Exploring Next-Generation Visual Object Detection and Tracking System

WEITAO FENG

B.Eng

Supervisor: Wanli Ouyang
Associate Supervisor: Luping Zhou

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Electrical and Information Engineering
Faculty of Engineering
The University of Sydney
Australia

Submitted on 27 February 2023
Revised on 29 April 2024

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Student Signature: Weitao Feng

Authorship Attribution Statement

Chapter 3 of this thesis contains materials published as *Multi-Object Tracking with Multiple Cues and Switcher-Aware Classification* (Feng et al. 2022a) in 2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA). I designed the study, implemented the algorithms, analyzed the data, and wrote the drafts of the paper.

Chapter 4 of this thesis contains materials published as *Similarity- and Quality-Guided Relation Learning for Joint Detection and Tracking* (Feng et al. 2024) in IEEE Transactions of Multimedia (TMM). I designed the study, implemented the algorithms, analyzed the data, and wrote the drafts of the paper.

Chapter 5 of this thesis contains materials first publicly available in ArXiv pre-print: *Towards Frame Rate Agnostic Multi-Object Tracking* (Feng et al. 2022b), which is subsequently accepted as a formal publication in the International Journal of Computer Vision (IJCV) (Feng et al. 2023). I designed the study, implemented the algorithms, analyzed the data, and wrote the drafts of the paper.

My thesis may also contain materials that are under confidential review for future publication in some conference proceedings and journals.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Student Signature: Weitao Feng

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor Signature: Wanli Ouyang

Supervisor Signature: Luping Zhou

Abstract

Recent advances in the field of object detection and tracking have enabled a wide range of industrial applications. In many video processing and analysis scenarios, object detection and tracking are often coupled, giving rise to the research topic of joint detection and tracking, which integrates the two tasks into a unified visual perception system. Currently, many research projects and applications are implementing visual perception modules using the joint detection and tracking approach. The growing interest in this joint task is due to its flexibility, efficiency, cost-effectiveness, and overall effectiveness. However, as a relatively new extension of object detection and tracking, the joint system faces a number of challenges. Some of these challenges are inherited from the original detection and tracking task, while others are unique to the joint design. Investigating the next generation of joint detection and tracking systems is an essential step toward robust video analysis techniques.

Within the context of the deep-learning era, addressing challenges requires a thorough examination across four foundational dimensions: **data**, **model architecture**, **learning procedure**, and **scenario abilities**. This thesis embarks on an exploration into the evolution of the next generation of joint detection and tracking systems, delineating four crucial subtopics within each of these dimensions: 1) **model architecture**: Investigating the impact of critical distractors and devising a model architecture explicitly tailored to effectively manage them, 2) **learning procedure**: Delving into the intricacies of relation learning within joint tasks and endeavoring to elucidate fundamental principles governing the learning process, 3) **scenario abilities**: Examining the frame rate robustness of the joint task and formulating a novel pipeline to adeptly handle dynamic frame rates, and 4) **data**: Exploring unsupervised learning methodologies tailored for the joint task. These carefully chosen subtopics align with and address four critical challenges, namely the distractor problem, learning representations from videos, robust tracking, and the application of unsupervised learning in the context of videos.

This research aims to contribute significant insights and advancements to the landscape of joint detection and tracking systems in the dynamic realm of deep learning.

To address these challenges, several new methods have been raised in this thesis. For the distractor problem, the concept of 'switcher' has been proposed and a switcher-aware association has been introduced to utilize the information of distractors. For the relation learning problem, a similarity- and quality-guided attention mechanism has been introduced for more effective feature refinement. For the frame rate robustness problem, a frame rate agnostic framework has been proposed to conduct joint detection and tracking with unseen frame rate inputs. For the unsupervised learning problem, a video-based pseudo-labeling method has been developed for model pretraining. To evaluate the effectiveness of these approaches, both qualitative and quantitative experiments have been conducted. The experimental results have shown satisfactory progress.

Acknowledgements

I am filled with gratitude for all the people who have supported me throughout my Ph.D. journey. I owe this thesis to their unwavering help and encouragement.

Above all, I express my heartfelt thanks to my supervisors, Associate Prof. Wanli Ouyang and Associate Prof. Luping Zhou. Their guidance, wisdom, and steadfast support have been indispensable to my Ph.D. Their feedback and encouragement have motivated me to reach new heights.

I am also grateful to my colleagues and collaborators, including Dr. Lei Bai, Dr. Dan Xu, and Dr. Baopu Li, for their technical assistance and contributions to my research. Dr. Lei Bai's insightful feedback has been instrumental in shaping my work.

My research has been made possible by the generous support of SenseTime Group Ltd. and the faculty of Electrical and Information Engineering at the University of Sydney. Their financial support has been crucial in enabling me to conduct my research.

I am immensely thankful to my friends and family for their unflagging support and encouragement during my Ph.D. Their love, understanding, and patience have sustained me through the challenging times and made the good times even sweeter.

Lastly, I am grateful to all the participants in my research studies, who have contributed their time, patience, and insights to my work. Without them, this thesis would not have been possible.

In conclusion, I owe a debt of gratitude to all the people who have supported me during my Ph.D. journey. Their contributions have been invaluable, and I will always be grateful for their help and encouragement.

Contents

List of Figures

CHAPTER 1

Introduction

Visual object detection and tracking are two fundamental tasks in the field of computer vision. Object detection aims to comprehend and locate visual concepts, while object tracking allows the identification of individual objects over time. These tasks collectively simulate how humans perceive the world.

In tandem with the comprehensive exploration of deep-learning models, significant advancements have been achieved in the realms of object detection and tracking in recent years, contributing to a variety of industrial applications such as video surveillance, unmanned aerial vehicle (UAV) auto-piloting, and autonomous driving. In many video analysis and processing situations, object detection and tracking are interrelated, which has led to the development of joint detection and tracking as a research topic. This approach integrates both tasks into a unified visual perception system. Currently, many research projects and applications are implementing visual perception modules using the joint detection and tracking approach (Kieritz et al. 2018; Zhou et al. 2020; Peng et al. 2020; Pang et al. 2020). The joint detection and tracking approach is becoming increasingly popular due to its flexibility, efficiency, cost-effectiveness, and overall effectiveness. However, as a relatively new extension of object detection and tracking, this joint system presents several challenges. Some of these challenges are inherited from the original detection and tracking tasks, while others are unique to the joint design. Investigating the next generation of joint detection and tracking systems is crucial to the development of robust video analysis techniques.

In the era of deep learning, addressing challenges necessitates a comprehensive examination across four foundational dimensions, namely, **data**, **model architecture**, **learning procedure**, and **scenario abilities**. Building upon these dimensions, this thesis seeks to tackle four

key areas in the exploration of the next generation of joint detection and tracking systems. The primary objective is to analyze the impact of critical distractors on the performance of joint tasks, a challenge inherited from traditional multi-object tracking methods. To tackle this challenge, there is a need for a novel model architecture design that incorporates explicit mechanisms to effectively manage distractors. The second goal is to delve into relation learning within joint detection and tracking, presenting a unique challenge stemming from the joint design. Addressing this challenge requires establishing fundamental rules for learning supervision. The third goal involves investigating the frame rate robustness of joint detection and tracking, a novel challenge not previously addressed by traditional methods. To overcome this challenge, a comprehensive exploration of the entire scenario pipeline and the development of new effective designs are essential. Lastly, the fourth goal aims to explore unsupervised learning for joint detection and tracking, which is crucial to overcoming the limitations associated with labeled data. Successfully achieving these four goals will significantly contribute to the development of more flexible, effective, and robust techniques for video analysis.

1.1 Modeling Critical Distractors

Multi-object tracking (MOT) aims to find out the relationships between each detected object and link them to form trajectories, where every single trajectory associates a unique object (e.g., a person) at different time steps (frames). Despite the impressive progress that has been achieved by recent studies, identity issue (also called ID-Switch or fragment), i.e.an object is wrongly associated with another object of a different identity, is still challenging because of cluttered background, occlusion, abrupt motion, etc. Most recent works that mainly focus on building up more effective association methods (Brasó and Leal-Taixé 2020; Xu et al. 2019b; Hornakova et al. 2020; Tang et al. 2017; Long et al. 2018) either prefer to develop more efficient graph partition methods or prefer to improve the quality of appearance feature representations. However, from another angle of view, we can take more information into consideration to alleviate this issue. Therefore, we argue that two factors should be paid great attention to.

FIGURE 1.1. A typical identity issue caused by occlusion. When the lady is occluded by the gentleman, the tracklet of the lady was wrongly associated with the gentleman, making the re-appearing unoccluded lady in the last frame considered as another new identity. Rectangles of the same color correspond to the same identity predicted by the MOT algorithm. In this case, each step from left to right is actually quite 'smooth'. Every choice based on a single visual cue is optimal, however, an identity issue happens. This kind of identity issue could be handled by including multiple cues in this paper.

First, we need to identify the potential identity issue that may happen. To this end, the single cue is not sufficient. Fig. 1.1 illustrates a typical identity issue caused by occlusion. In such a case, the single visual cue is less effective due to occlusion. Each step from left to right in Fig. 1.1 is actually optimal based on a single visual cue. Unfortunately, an identity issue happens. Therefore, multiple cues are needed for identifying the situation of identity issue.

Second, when there is a potential identity issue, switcher, the target that confuses the tracker and leads to an identity issue, should be specially taken into consideration. Specifically, when the switcher is used as the extra input, the MOT can be aware of such potential target causing the identity issue and act correspondingly to relieve such issue. As illustrated in Fig. 1.2, the extra information from the tracklet and its states benefits to identify in what cases the single visual cue is not effective and will cause identity issues. This can be done by including extra information which encodes the tracklet and surrounding states (e.g., the tracklet state map we proposed). Then, further extra information from the potential switcher can facilitate a more accurate prediction of relationships between tracked targets and candidates. This can be done by dynamically adjusting the gating strategy w.r.t. the switcher.

Based on this motivation, we propose a new method for MOT, which exploits extra information from multiple cues and combines them in a unified way, for more robust tracking. We propose to introduce the tracklet state map as a feature descriptor to encode information about the environment of a target. Specifically, the tracklet state map is the concatenation of the

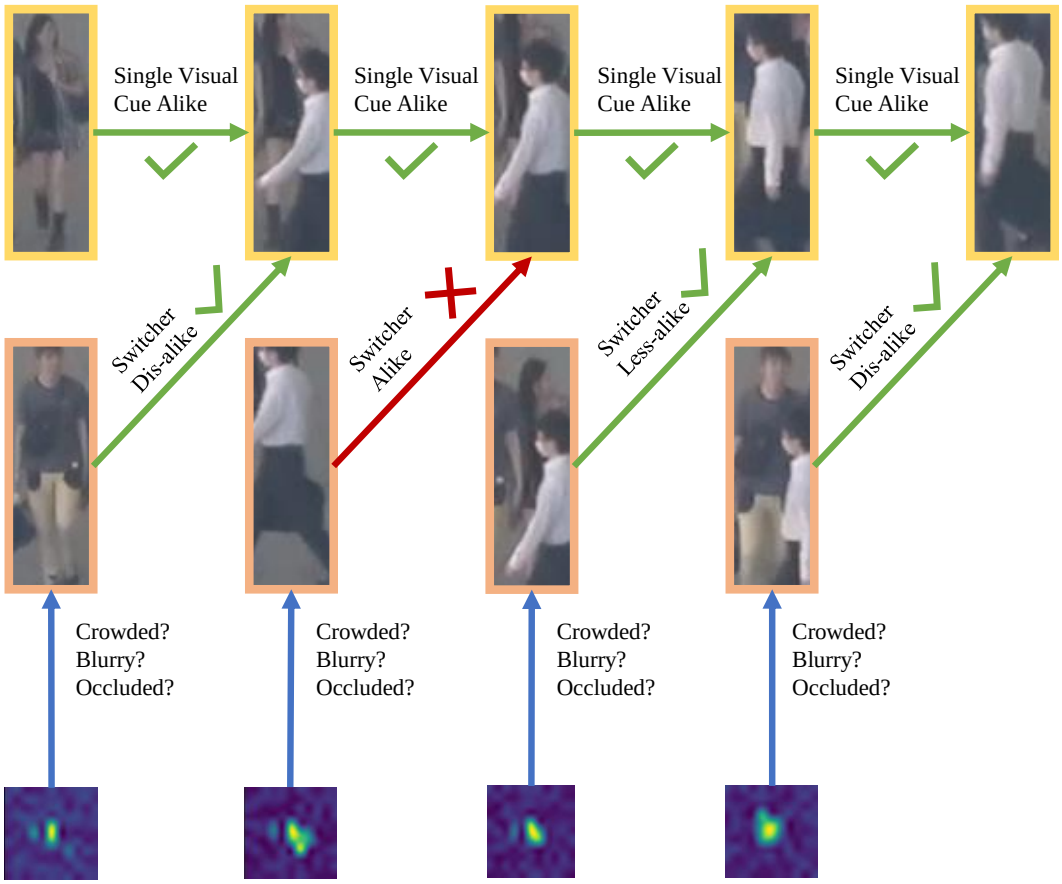


FIGURE 1.2. Including the tracklet state map and switcher as the extra information for the identity issue. When there is a low possibility of switching (frame 1), a single high response on the tracklet state map (bottom row heat-maps) is observed. When the tracklet state maps have multiple high responses, we can find that the target has a high potential to be switched to another object. Thus different strategies for prediction are needed for different identity-switching possibilities. The switcher information then provides more cues for handling potential identity issues.

detection heat map and the target correlation heat map. From the tracklet state map, we can find the spatial distribution of the surroundings, as well as the potential switcher that may cause identity issues. When the switcher is recognized, we include visual cues from the switcher, together with visual cues from the concerned target, and the tracklet state map as the input of the tracking classifier. The tracking classifier learns to make precise predictions based on abundant information and learns to perform different strategies in different situations. The details of the proposed method are presented in Section 3.1.

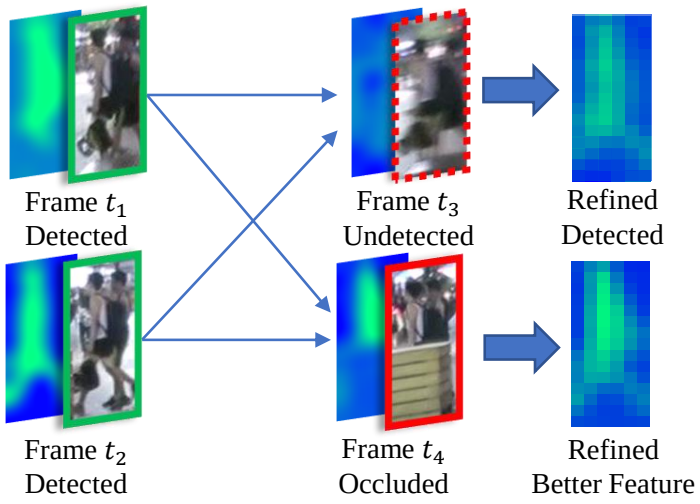


FIGURE 1.3. Examples of instance-level spatial-temporal semantic information that help refine features. Before refinement with features from frame t_1 and t_2, the feature in frame t_3 was not detected and the feature in frame t_4 was of low quality due to occlusion. After refinement, both of them could be corrected.

We also conduct experiments and present an analysis of how extra information from tracklet states and the switcher can facilitate making the tracking more robust. We find that the prediction based on the inclusion of extra information leans to perform differently when the surrounding environment is different. Compared with the single cue model, the proposed method provides a different confidence of prediction when the switcher is different, which provides a dynamic strategy for handling complicated situations. The detailed experiments and discussions are introduced in Section 3.2.

1.2 Exploring the Relation Learning for Joint Detection and Tracking

Recent studies (Kieritz et al. 2018; Zhou et al. 2020; Peng et al. 2020; Pang et al. 2020) on joint object detection and multi-object tracking have successfully constructed anchor-based (Peng et al. 2020; Pang et al. 2020) and anchor-free (Zhou et al. 2020) deep multi-task frameworks. These works have focused on setting up basic joint frameworks that enable simultaneous detection and tracking. Although the feasibility of the joint task has been verified, how to

effectively utilize the spatial-temporal information in video frames for this joint task remains an open problem. A carefully designed mechanism that utilizes instance-level spatial-temporal semantic information can improve the learning of feature representations for subsequent tasks, especially when the visual feature of a target is of low quality due to motion blur or occlusion, as shown in Fig. 1.3. Some recent studies have developed approaches that utilize low-level spatial-temporal information for joint detection and tracking. For example, Wang et al. 2021a provides a pixel-wise correlation learning method to enhance the low-level feature maps using neighboring pixels. Despite showing good effects, pixel-level correlation is not able to model high-level relations, and dense calculations make it difficult to utilize cues outside a local range. In contrast, using instance-level semantic information is a complementary, efficient, flexible, and more explainable approach that has been found to be capable of modeling high-level relations. The sparse distribution of instances compared with the number in pixel-level features allows us to aggregate useful cues from a larger range of spatial or temporal areas, and the high-level semantic features allow us to design meaningful guidance and make the model comprehensible.

Conventional relation learning that utilizes instance-level spatial-temporal semantic information, i.e., refining RoI features using reference RoI features from other frames of the video based on self-guided vanilla multi-head attention (Vaswani et al. 2017), has shown promising performance in some other video analysis tasks (e.g., relation networks by Deng et al. 2019; Chen et al. 2020b in video object detection). In connection with fixed detection MOT (not joint detection and tracking), Xu et al. 2019b also discussed the application of relation networks for re-identification (ReID) feature refinement but did not study the detection features. Obviously, because detection methods are evolving rapidly, involving both detection and tracking features for instance-level relation learning is critical for improving overall performance.

Based on the aforementioned motivation and inspired by the relation learning approaches in single detection **or** tracking tasks, our goal was to construct a relation learning pipeline for joint detection **and** tracking. However, new challenges are found in the joint task that make conventional relation learning ineffective. *First*, the optimization objectives are different for different sub-tasks. The goal of the detection is to minimize the distance between

two embeddings within the same category, but tracking increases the distance when the embeddings have different identities. *Second*, the scenarios of joint detection and tracking are different. In MOT scenarios (Milan et al. 2016; Dendorfer et al. 2020), the categories of interest are usually limited (e.g., vehicles and pedestrians), but the object density is high (up to 200 instances per image). In object detection (COCO, Lin et al. 2014 and ImageNet VID, Russakovsky et al. 2015), the number of categories varies but the density of the target is much lower. These differences make conventional relation networks for general detection less effective in joint detection and tracking tasks because 1) useful information from a different category or background, which is one of the main powers of the vanilla attention mechanism, is limited, and 2) a large number of instances from the same category contain noisy, low-quality references that should not be aggregated. In Section 4.1.2.1, an additional analysis of these two challenges is provided. In addition, previous relation networks involved with tracking-only tasks focused on building better re-identification features and were not designed for joint frameworks. Therefore, a more effective relation learning method is needed to capture relation information better in such joint tasks.

In this study, we developed a joint relation learning pipeline with a novel relation learning attention mechanism called Similarity- and Quality-Guided Attention (SQGA) for joint detection and tracking. Specifically, we placed different SQGA relation modules ahead of different prediction heads to refine instance features using (both spatial and temporal) neighboring instances. Feature aggregation was on the basis of relational affinities, which were factorized into similarity terms and quality terms in the SQGA relation modules. We designed two attention schemes (i.e., a similarity-based scheme and a quality-based scheme) to guide the relation learning. The goal of the similarity-guided attention scheme was to learn a rough similarity function, which was guided by the task annotations, to control the aggregation graph of different feature representations automatically. The goal of the quality-guided attention scheme was to choose fine-grained feature embeddings guided by the output loss magnitudes of task-specific optimization objectives. Both attention schemes were jointly embedded into the attention pipeline to learn the relation encoding of the two tasks. The principal rules provided by the two schemes (i.e., the similarity-based and quality-based schemes) enabled

us to develop task-specific attention loss functions with supervisory information about the two tasks.

To address the learning objective conflicts, different types of task-specific guidance are added by involving different modules to learn task-specific knowledge and to apply task-specific aggregation automatically. To address the issue of differences in category diversity and object density, the SQGA modules were designed to provide more fine-grained attention affinities via a similarity term that helps to avoid aggregation from meaningless distractors and a quality term that helps to avoid aggregation from low-quality noisy references.

Our investigation, described in Section 4.1.1, and experimental verification, described in Section 4.2.3.4 further show that the relational affinities in SQGA are more meaningful and explainable with the help of our new relation learning designs and thus provide better instance-level feature representations capturing spatial-temporal relationships among multiple frames.

