method on ReID, we conducted three experiments:

- **Experiment 1:** The goal of experiment 1 is to explore the feature map splitting stage mentioned in Section 3.2, the impact of the number of local feature splits on the model prediction performance. In detail, if the number of segmentations of video frames is increased to capture more detailed local features, whether the model obtains more accurate discriminative clues. On basis of the goal, we vary the number of splits of video frames on the experimental datasets, ranging from 3 to 8, to find the best splits that achieve the best performance of the model on different datasets.

- **Experiment 2:** Experiment 2 is to see if spatial information across frames can provide stronger discriminative cues than only considering temporal information across frames.

Based on the goal, we test our model and the other four baseline models on the MARS and iLIDS-VID datasets with real distractors and use the same metrics to compare our method to the baseline methods.

- **Experiment 3:** Experiment 3 aims to see whether our method can achieve better results than other latest baseline methods in solving partial and occlusion problems. Specifically, we test our model and the other four baseline models on the ILIDS, Partial-REID and Occluded-DukeMTMC datasets which are designed for occlusion tasks, and use the same metrics to compare our method to the baseline methods.

The detail of experimental data division is shown in Table 3. Specifically, according to the deep learning data division method, the data set is divided into three parts: training set, verification set, and testing set. In our experiments, we refer to the way of data division from baseline work [23], using 50% of the data as the training set and 50% of the data as the testing set. In addition, for the training set, we use 1/5 of the data in the training set as the validation set to fine-tune the model. Then, similar to other baseline works [2, 3, 19, 20, 22, 28, 23], the testing set is divided into Query and Gallery, where the query set is 1/5 of the testing set, and the gallery set is the remaining 4/5.

The experimental design is shown in Figure 7. We use a set of video sequences from different identities as the training set, where each video sequence is from an identity. Besides, in the training phase, we uniformly sample 1/5 of the data from each identity's video sequence as a validation set for fine-tuning and parameter optimization of the model. After training, we test the model on testing sets (i.e., Query and Gallery). The query set consists of video sequences that we consider unknown identities, i.e., the objects we want to find. Moreover, the gallery set consists of video sequences containing distractors of different identities, which we regard as video sequences simulating natural scenes. In the testing phase, we provide a video sequence of an unknown person from the query set. The training model needs to query all possible candidate video sequences from the gallery set, which are identified as having the same identity as the query person. In addition, the gallery set contains distractors (unknown identities) that are not existing in the training and query sets.

According to the query, the model's output is a probability matrix shown in Figure. 8. This matrix stores the probability distribution of predicted images (from the gallery set) matching each query. Specifically, the trained model will calculate the similarity of each query from the candidate sequences of the Gallery and return top-N similar members according to the degree of similarity. So top-1 is the most similar person among all candidates in the Gallery, then top-2 comes next, and so on. Finally, based on the probability matrix, we use three commonly used query metrics to evaluate the model's performance: Rank-N, Mean Average Precision, and Cumulative Matching Characteristic (CMC) curve. The details of the metrics are explained in Section. 4.4.

4.4 Evaluation protocols

In the experiments, we use three metrics commonly used in visual query tasks to evaluate the model's performance: Rank-N Accuracy, Mean Average Precision (MAP), and Cumulative

Figure 7: The experiments design.

Match Characteristic curve(CMC) [2, 3, 19, 20, 22, 28, 23].

Rank-N Accuracy: Rank-n, or top-n, is the most used metric in computer vision tasks. It represents the probability that, among the top n most likely answers of the model to the query, at least an answer exists that matches the expected answer [33]. Figure. 8 shows an example that when we query a video sequence of identities using the trained model, it returns the top 10 most likely candidates of the Gallery in order of similarity. In the figure, the items with the same color indicate that the predicted label matches the query label (i.e., the expected answer). Taking the previous three queries as an example,

- The rank-1 accuracy is (1+0+0)/3=33.3% because only the first prediction of query 1 matches the ground truth label;
- The rank-3 accuracy is (1+0+1)/3=75.0% because among the top 3 predicted answers, there are predicted correct answers in both query 1 and query 3;
- The rank-5 accuracy is (1+1+1)/3=100% because among the top 5 predicted answers, all three queries have predicted correct answers;

In rank-n accuracy, rank-1 is the most convincing and strict metric because it penalizes all strictly incorrect guesses, while a rank-n for $n > 1$ allows tolerance for some error.

Cumulative Match Characteristic curve(CMC): CMC curve [22, 28, 23], by drawing the curve of Rank-n accuracy as the parameter n changes, it can intuitively show the change of top-n hit probability, mainly used to evaluate the ranking results of the closed set correct rate. The y-axis of the curve represents the recognition accuracy rate, and the x-axis represents the top-n accuracy.