1.3 Robustness in Dynamic Frame Rate Scenarios

Despite the promising progress, we argue that current MOT algorithms are still not intelligent enough and cannot satisfy industrial demands since they all work on videos with a fixed frame rate. On the one hand, humans can track objects in videos with diverse frame rates, even in a situation lacking any information on the streaming frame rate. However, the recent State-Of-The-Art (SOTA) trackers (Zhou et al. 2020; Wu et al. 2021; Wang et al. 2021c; Zhang et al. 2021; Zhang et al. 2022) were not robust to frame rate changes. As shown in Fig. 1.4, we evaluated several recent trackers (dashed lines in Fig. 1.4) on MOT17 and MOT20 challenges with different input frame rates and have found that their tracking capability decreases a lot when the frame rate is reduced. The ByteTrack (Zhang et al. 2022) is much better than CenterTrack (Zhou et al. 2020) in the normal frame rate setting but is not advanced in lower frame rate settings, showing that some tracking techniques are frame rate sensitive. On the other hand, different applications require MOT models to work on different sampling rates. Some applications (e.g., autonomous driving) require the tracker to have low latency and

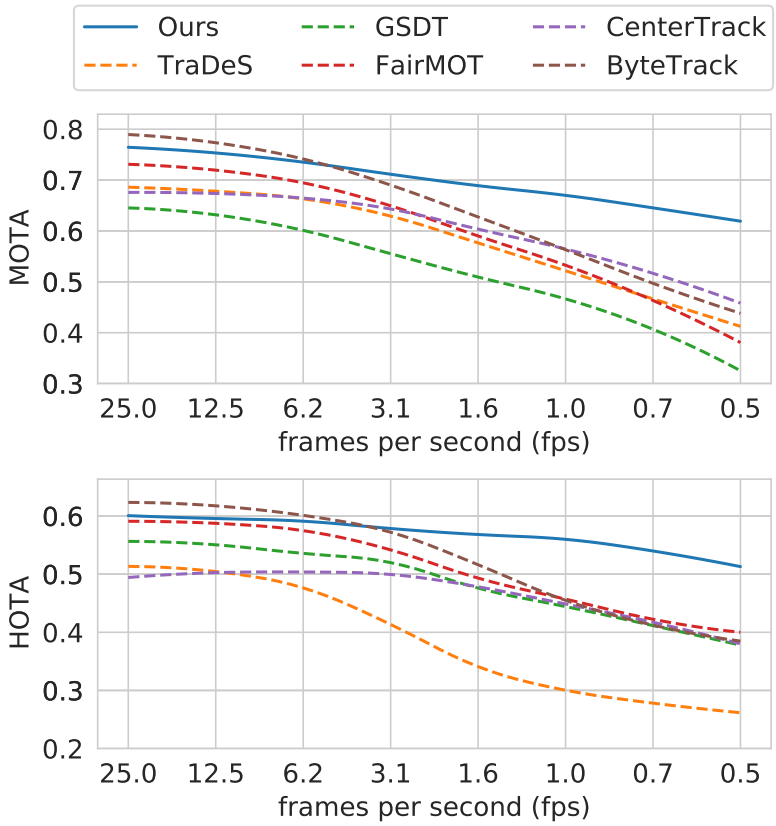


FIGURE 1.4. Performance of recent state-of-the-art trackers at multi-frame-rate settings. Both MOTA and HOTA scores drop dramatically when the frame rate is lowered. Our proposed method has a better capability of handling frame rate changes compared with previous methods.

provide input videos of very high sampling frame rates, but more others (e.g., large-scale trajectory retrieval and tracking-based crowd counting (Han et al. 2022)) only require the MOT algorithms to work on different lower frame rates. These applications care less about the frequent perception of object locations at every single time step but focus more on trajectory integrity along a greater temporal range. Although training and deploying a separate tracker for each frame rate is feasible, this trivial solution is neither convenient nor efficient because developing, selecting, and deploying the best tracker for each application and frame rate is laborious and expensive for large systems. Besides, it assumes that the frame rate is available during testing, which may not always be true. Thus, proposing more general, unified, and frame rate agnostic trackers that can understand videos with diverse frame rates like a human

is not only a reasonable objective for artificial intelligence but also a bandwidth-, storage-, and computation-efficient solution for many real-world scenarios.

To train a unified frame rate agnostic tracker, a straightforward manner is training a model of the classical design on the dataset with multiple diverse frame rates (i.e., frame rate agnostic training). However, this vanilla design does not work well due to the following two challenges.

First, the optimal matching rules of motion-appearance relations are different at different input frame rates. For example, motion cues are usually more reliable when the input frame rate is high, because of a smaller movement between adjacent frames. While in lower-frame-rate settings, appearance cues become more important. Two detections with similar appearances may be unlikely to be judged as the same object in higher frame rate videos given the large spatial distance but are more likely to be the same object in lower frame rate videos. Classical association models lack the mechanisms to well handle these complex motion-appearance relations.

Second, involving multi-frame-rate data in traditional frame-pair association training schemes leads to a larger gap between training and inference. Specifically, post-processing procedures that are not included in the training stage but applied in the inference stage will change the detected object locations, resulting in the input data of the association networks in the training stage being different from that in the inference stage. These changes are small in normal (higher) frame rates and thus can be ignored in the traditional training scheme. However, these changes are enlarged in the lower frame rates and are not negligible in the multi-frame-rate training. Section. 5.1.1 provides a more detailed analysis.

To tackle these challenges and achieve a unified frame rate agnostic tracker, we propose the Frame Rate Agnostic MOT framework with a Periodic training Scheme (FAPS), which consists of two effective techniques. *For the first challenge*, a unified Frame Rate Agnostic Association Module (FAAM) is proposed to handle various frame rate settings. Specifically, FAAM first employs the available frame rate information (e.g., the exact frame rate) to generate the frame rate embedding and infer the frame rate aware attention, which will further be multiplied with the prediction embedding in a channel-wise manner to predict

final association scores. For the cases where the exact frame rate is known, the frame rate embedding can be directly obtained by the frame rate cosine embedding. For the cases where the exact frame rate is unknown during testing, we propose to use the Inter-frame Best-matched Distance Vector (IBDV) to infer the frame rate information. To calculate the IBDV of two adjacent frames, we first calculate the normalized positional distance matrix, and then select the minimum distance of each row in the matrix, sorting them to form a vector. IBDV encodes the distribution of object movements and thus provides information to infer the frame rate. *For the second challenge*, a Periodic Training Scheme (PTS) is designed to enhance the frame rate agnostic training. Before starting the training, we sample the tracking patterns via running previous model checkpoints on the real inference pipeline including all post-processing steps. The tracking patterns record all information we need (i.e., positions, motion predictions, and cached features) to simulate the inference stage environment during training. We assume these patterns of the tracker to have negligible variation within a short period, and thus we divide the whole training procedure into several training periods and only update patterns between periods. During training, instances not matching those patterns will be discarded because they probably do not appear in inference time, thus reducing the difficulties of frame rate agnostic training. The remaining instances will be fused with the recorded patterns to reduce the input variance and will be turned into affinity features. With the proposed approaches, we successfully improve the accuracy of the tracker, especially at lower frame rate settings. Section 5.1 presents a detailed description of the proposed frame rate agnostic solution.

To make a fair evaluation for the FraMOT task, we simulate the multi-frame-rate settings using existing benchmarking MOT datasets and develop metrics for the evaluation. Moreover, we test the methods of FraMOT task in two modes, i.e., known-frame-rate mode and unknown-frame-rate mode, reflecting different deployment demands and bringing a more challenging yet practical MOT problem to the community. Section 5.2.1.1 shows more details of the evaluation method.

1.4 Unsupervised Pretraining for Joint Detection and Tracking

Despite the promising progress on supervised joint detection and tracking (JDT), currently, unsupervised learning for this challenging task seems underdeveloped. Although some previous research works have discussed unsupervised object detection (e.g., Xie et al. 2021; Dai et al. 2021) or unsupervised MOT (e.g., Karthik et al. 2020; He et al. 2019), they are significantly different from the unsupervised joint detection and tracking. Unsupervised object detection is not designed for video-based tasks and is not effective in scenarios with high object density, and some of them could only be applied to specific frameworks (e.g., Dai et al. 2021 is only for transformer-based detection frameworks). For unsupervised MOT, most previous works assume that the object detector is well-trained, and they only provide unsupervised learning of the appearance model. The authors of He et al. 2019 attempted to present a fully unsupervised perception system that automatically detects and tracks objects in videos. However, this method is only effective in very simple scenarios, rendering it unsuitable for real-world applications. Instead, a more realistic way is pre-training plus fine-tuning. The pre-training allows the model to learn from massive unlabeled data in an unsupervised manner, and the fine-tuning further defines the downstream task using limited labeled data.

Investigating unsupervised learning for JDT tends to be significant for the training of new video perception systems, which will be of great benefit for reducing labor costs and addressing challenges of wider scenarios, leading to a more intelligent solution for video processing.

In this study, we for the first time propose a simple but effective solution of self-supervised pre-training for JDT, making the first step towards unsupervised joint detection and tracking system utilizing large-scale unlabeled video data. Specifically, we propose to apply Foreground Trajectory Detection (FTD) on unlabeled videos to obtain trajectory-level pseudo labels for self-supervised JDT model pre-training.

In the proposed FTD method, the first step involves filtering all unlabeled videos by thresholding the difference ratio (DR) of adjacent frames. The larger adjacent DR indicates the larger camera motion. To obtain better trajectories, we choose those video sequences with a lower range of DR. Then we perform the foreground discovery method (e.g., the selective search by Uijlings et al. 2013 was used in our study) on multiple image differences of nearby frames and filter these proposal boxes using multi-frame consistency. Only proposals that appear in multiple frames will be recognized as stable foreground objects and these multiple proposals will naturally form trajectories. After the pseudo-labels of trajectories are obtained, then any joint detection and tracking models could be pre-trained with these pseudo-labels.

The experiments followed the standard pre-training and fine-tuning pipeline as those commonly used in unsupervised backbone pre-training or detector pre-training. Section 6.1 shows more details of the proposed self-supervised solution for JDT.

1.5 Summary of Contributions

This thesis addresses the aforementioned challenges in the joint detection and tracking task by presenting innovative methods that serve as a stepping stone towards the development of next-generation joint detection and tracking systems. To address the distractor problem, we have introduced the concept of a "switcher" and a switcher-aware association model to leverage distractor information. To tackle the relation learning problem, we have proposed an attention mechanism called similarity- and quality-guided attention (SQGA) for more effective feature refinement. To enhance frame rate robustness in multiple scenarios, we have proposed a frame rate agnostic framework that can perform joint detection and tracking with different frame rates. For unsupervised learning using un-annotated video data, we have developed a video-based pseudo-labeling method for model pre-training. We have evaluated these approaches through both qualitative and quantitative experiments, and the results have demonstrated their effectiveness in addressing the challenges.

CHAPTER 2

Literature review

In this section, a thorough review of previous studies in the field of visual object detection and tracking is presented.

2.1 Paradigms

The goal of visual object detection and tracking is to discover, locate, and identify any objects of interest in a video. There are mainly two paradigms to achieve this goal, i.e., tracking-by-detection and tracking-by-SOT (single object tracking).

2.1.1 Tracking by Detection

The tracking-by-detection (TBD) paradigm typically requires a detector and a data association procedure. The first step is to detect any objects of interest in the given video, and then all objects of the same identity are figured out through data association. Most methods in the related field followed this paradigm. Specifically, the TBD paradigm requires an object detector to provide object candidates and requires a tracker to implement the data association and trajectory management. The object detector usually consists of a feature extractor, some detector heads, and post-processings. The tracker takes care of the trajectory initialization, state updating, and termination. In joint detection and tracking systems, the detector and the association could be jointly trained and coupled.

2.1.2 Tracking by Multiple Single Object Tracker

The tracking-by-SOT paradigm does not conduct dense detection of objects throughout the video. Instead, only the first frame of a batch will be fed to the object detector. Because for some frames the detection results are missing, data association is not always possible. However, we can figure out the targets using multiple single-object trackers. Data association will only be conducted between the previous batch's last frame and the next batch's first frame. In other words, the TBD paradigm is a particular case where the batch size is set to one frame. In some situations, the detector is removed, and the targets to be tracked are given by the user. The core method of this paradigm is the single object tracker, which aims to figure out the target in future frames, given a template of the target.

2.1.3 Comparison between the Two Paradigms

The tracking-by-detection (TBD) and tracking-by-SOT paradigms each present distinct advantages and drawbacks. TBD excels in precise perception. By collaborating with an object detector, the tracker eliminates the need for a regression module to fine-tune object locations and mitigates background noise interference. This allows the model to concentrate solely on identity prediction and comparison. However, TBD encounters limitations with open set inputs where the object detector fails to provide accurate candidates. In contrast, tracking-by-SOT offers greater robustness, particularly in scenarios where the accuracy of the object detector is low. Moreover, TBD typically demands more computing resources compared to the tracking-by-SOT paradigm. Consequently, TBD is favored in stable scenarios with ample computing resources, while tracking-by-SOT is preferred in data-limited situations where low-latency algorithms are essential.

2.2 Feature Extractor

For either paradigm, a feature extractor is a critical component. The history of image feature extractors spans several decades. Prior to the advent of deep learning, researchers developed

numerous feature descriptors to generate feature representations. Later, with the introduction of large-scale learnable parameters, the learnable deep feature extractor was born.

2.2.1 Descriptors

Prior to 2012, researchers developed numerous image feature descriptors that were computationally efficient and demonstrated relatively good performance. These descriptors were developed based on the extensive experience of image-processing researchers. Classical hand-crafted descriptors, such as histograms of local orientation (Freeman and Roth 1995), shape contexts (Belongie et al. 2000), and local scale-invariant features (SIFT) (Lowe 1999), were among the many descriptors developed during this period. Subsequently, several studies improved upon these results and reduced computational costs. For example, Dalal and Triggs 2005 introduced the Histogram of Oriented Gradient (HOG) descriptor by combining the benefits of edge orientation histograms, SIFT descriptors, and shape contexts. Additionally, Calonder et al. 2010 developed an efficient feature point descriptor using binary strings, called Binary Robust Independent Elementary Features (BRIEF), while Rublee et al. 2011 introduced a more efficient alternative to SIFT called Oriented FAST and Rotated BRIEF (ORB). Despite the cost-effectiveness and appealing properties of these descriptors, they cannot fully utilize the more powerful computational resources available today and are unable to self-evolve as the data scale increases.

2.2.2 Simple Neural Networks

The concept of artificial neural networks has been around for a long time, but it only became functional after the demonstration of backward propagation by Rumelhart et al. 1986 for training a simple neural network. A basic neural network consists of several linear layers and some non-linear units, such as the Sigmoid function or the Rectified Linear Unit (ReLU) as described in Nair and Hinton 2010. Typically, these layers are divided into the input layer, the hidden layers, and the output layer, with a non-linear function after each layer. Initially, the performance of neural networks was limited by the scale of the parameters due

to computational power and the number of annotations. However, with the development of computing devices and larger datasets, neural networks have outperformed many traditional hand-crafted image descriptors. Despite this, simple neural networks have their limitations, such as the lack of a local attention mechanism, which has been shown to be effective when processing images.

2.2.3 Convolutional Neural Networks

Compared to simple neural networks, Convolutional Neural Networks (CNNs) are more powerful for image processing. The key idea of CNNs is local attention, implemented by convolution kernels. A neural unit in the next layer is only connected to nearby units in the previous layer. Fully convolutional neural networks further provide translation invariance. LeNet (LeCun et al. 1998) was one of the earliest CNNs, which established a CNN architecture of convolution, upsampling, and full connection. In 2012, AlexNet (Krizhevsky et al. 2017) ushered in a new era of deep networks, as it was trained with the first ultra-large image dataset ImageNet (Russakovsky et al. 2015). Later, larger and deeper networks with more parameters were proposed, such as VGGNet (Simonyan and Zisserman 2014), GoogLeNet (Szegedy et al. 2015), and ResNet (He et al. 2016). These deeper CNNs provide promising feature extraction capabilities, which largely improve downstream tasks.

2.2.4 Recurrent Neural Networks

Recurrent Neural Network (RNN) is another extension of neural networks for sequential tasks, e.g., natural language processing (NLP) and video processing. The core of RNN is the RNN cell, which updates the previous state using new inputs. Classical RNN cells include the Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber 1997) cell and the GRU (Cho et al. 2014). RNNs provide the capability of long-term and short-term memory. In visual tasks, RNNs are usually designed as a middle part of a framework, accepting CNN features and followed by classifiers.

2.2.5 Visual Transformers

Transformers (Vaswani et al. 2017) are token-based neural networks, which can process spatial and temporal attention in a unified way. Each pixel or image patch is regarded as an input token, and each output token of an attention layer (the core of the transformer) is the sum of the input token and the calculated attention vector. Moreover, some Feed-Forward Networks are appended after the attention layer to provide the ability for prediction and regression. Transformers can be used with CNNs or used alone. Dosovitskiy et al. 2020 developed the full transformer backbone for computer vision tasks. The proposed ViT model performs better than some classical CNNs.

2.3 Object Detector

In either Tracking-by-Detection or Tracking-by-SOT paradigm, the detector is necessary for automatic trajectory initialization.

2.3.1 Earlier Methods

In the early days of object detection, the task consisted of two sub-tasks: proposal of Regions of Interest (RoI) and classification. One straightforward method of proposing RoIs was to generate every possible RoI, which created many proposals. Later, hand-crafted unsupervised algorithms, such as the selective search method (Uijlings et al. 2013), were proposed to eliminate most impossible proposals. Once the proposals were generated, feature representations of these proposals were extracted from the corresponding image areas, and a classifier such as Support Vector Machine (SVM, Hearst et al. 1998) was trained to classify these feature vectors into different labels. R-CNN (Girshick et al. 2014) is a classical method in this category that uses Convolutional Neural Networks (CNNs) as the feature extractors and SVMs as classifiers. SPPnet (He et al. 2015) removed the repeated CNN feature extraction, instead generating a large feature map, and extracting the RoI features from this CNN feature map using the proposed SPP-layer. Although the number of proposals was significantly reduced

and CNN extraction was performed only once, the computational cost of this pipeline was still very high. Additionally, the system could not be learned in an end-to-end manner.

2.3.2 End-to-end Methods

In order to tackle the aforementioned issues, Girshick 2015 proposed a solution called Fast RCNN, which combined the benefits of RCNN and SPPnet. Fast RCNN utilized an RoI pooling layer to extract feature embeddings from the large CNN feature map and replaced the SVMs with a classification and regression branch implemented by neural networks, thus establishing the first end-to-end and rapid pipeline for object detection. Subsequently, Ren et al. 2015 proposed the region proposal network (RPN) that makes the proposal process learnable and removes more impossible proposals before the classification and regression stage. This framework established the first end-to-end anchor-based detection structure. In this phase, the manually designed region proposal was eliminated, speeding up the detection algorithm and enabling the entire detection model to be updated by a single supervision signal.

2.3.3 Anchor-free Methods

Although the end-to-end detection model was established, the definition of the anchors in RPN still remains a hyper-parameter that varies in different scenarios. In some cases, the anchors could not cover the targets perfectly. To address this issue, anchor-free methods are developed. Law and Deng 2018 and Duan et al. 2019 developed two point-based detection frameworks, respectively named corner-net and center-net, eliminating the anchor setting in anchor-based detection frameworks. The You-Only-Look-Once (YOLO) detection system series (Redmon et al. 2016; Redmon and Farhadi 2017; Redmon and Farhadi 2018; Bochkovskiy et al. 2020; Ge et al. 2021) also run in an anchor-free manner, but the purpose of them is to accelerate the detection and they did not figure out the actual drawbacks of anchor-based frameworks (except the speed). Both streams have made anchor-free detectors more popular.

2.4 Temporal Detectors

Other than still image detection, video object detection (VID) is also an important task. Video object detection typically uses several frames as input and models the temporal correlations between the frames, both of which are particularly important for improving the detection performance in videos. Several studies continued this line of thought by focusing on developing feature correlation models to build the temporal connections of the frames on either the pixel level (Zhu et al. 2017; Bertasius et al. 2018; Wang et al. 2018a) or the instance level (Xiao and Jae Lee 2018; Wang and Gupta 2018; Shvets et al. 2019; Deng et al. 2019; Chen et al. 2020b). Specifically, the flow-guided feature aggregation (FGFA) study (Zhu et al. 2017) proposed a flow-guided scheme to provide a dense warping between consecutive frame representations with the purpose of identifying the consistency along the time dimension. The goal was to obtain a pixel-level feature aggregation and develop the first deep-learning-based feature propagation method that could utilize spatial-temporal information in videos. This propagation enriches the feature maps and thus results in better detection accuracy in video object detection. However, the optical flow operation in this method is time-consuming and the warping range is thus limited. To achieve a larger warping range, the authors of the spatial-temporal sampling networks (STSN) method (Bertasius et al. 2018) proposed a different cross-frame warping method using deformable convolution rather than relying on the expensive optical flow computation.

Although modeling pixel-level relations via warping methods has led to improvements in accuracy, the method is still limited by its pixel-level correlation modeling, which is not able to model high-level relations, and its dense calculation makes it difficult to utilize cues outside a local warping range.

To obtain more flexible representation correlations, instance-level relation models have been further developed by researchers. Wang and Gupta 2018 introduced an attention model to learn the instance relationships across different frames and had a significant boost in model performance. Shvets et al. 2019 devised an appearance-based instance-level feature aggregation network using a non-local attention mechanism (Wang et al. 2018b); this method

has the ability to capture long-term temporal relationships for video frame representations. These two methods show the feasibility and flexibility of instance-level attention mechanisms in object detection. Later, Deng et al. 2019 and Chen Chen et al. 2020b established relation networks with more effective attention mechanisms, further simplifying the training process and improving accuracy.

In the field of self-supervised detection, some studies utilized correlations between frames as supervision signals of consistency learning. For example, Fang et al. 2022 developed a self-supervised traffic accident detection framework based on the spatial-temporal relation consistency of road participants.

2.5 Single Object Tracker

Single-object trackers aim to figure out the positions of the target of interest in the following frames.

2.5.1 Kernel Methods

In the pre-deep-learning stage, the single-object-tracking task was formulated as a kernel optimization problem (Bolme et al. 2010; Henriques et al. 2014). The target is to find the best parameters of a convolutional kernel that minimize the losses given some positive and negative samples of the target. Although having a high processing speed, the kernel methods were not robust to the changing of scales, shapes, and appearances, due to camera motion, switch of view, and occlusion.

2.5.2 Siamese Models

Later, the online learning kernels were replaced by deep features generated by the offline-trained deep models. Bertinetto et al. 2016 proposed Siamese-FC, which used identical two-branch networks for feature extraction of the template and search region. The correlation map is the outcome of the convolution operation between the template and search region features.

This approach provided a more robust model for appearance discrimination. However, the changing of scales and shapes was still difficult to handle. Later, Li et al. 2018a; Li et al. 2018b proposed the Siamese-RPN model series, which enabled the automatic regression of the bounding box and improved the accuracy.

2.6 Data Association

Most research works of Multi-Object Tracking (MOT) follow the tracking-by-detection paradigm, where data association is the core procedure. The core problems for data association are learning the matching scores on the edges of the graph and finding out the optimal graph partition based on these scores.

2.6.1 Matching Scores

To calculate the matching scores, two factors should be under consideration, i.e., what cues to include and how to generate effective feature embeddings. Son et al. 2017; Sadeghian et al. 2017; Milan et al. 2017; Huang et al. 2008; Bae and Yoon 2014; Bae and Yoon 2018; Zhou et al. 2019; Sun et al. 2021 emphasize improving the effectiveness of the features used in data association. Sadeghian et al. 2017; Milan et al. 2017; Kim et al. 2018 exploit RNNs in the MOT task. Milan et al. 2017 focuses on the utilization of positions and motions of the targets. Son et al. 2017 develop a new training method with ranking loss and regression loss to obtain higher accuracy. Sun et al. 2021 propose a joint learning network for both appearance feature and affinity learning.

However, they only focus on the improvement of a single cue and may be less robust when the single source cues are unreliable.

Xu et al. 2019a apply relation networks for better feature aggregation, which is another method for feature improvement. They try to figure out the general relationships among targets but still did not dive into the most confusing switchers' information, which we believe is helpful for MOT.

Sadeghian et al. 2017 combine appearance, motion, and interaction cues into a unified RNN network. The interaction cues encode spatial information from neighboring identities. However, they do not use information from the switcher and the spatial information of interaction cues is hand-crafted, which only preserves the density without the consideration of any visual cues. Some recent works (Long et al. 2018; Yoon et al. 2018) ensemble appearance features with position and motion features. Their designs in using these information sources are based on heuristic rules, but not based on learning.

Most of these works just regard data association as a pairwise matching problem between a single tracklet and the detection using a single source of cues. However, these works do not focus on how to exploit multiple (learnable) cues and the information of potential switchers for robust tracking, which is critical to solving identity issues.

2.6.2 Optimal Graph Partition

Given the matching scores that form a weighted graph, the next step of data association is to find out the partitions maximizing the objective score. Brasó and Leal-Taixé 2020; Xu et al. 2019b; Hornakova et al. 2020; Tang et al. 2017; Zhang et al. 2008; Tang et al. 2015; Tang et al. 2016; Pirsiavash et al. 2011; Zhang et al. 2010; Xiang et al. 2020; Sheng et al. 2020; Chen et al. 2017a formulate the data association process as various graph partition problems. Among them, Tang et al. 2017; Zhang et al. 2008; Zhang et al. 2010; Tang et al. 2015; Tang et al. 2016 formulate the problem as variants of graph segmentation problems and they need batch processing. Zhang et al. 2008 model the data association as a network flow problem, finding the relationships among targets by figuring out the maximum or minimum flow of a network graph. Tang et al. 2016; Tang et al. 2017; Tang et al. 2015 propose to solve the problem by sub-graph decomposition or multi-cut. Hornakova et al. 2020 further develop the multi-cut based on previous research. Most online processing methods use Hungarian Algorithm (Munkres 1957) to solve a bipartite graph matching problem. Moreover, Brasó and Leal-Taixé 2020; Xu et al. 2019b try to optimize the matching scores and figure out the best matches in an end-to-end manner. Sheng et al. 2018; Sheng et al. 2020 try to solve the MOT problem via multiple hypothesis tracking (MHT) with different graph settings. Xiang

et al. 2020 proposed a CRF method for association with end-to-end deep features learning. Chen et al. 2017a extend the graph optimization problem to multiple camera tracking. For now, most of these partition methods can figure out trajectories of good quality if the given weighted graph is accurate, therefore, the quality of the matching scores is the most critical part of data association.

2.7 Temporal Trackers

As an extended stream of the utilization of historical information, modeling temporal correspondence could better improve the tracker performance. To learn the temporal correspondence, several research studies proposed using attention mechanisms to model spatial-temporal relationships Chu et al. 2017; Zhu et al. 2018; Xu et al. 2019b for data association based on bounding-box representations. Chu et al. 2017 developed a quality-based attention mechanism to ignore low-quality samples (e.g., occluded pedestrians). Zhu et al. 2018 developed an attention mechanism to reduce noisy bounding boxes from given fixed detection inputs. More recently, Xu et al. 2019b developed a spatial-temporal relation network to learn the combination of different cues for more accurate similarity measurement. Instead of focusing on similarity learning of bounding-box representations, Wang et al. 2019 proposed modeling long-term spatial-temporal relationships on tracklets. Nevertheless, these works only considered relation modeling using the factors affecting tracking (assuming that fixed detection inputs are given) and focused more on modeling relations between different types of cues (e.g., motion, visual, and topological cues).

2.8 Reidentification

The problem of long-range recall in multiple object tracking is similar to the reidentification (ReID) task, except that the changing of point of view does not happen in MOT. Thus, some ReID techniques are borrowed to improve tracking robustness. In ReID, the loss function plays an important role in discriminative feature learning, for example, triplet loss and quadruplet loss (Chen et al. 2017b) can help make the feature embedding more discriminative via some

ranking mechanisms. Moreover, part-level features or pose key points were also helpful. (Zhao et al. 2017; Sun et al. 2018) developed part-level features for ReID that enables an aligned comparison between different targets. (Su et al. 2017) utilized the pose key points as additional features to aid the appearance comparison. However, in single-camera MOT scenarios, the most common approaches are adding a ReID sub-network or adding a ReID branch using some ranking losses like triplet loss, and the part-level features or key points are less effective due to the high stability of the camera view.

2.9 Joint Detection and Tracking

Recently, joint deep framework design has become a trending research direction that has attracted great interest from the research community (Peng et al. 2020; Sun et al. 2020; Zhang et al. 2021; Zhou et al. 2020; Wang et al. 2020; Wang et al. 2021a). Several different framework designs, either anchor-based or anchor-free, have been proposed for the joint detection and tracking task. These frameworks aimed to establish and verify basic joint detection and tracking functions, so they did not develop mechanisms to utilize spatial-temporal information in videos. For example, Peng et al. 2020 proposed a chained anchor-based method to perform tracking and detection simultaneously. To alleviate the complicated settings of anchors, Zhou et al. 2020 proposed an anchor-free point-based framework. Later, Zhang et al. 2021 further improved the anchor-free framework by leveraging the learning of the detection and tracking branches. These works focused on building basic frameworks for the joint task and did not exploit the spatial-temporal information from the video input. Some other studies attempted to exploit more cues across a spatial and temporal range but did not consider detection and tracking as a whole. For instance, the authors of the D&T study (Feichtenhofer et al. 2017) proposed a two-branch network that used the predicted instance tracking and classification results to identify dense correlations between representations from consecutive frames. This study used tracking as a feature propagation method for detection but did not actually consider the tracking scenarios. To obtain discriminative feature learning and effective object association, Wang et al. 2020 introduced a graph convolution network (GCN) to model the relationships between tracked objects and detections in the spatial domain;

this process benefited the association but did not improve the detection quality. Another way to model temporal cues is 3D convolution. Pang et al. 2020 established a 3D CNN framework for joint detection and tracking. However, it was designed as tracklet detector and did not truly learn a tracking-specific feature, which is critical for robust tracking. The works mentioned above did not establish an effective design for utilizing spatial temporal information for both detection and tracking in a joint learning manner. More recently, Wang et al. 2021a proposed a correlation network for joint detection and tracking that focused on task-irrelevant pixel-level spatial-temporal feature aggregation. This method learned low-level feature maps better, leaving high-level task-specific relations still not utilized. Utilizing such high-level task-specific relations can provide a more flexible, comprehensible, and powerful mechanism for aggregating information from a video. Because transformers have been shown to be effective in many tasks, Yu et al. 2022 applied transformers to model relations between low-level tokens. However, their method still did not include instance-level relation learning, and it learned relations for only the tracking branch. None of these mentioned methods investigated how specifically designed attention mechanisms can be effectively adapted; nor were they able to model instance-level relations for both the detection and tracking tasks.

Moreover, because the scenarios of MOT are significantly different from object detection or single object tracking tasks, new mechanisms for relation learning are desirable.

2.10 Trajectory Management

Trajectory management includes three different stages, i.e., trajectory initialization, state updating, and termination, all of which are processed based on the data association results.

2.10.1 Initialization and Termination

Most algorithms follow a straightforward rule of tracklet initialization, i.e., a detection will be initialized as a new tracklet if it is 1) not associated with any existing tracklet, and 2) not below the confidence threshold. Some methods may have more fine-grained conditions, such

as only accepting tracklets longer than a length threshold. To terminate a tracklet, the most common rule is to remove those not updated for a certain number of frames.

2.10.2 State Updating

Based on the results of the data association, there will be two different types of updating procedures, i.e., updating with an associated new instance, or updating without any matching instance. In an updating procedure, the position, the feature template, and the status will usually be modified according to new input (a matched target or nothing). For position, the most common updating method is the Kalman Filter, with pre-defined estimation noise and system noise. For feature templates, a FIFO queue is usually helpful. Some methods have their particular designs for the trajectory status, for example, a classical status configuration will be New, Tracked, Occluded, and Lost. Different strategies are applied to trajectories with different statuses. This part of post-processing is based on hand-crafted strategies and is usually not learnable.

2.11 Datasets

Currently tracking datasets include MOT16/17/20 series (Milan et al. 2016), KITTI(Geiger et al. 2013), HIE(Lin et al. 2020), SOMPT22(Simsek et al. 2022), and so on. The MOT17/20 datasets are the most frequently utilized tracking datasets, which include identity labeling. However, manually labeling identities in tracking datasets can be time-consuming. The number of identities in tracking datasets varies between hundreds and thousands, which is usually much less than the number of bounding boxes (which can be up to 1 million boxes). The video resolutions typically range between 720p and 1080p, and the sampling frame rates typically range from 20 to 30 frames per second. The majority of datasets consist of numerous stationary or moving perspectives of urban environments, where the objects of interest are generally pedestrians or vehicles.

Despite the fact that video datasets provide a wealth of information, prior research studies have paid little attention to their resilience when utilizing these datasets. Specifically, previous

studies have presumed that the frame rates of input videos range between 25 to 30 fps, which may not be the case in real-world industrial settings.

Besides tracking datasets, some other datasets also contributed to the MOT task either in training or in testing. For example, object detection datasets, e.g., COCO(Lin et al. 2014) and CrowdHuman(Shao et al. 2018) are used to improve the detector performance; some ReID datasets, e.g., CUHK(Li et al. 2012; Li and Wang 2013; Li et al. 2014) and Market-1501(Zheng et al. 2015) are used to improve the appearance feature (typically for pedestrian tracking).

2.12 Data Augmentation and Unsupervised Learning

The data augmentation strategies employed in the joint detection and tracking domain are akin to those utilized in object detection and re-identification. These methods involve flipping, transforming, mixing, randomly cropping, and combining images, thereby enhancing the labeled training data.

Looking at it from a different perspective, unannotated data may also prove beneficial. A novel learning pipeline for contemporary AI applications involves pre-training on extensive unlabeled datasets and then finetuning on limited labeled datasets. In terms of backbone pre-training, various unsupervised learning methods have been proposed utilizing unstructured images (such as those from ImageNet). Several of these methods utilize distinct variants of contrastive learning, treating the original image or its augmented versions as positive samples and the remaining as negative samples (e.g., Chen et al. 2020a; He et al. 2020; Caron et al. 2020). There are only a limited number of investigations into pre-training detection heads utilizing a general configuration that is agnostic to the detection framework it is being employed with (e.g., Wang et al. n.d.). On the other hand, the majority of these studies are based on particular detector frameworks, such as DETR(Carion et al. 2020). For example, UP-DETR(Dai et al. 2021) was designed for DETR. Regarding pre-training of reidentification models, Fu et al. 2021 recommended executing contrastive learning on a vast

multi-view unlabeled dataset named LU-Person, while Fu et al. 2022 suggested leveraging pre-trained human detectors and manually-crafted trackers (non-parametric) for weakly supervised learning.

Nonetheless, the aforementioned investigations failed to furnish a comprehensive evaluation of unsupervised learning for joint detection and tracking models with multiple branches, designed to perform diverse tasks. The conundrum of training all the trainable parameters of joint detection and tracking models without a robust label, and conducting unsupervised learning in this domain, is still an open challenge that necessitates additional research and deliberation.

2.13 Evaluation Methods

To evaluate the ability of a joint detection and tracking system under a general definition, the precision, recall, localization accuracy, and identity consistency should be considered and balanced. CLEAR metrics (Bernardin and Stiefelhagen 2008), the most popular MOT evaluation metrics up to now, propose to use Multi-Object-Tracking Accuracy (MOTA) as the performance indicator for MOT. MOTA considers the False Positive (FP), False Negative (FN), and IDentity SWitch (IDSW) as three critical terms for MOT. However, the IDSW number is usually far smaller than the FN number, which makes detection quality dominate the task. Later then, the Identity metrics (Ristani et al. 2016) and Higher Order Tracking Accuracy (HOTA) (Luiten et al. 2021) metrics are proposed to solve such a problem. The core metric of Identity metrics, IDF1 score, is designed to evaluate MOT in the multi-camera MOT task. IDF1 basically indicates the largest ratio of consistent tracking. Due to the optimization problem in Identity metrics, IDF1 takes a long time to compute, especially when the video to be evaluated contains thousands of frames and hundreds of objects. HOTA is then proposed to solve these drawbacks of MOTA and IDF1. In the experiment of the HOTA metric, surveys were conducted and it shows that HOTA can reflect the judgments of human beings about tracking quality in a more accurate way.

Although these metrics provide a general definition of tracking quality, it is still difficult to predict the actual performance in some complicated scenarios, e.g., a so-called better tracker may become much worse when the environment changes. One usual environment complex is the change of input frame rate. Unlike the change of resolution size, changing the frame rate leads to a more complicated setting which usually involves dozens of modifications of the method details. Therefore how to design evaluation metrics for joint detection and tracking system is still an open problem.

2.14 Summary and Discussion

2.14.1 Identity Issues

Previous studies had a very limited discussion on what causes an identity issue and what more cues we could utilize to aid the task. The majority of efforts are directed towards improving detectors or feature extractors, a crucial endeavor but one that falls short of addressing the full scope of the challenge. Errors tend to accumulate throughout the processing pipeline, leading to identity switches, yet research on effectively addressing these issues remains relatively under-explored. In our research about critical distractors (switchers), we introduce a new method that combines multiple cues and the switcher information for robust tracking. Our proposed method takes the most possible switcher into training and inference and combines the tracklet state cues and historical appearance cues in a data-driven way. With extra information from tracklet states and potential switchers, the tracker is able to handle the challenging identity issues.

2.14.2 Relation Learning

Previous studies in the field of video object detection and visual object tracking clearly demonstrated the importance of instance-level correlation learning from videos; however, these explorations of instance-level relation learning were only conducted on a single task, i.e., video object detection. In addition, Qiu et al. 2020 proposed aggregating hierarchical

segmentation information into detection features using attention mechanisms. This was an attempt to model cross-task relations into a detection task. However, it only focused on low-level features and is not feasible for joint detection and tracking. Because their application scenarios and overall objectives are different, directly applying relation networks to joint detection and tracking does not improve accuracy, so it remains unclear how to perform instance-level relation learning effectively in the joint task of detection and tracking. In our proposed SQGA approach, both the spatial and temporal visual cues from detection and tracking were coupled in a unified deep framework with well-designed relation modules and fine-grained attention mechanisms.

2.14.3 Frame Rate Robustness

Our stance is that a strong tracker must possess the ability to accommodate varying frame rates. Nevertheless, past approaches have relied on the presumption that the input videos maintain a high sampling frame rate, rendering them susceptible to frame rate reductions.

Our investigation into frame rate agnostic trackers involved addressing the issue through a redesign of the association model architecture and the implementation of a post-processing aligned training scheme.

Furthermore, due to the inadequacy of current evaluation methods for frame rate agnostic multi-object tracking (MOT), we sought to suggest new evaluation methods and provide a comprehensive assessment of tracking quality in various application scenarios.

2.14.4 Unsupervised Pre-training

Previous research has not emphasized unsupervised pre-training of joint detection and tracking models, which is a crucial aspect of this field, particularly when the task of labeling tracking datasets is laborious. In our approach, we introduced a novel method that harnesses video-level unlabeled data for trajectory-level pseudo-label generation. Our technique exhibits better performance than unsupervised image-based approaches and offers a comprehensive

unsupervised pre-training solution that can be easily adapted to most joint detection and tracking frameworks.

CHAPTER 3

Switcher-aware Multi-Object Tracking with Multiple Cues

This chapter introduces our study on the distractor problem and our switcher-aware MOT model framework.

3.1 Methodology

An overview of the proposed framework is shown in Fig. 3.1. For every frame t, the image and detection results first pass through a pre-processing module to eliminate detection results corresponding to background regions (Section 3.1.1). Then the proposed switcher aware association module (SAA) predicts more reliable matching probabilities of the detection results in frame t and the tracked targets in frame $t-1$ (Section 3.1.2). Based on the matching probabilities predicted by the SAA, Hungarian Algorithm(Munkres 1957) is used for associating the tracked targets in frame $t-1$ and the detection results in frame t. The target management initializes the new targets and terminates the disappeared targets. The procedure above for frame t will loop for the next frame until no more frames are left.

Mathematical formulation. At frame t, given the trajectory set $\mathcal{T}$ and detection result set $\mathcal{D}$, the trajectory of one tracked target can be denoted by $\mathbf{X} = \{\mathbf{x}_k | 1 \leq k < t\} \in \mathcal{T}$, where $\mathbf{x}_k$ is the bounding box of $\mathbf{X}$ at frame k. $\mathcal{I}$ is the whole image input. $\mathcal{I}^{\mathbf{X}} = \{\mathcal{I}_{\mathbf{x}_k}\}$ is the set of image regions of the target for the trajectory $\mathbf{X}$ at different frames. For a detection result at the frame t, its bounding box is denoted by $\mathbf{d} \in \mathcal{D}$ and its image region is $\mathcal{I}_{\mathbf{d}}$. Our objective is to build a scoring function $\mathcal{S}(\mathbf{X}, \mathbf{d} | \mathcal{I}, \mathcal{T}, \mathcal{D})$ that calculates a reliable score about

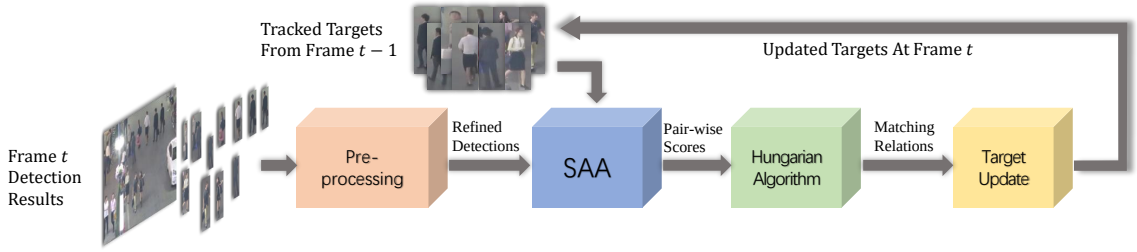


FIGURE 3.1. Overall design of the proposed MOT framework. It consists of four major parts, the pre-processing module for eliminating false positives (FPs), the Switcher-Aware Association (SAA) module for associating detection results based on multiple cues, the Hungarian Algorithm, and the target management. At every frame t, the pre-processing module accepts raw detection results as input and outputs refined detections for SAA. SAA generates the pair-wise scores between the refined detections and tracked targets. Then Hungarian Algorithm is applied to figure out matching relationships and then all tracked targets are updated with the new detections.

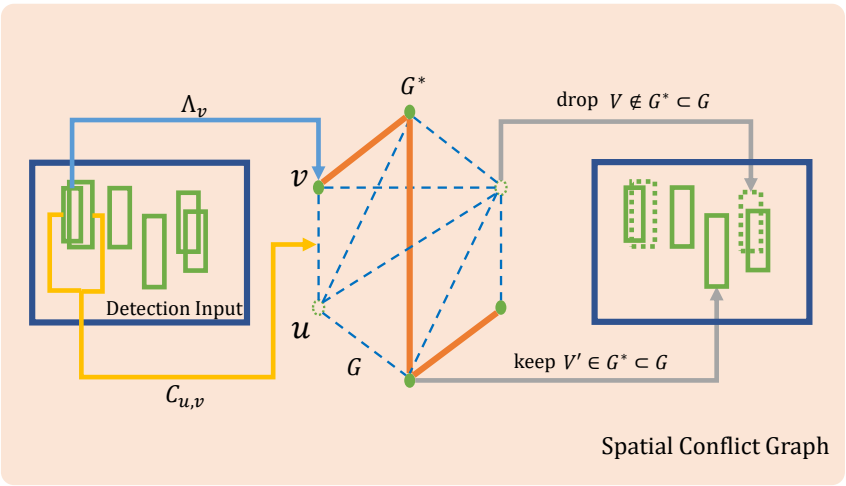


FIGURE 3.2. The outline of Spatial Conflict Graph (SCG) for eliminating false positives. SCG is built with each detection result as a node and the conflict between two detection results as an edge. For detection u and v, node scores Λ_v and edge scores $C_{u,v}$ are calculated for the graph. The original graph G contains both dashed and solid lines for edges and nodes. Then the optimal subgraph G^* and its corresponding detection results are figured out. The optimal subgraph G^* ($\subset G$) contains only solid lines for nodes and edges. The nodes in G^* correspond to the refined detection results.

the matching probability of tracked target $\mathbf{X}$ and detection $\mathbf{d}$. Then, the online MOT problem can be solved by Hungarian Algorithm as a bipartite graph matching problem.

3.1.1 Pre-processing

For a better input of the Switcher-Aware Association (SAA) module, we first need to filter out the false positives (FPs) of the detection input, which are background regions falsely predicted to be objects in the detection results. Otherwise, the FPs will be treated as switchers and confuse the SAA module. We build a Spatial Conflict Graph (SCG) model to handle the FPs. The pipeline of Spatial Conflict Graph (SCG) for eliminating FPs is shown in Fig. 3.2, which can be considered as an optimal subgraph searching problem. In order to obtain the refined detection result $\mathcal{D}$, we build a conflict graph $G =< V, E; \hat{\mathcal{D}} >$ on the raw detection results $\hat{\mathcal{D}}$ from the external detector. Each node in the vertex set V of G stands for one detection result and each edge in the edge set E means a conflict between two nodes. Then, we figure out the optimal subgraph $G^* =< V^*, E^*; \mathcal{D} > \subset G$ with the following formulation:

$$G^* = \underset{G'=<E',V'>\subset G}{\arg\max} \sum_{u,v\in E'} C_{u,v} + \sum_{v\in V'} \Lambda_v^2, \tag{3.1}$$

where $C_{u,v} \leq 0$ is the conflict score between two nodes u and v, and $\Lambda_v \geq 0$ is the confidence score of node v, which are detailed in Section 3.1.1.1 and 3.1.1.2, respetively. The formulation of obtaining the optimal subgraph in Eq. (3.1) can be considered as finding the most confident nodes with fewer conflicts between nodes. The corresponding detection results $\mathcal{D}$ in the optimal subgraph G^* are the refined detection results.