Mean Average Precision (MAP): Mean Average Precision(mAP) is a commonly used metric for evaluating object detection models [2, 3, 19, 20], it is obtained by weighting the

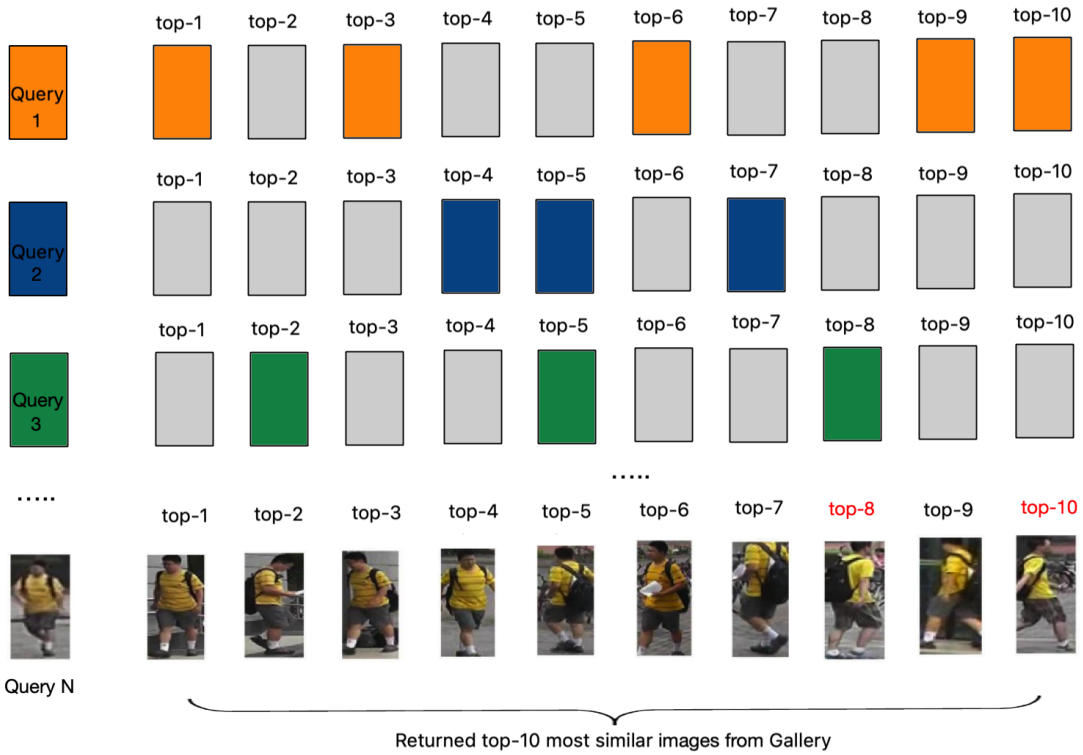


Figure 8: The returned results (probability array) of our model: for each query image it returned top-10 most likely images from gallery (in this case) based on the probability from high to low. The ground truth images are represented in colored block who has the same color with the query image. The grey blocks (or red title) denote the results of those returned images whose real identity is not same as the query image.

average accuracy (AP) of all class detection. Average Precision (AP) measures the quality of the learned model in a single class, while mAP measures the quality of the learned model in all classes. The calculation of mAP is to take the average of AP across all the classes, i.e.,

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i \tag{14}$$

In equation (14), k denotes the number of all classes, i represents a certain class. In a query task, the meaning of average precision is where the predicted correct answer (ground truth) should appear in the returned sequence. Taking the first three queries of Figure 8 as an example,

- The average precision (AP) for query 1 is $(1+2/3+3/6+4/9+5/10)/5 = 0.62$ because the 5 correct answers predicted in the first query appear in the 1st, 3rd, 6th, 9th, and 10th positions of the returned sequence, respectively.
- The average precision (AP) for query 2 is $(1/4 + 2/5 + 3/7)/3 = 0.36$ because the 3 correct answers predicted in the second query appear in the 4th, 5th, and 7th positions of the returned sequence, respectively.
- The average precision (AP) for query 3 is $(1/2 + 2/5 + 3/8)/3 = 0.425$ because the 3

correct answers predicted in the third query appear in the 2nd, 5th, and 8th positions of the returned sequence, respectively.

- Hence the mAP of first three queries is $(AP1 + AP2 + AP3)/3 = (0.62 + 0.36 + 0.425)/3 = 46.8\%$.

4.5 Implementation Details

ResNet50 : We adopt ResNet-50 [34] model pre-trained on ImageNet for coarse extraction of input videos. The ResNet50 has 50 layers and 3 bottlenecks. Each input video frame (image) is resized to $3 \times 256 \times 128$, representing the number of RGB channels (i.e., red, green, and blue), the image's height and width. The channel's value ranges from 0 to 255. To enable the size of the obtained feature map has the same size as the input frame, the stride of the last down sample layer is set to 1 [27]. The detailed parameters' setting for the ResNet50 shows in Table. 4.

layer name	output size	ResNet-50	parameter setting
conv1	$64 \times 256 \times 128$	$7\times7, 64$	kernel(k)=7, stride(s)=1, padding(p)=3
		3×3 max pool	k=3, s=1, p=1
conv2_x	$256 \times 256 \times 128$	$\begin{pmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{pmatrix} \times 3$	k=3, s=1, p=0 k=3, s=1, p=1 k=3, s=1, p=0
conv3_x	$512 \times 256 \times 128$	$\begin{pmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{pmatrix} \times 4$	k=3, s=1, p=0 k=3, s=1, p=1 k=3, s=1, p=0
conv4_x	$1024 \times 256 \times 128$	$\begin{pmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{pmatrix} \times 6$	k=3, s=1, p=0 k=3, s=1, p=1 k=3, s=1, p=0
conv5_x	$2048 \times 256 \times 128$	$\begin{pmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{pmatrix} \times 3$	k=3, s=1, p=0 k=3, s=1, p=1 k=3, s=1, p=0

Table 4: ResNet-50 Parameters Setting.

GCN Setting: For the GCN modules shown in Figure. 9, we use the 3 layers graph convolution network. In addition, we select the optimal number of feature map segments corresponding to each dataset for experiments i.e., 5 segments splitting are used on experiments of MARS, Partial-iLIDS, Occlude-reID datasets and 6 segments splitting is used on experiments of iLIDS-VID dataset. Besides, we choose ReLu as the activation function of GCN.

Train/Test Setting: In the training phase, we set each mini-batch to contain 16 person IDs, each ID contains 4 input video clips representing the 4 different tracklets, and each video clip consists of 8 frames sampled by a restricted random sampling strategy [35]. Therefore the size of a mini-batch is $16 \times 4 \times 8 = 512$ images. Next, all the video images are resized to 256×128, and we adopt a random flipping strategy [36] for data augmentation. Next, the model is trained by 800 epochs with initial $3e^{-4}$ as the learning rate, and for every 40 epochs, the learning rate decay by 0.1. Following [37], we leverage Adam as a network optimizer. Finally, in the testing phase, for each identity, we select 4 queries from different tracklets, find matching images from the gallery set and keep only the top-10 similar candidate identities.

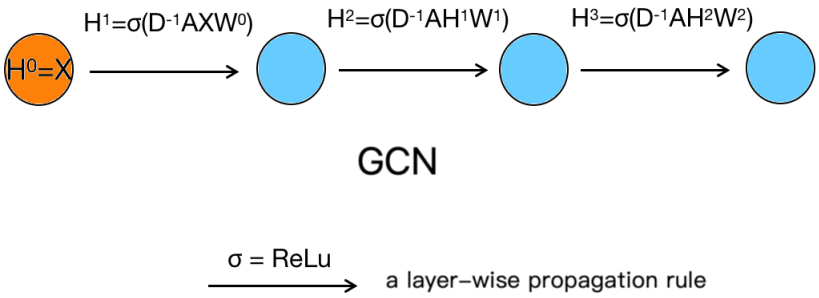


Figure 9: The GCN parameter Setting.

Experiments Environment: All the experiments are running under the following conditions: we perform the experiments by google Colab with GPU acceleration. The programming language is 'Python 3.8.14'. The ResNet50 module[3] is refereed by Pytorch official library [38] and the installed pytorch versions are 'pytorch==1.12.0', 'torchvision==0.13.0' and 'torchaudio==0.12.0'. Besides, to accelerate the training process for video tasks, we enable the CUDA toolkit for using GPU to accelerate the calculation process and the CUDA version is 'cudatoolkit=10.2', the GPU model used in Colab is 'Tesla K80' with 12GB of video memory. For the GCN, we use the author's original implementation in PyTorch [4] and extend the network from 2 to 3 layers.

5 Results and Analysis

5.1 Comparison with the State-of-the-art Methods

In this Chapter, we present the results of the three experiments mentioned in Section.4.3 and analyze the potential causes of the results.

The result of Experiment 1: Experiment 1 aims to explore the effect of the number of feature map splits (number of local features) on model performance. In our experiments, we varied the number of splits of the feature map from 3 to 8, then recorded the Rank-1 accuracy and mAP values for all datasets. For experiment 1, we provide two experimental results, one is the effect of the number of feature map splits on rank-1 accuracy on all datasets (see Figure. 10), and the other is the effect of the number of feature map splits on the MARS dataset on the rank-1 and mAP metrics (Figure. 11).

From Figure. 10, it can observe that for most datasets, the optimal feature map splits' number is between 5 and 6, except the P-DukeMTMC-reID, which is 7. Furthermore, it can be seen from Figures. 10 and 11 that in all data sets, the impact of the splits' number on both metrics (i.e., rank-1 accuracy and mAP) shows a trend of increasing first and then slightly decreasing, which indicated that in our STG-GCN model, too many or too few feature map division would cause the GCN underperformance. Because when the number of splits decreases, the local area

[3]ResNet in PyTorch:https://pytorch.org/hub/pytorch$_vision_resnet$/

[4]Graph Convolutional Networks in PyTorch: https://github.com/tkipf/pygcn

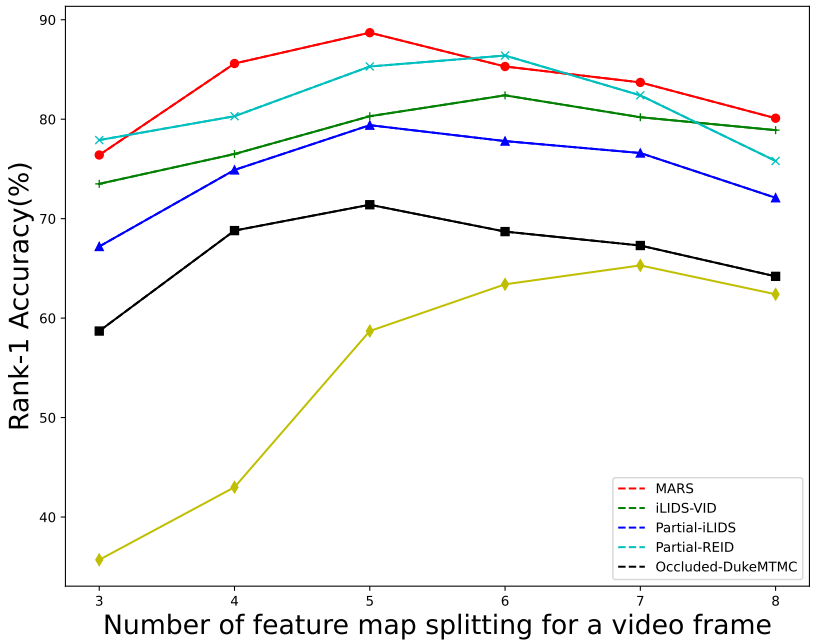


Figure 10: The effect of the number of feature map splits (3 to 8) on the Rank-1 accuracy in the experiment.

of each segment becomes larger, causing the local features to ignore some subtle but valuable discriminative clues, thus reducing the model's accuracy. On the contrary, continuously increasing the number of splits does not continually improve the model recognition rate because when segments increase, the area of each segment becomes smaller and continuously refined, which will cause the segments over-focus on local details and lose the generation ability to global representation.