3.1.1.1 Conflict score

For the graph G, the conflict score $C_{u,v}$ for $u, v \in V$ can be calculated with:

$$C_{u,v} = -\alpha_1 M_{u,v} - (1 - \alpha_1) A_{u,v}, \tag{3.2}$$

where α_1 is a weight coefficient balancing the $M_{u,v}$ and $A_{u,v}$. For the features F_a^u and F_a^v of detection results u and v respectively generated by the appearance feature extractor in SAA, $A_{u,v}$ denotes their cosine similarity as follows:

$$A_{u,v} = \frac{F_a^u \cdot F_a^v}{\|F_a^u\| \|F_a^v\|}. \tag{3.3}$$

$M_{u,v}$ measures the occlusion confidence as follows:

$$M_{u,v} = \frac{Int(Core(u), \quad Core(v))}{Area(Core(o))},$$
$$\text{where } o = \begin{cases} u, \text{ if } u \text{ is the occludee,} \\ v, \text{ if } v \text{ is the occludee,} \end{cases} \tag{3.4}$$

where $Core(\cdot)$ in Eq.(3.4) is the central region of the detection, which has 60% of the width and height of the raw detection bounding box. $Int(\cdot, \cdot)$ measures the area of the intersected region for two input regions. $Area(\cdot)$ denotes the area of a region and o denotes the occludee. The occlusion confidence $M_{u,v}$ describes the occlusion ratio of the core area of the occludee. Denote the bottom y-axis coordinate of u by y_u^{bottom} and similarly for v. When $y_u^{bottom} < y_v^{bottom}$, we assume that the detection result for node u is the occludee occluded by node v because v is in front of u from a camera view, and vice versa. Note that the vanilla IoU is not able to describe such a relationship between occluders and occludees in MOT scenes. Since $Int(Core(u), \quad Core(v)) \leq Area(Core(o))$, we have $M_{u,v} \leq 1$.

3.1.1.2 Confidence score

For the graph G, the node score Λ_v for the node $v \in V$ can be chosen as follows:

$$\Lambda_v = \beta_1 \Omega_v + (1 - \beta_1) Z_v, \tag{3.5}$$

where β_1 is a coefficient balancing the terms Ω_v and Z_v. Ω_v, the consistency score for node v, is the max overlap between node v and the detection results $\mathcal{D}_{t-1}$ in the previous frame defined as below:

$$\Omega_v = \max_{d \in \mathcal{D}_{t-1}} IoU(d, v). \tag{3.6}$$

Z_v, the detection score of v, is the classification score obtained from another binary classifier as done in the experiments of Long et al. 2018; Xu et al. 2019a.

The optimal subgraph can be solved in a fast way, with the computational time required on par with the time required by Hungarian Algorithm on MOT17. We describe how to solve the optimal subgraph problem in Eq. (3.1) and analyze its complexity of it in Appendix A.

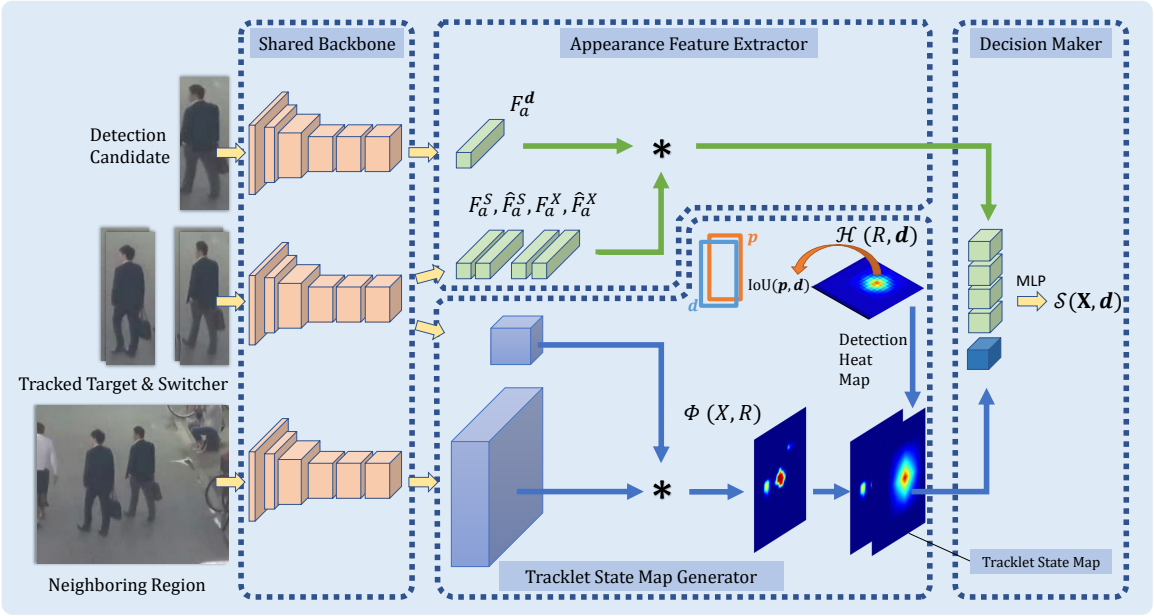


FIGURE 3.3. Pipeline of the proposed switcher-aware association (SAA). There are four sub-nets in SAA: the shared backbone, the appearance feature extractor, the tracklet state map generator, and the decision-maker. The shared backbone yields general feature maps for the given images. The tracklet state map generator takes feature maps from the template image of a target and the neighboring region as inputs. The feature of the template image is used as the kernel for convolution with the features of the neighboring region. The appearance feature extractor takes several historical target images as inputs and uses the same weight-sharing backbone CNN and some extra header layers for extracting their historical appearance features. The decision-maker consists of some convolution layers and fully-connected (FC) layers. It combines the heat maps and the historical appearance features to make the final judgment.

3.1.2 Switcher-Aware Association

The Switcher-Aware Association (SAA) module is designed to utilize multiple cues and the information from the potential switcher for robust tracking. The overview of SAA is shown in Fig. 3.3 and the whole network can be trained end-to-end. SAA is composed of four parts: the shared backbone CNN, the tracklet state map generator, the appearance feature extractor, and the decision-maker. 1) The backbone CNN takes an image patch as input and generates the feature maps of the image patch. The image patch can be detection candidates, tracked targets, switchers, or the neighboring regions of the target. The backbone CNN has shared weights for these different types of image patches (Section 3.1.2.1). 2) The tracklet state

map generator takes the CNN feature maps of a target and its neighboring region as inputs, and it first generates the surrounding heatmap and then concatenates it with the detection heat map and outputs the tracklet state map. At the same time, the potential switcher is figured out through the surrounding heat map(Section 3.1.2.2). 3) The appearance feature extractor takes all the current and historical feature maps of image patches of the targets, the potential switcher and the detection, then outputs their relationship embeddings, which are a series of similarities (Section 3.1.2.3). 4) Finally the tracklet state maps and the relationship embeddings are then concatenated and fed to the decision maker and used for predicting the final matching scores of the target and the detection candidate (Section 3.1.2.4).

3.1.2.1 Backbone CNN

We use a modified ResNet-18 (He et al. 2016) as the weight-sharing backbone of the appearance feature extractor and the tracklet state map generator.

3.1.2.2 Tracklet state map generation

We propose to use a tracklet state map descriptor as an extra cue for data association. The tracklet state map is the concatenation of the surrounding heat map and the detection heat map.

Generating the surrounding heat map. The feature maps from block2, block3 and block4 of the backbone are denoted by F_2, F_3 and F_4, respectively. The feature maps of the cropped image $\mathcal{I}_X$ of region X are denoted as F_2^X, F_3^X, and F_4^X. At frame t, we choose X to be the bounding box at time $t-1$, i.e. $X = \mathbf{x_{t-1}}$, to generate the surrounding heat map. And the feature maps from the neighboring region R at time $t-1$ are represented by F_2^R, F_3^R, and F_4^R. Then, the output of the surrounding heat map generator can be described with the following equation:

$$\Phi(X, R) = \text{sigmoid}(\frac{1}{3} \sum_{i=2,3,4} F_i^X * F_i^R), \tag{3.7}$$

where $F_i^X * F_i^R$ denotes the convolution between the features of the cropped region F_i^X and all possible locations in F_i^R, and is used for obtaining the similarity between the template and all possible locations in the neighboring region. The surrounding heat map is supervised by binary cross entropy loss on map L_h, which is similar to the study of Li et al. 2018b.

Generating the detection heat map. To embed the position of the target and detection result $\mathbf{d}$, we generate another detection heat map on the same scale as the surrounding heat map and include it as part of the input of the decision-maker. The detection heat map $\mathcal{H}(R, \mathbf{d}) : \mathbb{D} \rightarrow \mathbb{R}$ is defined as follow:

$$\mathcal{H}(R, \mathbf{d})[\mathbf{p}] = IoU(\mathbf{p}, \mathbf{d}), \tag{3.8}$$

where $\mathbf{p} \in \mathbb{D}$ denotes the position on the map. As shown in Fig. 3.3, the detection heat map indicates the relative position of the detection result to the target.

Finally, the tracklet state map $\mathcal{K} = \Phi(X, R, \mathcal{H}(R, \mathbf{d})$ is the concatenation of the two heat maps and will be used in the decision maker.

Finding the potential switcher from the surrounding heat map. We identify the potential temporal switcher from the surrounding heat map using a modified Non-Maximum Suppression (NMS) algorithm. We first find out the second-best position (and its corresponding bounding box having the same size as the template at the position) which satisfies the following two conditions: 1) its bounding box overlap with the one of the first largest-response positions is smaller than 0.5; 2) it has the largest response score under condition 1). If the response score of the second-best position is larger than a threshold of 0.5, then we use the target that is nearest to the second-best position in the previous frame as the potential switcher; otherwise, it is considered that no potential switcher is found and we will use the IoU of target and detection result as the matching probability instead for efficiency. After the potential switcher is identified, its appearance features will be extracted and used as part of inputs to the final decision-maker.

3.1.2.3 Appearance feature extractor.

The feature maps from block4 of the backbone are then passed through a group of header layers. After the header layers, the appearance feature of a target is calculated. For a tracked target X, F_a^X is the appearance feature of the target, F_a^S is the appearance feature of the potential switcher and $F_a^{\mathbf{d}}$ is the appearance feature of the detection result. To supervise the feature learning, we add another binary cross entropy (BCE) loss

$$L_a = \frac{1}{2} L_{BCE}(\eta F_a^X * F_a^{\mathbf{d}}, y_x) + \frac{1}{2} L_{BCE}(\eta F_a^X * F_a^S, y_s) \tag{3.9}$$

for the appearance feature extractor. Here η is the rescaling weight, and y_x is the corresponding label of whether the target and the detection are identical. y_s have similar meanings for the target and potential switcher.

Due to frequent interactions of the targets and occlusions, we find that in some critical frames, the bounding boxes are inaccurate and therefore the corresponding appearance features may not be reliable. To handle these cases, we include extra appearance features as part of the inputs of the decision-maker. We use $\hat{F_a^X}$ and $\hat{F_a^S}$ respectively to denote the saved appearance features of the target and potential switcher from the history, i.e., in the frames before $T-1$.

3.1.2.4 Final decision-maker

To make the final decision on whether the relationship of the pair of targets should be considered matched or separated, the final decision-maker combines all cues generated in the previous steps. As such, the input set can be denoted by

$$\Gamma = \{\mathrm{state\_head}(\mathcal{K}), F_a^X \cdot F_a^{\mathbf{d}}, \hat{F_a^X} \cdot F_a^{\mathbf{d}}, F_a^S \cdot F_a^{\mathbf{d}}, \hat{F_a^S} \cdot F_a^{\mathbf{d}}\}, \tag{3.10}$$

where $\mathcal{K}$ is the tracklet state map, F_a^X, F_a^S, $F_a^{\mathbf{d}}$, $\hat{F_a^X}$ and $\hat{F_a^S}$ are the appearance features introduced before, $\cdot$ is the dot product of two vectors, $\mathrm{state\_head}$ is a tiny neural network head which consists of convolution and batch norm layers.

For the features $F_a^{\mathbf{d}}$ of a detection candidate, we calculate its cosine similarity to the tracked target feature F_a^X, the historical feature $\hat{F_a^X}$, similarly for the potential switcher. We simply

calculate the dot product of two appearance features and obtain the cosine similarity because all these features are normalized with the length of 1. For the tracklet state map, we apply a state head, which consists of two convolution layers with batch normalization layers, to generate another embedded feature. We concatenate all these similarities and the embedded features of the tracklet state map, and then feed them to fully-connected layers with ReLU(Nair and Hinton 2010), as shown in Fig. 3.3. Finally, $\mathcal{S}(\mathbf{X}, \mathbf{d}|\mathcal{I}, \mathcal{T}, \mathcal{D}) = \mathrm{MLP}(\Gamma)$ and the output is the matching probability of the detection candidate and the target, which will be used in the data association procedure.

We add the third binary cross entropy loss L_{dm} for the decision-maker as below,

$$L_{dm} = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)], \tag{3.11}$$

where $p = \mathcal{S}(\mathbf{X}, \mathbf{d})$ is the output probability and y is the label.

In this way, the tracklet state information, as well as the appearance features of the potential switcher, are encoded in the inputs of the decision-maker, which is expected to learn to judge based on all these features. As such, the proposed framework is switcher-aware and becomes more robust than merely depending on a single appearance feature.

The overall loss L is the sum of all losses of different parts:

$$L = L_h + L_a + L_{dm}, \tag{3.12}$$

where L_h, L_a, L_{dm} are the losses of the surrounding heat map, appearance feature extractor, and decision maker, respectively.

3.2 Experiments

3.2.1 Implementation Details

The network is trained in three stages. In the first stage, the appearance feature extractor and the backbone are trained on ReID datasets market-1501(Zheng et al. 2015). The backbone

Method	**MOTA** ↑	MOTP ↑	**IDF1** ↑	IDP ↑	IDR ↑	FP ↓	FN ↓	**IDS** ↓
FAMNet(Chu and Ling 2019)	**52.0%**	76.5%	48.7%	66.7%	38.4%	**14138**	253616	3072
STRN_MOT17(Xu et al. 2019a)	50.9%	75.6%	**56.0%**	74.4%	**44.9%**	25295	**249365**	2397
MOTDT17(Long et al. 2018)	50.9%	**76.6%**	52.7%	70.4%	42.1%	24069	250768	2474
MTDF17(Fu et al. 2019)	49.6%	75.5%	45.2%	58.1%	37.0%	37124	**241768**	5567
PHD_GM(Sanchez-Matilla and Cavallaro 2019)	48.8%	**76.7%**	43.2%	58.2%	34.3%	26260	257971	4407
DMAN(Zhu et al. 2018)	48.2%	75.7%	55.7%	**75.9%**	44.0%	26218	263608	**2194**
Ours	**52.4%**	**76.7%**	**56.3%**	**76.9%**	**44.4%**	**14081**	252236	**2166**
Tracktor17(Bergmann et al. 2019)	53.5%	**78.0%**	52.3%	71.1%	41.4%	12201	248047	**2072**
Ours with FPN	**55.0%**	77.9%	**57.2%**	**76.8%**	**45.6%**	**11502**	**240342**	2126

TABLE 3.1. Comparison of the proposed MOT framework with other online processing state-of-the-art (SOTA) methods in MOT17 whose MOTA is larger than 48%. red means the best result and blue denotes the runner-up. ↑ means higher is better and ↓ means lower is better. We compare with *Tracktor17* separately by adding Feature Pyramid Network (FPN, Lin et al. 2017) as post-processing, **bold** for the better result between ours(w. FPN) and *Tracktor17*.

Method	MOTA↑	IDF1↑	Recall↑	Prcn↑	MT↑	ML↓	FP↓	FN↓	IDs↓	MOTP↑
GNN(Wang et al. 2020)	56.4	42.0	60.3	**95.1**	394	961	**17421**	223974	4572	-
Tube_TK(Pang et al. 2020)	63.0	58.6	68.5	93.5	735	469	27060	177483	4137	-
CTrackerV1(Peng et al. 2020)	66.6	57.4	71.6	94.8	759	570	22284	160491	5529	78.2
CenterTrack(Zhou et al. 2020)	67.3	59.9	71.9	94.6	822	584	23031	158676	**2898**	-
Ours w. private detector	**71.0**	**64.5**	**76.9**	93.6	**912**	**387**	29862	**130386**	3183	77.2

TABLE 3.2. Results for private detector track in MOT17 benchmark. Red for best result. ↑ for higher is better, ↓ for lower is better.

is pretrained on ImageNet(Russakovsky et al. 2015) and other parameters are randomly initialized. Afterward, the mAP of the appearance feature extractor on the market-1501 dataset is 62%. For the second stage, the switcher heat-map generator is trained on Youtube-BB(Real et al. 2017) dataset but only pedestrians are considered. In the third stage, the whole network is trained on MOT16(Milan et al. 2016) dataset supervised by the losses mentioned in Sec. 3.1.2.3. MOT datasets are only used in the last stage and the first two stages are pretraining steps for the final decision-maker.

In all stages of training, we used an SGD optimizer with a momentum of 0.9. In the third stage, we applied an initial learning rate of 0.1 and decayed by 0.1 at epochs 6, 10, and 15. We implemented training for 20 epochs in total. It is vital in the third training stage to generate the training data using detection-based bounding boxes because the ground truth is much more precise than practical detections. Training on ground truth will lead to an over-fitting

problem to easy cases, and thus the network will not perform well in the real detection-based scene.

3.2.2 Evaluation on MOT benchmarks

3.2.2.1 Datasets

The proposed framework is evaluated with the MOT17 benchmark (Milan et al. 2016), which shares the same training and test videos with MOT16 but offers different detection inputs. MOT17 has made the ground truth more accurate. Besides, it requires the algorithm to be evaluated under three different detectors simultaneously, making the results more convincing. The test video sequences include various complicated scenes and are still a great challenge.

3.2.2.2 Evaluation Metrics

The benchmark uses CLEAR MOT Metric and IDF1 scores for evaluation. The main metrics are MOTA and IDF1. MOTA score considers False Positives(FPs), False Negatives(FNs), and IDentity Switches(IDS) in an average manner. IDS counts the number of wrongly assigned events. However, in normal situations, the FN number is much larger than the IDS number, so the metric MOTA can hardly reflects the robustness of the tracker. In contrast, the IDF1 score estimates the average maximum matching rate of the tracker, which is more likely to reflect the tracking ability of a tracker than the MOTA score. We usually compare both two metrics simultaneously to gain an objective evaluation result.

3.2.2.3 Results

Table 3.1 illustrates the results for state-of-the-art (SOTA) online processing methods and our approach in the MOT17 testing set (public detection input). The results show that our framework has the highest MOTA when compared to some popular methods. At the same time, our framework has the least IDS number and the highest IDF1 scores. In addition, our MOTA, IDS, and IDF1 scores are in the leading positions for MOT17 for all the online processing algorithms considered. High IDF1 and low IDS also demonstrate that our method works

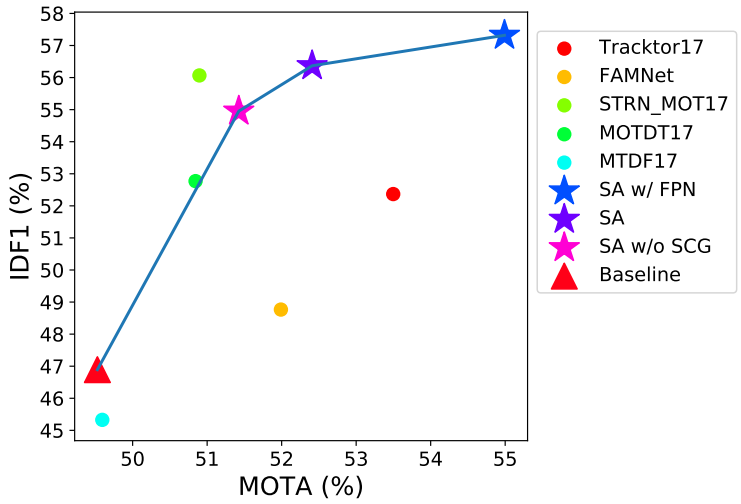


FIGURE 3.4. Ablation study on our framework. SA w/o SCG in pink star stands for using SAA only. SA in purple star means using both SAA and SCG. SA w/ FPN represents using SAA and SCG with FPN as post-process.

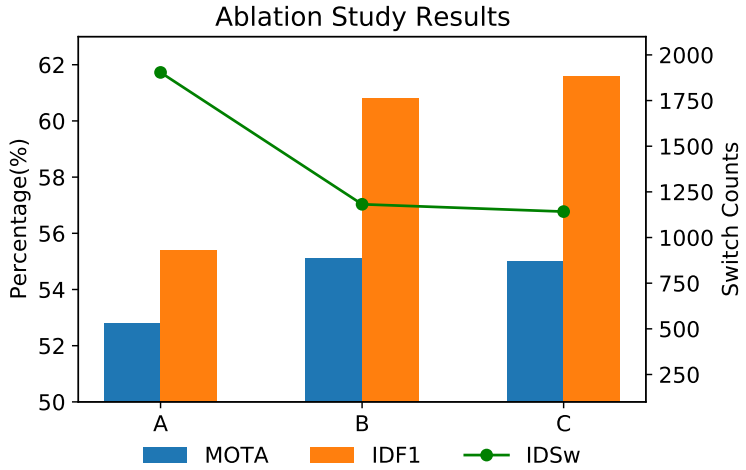


FIGURE 3.5. Ablation study on including different cues for MOT. *A:* Single appearance cues only. *B:* Adding tracklet state map to the single appearance cues in the baseline A. *C:* Add the switcher to the implementation B, where all appearance cues from the targets and its switcher, as well as the tracklet state map, are included (Our final setting). The bars in the figure correspond to MOTA and IDF1. The green points correspond to the IDSw (ID switch).

well in solving identity issues. When comparing with the method *Tracktor17*(Bergmann et al. 2019), we find that a Feature Pyramid Network (FPN)(Lin et al. 2017) is used in the work. For a fair comparison, we also add the same FPN from *Tracktor17* as post-processing for our result. Results show that our work outperforms *Tracktor17* under the FPN setting in terms of

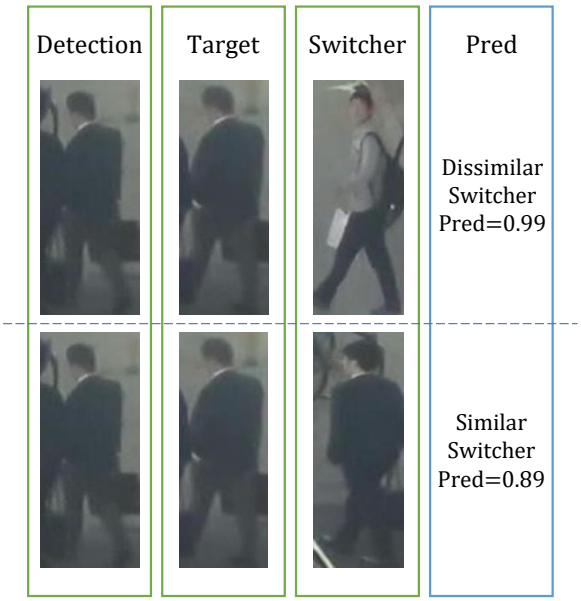


FIGURE 3.6. Different prediction scores (Pred) obtained by the decision-maker on the same matching target-detection pair with different potential switchers.

most measurements such as MOTA, IDF1, and so on. As for MOTP and IDS, our algorithm is almost comparable to *Tracktor17*.