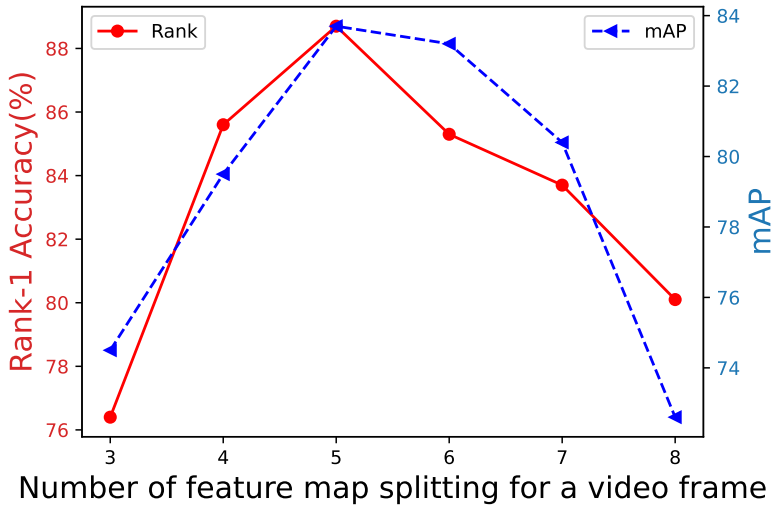


Figure 11: The effect of the number of feature map splits (3 to 8) on Rank-1 and mAP on the MARS dataset.

Therefore, in Experiments 2 and 3, we set the number of feature map splits of the trained model to the optimal value corresponding to each dataset, that is, for MARS, Partial-iLIDS, and Occluded-DukeMTMC, the number of feature map splits is set to 5, 6 for iLIDS-VID,

Partial-REID dataset, and 7 for P-DukeMTMC-reID dataset.

The result of Experiments 2: In Experiments 2, we validate our method on two general ReID datasets. i.e,MARS and iLIDS-VID, and comparing the result with four other state-of-art baseline works, the result is shown in Table. 5, and the corresponding CMC curves chart can be found in Figure. 12 and Figure. 13.

Method	reference	MARS				iLIDS-VID			
		Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
TKP	ICCV'18	78.5	90.9	93.9	70.5	68.3	75.7	83.1	61.4
temp-RN	AAAI' 20	80.1	88.3	92.4	74.6	75.4	83.6	90.4	65.7
TCRL	CVPR'22	86.0	92.5	94.2	80.1	77.3	94.7	96.7	73.4
TVCAN	ICMR'22	89.0	97.0	98.1	85.7	88.5	98.1	100	82.3
STG-GCN(Ours)	-	**88.7**	**95.3**	**98.4**	**83.7**	**82.4**	**95.6**	**97.6**	**79.4**

Table 5: Performance comparison of our STG-GCN model with 4 other baseline works on MARS and iLIDS-VID data. The bolded data are the experimental results of our model; The data with a green background representing our metric results are superior to baseline work results; The data with a red background indicates our metric results are inferior to baseline work results.

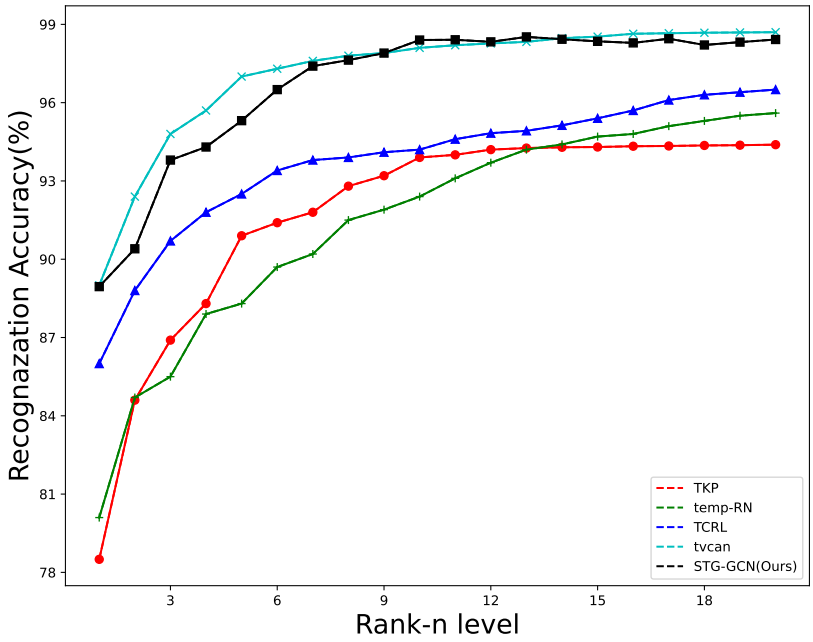


Figure 12: The baseline comparison's cumulative match curve (CMC) on MARS. The X-axis represents the variation of rank-n accuracy with parameter n, while the Y-axis represents accuracy; The area under the Cumulative Matching Characteristic (CMC) curve refers to the average probability of correct matches from the top 1 to top-n ranked images, where n represents the size of the image gallery.

In both the CMC curves and Table. 5 can be seen that on the general ReID datasets (i.e., MARS, iLIDS-VID), our model outperforms most baseline works considering only temporal clues, However, some metrics underperform for the work TVCANZ [20]. Specifically, for the

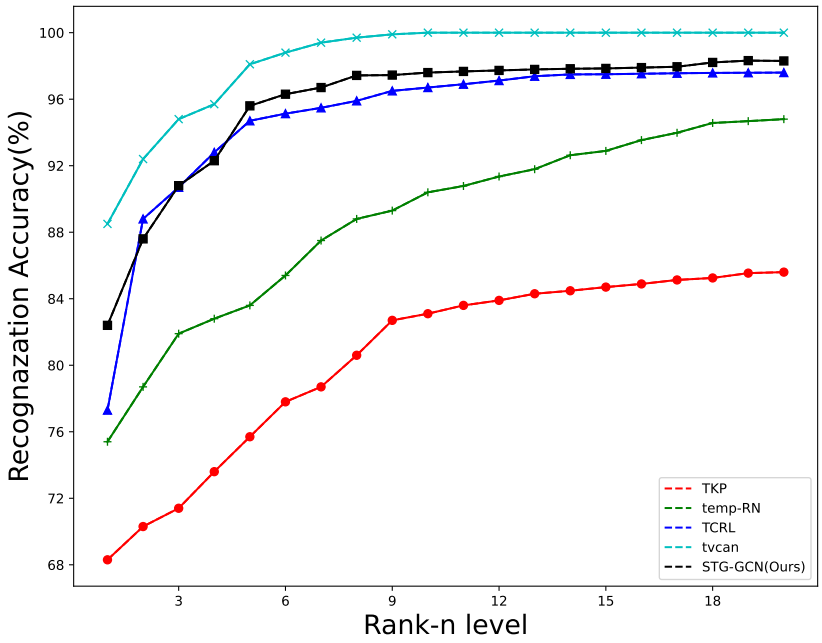


Figure 13: The baseline comparison's cumulative match curve (CMC) on ILIDS_VID. The X-axis represents the variation of rank-n accuracy with parameter n, while the Y-axis represents accuracy;

TKP approach [3], it adds a Non-local block to ResNet 50 to model temporal features, which then be taken into account in the loss function. However, since TAP focuses on extracting the average global information over each video frame, it may ignore the correlation of different parts within a video frame and thus cannot be adapted to solve the occlusion scenario in the video. Besides, for the temporal-relational network [2], it uses local segmentation within each video frame to obtain spatial features between different parts of the human body, but this spatial information is limited to a single video frame, ignoring the relation between local feature segments of the entire video sequence. And for the TCRL method [19], it uses reinforcement learning in which the Agent dynamically selects an appropriate number of frames from the gallery video to accumulate temporal information when it encounters a query. However, in TCRL, the Agent can only use the salient regions of the previous frame as supplementary information to enhance the representation of the next frame, which leads to interdependence of supplementary knowledge from adjacent frames. This may result in video frames being occluded and the occluded region may be used as a salient region and considered as valid supplementary information for the next frame, thus introducing interference in the model. In contrast, the temporal information extracted by our model can be obtained from non-adjacent but close frames, thus mitigating the noise caused by temporary obstacle occlusions, resulting in an improvement of 2.7% for rank-1 and 3% for rank-5 in comparison. From Figure 14, it can be observed that our model, which employs dual clues (i.e., temporal and spatial), is effective in alleviating occlusion and visual blur when compared to the latest baseline work which only considers a single clue.

However, when compared with the most recent work TVCAN [20], it can be seen that the key metrics rank-1 to rank-5 are inferior on both the MARS and ILIDS/VID datasets, but the rank-10 metric is superior to TVCAN. However, when compared to the latest work TVCAN, it