Besides, the backbone we used is ResNet-18, while many other works (Zhu et al. 2018; Long et al. 2018) use larger networks such as ResNet-50 for appearance feature extraction. We achieve competitive results on multiple measures by a smaller network, which also indicates the effectiveness of switcher information. With the help of deeper networks and spatial-temporal techniques on feature extracting, we believe the switcher information may yield larger improvement in the future.

At the same time, we evaluate our method on private detection input so that we can compare it with recent joint detection and tracking methods. Table 3.2 shows the results of our method on the MOT17 testing set (private detection input). Compared with several joint detection and tracking method, we still achieve competitive results in all three main metrics.

3.2.3 Ablation Study and Discussion

3.2.3.1 How Different Components Impact the Tracking Performance?

Fig. 3.4 shows the impact of different components of the framework. We test on three detectors in the MOT17 test dataset and the results are the average of them. The baseline method utilizes a single target appearance association with a binary classifier to refine the detection score. As shown in Fig. 3.4, without the help of SCG, the performance decreases by 1.0% MOTA and 1.4% IDF1, and without the help of SAA (the baseline in Fig. 3.4), the performance decreases by 1.9% MOTA and 8.0% IDF1. With the help of FPN, we further gain another 2.6% MOTA and 0.9% IDF1 improvement. With SAA, the edges in the association are more discriminative so that more identity issues are solved and thus the IDF1 is 8.0% higher. With SCG, more confusing spatial switchers are eliminated so that the IDF1 increases another 1.4%.

Fig. 3.5 further analyzes the impact of different components inside the SAA module. The experiment was conducted on the MOT17 training set. Setting *A* uses a single appearance feature only. Based on the single appearance features of a detection candidate and the tracklet in Setting A, Setting *B* further uses the tracklet state map. Setting *C* is our final setting, which includes the tracklet state map and the appearance features from the tracklet in Setting B, as well as the appearance features of the potential switcher. Compared with setting A, setting B performs better in all MOTA, IDF1, and IDS metrics, with improvements of 2.3% and 5.4% for MOT and IDF1 respectively, with a decrease in IDS number of more than 700. This shows that the tracklet state map is useful multi-cue information for improving tracking results. Compared with setting B, setting C further improve IDF1 by 0.8%, while MOTA and IDS keep a similar level, showing that the appearance features of the potential switcher are beneficial for reducing ID switch. The experiment shows that the tracklet state map and the information of potential switcher are helpful and make the tracker more robust compared with baseline *A*.

(A) similar switcher: before the identity issue happened

(B) similar switcher: after the identity issue happened

(C) dissimilar switcher: before the identity issue happened

(D) dissimilar switcher: after the identity issue happened

FIGURE 3.7. Screen-shot of a demo video. Green box for highlighted target, red box for wrongly assigned target.

3.2.3.2 In What Cases Can the Switcher Information Help the Tracking?

With switcher information, the tracker is able to make a comparison between the target and its potential switcher, thus it has a higher accuracy when matching. In some cases where nearby targets have a similar appearance, the decision-maker is trained to learn which one is better. In the cases that we could not find a similar potential switcher, the decision-maker learns to use a lower threshold for the appearance feature, which may be of low quality due to occlusions by background objects. As shown in Fig. 3.6, the decision-maker is trained to yield a higher matching score if the potential switcher is not similar to the target.

3.2.3.3 Visualization

As shown in Fig. 3.7, it compares the 'Non-SA' (baseline) with 'SA' (baseline with our switcher-aware design). In the video, green boxes stand for highlighted targets and red boxes denote identity issues. These identity issues happening in Non-SA show that confusing switchers are one of the important error sources that cause identity issues. The proposed scheme can effectively overcome the identity issues in the video demonstration.

Similar Switchers. For the case shown in Fig. 3.7 (a) and (b), the identity 37 drifted to another target in Non-SA but the identity keeps unchanged in SA. In fact, identity 26 is a temporal switcher from previous frames, the inclusion of the potential switcher in our approach results in more reliable matching relationships. The decision-maker is trained to make a fair judgment based on all appearance features and heat maps.

Dissimilar Switchers. For the case shown in Fig. 3.7 (c) and (d), the identity is changed from 52 to 57 for Non-SA but the identity is not changed for SA. This is an example where the similarity of the features for the highlighted target is low due to partial occlusion. Comparison to a dissimilar potential switcher allows a lower threshold to be used in our SA for matching the correct pair of targets in adjacent frames and thus the matching probability is increased relatively. It reflects another situation discussed in the ablation study.

Failure Cases. Although the inclusion of the switcher information solves many identity issues compared to non-switcher-aware methods, there are still some cases remaining unsolved even though we consider the potential switcher. This is probably because the quality of the appearance feature is extremely low, which may need more research in the future.

CHAPTER 4

Similarity- and Quality-Guided Relation Learning for Joint Detection and Tracking

---

In this chapter, we explore the solution of performing relation learning in the joint detection and tracking task to utilize instance-level spatial-temporal information in videos.

4.1 Methodology

4.1.1 Exploring Instance-Level Relation Learning for Joint Detection and Tracking

We first explored and designed the instance-level relation learning pipeline for joint detection and tracking by integrating relation modules and task-specific branches into the popular Faster R-CNN (Ren et al. 2015) detection framework. Specifically, our joint framework consisted of backbone extractor networks, a regional proposal network (RPN) (Ren et al. 2015), an RoI pooling layer (Ren et al. 2015), a relation module, and task-specific branches. Please note that, although the relation modules were applied on the Faster RCNN framework, they are decoupled from the detection framework, which means that the backbone and RPN/RoI can be replaced by more advanced components, e.g., YoloX (Ge et al. 2021).

There are three main steps in the pipeline. First, each input image frame I_t of a video is fed to the backbone networks to generate feature maps for the image. This is followed by the region proposal network (RPN) and RoI pooling layer to predict and extract a set of RoI

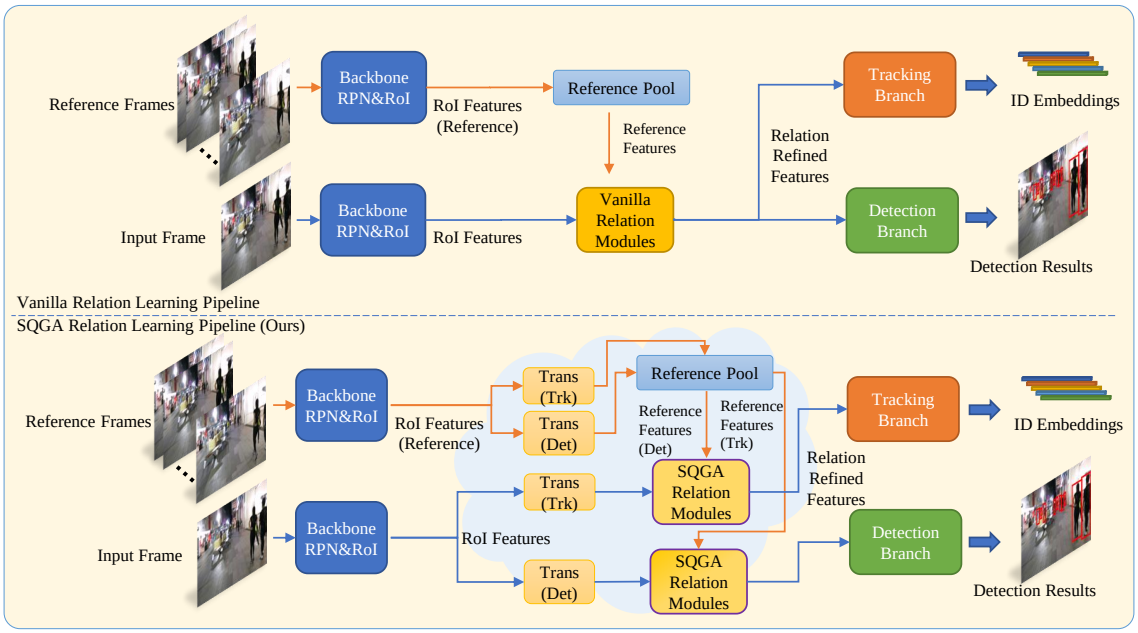


FIGURE 4.1. *Top row:* Conventional relation learning. Conventional relation modules take common RoI features as inputs and refine them using reference features in the reference pool. *Bottom row*: Relation learning with Similarity- and Quality-Guided Attention (SQGA) for joint detection and tracking. First, common RoI features are transformed into task-specific features (i.e., 'Trans' in the figure, 'Trk' is short for tracking, and 'Det' is short for detection) and then are fed into the proposed SQGA relation modules. The SQGA modules for the tracking branch refine features for the tracking task; this is done similarly for detection. Finally, the detection results from the detection branch are used for detection. ID embeddings of the tracking branch and detection results from the detection branch are used to match and form trajectories in MOT.

features (i.e., the features of regions of interest). We denote the set of RoI features in frame t as $F_t = \{F_t^i\}, i = 1, 2, \ldots, N$ with N as the number of RoIs in the frame.

Second, the relation module, denoted by $\mathcal{N}_R$, takes the RoI feature embeddings F_t as inputs and refines them using the reference embeddings F_r from a reference pool storing RoI feature embeddings from reference frames. We denote the set of M reference RoI features from the reference pool as $F_r = \{F_r^1, F_r^2, \ldots, F_r^M\}$; these are stored in a matrix with a shape of $\mathbb{R}^{M \times D}$, where D is the feature dimension. Then, the refined features $\hat{F}_t = \{\hat{F}_t^i\}, i = 1, 2, \ldots, N$ are calculated as follows based on the multi-head attention proposed by Vaswani et al. 2017:

$$\hat{F}_t^i = \mathcal{N}_R(F_t^i, F_r) = F_t^i + \mathrm{MultiHead}(F_t^i, F_r) \times \mathbf{W}_v, \tag{4.1}$$

$$\mathrm{MultiHead}(F_t^i, F_r) = \underset{h=1,2....,H}{\mathrm{Concat}}(\mathrm{Head}_h), \tag{4.2}$$

$$\begin{aligned} \mathrm{Head}_h &= \mathrm{Attention}_h(F_t^i, F_r) \\ &= \mathrm{Softmax}(\frac{Q_h(F_t^i)K_h(F_r)^T}{\sqrt{d_h}}) \times F_r, \end{aligned} \tag{4.3}$$

where $\mathbf{W}_v \in \mathbb{R}^{HD \times D}$ is the adjustment matrix, Concat is the channel-wise concatenation, Q_h and K_h are the h-th respective functions for query and key, which are implemented by the fully connected (FC) layers. The dimensions for Head_h and $\mathrm{MultiHead}(F_t^i, F_r)$ are $\mathbb{R}^{1 \times D}$ and $\mathbb{R}^{1 \times HD}$, respectively. d_h is the feature dimension for every head of the query and key. The Softmax function is calculated over all the reference feature embeddings in a row of the matrix. The matrix multiplication related to the adjustment matrix $\mathbf{W}_v \in \mathbb{R}^{HD \times D}$ in Eq. (4.1) is implemented by an FC layer with zero bias. In other words, the relation module uses multi-head attention to generate messages and refines the input features F_t^i by adding transformed messages to the input in Eq. (4.1). The transformed message is obtained from a weighted sum of the reference features in Eq. (4.3).

Finally, the refined features of $\hat{F}_t$ are used for detection and tracking prediction. The output of the detection branch is a set of detection results consisting of bounding boxes and confidence scores. The output of the tracking branch is the identity embeddings for the bounding boxes. These results are used to form trajectories. The whole pipeline is illustrated in the top row of Fig. 4.1. During training, detection loss L_{det} and tracking loss L_{trk} are separately added to the two branches. We implemented the detection loss L_{det} as the widely used cross-entropy and smooth L1 loss and implemented tracking loss L_{trk} as the binary cross-entropy loss with label=1 when the identities are the same.

As an initial design involving relation learning for joint detection and tracking, we straightforwardly applied conventional relation modules with multi-head attention on the common RoI features extracted from the RoI pooling layer. Taking the RoI features refined by the relation

modules as inputs, the detection branch and the tracking branch extract different task-specific features by their own linear layers to obtain tracking and detection results separately.

4.1.2 Analysis for Conventional Relation Learning

.

4.1.2.1 The Problems

Directly using this conventional design does not work for the complex joint detection and tracking task for two reasons:

(i) *Detection and tracking share the same relation mechanism.*

Detection and tracking are different tasks that have different learning objectives and thus require different attention mechanisms. In detection, the relation module can attend to both objects and context so that the refined features are better at distinguishing different classes of objects. In MOT (e.g., Xu et al. 2019b), the relation module only attends to objects (pedestrians). This reminds us that the attention mechanisms are very likely to be different in detection and tracking. Based on this observation, in joint detection and tracking, relation modules should not be applied to common RoI features but should be applied to task-specific features with exclusive parameters for different tasks.

(ii) *Vanilla self-guided attention mechanism is not suitable for MOT scenarios.* Detection proposals usually contain a variety of regions from both possible foreground and background objects or contexts. In MOT scenarios, two significant differences are that fewer categories of interest but more instances are involved. In this case, information from different categories or the background is extremely limited. The self-guided nature of the conventional relation modules makes it difficult for them to generate the type of distinguishing attention that can avoid most aggregations obtained from a different category or background. Our experiment shows that (see Fig. 4.4a) the affinity matrix of conventional relation learning did not show great differences for clear objects (the first 3 columns), unclear objects (the fourth column),

and context (the last two columns). Given that the number of context regions is much larger than that of the foreground regions, the vanilla attention mechanism actually focuses mainly on the context (see Fig. 4.3). The experimental results above coincide with our induction, revealing that the vanilla attention mechanism is not effective in MOT even though it is effective in detection tasks. More specifically, in video object detection, context is related to class type to some extent. For example, cats and lions are sometimes confusing. However, the contextual background where they appear (lions in the jungle while cats at home) is useful for distinguishing them; thus, conventional relation learning without guidance can distinguish them more easily. Things change in joint detection and tracking scenes (e.g., pedestrian detection and tracking). The contextual background is often almost the same (e.g., static street scene) and thus becomes less discriminative. This makes the vanilla attention mechanism less effective. Moreover, the massive number of instances in the same category inevitably include a certain number of noisy and low-quality objects, which might not worth being aggregated into some other instances of good quality. Based on the above analysis, we concluded that more fine-grained attention schemes are needed to effectively aid the relation learning and that task-specific guidance is desirable for learning a fine-grained attention in the joint task.

4.1.2.2 A Better Attention Guidance

From another perspective, attention is a more complicated and dynamic weighting strategy. Consider a simplified case of a relation module, i.e., a weighted mean of a feature F_x and another feature F_y, where the refined feature $\hat{F}_x$ is given as:

$$\hat{F}_x = (1 - \alpha)F_x + \alpha F_y. \tag{4.4}$$

Here $\alpha = \mathcal{A}(F_x, F_y) \in (0, 1)$ is a dynamic weight decided by F_x and F_y together, assuming that the optimization targets of F_x and F_y are γ_x and γ_y , respectively. The supervision from the loss L is added after a linear predictor ϕ. Then, we have three different losses, i.e., $L(\phi(F_x), \gamma_x)$, $L(\phi(F_y), \gamma_y)$, and $L(\phi(\hat{F}_x), \gamma_x)$. If the feature aggregation helps, we should

have:

$$\begin{aligned} L(\phi(F_x), \gamma_x) &> L(\phi(\hat{F}_x), \gamma_x) \\ &= L(\phi((1-\alpha)F_x + \alpha F_y), \gamma_x) \\ &= L((1-\alpha)\phi(F_x) + \alpha\phi(F_y), \gamma_x). \end{aligned} \tag{4.5}$$

One solution for Eq. (4.5) is to satisfy the following three conditions:

$$\gamma_x = \gamma_y, \tag{4.6}$$

$$L(\phi(F_y), \gamma_y) < L(\phi(F_x), \gamma_x)), \tag{4.7}$$

$$\begin{aligned} L((1-\alpha)Y_a + \alpha Y_b, \gamma_z) &< \max(L(Y_a, \gamma_z)), L(Y_b, \gamma_z)) \\ \forall\, 0 < \alpha &< 1, Y_a \neq Y_b. \end{aligned} \tag{4.8}$$

which provides a similarity guidance, as described in Eq. (4.6)), and a quality guidance, as in Eq. (4.7). The similarity guidance in Eq. (4.6) requires that the labels of the samples (i.e., x and y) be consistent, such that samples from the same category verify the existence of each other. The quality guidance in Eq. (4.7) requires that the loss of sample y be smaller than the loss of sample x, meaning that y has better quality than x, so that y can be used to refine the feature x. Eq. (4.8) requires the loss to have special characteristics. However, it is easy to show that the widely used binary cross-entropy (BCE) loss and smooth L1 loss satisfy Eq. (4.8).

Based on the investigation above, we concluded that *two factors of guidance, i.e., which kinds of features to aggregate and what levels of qualities to aggregate,* should be considered in the joint modeling of the two tasks. Note that, although the guidance explicitly limits the relations between targets, the affinity matrix is not fully decided by this guidance. The gradients from the original task loss still affect the affinity, and the context is not eliminated but just balanced.

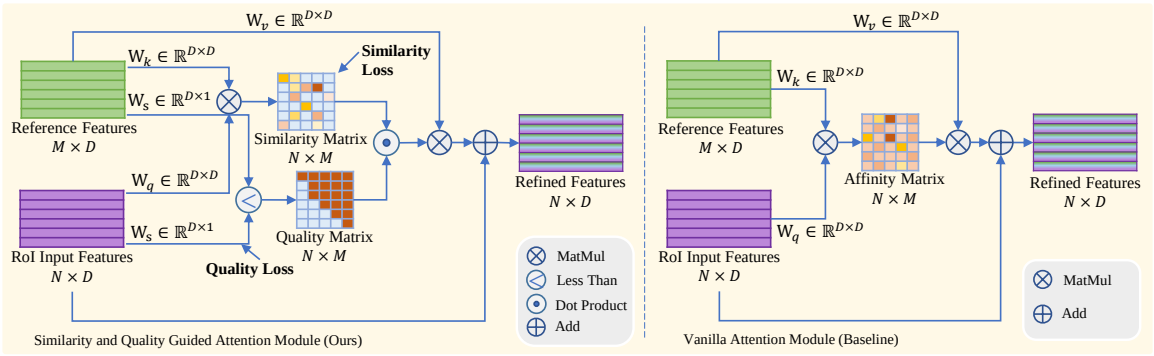


FIGURE 4.2. Proposed Similarity- and Quality-Guided Attention (SQGA) on the left and vanilla attention on the right. In SQGA, the affinity matrix is factorized to the similarity term (similarity matrix in the figure) and quality term (quality matrix in the figure). Similarity loss and quality loss guide the learning of these two terms. For simplification, the head number H is set to 1 and the head feature dimension d_h is set to D in the figure. W_q, W_k, W_s are respectively the parameters for the query, key, and quality transformations.

4.1.3 Unified Relation Learning Pipeline with SQGA

4.1.3.1 Dual Relation Modules

Based on the first reason discussed in the previous section, we designed a dual relation module pipeline for joint detection and tracking, as shown in the bottom row of Fig. 4.1. Specifically, for the detection process, the common RoI features are first transformed to task-specific features (det-features for detection and trk-features for tracking). Then they are fed to the relation modules for detection, using the reference features for detection and using a similar process for tracking. Finally, the task-specific relation-refined features are used in different task branches. The losses of the detection and tracking branches remain the same as in the vanilla design.

4.1.3.2 Similarity- and Quality-Guided Attention

Based on the second reason and the better guidance rules in Section 4.1.2.1, we implemented the relation modules using our proposed Similarity- and Quality-Guided Attention (SQGA).

We replaced the attention heads Head_h in Eq. (4.3) as follows:

$$\begin{aligned} \mathrm{Head}_h &= \mathrm{SQGA}_h(F_t^i, F_r) \\ &= \underset{j=1,2,\ldots,M}{\mathrm{Concat}}(\mathcal{A}_h(F_t^i, F_r^j)) \times F_r, \end{aligned} \tag{4.9}$$

$$\mathcal{A}_h(F_t^i, F_r^j) = \frac{\exp(\mathrm{Sim}_h(F_t^i, F_r^j)) \cdot \mathrm{Qlt}(F_t^i, F_r^j)}{\sum_{F_r^k \in F_r} \exp(\mathrm{Sim}_h(F_t^i, F_r^k)) \cdot \mathrm{Qlt}(F_t^i, F_r^k)}, \tag{4.10}$$

$$\mathrm{Sim}_h(F_x, F_y) = \frac{Q_h(F_x) K_h(F_y)^T}{\sqrt{d_h}}, \tag{4.11}$$

$$\mathrm{Qlt}(F_x, F_y) = [\mathcal{N}_q(F_x) \leq \mathcal{N}_q(F_y)], \tag{4.12}$$

where the operation Concat concatenates the scalars $\mathcal{A}_h(F_t^i, F_r^1), \ldots, \mathcal{A}_h(F_t^i, F_r^M)$ to a row vector, $\mathcal{A}_h(F_t^i, F_r^j)$ ($j = 1, 2, \ldots, M$) is the attention weight connecting F_t^i and the j-th reference feature F_r^j. Sim_h is the similarity term of the h-th head, and Q_h, K_h, and d_h share the same denotations as in Eq. (4.3). Qlt is the quality term, and $\mathcal{N}_q$ is implemented by a tiny neural network, which consists of two fully connected layers with ReLU (Nair and Hinton 2010) for predicting the quality. The symbol $[\cdot]$ in Eq. (4.12) is 1 if the condition is satisfied and is 0 otherwise.

Fig. 4.2 illustrates the differences between our attention design and the vanilla design. In SQGA, we factorize the affinities to similarity terms $\mathrm{Sim}_h(F_x, F_y)$ and quality terms $\mathrm{Qlt}(F_x, F_y)$. The similarity term measures the similarity between features F_x and F_y, while the quality term measures the relative quality between input F_x and the reference feature F_y. If the estimated quality $\mathcal{N}_q(F_y)$ of sample y is worse than the quality of x, then the affinity term (i.e., $\mathcal{A}_h(F_x, F_y)$) is 0, and the feature F_y is then not used to refine the feature F_x.