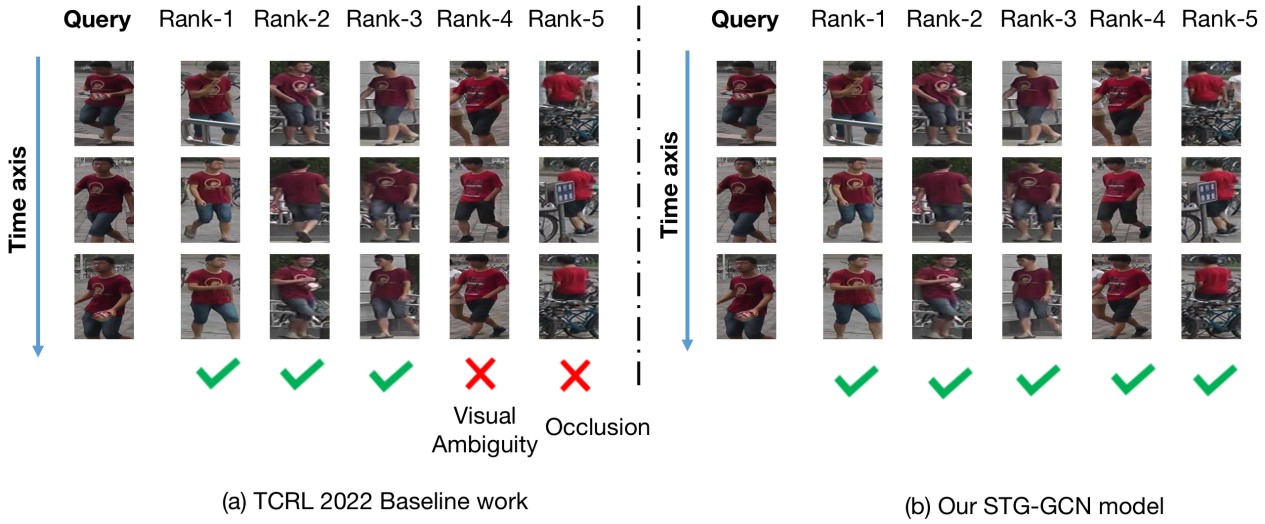


Figure 14: (a) and (b) correspond to the top-5 retrieval results of the TCRL 22' baseline model and our proposed method on the MARS dataset, respectively.

can be found that on the MARS and ILIDS/VID datasets most of the main metrics are inferior, with only rank-10 and mAP values on the MARS dataset our model performs well. This may be because TVCAN's approach to solving the occlusion problem is divided into two steps. First, it uses a series of detectors to extract consistent cues between frames. It then uses a selection algorithm to pick out a few salient cues that are most consistent and applies a focus mechanism to each frame feature to focus on these cues. In other words, the TVCAN method does not directly extract association information on the video sequence. It first extracts the original video sequence cues containing interference factors, then obtains consistent cues on the original video cues, and then allows the model to focus only on these consistent cues to achieve complete disregard for interference noise. However, since our method uses continuous spatial-temporal information as supplementary information to mitigate the occlusion interference, it still directly extracts the information from the video sequence and cannot eliminate the interference factors. For example, in extracting feature maps by GCN, the occluded feature segments are still considered in the GCN network for learning, which inevitably introduces transient interference elements and the momentary interference elements, e.g., occlusion. In contrast, TVCAN performs feature extraction on clues with consistent features, eliminating transient interference factors.

The Results of Experiment 3: In Experiment 3, we validate our method on three datasets that are specific for the occlusion scenario, i.e., Partial-ILID, Partial-REID, and Occluded-DukeMTM, and compare metrics result with three other baseline works, the result is shown in Table. 6, and the CMC curves chart can be found in Figure. 15 and Figure. 16
Overall, our model performs better on the Occluded dataset than on the Partial dataset. On the Occluded-DukeMTM dataset, our model is 6% higher in rank-1 and 13% higher in mAP than the cutting-edge method POS [23]. However, on the partial dataset, our method only outperforms the PGFA [21] method across the board but still lags behind ADGC-CGEA [22] and POS on the critical metric rank-1,rank-5 accuracy.

This result is probably because when the PGFA [21] method encounters occlusion, it directs the attention map to focus only on non-occluded areas. However, most occlusions in images

Method	reference	Partial-ILIDS				Partial-REID				Occluded-DukeMTMC			
		Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
PGFA	ICCV'19	69.1	80.9	89.3	63.7	68.0	80.0	85.4	67.4	51.4	68.6	74.9	37.3
ADGC-CGEA	CVPR' 20	72.6	86.4	91.3	67.5	85.3	91.0	94.8	74.6	55	70.2	80.4	43.8
POS	ELSEVIER '22	86.0	88.6	94.2	70.3	86.1	91.3	96.3	80.9	65	79	83.8	54
STG-GCN(Ours)	-	**79.4**	**85.8**	**92.4**	**68.5**	**84.7**	**92.9**	**95.3**	**79.4**	**71.4**	**83.6**	**88.6**	**67.0**

Table 6: The experiments result for our STG-GCN and other baseline methods on 3 Occluded-ReID dataset. The bolded data are the experimental results of our model; The data with a green background representing our metric results are superior to baseline work results; The data with a red background indicates our metric results are inferior to baseline work results.

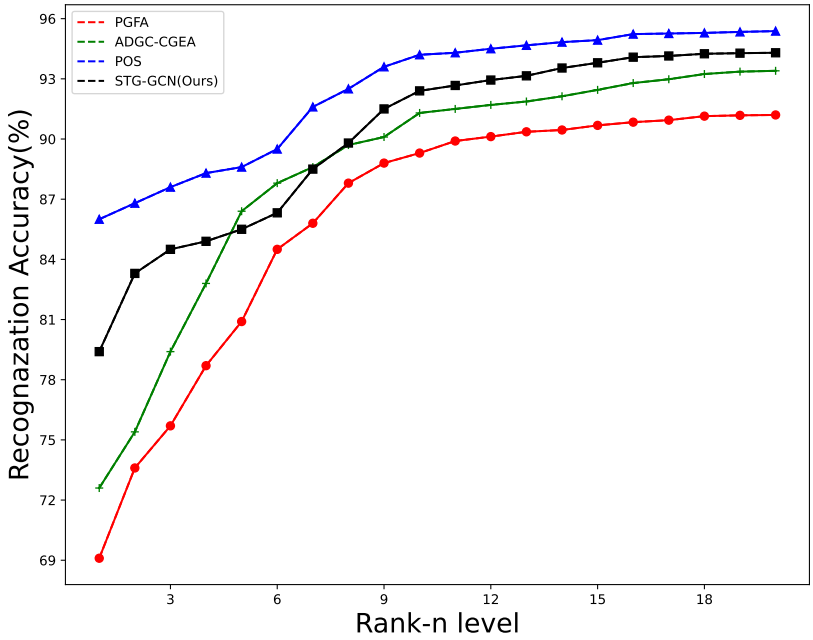


Figure 15: The baseline comparison's cumulative match curve (CMC) on Patrial_ILIDS. The X-axis represents the variation of rank-n accuracy with parameter n, while the Y-axis represents accuracy.

are only partially occluded and not fully occluded. Nevertheless, once the method encounters an occlusion, it treats it as a fully occluded situation and excludes it, thus ignoring the valid information that many partially occluded parts of the image are also present.

Furthermore, the ADGC-CGEA [22]and POS [23] methods solve the occlusion problem by predicting the key point map of the human body to recover the complete topology of the body. However, these methods only operate on each video frame and they didn't consider passing the spatial topological structure information across frames, so the spatio-temporal relationships of different parts of the human body across frames are not taken into account. Since our model considers both the spatiotemporal information transfer of different video parts within and across frames, our approach outperforms them in all respects on the occluded-dukeMTMC dataset.

However, our model's key metrics rank-1 and rank-5 accuracy on the two partial datasets are inferior to the baseline work. The reason may be that, on the one hand, our model was trained

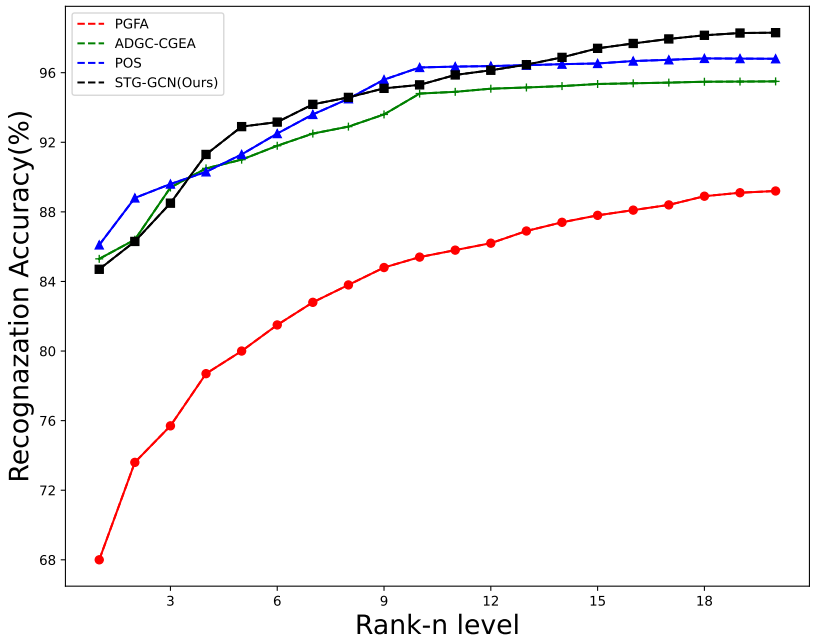


Figure 16: The baseline comparison's cumulative match curve (CMC) on Patrial_ReID. The X-axis represents the variation of rank-n accuracy with parameter n, while the Y-axis represents accuracy.

and predicted on the Occluded-dukeMTMC dataset, but on Partial-iLIDS and Partial-REID we only make predictions instead of training, so the performance of the model was degraded due to the lack of knowledge of the dataset training. On the other hand, the content of the partial datasets and the occluded datasets are different. Specifically, a Query of the partial dataset is a partial human body part that manually cropped the occluded target person, and the gallery set is the complete person image. But for Occluded Duke MTMC, both the Query and Gallery are complete images. Therefore, to fit our model, we do not need to perform image padding on the Occluded datasets, but we need to perform image padding on the partial dataset so that the cropped images fit into the 256*128 size. However, since the image padding is performed using the uniformed grayscale image, as shown in Figure, compared with the Occluded case, the partial dataset image padding introduces great interference. For example, the occluded body part will not be reappeared in subsequent frames, thus making our model unable to utilize continuous temporal and spatial supplementary information, which is the main reason for our model performance to degrade on partial datasets.

6 Conclusion

In this paper, we propose a video-based person re-identification method that uses a spatial-temporal graph (STG) to represent a video sequence. It models the relationships between different body parts and their interactions over time, which allows it to learn more discriminative features for person ReID than existing methods. Furthermore, our model shows good effectiveness in alleviating occlusion and visual ambiguity problems. Specifically, the Spatial-Temporal Graph (STG) consists of two graph modules. i.e., temporal graph module and spatial graph module. Both graph modules regard the segment features as nodes. Temporal graph is built by connecting nodes of the same body part across frames, and it learns the temporal

relation between consecutive frames for alleviating occlusion problem. Spatial graph is built by connecting nodes of different body parts in each video frame in top-to-bottom order, and it learns the structural information of human body for alleviating visual ambiguity problems. Besides, we integrate temporal and spatial graphs in one spatial-temporal graph and apply GCN to obtain a spatial-temporal feature to optimize the model jointly.

Our model has been validated on multiple datasets, demonstrating outstanding performance on the regular ReID dataset and the Occluded ReID dataset. However, its performance declines when faced with the partial ReID problem. On the one hand, due to the lack of temporal continuity in the dataset, our model failed to obtain temporal clues. On the other hand, the padding method used for partial images inevitably introduces interference, which results in a reduction in the model's performance.