4.1.3.3 Loss Functions for Guiding SQGA

During training, we further added loss functions to guide the generation of the attention weights in the SQGA. For the factorized similarity term $\mathrm{Sim}_h(F_x, F_y)$ and quality term $\mathrm{Qlt}(F_x, F_y)$ in Eqs. (4.11) and (4.12), we designed the similarity loss L_{sim} and quality loss

L_{qlt} to guide them as follows:

$$\begin{aligned} & L_{sim}(F_x, F_y, \gamma_x, \gamma_y) \\ =& L_{bce}(\frac{1}{H}\sum_h \mathrm{Sim}_h(F_x, F_y), \zeta_{sim}(\gamma_x, \gamma_y)), \end{aligned} \tag{4.13}$$

$$L_{qlt}(F_x, \gamma_x) = L_{bce}(\mathcal{N}_q(F_x), \zeta_{qlt}(\phi(F_x), \gamma_x)), \tag{4.14}$$

where L_{bce} denotes the binary cross-entropy loss, and the sigmoid function is applied to both the quality prediction $\mathcal{N}_q(F_x)$ and the similarity prediction $\mathrm{Sim}(F_x, F_y)$ to obtain the binary cross entropy loss. $\zeta_{sim}(\gamma_x, \gamma_y)$ denotes the function used to map the difference between task labels γ_x and γ_y to a value in the range of $[0, 1]$, and analogously for $\zeta_{qlt}(\cdot, \cdot)$. ϕ is a function for predicting the task objective value. The two relation modules in the two branches have the same structure, while ζ_{sim}, ζ_{qlt}, and ϕ differ in these two branches. The detailed implementation of ζ_{sim}, ζ_{qlt}, and ϕ for the two branches is given below.

Losses for the Detection Branch. In the detection branch, the relation refined features are used for a classification task that distinguishes objects from the background. Therefore, the losses $L_{sim,det}$ and $L_{qlt,det}$ in Eqs. (4.13) and (4.14) are implemented by $\zeta_{sim,det}$ and $\zeta_{qlt,det}$ as follows:

$$\zeta_{sim,det}(\gamma_{x,det}, \gamma_{y,det}) = \begin{cases} 1, \text{if } \gamma_{x,det} = \gamma_{y,det}, \\ 0, \text{otherwise}. \end{cases} \tag{4.15}$$

$$\zeta_{qlt,det}(\phi_{det}(F_x), \gamma_{x,det}) = \phi_{det}(F_x)_{\gamma_{x,det}}, \tag{4.16}$$

where $\gamma_{x,det}, \gamma_{y,det} \in \{1, \dots C\}$ are the object detection labels for F_x and F_y, $\phi_{det}(\cdot)$ is a linear predictor (implemented by an FC layer and supervised by the detection loss) that predicts the detection label of a detection feature F_x, which outputs a vector with C dimensions denoting the predicted category distribution, and C is the number of classes; $\phi_{det}(F_x)_{\gamma_{x,det}}$ denotes the $\gamma_{x,det}$-th value of $\phi_{det}(F_x)$ (i.e., the predicted possibility of a ground-truth category). The task losses L_{det} at the detection branch are identical to the one in the experiments of Ren et al. 2015, i.e., a multi-class cross-entropy loss for classification and a smooth L1 loss for the bounding box regression.

Losses for the Tracking Branch. In the tracking branch, the goal is to distinguish objects of different identities. In this branch, the similarity loss $L_{sim,trk}$ is similar to Eq. (4.15) but uses the labels of the identities. The loss $L_{qlt,trk}$ for guiding the quality term in Eq. (4.14) is implemented by $\zeta_{qlt,trk}$ as:

$$\begin{aligned} &\zeta_{qlt,trk}(\phi_{trk}(F_x), \gamma_{x,trk}; \mathcal{P}) \\ =&\frac{1}{|\mathcal{P}_{pos}|} \sum_{F_{k1} \in \mathcal{P}_{pos}} \sigma(\phi_{trk}(F_x) \cdot \phi_{trk}(F_{k1})) \\ &- \frac{1}{|\mathcal{P}_{neg}|} \sum_{F_{k2} \in \mathcal{P}_{neg}} \sigma(\phi_{trk}(F_x) \cdot \phi_{trk}(F_{k2})), \end{aligned} \tag{4.17}$$

where ϕ_{trk} is a transformation (implemented by an FC layer) for adjusting the dimensions of the tracking embeddings, $\mathcal{P}$ is a pool of RoI feature samples used for evaluating the quality. $\mathcal{P}_{pos} \subset \mathcal{P}$ contains a set of RoIs with the same identity as $\gamma_{x,trk}$, while $\mathcal{P}_{neg} \subset \mathcal{P}$ contains a set of RoIs with identities different from $\gamma_{x,trk}$; $\mathcal{P}_{pos} \cup \mathcal{P}_{neg} = \mathcal{P}$ and $\mathcal{P}_{pos} \cap \mathcal{P}_{neg} = \emptyset$; the ratio between positives and negatives is 1:3. $\sigma(\cdot)$ is a sigmoid function. If $\zeta_{qlt,trk}$ is less than zero, it is truncated to zero. For the tracking branch, the task loss L_{trk} for the task of tracking is another binary cross entropy loss to supervise the scaled dot product of the two identity embeddings.

Overall Loss. Overall, the loss for the whole framework is:

$$L = L_{det} + L_{trk} + L_{sim,det} + L_{qlt,det} + L_{sim,trk} + L_{qlt,trk}, \tag{4.18}$$

where L_{det} and L_{trk} are task losses for the detection and tracking branches, respectively; $L_{sim,det}$ and $L_{sim,trk}$ are similarity losses that can guide the relation modules for detection and tracking; $L_{qlt,det}$ and $L_{qlt,trk}$ are the quality losses for the two tasks.

4.1.4 Other Necessary Modules.

4.1.4.1 Reference Feature Pool Implementation

The reference feature pool uses the top K features with the highest confidence scores from each selected frame among all the B frames, resulting in a total of $M = B \times K$ reference feature embeddings.

In the training phase, the reference feature embeddings pool is built by randomly sampling B frames within the same video. Similar to the study by Chen et al. 2020b, there are two types of sampled frames, i.e., local and global frames. Local frames are sampled within a short range near the training frame. Specifically, for training frame T, local frames are selected randomly from frame $T - \tau_s$ to frame $T + \tau_s$. Global frames are sampled uniformly along the video. In the testing phase, the feature embeddings pool is built as a first-in-first-out (FIFO) queue with a fixed length of B. When a new frame arrives, it is pushed into the queue, and if the queue is full, the earliest frame in the queue is popped out.

4.1.4.2 Tracklets Management

The outputs of the framework are the detection results and identity embeddings, which are fed to a post-processing procedure called tracking management. Some previous works (e.g., Stadler and Beyerer 2021a; Wang et al. 2022) have proposed some post-processing techniques. However, to reduce the influence of the post-processing procedure, we used a simple but effective tracklet management strategy to form the tracking results in this work. After being filtered by a threshold of the confidence score, the detection results are used for matching previous tracklets.

For an existed tracklet $\mathcal{T}$ and a detection result $\mathcal{D}$, the matching score is defined as:

$$\mathrm{C}(\mathcal{T}, \mathcal{D}) = \mathrm{Affinity}_{pos}(\mathcal{T}, \mathcal{D}) \cdot \mathrm{Affinity}_{apr}(\mathcal{T}, \mathcal{D}) \cdot s_{\mathcal{D}}, \tag{4.19}$$

$$\mathrm{Affinity}_{apr}(\mathcal{T}, \mathcal{D}) = \max\left(\frac{F^a_{\mathcal{T}} \cdot F^a_{\mathcal{D}}}{|F^a_{\mathcal{T}}||F^a_{\mathcal{D}}|} - \theta_a, 0\right) \cdot \frac{1}{1 - \theta_a}, \tag{4.20}$$

$$\text{Affinity}_{pos}(\mathcal{T}, \mathcal{D}) = 1 - \frac{2}{\pi} \arctan(\theta_d \text{dist}(\mathcal{T}, \mathcal{D})). \tag{4.21}$$

$s_{\mathcal{D}}$ is the confidence score of $\mathcal{D}$. $F^a_{\mathcal{T}}$ and $F^a_{\mathcal{D}}$ are identity embeddings for $\mathcal{T}$ and $\mathcal{D}$, respectively. $\text{dist}(\mathcal{T}, \mathcal{D})$ is the normalized euclidean distance between the center points of $\mathcal{T}$ and $\mathcal{D}$. θ_a and θ_d are constant hyperparameters. Then the Hungarian algorithm (Munkres 1957) is applied to find the optimal matching relationships. If a tracklet is matched with a detection result, then its appearance embeddings and position coordinates are updated. If a tracklet is not matched, it will be dropped if it is not matched again after a time interval. If a detection result is not matched, it is considered to be an initial tracklet. Similar to many previous methods, the Kalman filter is used to obtain smooth trajectories, and non-maximum-suppression is performed to eliminate duplicate bounding boxes.

4.2 Experiments

4.2.1 Implementation Details

4.2.1.1 Datasets for Training and Data Augmentation

MOT16/17 (Milan et al. 2016), MOT20 (Dendorfer et al. 2020), CrowdHuman (Shao et al. 2018), and HIE (Lin et al. 2020) were used for training in this work. The MOT16/17, MOT20, and HIE datasets are video datasets, and CrowdHuman is an image dataset. There are 7 videos in the MOT16/MOT17 training set, 4 videos in MOT20 training set, and 19 videos in the HIE training set. The MOT17 dataset contains the same videos as the MOT16 dataset, thus we refer to it as MOT16/17 in our experiments. To train a model to be tested on the MOT16/17 testing set, we used the training sets of MOT16/17, the full dataset of CrowdHuman, and the training set of the HIE dataset. To train a model to be tested on the MOT20 testing set, we used the MOT20 training set and the full CrowdHuman dataset. For the MOT16/17 validation experiments, we divided the MOT16/17 training set into two equal parts, one for training (together with the CrowdHuman dataset) and the other for validation. We treated each frame of the CrowdHuman dataset as an individual video. Following Peng et al. 2020, we applied random crop, photometric distortion, and random flip for data augmentation.

4.2.1.2 Networks Settings

We used ResNet-101 (He et al. 2016) with FPN (Lin et al. 2017) as the backbone network. The dimensions of the feature embeddings were set to 1024 and 512 for the detection and tracking branches, respectively.

4.2.1.3 Training

We used SGD as the optimizer, with a learning rate of 0.002 and a batch size of 8 images. The network was trained for 10 epochs (about 480,000 iterations). For both training and testing, the size of the feature embedding pool was set to $B = 7$ frames where 4 frames are randomly selected from the neighboring 25 frames ($\tau_s = 12$) and 3 frames are randomly selected from the whole video. Only the top $K = 375$ RoIs with the highest scores in each frame were kept for reference. The momentum was set to 0.9, and the weight decay was 0.0001. Other hyperparameters in the detection branch were the same as popular settings for the Faster-RCNN-COCO configuration (e.g., configuration from mmdetection (Chen et al. 2019)).

4.2.1.4 Inference

We filtered the detection results using a confidence threshold of 0.9. Hyperparameters θ_a and θ_d for tracklet management were 0.5 and 1 respectively in our default setting. All tracklets were cached for 30 frames at least. The reference pool was set to be the recent 7 frames with the top 375 RoIs per frame during inference.

4.2.2 Evaluation

4.2.2.1 Datasets and Metrics

The proposed approach was evaluated on the testing sets of MOT16, MOT17, and MOT20 benchmarks (Milan et al. 2016; Dendorfer et al. 2020). There are 7 testing videos in MOT16/17 and 4 testing videos in MOT20. MOT16/17 contains both static and hand-held camera views,

Online	Det	Joint?	Det Type	Method	MOTA↑	IDF1↑	Recall↑	Prcn↑	MT↑	ML↓	IDs↓
No	Private	No	-	NOMT(Choi 2015)	62.2	62.6	65.3	95.9	247	236	**406**
				MCMOT(Lee et al. 2016)	62.4	51.6	68.6	92.7	239	184	1394
				LM_CNN(Babaee et al. 2019)	67.4	61.2	73.4	93.0	290	146	931
				KDNT(Yu et al. 2016)	68.2	60.0	75.0	92.3	311	144	933
				LMP_p(Tang et al. 2017)	71.0	70.1	75.6	94.6	356	166	434
Yes				VMaxx(Wan et al. 2018)	62.6	49.2	69.2	92.2	248	160	1389
				RAR16(Fang et al. 2018)	63.0	63.8	70.8	90.4	303	168	482
				TAP(Zhou et al. 2018)	64.8	73.5	72.2	91.0	292	164	571
				CNNMTT(Mahmoudi et al. 2019)	65.2	62.2	69.3	95.1	246	162	946
				POI(Yu et al. 2016)	66.1	65.1	69.3	**96.2**	258	158	805
Yes	Private	Yes	Anchor-based	Tube_TK(Pang et al. 2020)	66.9	62.2	73.9	92.1	296	122	1236
Yes				CTrackerV1(Peng et al. 2020)	67.6	57.2	73.5	93.8	250	175	1897
No				QuasiDense(Pang et al. 2021)	69.8	67.1	75.8	93.3	316	150	1097
Yes				Ours	70.9	63.8	77.6	92.8	284	**100**	1223
Yes			Anchor-free	TraDeS(Wu et al. 2021)	70.1	64.7	75.2	94.4	283	152	1144
				GSDT(Wang et al. 2021c)	74.5	68.1	80.0	94.2	313	131	1229
Yes				FairMOT(Zhang et al. 2021)	74.9	72.8	-	-	339	120	1074
				GRTU(Wang et al. 2021b)	76.5	**75.9**	**83.1**	93.0	**391**	129	584
				Ours+	**76.8**	74.2	82.6	94.0	332	40	984

TABLE 4.1. Results of the MOT16 benchmark. The best result is **bolded**. ↑ indicates that higher is better, and ↓ indicates that lower is better.

Online	Det	Joint?	Det Type	Method	MOTA↑	IDF1↑	Recall↑	Prcn↑	MT↑	ML↓	IDs↓
Yes	Public	No	Mixed	STRN(Xu et al. 2019b)	50.9	56.0	55.8	92.6	446	797	2397
				Tracktor++v2(Bergmann et al. 2019)	56.3	55.1	58.3	**97.4**	498	831	1987
				GSM(Liu et al. 2020)	56.4	57.8	59.2	95.9	523	813	1485
No				MPNTrack(Brasó and Leal-Taixé 2020)	58.8	61.7	62.1	95.3	679	788	**1185**
				Lif_T(Hornakova et al. 2020)	60.5	65.6	63.4	96.0	637	791	1189
Yes	Private	Yes	Anchor-based	Tube_TK(Pang et al. 2020)	63.0	58.6	68.5	93.5	735	469	4137
Yes				CTrackerV1(Peng et al. 2020)	66.6	57.4	71.6	94.8	759	570	5529
No				QuasiDense(Pang et al. 2021)	68.7	66.3	74.0	94.0	957	516	3378
Yes				Ours	70.3	63.1	76.0	93.8	855	**354**	3861
Yes			Anchor-free	CenterTrack(Zhou et al. 2020)	67.3	59.9	71.9	94.6	822	584	2898
				TraDeS(Wu et al. 2021)	69.1	63.9	73.4	95.2	858	507	3555
				PermaTrack(Tokmakov et al. 2021)	73.8	68.9	79.6	93.9	1032	405	3699
				GRTU(Wang et al. 2021b)	74.9	**75.0**	**80.9**	93.4	**1170**	444	1812
				Ours+	**76.3**	73.4	**80.9**	95.2	996	468	3069

TABLE 4.2. Results of the MOT17 benchmark. The best result is **bolded**. ↑ indicates that higher is better, and ↓ indicates that lower is better.

but MOT20 only contains static camera views. However, MOT20 has a higher object density than MOT16/17. MOT17 shares the same training and testing videos with MOT16 but has provided three public detectors and made the ground truth more accurate. Therefore, the testing results of MOT17 are similar to those of MOT16 in terms of percentages but the absolute numbers in the results from MOT17 are about 3 times those from MOT16. The ablation studies were conducted on MOT16 to avoid duplicate testing on MOT17. The test

video sequences contain a variety of complex scenes, which are sufficient to demonstrate the performance of modern detectors and trackers.

The benchmark uses the CLEAR MOT metric and IDF1 scores for evaluation. The main indicators are the MOTA and IDF1 scores. The MOTA score treats the error sources, i.e., false positives (FP), false negatives (FN), and identity switches (IDS), with equal weight. Given that IDS (assignment errors) are usually much fewer than the number of FNs, MOTA is relatively unaffected by the identity assignments in the tracking process, but can fairly indicate the precision and recall of joint detection and tracking. IDF1 tends to calculate the average maximum tracked ratio of trajectories and can thus better indicate the tracking ability of an algorithm. The reason we choose the MOT benchmark is that the evaluation metrics MOTA and IDF1 in MOT can simultaneously indicate the performance of detection and tracking, whereas mAP in VID is for detection only. The MOT benchmark also provides some other metrics for detailed analysis, for example, MT and ML represent mostly tracked and mostly lost, respectively. Mostly tracked counts the number of trajectories that are 80% or more correctly tracked. Mostly lost counts the number of trajectories which are 20% or less correctly tracked.

4.2.2.2 Overall Results

Table 4.1 shows the results of recently published methods and our approach based on the MOT16 testing set. Of all the published methods of joint detection and tracking TraDeS (Wu et al. 2021), GSDT (Wang et al. 2020) FairMOT (Zhang et al. 2021) and GRTU (Wang et al. 2021b) are based on an advanced anchor-free detection framework, whereas TubeTK (Pang et al. 2020), CTrackerV1 (Peng et al. 2020), QuasiDense (Pang et al. 2021) are based on an anchor-based RPN detection framework (Faster RCNN). Specifically, TubeTK (Pang et al. 2020) uses the 3D Conv backbone, whereas GRTU utilizes synthetic datasets. The other methods are not based on joint detection and tracking frameworks but are either based on independent detector results or public detection results. To compare different MOT trackers in a relatively fair way, we obtained two testing results for MOT16. Specifically, “Ours" represents our approach using the Faster-RCNN framework (anchor-based), while “Ours+"

represents our approach using the YoloX (Ge et al. 2021) framework (anchor-free). Compared with methods based on an anchor-based RPN (Faster RCNN), “Ours" achieved the highest 70.9% MOTA. Given that QuasiDense does not have online processing, causing it to have a higher IDF1, our approach still outperformed other online processing trackers. Compared with advanced anchor-free methods, “Ours+" with YoloX (anchor-free) achieved the best MOTA and the second-best IDF1 of all the methods. GRTU uses additional synthetic data and thus had a higher IDF1 than ours.

Table 4.2 shows the results of recently published methods and our approach on the MOT17 testing set. Of all the published methods that had private detectors, CenterTrack (Zhou et al. 2020), TraDeS (Wu et al. 2021), PermaTrack (Tokmakov et al. 2021), and GRTU (Wang et al. 2021b) are based on an advanced anchor-free detection framework. Moreover, PermaTrack and GRTU utilized synthetic datasets. We also have two testing results on MOT17. As above, “Ours" represents our approach using the Faster-RCNN framework, while “Ours+" represents our approach using the YoloX (Ge et al. 2021) framework. Compared with methods based on anchor-based RPN (Faster RCNN), “Ours" achieved the highest 70.3% MOTA. Similar to the results for MOT16, QuasiDense does not have online processing and thus has a higher IDF1. Compared with advanced anchor-free methods, “Ours+" with YoloX (anchor-free) achieved the best MOTA and the second-best IDF1 among all the methods. GRTU uses additional synthetic data and thus had a higher IDF1 than ours.

Table 4.3 shows the results of recently published online processing methods with private detectors and our approach in the MOT20 testing set. Our approach (with the YoloX framework) performed better than most of the recently published methods, including TLR, RelationTrack (pixel-level correlation), and GSDT(GNN), which also includes information from adjacent frames.

We noticed that our method did not perform the best in terms of some side metrics such as MT and ML. These metrics are designed to analyze the characteristics of a tracker and are not applicable to performance comparison. For instance, a higher MT number indicates that more trajectories are at least 80%+ correctly tracked, but does not tell whether they are 85%, 90%, or 95%. Our method had a higher MOTA, but the lower MT might indicate that we got fewer

Online	Det	Joint?	Det Type	Method	MOTA↑	IDF1↑	Recall↑	Prcn↑	MT↑	ML↓	IDs↓
				TMOH(Stadler and Beyerer 2021a)	60.1	61.2	67.9	90.2	580	221	2342
				FairMOT(Zhang et al. 2021)	61.8	67.3	82.8	80.6	**855**	**94**	5243
				TLR(Wang et al. 2021a)	65.2	69.1	**81.5**	84.2	825	111	5183
Yes	Private	Yes	Anchor-free	GSDT(Wang et al. 2021c)	67.1	67.5	73.8	92.4	660	164	3230
				RelationTrack(Yu et al. 2022)	67.2	**70.5**	79.8	87.1	773	111	4243
				CrowdTrack(Stadler and Beyerer 2021b)	70.7	68.2	75.5	**94.7**	682	150	3198
				Ours+	**73.4**	69.3	78.2	94.6	804	151	**1684**

TABLE 4.3. Results of the MOT20 benchmark (with private detector). The best result is **bolded**. ↑ indicates that higher is better, and ↓ indicates that lower is better.

MostTrack (80%+ tracked) trajectories but that most of the trajectories had a higher average accuracy, resulting in higher MOTAs and IDF1s. Another problem is the recall-precision balance. By adjusting the confidence thresholds, we could get different recall-precision pairs. We present the most balanced results here, and thus our results did not always get the highest precision and recall. Our primary objective was to obtain a higher MOTA while obtaining a higher IDF1 since these are the main indicators of detection and tracking performance.

Compare with methods using the same detection frameworks and backbones. We built our own baseline using Faster-RCNN based on the ResNet-101 backbone. However, of the methods with private detectors, many methods, e.g., PermaTrack (Tokmakov et al. 2021), GRTU (Wang et al. 2021b), FairMOT (Zhang et al. 2021), TLR (Wang et al. 2021a), and GSDT (Wang et al. 2021c) used better object detection frameworks (e.g., point-based detection), synthetic data, or other training techniques, which are complementary to our relation module design. As shown in Tables 4.2 and 4.3, when we implemented our approach with a better detection framework, e.g., YoloX, our method performed better. For a fairer comparison under the same framework and backbone settings as our baseline, we chose CTracker (Peng et al. 2020), which is also based on the Faster RCNN and ResNet series, to conduct further experiments. We re-ran CTracker using ResNet-101 and compared the result with our baseline in the MOT16 **testing set**. The results are presented in Table 4.4. These results show that under very similar conditions, i.e., both based on a Faster-RCNN and a ResNet-101 backbone, our baseline was close to CTracker. The results for ResNet-50 and ResNet-101 using CTracker also showed that the use of ResNet-101 did not bring much improvement (0.4% improvement in MOTA) compared with ResNet-50. On the other

method	backbone	MOTA	time (ms)
CTrackerV1(Peng et al. 2020)	ResNet-50	67.6	120
CTrackerV1(Peng et al. 2020)	ResNet-101	68.0	160
Our baseline	ResNet-101	68.4	130
Ours with SQGA	ResNet-101	70.9	175

TABLE 4.4. Time cost and performance comparison with Peng et al. 2020. Tested on MOT16 testing set using single NVIDIA TITAN XP and Intel Xeon CPU E5-2620 @ 2.1 GHz.