7 Limitation and future work

As the Chapter. 5, our STG-GCN model exhibits reduced performance when faced with the partial ReID problem, with the main reasons for the performance degradation stemming from two factors. Firstly, the partial ReID dataset itself is not a continuous video sequence, resulting in a lack of temporal continuity and the failure of our model to obtain the necessary temporal clues. This indicates that our method is limited to the use of data with temporal continuity. Secondly, in order to train our model using inputs of a uniform size, we used grayscale images to fill the missing parts of partial images. This introduced significant interference, leading to a reduction in model performance. With respect to the second factor, we can make two improvements in the future. Firstly, for cropped partial image pictures, we can use pixel interpolation methods (such as mosaic) instead of cropping the image. The advantage of using mosaic to mask the partial image is that it preserves the color or brightness values that are the same or similar to the existing pixels, which can reduce the interference caused by filling in the missing parts of the image for machine learning models. Secondly, inspired by the approach taken by TVCAN, we can use a video detector to extract consistent clues between frames, and then apply our STG-GCN method on the consistent clue features. This will enable our model to obtain an attention clue before extracting spatial-temporal clues, thereby minimizing the interference factors and achieving the desired performance.

8 References

[1] Manisha Talware and Sanjay Koli. Video-based person re-identification: methods, datasets, and deep learning. *Int J Eng Adv Technol (IJEAT)*, 9(3):4249–4254, 2020.

[2] Hyunjong Park and Bumsub Ham. Relation network for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11839–11847, 2020.

[3] Xinqian Gu, Bingpeng Ma, Hong Chang, Shiguang Shan, and Xilin Chen. Temporal knowledge propagation for image-to-video person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9647–9656, 2019.

[4] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[5] Luo Hao, Jiang Wei, Fan Xing, and Zhang Sipeng. Research progress of person re-identification based on deep learning. *Journal of Automation*, 45(11):2032–2049, 2019.

[6] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2360–2367. IEEE, 2010.

[7] Tetsu Matsukawa and Einoshin Suzuki. Person re-identification using cnn features learned from combination of attributes. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2428–2433. IEEE, 2016.

[8] Arne Schumann and Rainer Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 20–28, 2017.

[9] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, 2019.

[10] Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016.

[11] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European conference on computer vision*, pages 791–808. Springer, 2016.

[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[13] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.

[14] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the iEEE conference on computer vision and pattern recognition*, pages 1335–1344, 2016.

[15] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017.

[16] Hao Liu, Zequn Jie, Karlekar Jayashree, Meibin Qi, Jianguo Jiang, Shuicheng Yan, and Jiashi Feng. Video-based person re-identification with accumulative motion context. *IEEE transactions on circuits and systems for video technology*, 28(10):2788–2802, 2017.

[17] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. Recurrent convolutional network for video-based person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2016.

[18] Jiyang Gao and Ram Nevatia. Revisiting temporal modeling for video-based person reid. *arXiv preprint arXiv:1805.02104*, 2018.

[19] Wei Wu, Jiawei Liu, Kecheng Zheng, Qibin Sun, and Zheng-Jun Zha. Temporal complementarity-guided reinforcement learning for image-to-video person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7319–7328, 2022.

[20] Bingliang Jiao, Liying Gao, and Peng Wang. Temporal-consistent visual clue attentive network for video-based person re-identification. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 72–80, 2022.

[21] Yiming Wu, Omar El Farouk Bourahla, Xi Li, Fei Wu, Qi Tian, and Xue Zhou. Adaptive graph representation learning for video person re-identification. *IEEE Transactions on Image Processing*, 29:8821–8830, 2020.

[22] Guan'an Wang, Shuo Yang, Huanyu Liu, Zhicheng Wang, Yang Yang, Shuliang Wang, Gang Yu, Erjin Zhou, and Jian Sun. High-order information matters: Learning relation and topology for occluded person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6449–6458, 2020.

[23] Shujuan Wang, Bochun Huang, Huafeng Li, Guanqiu Qi, Dapeng Tao, and Zhengtao Yu. Key point-aware occlusion suppression and semantic alignment for occluded person re-identification. *Information Sciences*, 2022.

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] 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.

[27] Jinrui Yang, Wei-Shi Zheng, Qize Yang, Ying-Cong Chen, and Qi Tian. Spatial-temporal graph convolutional network for video-based person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3289–3299, 2020.

[28] Jiaxu Miao, Yu Wu, Ping Liu, Yuhang Ding, and Yi Yang. Pose-guided feature alignment for occluded person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 542–551, 2019.

[29] Springer. *MARS: A Video Benchmark for Large-Scale Person Re-identification*, 2016.

[30] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *Proceedings of the European conference on computer vision (ECCV)*, pages 737–753, 2018.

[31] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR 2011*, pages 649–656. IEEE, 2011.

[32] Wei-Shi Zheng, Xiang Li, Tao Xiang, Shengcai Liao, Jianhuang Lai, and Shaogang Gong. Partial person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4678–4686, 2015.

[33] Accuracy and loss: Things to know about the top 1 and top 5 accuracy. https://towardsdatascience.com/accuracy-and-loss-things-to-know-about-the-top-1-and-top-5-accuracy-1d6beb8f6df3. Accessed: 2023-02-26.

[34] 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.

[35] Shuang Li, Slawomir Bak, Peter Carr, and Xiaogang Wang. Diversity regularized spatiotemporal attention for video-based person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 369–378, 2018.

[36] Chih-Ting Liu, Chih-Wei Wu, Yu-Chiang Frank Wang, and Shao-Yi Chien. Spatially and temporally efficient non-local attention network for video-based person re-identification. *arXiv preprint arXiv:1908.01683*, 2019.

[37] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. *Advances in neural information processing systems*, 31, 2018.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.