Model	Detection Branch Vanilla	Detection Branch Similarity	Detection Branch Quality	Tracking Branch Shared	Tracking Branch Vanilla	Tracking Branch SQGA	MOTA↑	IDF1↑	IDs↓
Non-Relation-Learning							64.3	65.2	322
Shared-Relation-Learning	✓			✓			62.1	64.3	608
Vanilla-Attention-Det	✓						64.1	66.3	351
Vanilla-Attention-Both	✓				✓		63.7	65.9	340
Sim-Attention-Det		✓					65.3	68.9	355
Qlt-Attention-Det			✓				65.1	66.9	360
SQGA-Det		✓	✓				66.2	67.1	356
SQGA-Both		✓	✓			✓	**66.4**	**68.9**	337

TABLE 4.5. Ablation study on MOT16 validation set for different types of attention heads.

hand, our proposed method (with Faster-RCNN and ResNet-101) improved MOTA by 2.5%, showing that the improvement from our design is more obvious than those obtained by the use of more powerful backbone models. In addition, SQGA required only 10% more execution time than the ResNet-101 version, which we developed based on Peng et al. 2020 and about 30% more execution time than our baseline, both of which are acceptable.

4.2.3 Ablation Study and Analysis

Quantitative experiments were conducted on the MOT16 validation set to investigate the components of the proposed approach.

4.2.3.1 Comparison with Different Attention Mechanisms

Table 4.5 shows different types of attention heads. The Non-Relation-Learning model is a simple anchor-based multi-task baseline (Faster-RCNN with ResNet-101 and multi-branches) for joint detection and tracking without utilizing any spatial-temporal information across video

	Model	Guidance	MOTA↑	IDF1↑	IDs↓
Sim	A	latent	64.1	66.3	351
	B	identity	64.9	65.7	397
	C	class	**66.2**	**67.1**	356
Qlt	D	thresholding	65.4	66.9	371
	E	comparison	**66.2**	**67.1**	356
	F	similarity	65.8	66.6	369

TABLE 4.6. Comparison using different forms of similarity and quality guidance rules. 'Sim' indicates that similarity-guided attention was used, 'Qlt' indicates that quality-guided attention was used. 'latent' indicates no direct supervision. 'identity' indicates that only samples with the same identity were selected, while 'class' selected samples of the same object/background class. 'thresholding' indicates that features with a lower quality than a threshold (0.5 in this experiment) were ignored. 'comparison' indicates that comparatively higher quality features were used. 'similarity' indicates that similarity ranking was used (ignoring some features with the lowest ranking). Our final model chose model C for the similarity guidance and model E for the quality guidance.

Frames	4	7	10	13
RoIs	1500	2625	3750	4875
MOTA↑	65.8	66.2	66.1	65.5
IDF1↑	67.4	67.1	66.5	67.6
IDs↓	317	356	374	373

TABLE 4.7. Performance comparison based on the different numbers of frames used for training.

frames. The Shared-Relation-Learning model performs a single shared vanilla multi-head attention relation module (top row in Fig. 4.1) for both tasks. The Vanilla-Attention-Det model performs a vanilla multi-head attention relation module on the detection classification branch (not shared with the tracking branch). The Vanilla-Attention-Both model performs two different vanilla multi-head attention relation modules on both the detection and tracking branches (parameters not shared). Compared with the Non-Relation-Learning baseline, all the vanilla attention designs failed to improve the MOTA because self-guided multi-head attention is not well-learned in the joint task. A comparison between Vanilla-Attention-Both and Shared-Relation-Learning shows that the dual relation modules performed better than the single shared relation module. A comparison between Vanilla-Attention-Both and Vanilla-Attention-Det

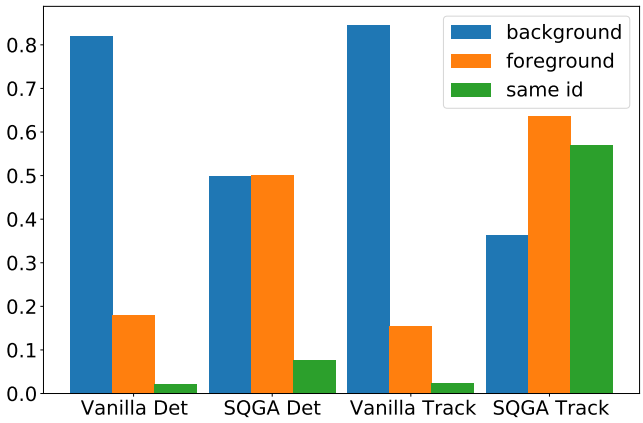


FIGURE 4.3. Statistics associated with what was attended to with respect to foreground objects in different tasks with different attention mechanisms. SQGA showed the most balanced and task-specific attention.

shows that involving a vanilla attention relation module for the tracking branch even resulted in a decrease in terms of MOTA. The results of the Sim-Attention-Det, Qlt-Attention-Det, and SQGA-Det models reveal the ablation study of the proposed SQGA mechanism on the detection classification branch. The Sim-Attention-Det model imposes supervision only on the similarity term. The Qlt-Attention-Det model only imposes quality supervision. The SQGA-Det model performs full similarity and quality-guided attention. None of these models share the relation module parameters between branches. The results show that the SQGA-Det model, which uses both similarity and quality for guiding attention, achieved a better MOTA than either similarity-guidance-only or quality-guidance-only (improving MOTA by 0.9%). When we further applied the SQGA relation module to the tracking branch, the resulting model SQGA-Both (bottom row in Fig. 4.1)further improved the IDF1 by 1.8% and slightly improved the MOTA score as well. Compared with a Non-Relation-Learning baseline, our approach (SQGA-Both) improved MOTA by 2.1% and improved IDF1 by 3.7%. Compared with vanilla multi-head attention dual relation modules (Vanilla-Attention-Both), our method (SQGA-Both) improved by 2.7% in MOTA and 3.0% in IDF1. More importantly, involving the SQGA module on the tracking branch did not lead to a performance drop. The experiments above verified the effectiveness of the proposed approach. The SQGA module on the detection branch improved both the MOTA and IDF1 scores, whereas the SQGA module on the tracking branch mainly improved the IDF1 score.

4.2.3.2 Comparison between Different SQGA Forms

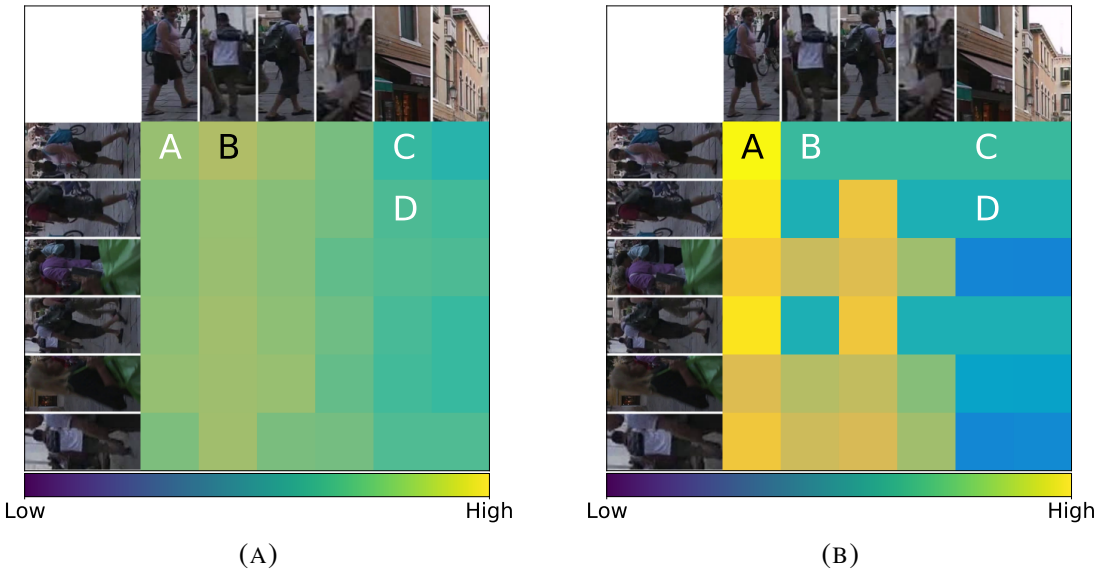
We further investigated several forms of similarity and quality guidance rules that are different from those proposed in Section 4.1.3. These experiments were done on the detection classification branch. Model A in Table 4.6 utilizes vanilla multi-head attention that is self-learned in latent feature space. Model B uses identity supervision on the similarity term and only aggregates a subset of features compared with model C, i.e., class label supervision. A comparison between B and C shows that correct similarity guidance increased the gain in overall performance. Models D, E, and F show 3 different forms of quality guidance including thresholding (i.e., ignoring features with quality less than a threshold of 0.5), comparison (the way we designed in Eq. (4.12), and similarity (top 50% features with highest similarity scores). These experiments showed that the two quality-guided forms in Models D and E are better than the similarity ranking strategy in Model F, especially in terms of IDF1. We chose the comparison strategy in this work because it does not have a hyperparameter and provided a greater gain in terms of MOTA. In contrast, the thresholding strategy includes the extra hyperparameter of a threshold, which would vary in different scenarios and thus would provide less total improvement. These experiments clearly demonstrate the benefits of employing correct similarity and quality guidance.

4.2.3.3 Comparison between Different Number of Frames

Table 4.7 shows the performances of the models with different numbers of frames and RoIs. For every frame and every level (5 levels in total) in the feature pyramid feature networks (FPNs), only the top 75 RoIs were reserved (resulting in $K = 375$ features per frame). This shows that too few or too many frames (RoIs) harmed the performance. When the number of frames was set to about 7 - 10, and the number of RoIs was about 3000, the method produced a satisfactory performance.

4.2.3.4 Visualization of the Learned Affinities

We visualized the learned affinities of our method and vanilla multi-head attention for further analysis.



(A) (B)

FIGURE 4.4. Visualization of the affinity matrices in the detection branch. SQGA (B) learns better affinities than the vanilla design (A). Comparing A and B in (B) and (A) shows that SQGA focuses more on objects with the same identity as the one to be refined. Comparing C and D in (B) and (A) shows that SQGA learns to ignore irrelevant background (best be viewed by zooming in).

The statistics in Fig. 4.3 show that about 80% of the features aggregated to a foreground object are actually from background areas using vanilla multi-head attention (Fig. 4.3 vanilla Det and vanilla Track). Because the attention value is non-negative, aggregating too many background features to foreground features will distort the foreground features rather than refine the features. In SQGA, the ratio between the background and foreground features fell to less than 50% in the detection branch, i.e., SQGA Det in Fig. 4.3and fell further to 36% for the SQGA in the tracking branch, i.e., SQGA Track in Fig. 4.3. Fig. 4.3 also shows that the SQGA for detection and the SQGA for tracking learned to perform different relation connections. The context draws more attention to detection when objects of the same identities are favored in tracking. This visualization verifies that our design based on *Reason* 1 (dual relation modules in different tasks) and *Reason* 2 (guidance for the attention to fit the need of specific tasks) in Section 4.1.2.1 are necessary and achieved by the SQGA.

Fig. 4.4 further shows an actual case of relation modeling of the detection branch. For the vanilla attention in Fig. 4.4a, the difference between the reference regions is small and, thus, did not provide distinctive information. These noisy affinity values actually confused the task.

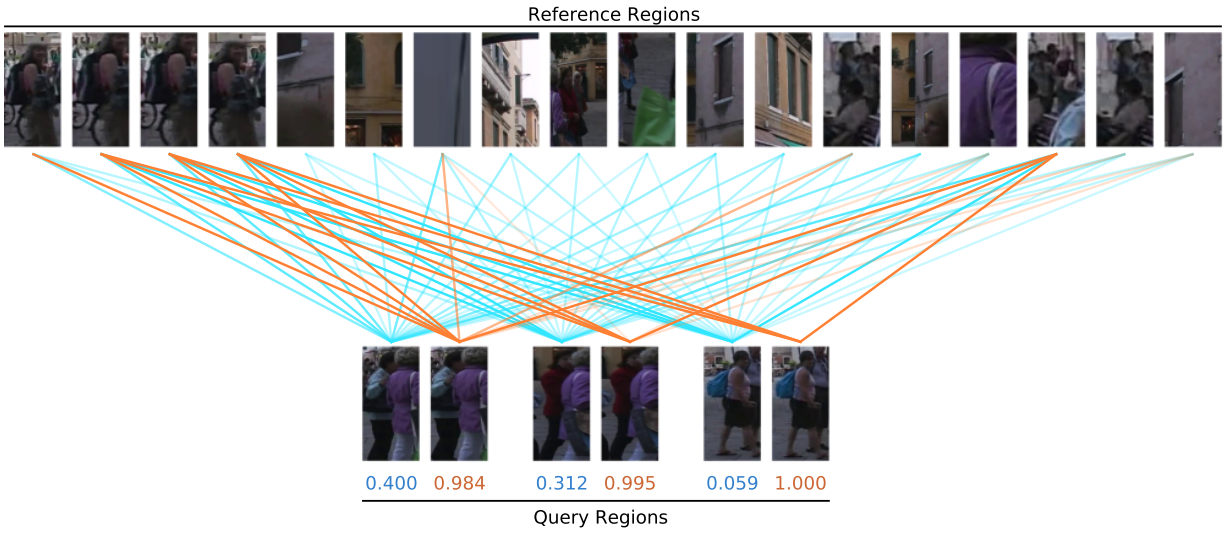


FIGURE 4.5. Comparing SQGA cases with vanilla attention cases. Blue indicates the vanilla design results and orange indicates the SQGA results. Deeper colors of lines indicate higher attention weights. All these cases were wrongly classified by vanilla attention design but were corrected by the proposed SQGA (i.e., confidence scores changed from < 0.5 to > 0.9). In vanilla design, most regions are weighted with quite small weights, thus leading to noisy results. In SQGA, the attentions are better balanced and thus better results are achieved.

In Fig. 4.4a, some background regions still have observable positive weights for the objects (C and D in Fig. 4.4a), while SQGA in Fig. 4.4b can have much lower weights. When the identical person appears in both the target and the reference frames, SQGA can identify it (A in Fig. 4.4b) and can help to enhance the feature for this person. But vanilla attention has a higher weight for a separate identity (B in Fig. 4.4a), which could make the primary feature in the target frames blurry and more difficult to detect.

Fig. 4.5 illustrates several examples of how SQGA achieves better results by effective relation learning. All these cases were wrongly classified by vanilla attention design but were correctly classified by the proposed SQGA (i.e., the confidence score changed from < 0.5 to > 0.9). In vanilla design, most regions are weighted by quite small weights. Context regions are too focused (from the 5-th to the 12-th reference regions from the left)), leading to noisy results. In SQGA, the attention is more balanced, leading to better results. (Meaningful left-most 4 reference regions are more focused.)

CHAPTER 5

Towards Frame-Rate Agnostic Multi-Object Tracking

5.1 Methodology

5.1.1 Frame Rate Agnostic MOT

In this section, we introduce the Frame Rate Agnostic Multi-Object Tracking (FraMOT) problem for a more flexible, intelligent, and robust tracking solution. As an extended challenge of traditional MOT, the desired output of the FraMOT problem is the same as the classical MOT problem, i.e., finding out all objects of interest along the input video with each detection result having a bounding box and an identity number. The identity number should be the same for detection results in different frames as long as they belong to the same identity instance. The only difference between FraMOT and the normal MOT problem is the input, i.e., input videos of the normal MOT are all sampled in a fixed small range of frame rates (e.g., 25 frames per second), while in FraMOT, the input frame rate is agnostic and could vary according to different situations. Besides, we usually know the exact frame rate in the classical MOT problem, but this may not be true in FraMOT due to dynamic sampling strategies or some other pre-processing steps. Therefore, we consider two different testing modes in FraMOT, i.e., known frame rate and unknown frame rate, to simulate different deployment scenarios.

New challenges of FraMOT. The direct challenge that FraMOT brings is how to handle agnostic streaming frame rates, which requires the trackers to handle more complex and diverse motion and appearance shifts, and makes most previous techniques and heuristic strategies (i.e., classical association model design, training scheme, and target management

methods) not working anymore. First, it is more difficult to train the association module following the previous most commonly used network design, which never considers multi-frame-rate inputs and thus lacks the mechanisms to process dynamic motion-appearance relations. Second, previous training schemes are based on the assumption that post-processing and target management do not influence the performance of the association module, which might be true in high frame rate tracking but is false in FraMOT. In classical MOT training, the inputs of the association network are the bounding boxes and appearance features provided by the ***raw*** *detection results* for adjacent key frame pairs. However, in the inference stage, the inputs of the association network (bounding boxes and appearance features) are provided by *the target management*, which is different from raw detection results. The reason is that a lot of post-processing steps, such as trajectory cache strategy, occlusion status handling, and feature and motion smoothing techniques, are applied. These post-processing procedures performed only in the inference stage will change the locations of the bounding boxes used as the input data to the association networks in the following frames and thus will change the association results in the future frames. For example, Kalman Filter, a commonly used motion smoothing technique, will predict the movement of each trajectory based on historical locations, which provides the association networks with a predicted object location instead of the location from the original detection result in the following frames. Another example is that the caching strategy will provide/add temporarily lost targets (object bounding boxes) to the association networks, which are not present during the training stage. Considering the small sampling interval that leads to small object movements, the changes of the input data to the association network might be negligible in normal frame rate scenarios and thus have a minor impact in traditional association training. However, when video frame rates become lower (sampling interval is increased), the object movements become larger, and thus the influence of this gap could be significant (e.g., the predicted movement is larger and less accurate and there are more temporally lost cached targets in lower frame rate scenarios). The enlarged changes lead to an enlarged gap between the training stage and inference stage and result in an accuracy drop during inference. Moreover, these post-processing steps are not learnable but are necessary to manage the trajectories and improve the overall accuracy, which

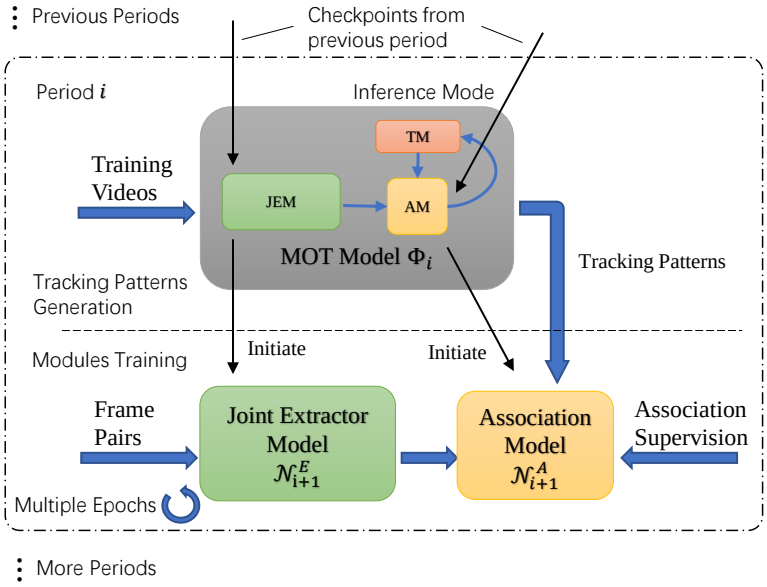


FIGURE 5.1. The training pipeline of the proposed **F**rame-rate **A**gnostic MOT framework with a **P**eriodic Training **S**cheme (FAPS) at top level. There are multiple training periods in the framework, each period contains two stages, i.e., tracking patterns generation and module training. In tracking pattern generation, we generate the tracking patterns using the previous MOT model. Then these tracking patterns will be used for module training, providing brief information on the testing environment. 'JEM', 'AM', and 'TM' are short for Joint Extractor Module, Association Module, and Target Management, respectively.

makes it hard to eliminate the gap between training and inference. Thus, new approaches for data association network design and training schemes are needed.

5.1.2 Frame Rate Agnostic MOT Framework with a Periodic Training Scheme

In this section, we introduce our Frame Rate Agnostic MOT framework with a Periodic training Scheme (FAPS) to address the new challenges of FraMOT.

5.1.2.1 Overview

There are three different modules in the framework, i.e., Joint Extractor Module (JEM), Association Module (AM), and Trajectory Management Module (TM). The JEM generates

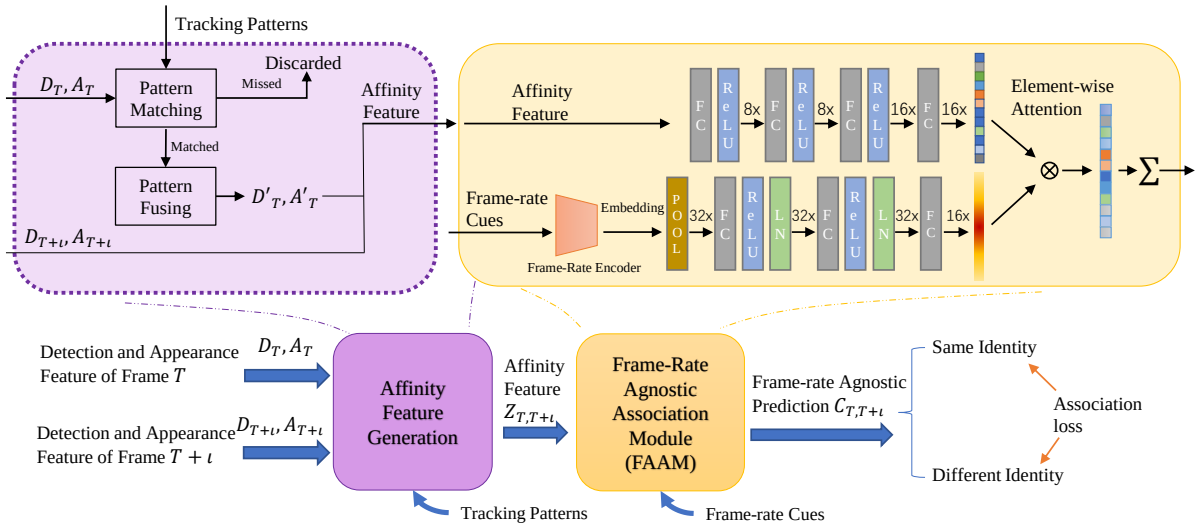


FIGURE 5.2. The training pipeline of the Association Model (AM). The detection bounding boxes $D_T, D_{T+\iota}$ and corresponding appearance features $A_T, A_{T+\iota}$ of Frame T and $T+\iota$ generated by the joint extractor (e.g., YOLOX detector with an appearance embedding branch) are firstly fed to the Affinity Feature Generation Module and are matched with tracking patterns. Those matched and fused with some patterns (i.e., D'_T and A'_T) form pairs with $D_{T+\iota} and A_{T+\iota}$ and generate the affinity features $Z_{T,T+\iota}$. Then affinity features $Z_{T,T+\iota}$ will be fed to the Frame Rate Agnostic Association Module (FAAM) and fused with the frame rate embedding generated from the provided frame rate cues. Finally, the voted prediction $C_{T,T+\iota}$ of the module is supervised by the association loss.

the detection results and corresponding appearance feature embeddings from raw images. The AM associates the new detection results with existing trajectories. The TM determines the initiation and termination of all trajectories, making them smoother and handling their status.

The core module of the proposed framework is the Association Module. We design a new Frame Rate Agnostic Association Module with mechanisms to encode frame rate information, providing the capability to handle complex motion-appearance relations of the various frame rates. At the same time, the framework is trained with the proposed Periodic Training Scheme (PTS), which takes all post-processing steps into consideration, providing a simulation of the inference stage environment and thus reducing the gap of data association between training and inference.

Fig. 5.1 illustrates the overview of the proposed framework. The training pipeline follows the proposed PTS consisting of several training periods, each period contains two stages, i.e., tracking patterns generation stage, and tracking module training stage. Specifically, the tracking patterns generation stage conducts a forward pass using the model checkpoints from the last period and generates the tracking patterns. The tracking patterns include some inference stage information that helps simulate the inference run-time (details in 5.1.2.2). In the module training stage, the Association Model takes the output of the JEM and the tracking patterns as input, generates the affinity features, predicts the association scores using the proposed Frame Rate Agnostic Association Module (FAAM), and is supervised by the corresponding association ground-truth signals, as shown in Fig. 5.2. Especially, during training, instead of passing the input data to the FAAM directly, we design an affinity feature generation module to adjust the affinity features utilizing the generated tracking patterns via pattern matching and fusion. Then the adjusted affinity features will be passed through the FAAM. The FAAM networks utilize the frame rate information to infer the frame rate aware attention and enhance the association prediction. During Inference, the model checkpoints of the last period will be used. The association model only takes the output of the JEM as input, the tracking patterns are no longer needed and the pattern matching and fusion step is removed. The inference pipeline is the same as the tracking patterns generation pipeline.

5.1.2.2 Periodic Training Scheme

As mentioned in Section 5.1.1, one of the challenges that FraMOT brings to us is that the diverse video frame rates enlarge the gap between frame-pair association training and actual inference run-time. The most straightforward solution is adding these inference stage post-processing procedures to the training stage, which is hard to implement since most of these procedures can not be backward propagated. To address this issue, we propose to reflect the inference time post-processing steps during training by considering the tracking patterns in the association network (detailed in Section 5.1.2.3), which is generated via the Periodic Training Scheme.

Here, the tracking patterns are some records that can reflect the inference time patterns, which depend on the post-processing steps of the tracker. In this paper, tracking patterns include the location of each tracked target at every frame (denoted by p_i^{loc}), the Kalman-Filter-predicted movement based on the previous trajectory, the cached appearance feature embedding (denoted by p_i^{pred} and p_i^{apr}, respectively), as well as the level index (denoted by p_i^{lvl}) of the two-stage association strategy because we use a cascade association strategy as some previous works (Yu et al. 2016; Zhang et al. 2021). Other patterns such as the tracklet occlusion status can also be included if the tracker does have a carefully-designed occlusion management.

The Periodic Training Scheme aims at generating these tracking patterns in a periodic manner considering that the inference stage tracking behaviors are not going to change significantly within a short training period. As shown in Fig. 5.1, we set up multiple training periods and re-sample the tracking patterns before each period starts. In the tracking pattern generation stage of a new period, we utilize the trained checkpoints of the previous period to build an MOT tracker, conduct tracking on full training videos, and generate the tracking patterns that will be used in the affinity feature generation pipeline (see Section 5.1.2.3). Then we train all the models using the training data and tracking patterns. When a training period is done, we then have the updated checkpoints to generate tracking patterns for the next period. For the first period, we use a pre-trained joint extractor and a trivial hand-crafted association module to generate the patterns instead of using randomly initialized checkpoints. For example, in this paper, we use the pre-trained YOLOX detection framework (Ge et al. 2021) with appearance embedding branch as the JEM, a linear combination of the spatial distances and appearance similarities as the AM, and the commonly used Hungarian Algorithm and Kalman Filter with thresholding strategies for target initiation and termination as the TM (detailed in Section 5.1.2.4). For the later periods, we use the proposed Frame Rate Agnostic Association Module (FAAM), and the architecture of JEM and TM remain the same.

The whole pipeline is shown in Procedure 1. $L(\mathcal{X}, P_t; \mathcal{N}_t)$ is the overall loss which combines detection losses and id loss from the JEM, and association loss from the AM (see Section 5.1.2.3). λ_E and λ_A are learning rates.

Procedure 1 Periodic Training Scheme for MOT

Input: Period Number N_p, Training Data $\mathcal{X}$
Output: Trained Model Φ_{N_p}
  Initialize $\mathcal{N}_0^E$ with pre-trained joint extractor;
  Initialize $\mathcal{N}_0^A$ with random parameters;
  $\Phi_0 \leftarrow (\mathcal{N}_0^E, \gamma)$; {trivial association module γ}
  for t:=1 to N_p **do**
    $P_t \leftarrow \text{Track}(\mathcal{X}, \Phi_{t-1})$;
    $\mathcal{N}_t^E, \mathcal{N}_t^A \leftarrow \mathcal{N}_{t-1}^E, \mathcal{N}_{t-1}^A$
    for i:=1 to τ **do**
      $\mathcal{N}_t \leftarrow \mathcal{N}_t^E, \mathcal{N}_t^A$
      $\mathcal{N}_t^E \leftarrow \mathcal{N}_t^E - \lambda_E \nabla_{\mathcal{N}_{t-1}^E} L(\mathcal{X}, P_t; \mathcal{N}_t)$
      $\mathcal{N}_t^A \leftarrow \mathcal{N}_t^A - \lambda_A \nabla_{\mathcal{N}_{t-1}^A} L(\mathcal{X}, P_t; \mathcal{N}_t)$
    end for
    $\Phi_t \leftarrow (\mathcal{N}_t^E, \mathcal{N}_t^A)$;
  end for
  Return Φ_{N_p}

5.1.2.3 Frame Rate Agnostic Association

The Association Model (AM) takes the detection results and corresponding appearance feature embeddings from the JEM and the tracking patterns (for training only) provided by the PTS as inputs and generates the affinity scores supervised by ground-truth labels. As shown in Fig. 5.2, there are two parts in the AM, i.e., Affinity Feature Generation Module and Frame Rate Agnostic Association Module.

Affinity Feature Generation. Given a frame pair of frame T and $T + \iota$, where ι is **not** a constant (i.e., the sampling interval is not fixed), the JEM generates the detection results $D_T = \{B_{T,i} \mid i = 1, 2, \ldots, N_T\}$, $D_{T+\iota} = \{B_{T+\iota,i} \mid i = 1, 2, \ldots, N_{T+\iota}\}$, and the corresponding appearance features $A_T = \{f_{T,i} \mid i = 1, 2, \ldots, N_T\}$, $A_{T+\iota} = \{f_{T+\iota,i} \mid i = 1, 2, \ldots, N_{T+\iota}$, where N_T denotes the numbers of proposals of frame T, $B_{T,i}$ contains the bounding box and confidence score of proposal i in frame T, and $f_{T,i}$ represents the appearance feature vector of proposal i in frame T, respectively. At the same time, we have the tracking patterns from frame T, denoted by $P_T = p_j \mid j = 1, 2, \ldots, N_T^p$, where $p_j = \{p_j^{loc}, p_j^{pred}, p_j^{apr}, p_j^{lvl}\}$ consists of a bounding box location p_j^{loc}, an inference stage temporal prediction p_j^{pred} based on the inference environment, a saved appearance feature embedding p_j^{apr} from the inference time,

as well as the matching level index p_j^{lvl} in the two-stage association. We then construct a pattern matching and fusing pipeline to simulate the inference stage tracking environment for frame rate agnostic training. As shown in the left part of Fig. 5.2, only detections that are matched with a certain p_j will be chosen for the training, otherwise, it will be discarded because they do not appear in the inference stage (e.g., being filtered out). In this work, we define a $B_{T,i}$ is matched with a pattern p_j of the same frame if and only if their Intersection Over Union $\mathrm{IoU}(B_{T,i}, p_j^{loc})$ is larger than a threshold (e.g., 0.7 as used in our work). Then the matched instances are fused with the tracking patterns and it generates

$$\begin{aligned} D'_T = \{&B_{T,i} + p_j^{pred} \mid \\ &p_j \in P_T \wedge B_{T,i} \in D_T \\ &\wedge p_j \text{is matched with} B_{T,i}\} \\ \cup\{&p_j^{loc} + p_j^{pred} \mid p_j \in P_T \\ &\wedge p_j \text{is } \textbf{NOT} \text{ matched}\}. \end{aligned} \tag{5.1}$$

The corresponding appearance features of D'_T are denoted by A'_T. Then the corresponding embedding $f'_i \in A'_T$ of $B'_i \in D'_T$ is

$$f'_i = \begin{cases} f_{T,l} & \text{if} B'_i \text{ is from some } B_{T,l}, \\ p_l^{apr} & \text{if} B'_i \text{ is from some } p_l^{loc}. \end{cases} \tag{5.2}$$

After matching and fusing, the D'_T, A'_T and $D_{T+\iota}, A_{T+\iota}$ are then used to calculate some affinity features for the association networks. We include three distance measurements and one environment variable as the affinity features, denoted by $Z_{T,T+\iota}$, where

$$\begin{aligned} Z_{T,T+\iota} = \{&\hat{\mathcal{D}}(B'_i, B_{T+\iota,j}), \mathrm{IoU}(B'_i, B_{T+\iota,j}), \\ &\frac{f'_i \cdot f_{T+\iota,j}}{\mid f'_i \mid \cdot \mid f_{T+\iota,j} \mid}, \mathrm{level}_i \\ &\mid \forall B'_i \in D'_T, B_{T+\iota,j} \in D_{T+\iota}\}. \end{aligned} \tag{5.3}$$

$\hat{\mathcal{D}}(\cdot)$ denotes the normalized euclidean distance and level_i is the matching level index p_j^{lvl} in the two stage association where B_i' appears in the tracking patterns.

Frame Rate Agnostic Association Module. The affinity features $Z_{T,T+\iota}$ (in shape of $N_s \times D_a$, N_s is the sample amount and D_a is the dimension of each sample) are then fed into a 4-layer neural network $\mathcal{N}_{aff}$ and transformed to more discriminative features f^{aff} (in the shape of $N_s \times 16D_a$), as shown in the right part of Fig. 5.2. On the other branch generates the frame rate aware attention. The frame rate cues are first encoded into a frame rate embedding $\hat{\sigma}$ by the frame rate encoder (see the next paragraph), then the embedding is pooled into the shape of $N_s \times 32D_a$, followed by the 3-layer frame rate sub-net $\mathcal{N}_{att}$ and finally the channel size is reduced to the same as the feature branch. The output of this branch is denoted by f^{att} (in the shape of $N_s \times 16D_a$). The final output prediction of the association module is

$$C_{T,T+\iota} = \sum_i \frac{f_i^{aff} \cdot \exp(f_i^{att})}{\sum_j \exp(f_j^{att})}. \tag{5.4}$$

f_i^{aff} denotes the i-th element of f^{aff} and f_j^{att} denotes the j-th element of f^{att}. Finally, we apply a binary cross entropy loss L_{assoc} for the prediction $C_{T,T+\iota}$ with label 1 for the same identity and 0 for a different one. The overall loss $L = L_{det} + \alpha_2 L_{id} + \beta_2 L_{assoc}$, where L_{det} and L_{id} is the detection losses and id loss for the joint extractor, following the similar design of Zhang et al. 2021.

Frame Rate Encoder. As discussed in Section 5.1.1, the frame rate could either be known or unknown during the deployment stage. Thus, two frame rate encoders are introduced correspondingly.

For known frame rate, we simply use the cosine embedding of the given frame rate. Specifically, the i-th element of the frame rate embedding $\hat{\sigma}_i = cos(\frac{i \cdot s \cdot F}{D_{\hat{\sigma}}})$, where F is the frame rate number, s is a constant scaling factor, and $D_{\hat{\sigma}}$ is the embedding dimension.

For unknown frame rate, we propose to utilize the Inter-frame Best-matched Distance Vector (IBDV) as the frame rate indicator. Specifically, IBDV describes the distribution of the normalized distances between instances in the two frames if the instances are best matched. This distribution roughly encodes the frame rate information, because best-matched distances

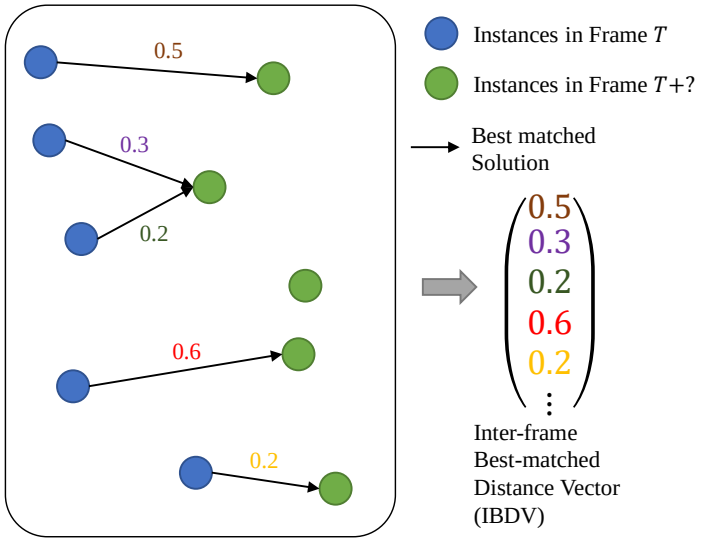


FIGURE 5.3. Illustration of Inter-frame Best-matched Distance Vector (IBDV). We calculate the normalized distances of the best matching pairs and generate the IBDV.

will increase when the frame rate is reduced. As shown in Fig. 5.3, we first find the best matching object pairs through some pre-defined criteria, e.g., minimizing spatial distances or appearance feature distances, and then put all normalized distance values of these pairs into a vector. To make its dimension unchanged, we sort the values in the vector and interpolate them into a fixed shape. We investigate the effects of different criteria to find the best matching pairs in 5.2.4.3. In our final solution, according to the results on MOT datasets, we choose the 'spatial distances" as our criterion, i.e., we choose the pairs minimizing the sum of the spatial distances as our best matching pairs. Mathematically, when the frame rate is unknown,

$$\hat{\sigma} = \mathrm{interp}(\mathrm{sort}(\{\hat{\mathcal{D}}(B'_i, B_{T+\iota,j_i}), i = 1, 2, \ldots, N'\})) \tag{5.5}$$

where $B'_i, B_{T+\iota,j_i}$ is the i-th matched pair, interp denotes the linear interpolation and sort denotes sorting the values of a vector in non-decreasing order.

5.1.2.4 Online Tracking

For online tracking, we adopt a simple but effective pipeline that consists of steps including detection filtering, Kalman-Filter-based movement prediction, and a two-stage association strategy. The target management cycle is stated in Procedure 2. We first filter out background

Procedure 2 Target Management Cycle

Input: Video Frames Set $I = I_1, I_2, \ldots, I_T$
Output: Trajectories Set $\mathcal{T} = o_1, o_2, \ldots, o_{N_\mathcal{T}}$
  $\mathcal{T} \leftarrow \emptyset$;
  for t:=1 to T **do**
    $D_t, A_t \leftarrow \mathcal{N}^E(I_t)$;
    $D_t, A_t \leftarrow \mathrm{Filter}(D_t, A_t, \lambda_{low})$;
    $D_t^{high}, A_t^{high} \leftarrow \mathrm{Filter}(D_t, A_t, \lambda_{high})$;
    $D_t^{low}, A_t^{low} \leftarrow D_t - D_t^{high}, A_t - A_t^{high}$;
    $\mathcal{T}' \leftarrow \mathrm{KalmanFilter}(\mathcal{T})$;
    $\mathcal{T}_1, D_t^{rem}, A_t^{rem} \leftarrow$
    $\mathrm{Match}_1(\mathcal{T}', D_t^{high}, A_t^{high}; E^A)$;
    $\mathcal{T}_2 \leftarrow \mathrm{Match}_2(\mathcal{T}' - \mathcal{T}_1, D_t^{low}, A_t^{low}; E^A)$;
    $T_{rem} \leftarrow \mathrm{InitTrack}(D_t^{rem}, A_t^{rem})$
    $\mathcal{T} \leftarrow \mathcal{T}_1 \cup \mathcal{T}_2 \cup (\mathcal{T}' - \mathcal{T}_1 \cup \mathcal{T}_2) \cup \mathcal{T}_{rem}$;
    $\mathcal{T} \leftarrow \mathrm{RemoveMissing}(\mathcal{T}, \lambda_i)$;
    {PTS collects $p_i^{loc}, p_i^{pred}, p_i^{apr}$ from $\mathcal{T}$}
  end for
  Return $\mathcal{T}$

objects with a threshold λ_{low}, then predict the movement for each tracked target using KalmanFilter. In the next, we perform two different stages of association, which is a commonly used technique to reduce fragments (Yu et al. 2016; Zhang et al. 2021). For the first stage, only detection results of high confidence (confidence score greater than λ_{high}) will be put into the matching pool. For the second stage, the remaining unmatched tracked targets will be again put into the matching pool and matched with the less confident detection results (confidence score less than λ_{high} but greater than λ_{low}). The association step will find out the best matching pairs minimizing the sum of pair distances (or maximizing the sum of pair scores). This can be easily calculated by the Hungarian Algorithm. Finally, the remaining unmatched detection results of high confidence (confidence score greater than λ_{high}) will be appended to the tracked target list. If a target in the tracked target list is unmatched for a long interval λ_i, then it will be dropped from the list.

5.2 Experiments

In this section, we provide quantitative experiment results, ablation studies, and in-deep analysis of the proposed approaches to tackle the FraMOT problem.

5.2.1 Evaluation Datasets and Metrics

5.2.1.1 Datasets

For evaluating FraMOT, a multi-frame-rate dataset is necessary for training and evaluating the algorithm. Instead of collecting and re-annotating extra data, we simulate the multi-frame-rate inputs from existing high-frame-rate MOT datasets. We do not consider a higher frame rate than 30 fps, because i) currently most cameras work at a frame rate of 30 fps, and streaming devices of higher frame rate are rare, and ii) to reduce computational cost most industrial scenarios tend to reduce the frame rate while not increase the frame rate, and iii) higher frame rate usually does not bring extra challenge. Given a video with F frames per second and N frames in total, the frames are denoted by $I_1, I_2, \ldots, I_N$. The target frame rate is $F'(F' < F)$ frame per second and we assume that F is divisible by F'. Our simulation method re-samples the original video and re-composes the frames into new videos, satisfying the target frame rate. Specifically, we generate $k = \frac{F}{F'}$ new videos, the i-th of which consists of frames $\hat{I}_{ij}, j = 1, 2, \ldots, \frac{N}{k}$ where

$$\hat{I}_{ij} = I_{(j-1)k+i}. \tag{5.6}$$

We call k the sampling factor of a simulation. As shown in Fig. 5.4, the original video is decomposed into a matrix of k rows and $\frac{N}{k}$ columns, with each row generating a new video at the target frame rate.

Considering that most videos nowadays have a default frame rate of 25 fps, we start from 25 fps and sample videos with 12.5 fps, 6.25 fps, 3.125 fps, 1.5625 fps, 1 fps, 0.722 fps, and 0.5 fps for a thorough evaluation of the trackers' performance. That means we choose 1, 2, 4, 8, 16, 25, 36, and 50 for k in the Eq. 5.6, leading to eight different settings in total. Although it is possible to sample videos of frame rates lower than 0.5 fps, a 0.25 fps setting will generate

FIGURE 5.4. Illustration of multi-frame-rate dataset generation. We obtain k new videos with $\frac{F}{k}$ fps given the original video with F fps.

	k=1		k=2		k=4		k=8		k=16		k=25		k=36		k=50	
Dataset	fr.	id.	fr.	id.	fr.	id.	fr.	id.	fr.	id.	fr.	id.	fr.	id.	fr.	id.
MOT17-train	759	78	380	78	190	77	95	76	47	73	30	71	21	68	15	64
MOT17-test	846	-	423	-	211	-	106	-	53	-	34	-	23	-	17	-
MOT20-train	2233	554	1116	553	558	552	279	551	140	547	89	543	62	539	45	534
MOT20-test	1120	-	560	-	280	-	140	-	70	-	45	-	31	-	22	-

TABLE 5.1. Statistics of the generated videos. ‘fr." and ‘id." represent the number of frames and identities per video, respectively. “-" denotes lacking of data since the testing dataset is hidden by the organizers to avoid misuse.

videos with less than 10 frames, leading to weak test cases. This range may also be extended in the future as the demands change.

Table 5.1 shows the statistics of the simulation datasets. Every single frame from the original dataset is perfectly included in the new dataset. Lower target frame rate setting results in a larger number of videos but a shorter length for each generated video. However, the average number of different identities in each video remains stable.

5.2.1.2 Metrics

Following the CLEAR MOT metrics and HOTA metrics, we define the corresponding mean metrics over multi-frame-rate settings to reflect the overall performance of trackers on videos

with diverse frame rates. Specifically, we use mean-HOTA (mHOTA) and mean-MOTA (mMOTA) as the primary indicators of frame rate agnostic tracking performance, where

$$\mathrm{mHOTA} = \frac{1}{\mid F_{all} \mid} \sum_{F' \in F_{all}} \mathrm{HOTA}_{F'}, \tag{5.7}$$

$$\mathrm{mMOTA} = \frac{1}{\mid F_{all} \mid} \sum_{F' \in F_{all}} \mathrm{MOTA}_{F'}. \tag{5.8}$$

F_{all} is the set of all target frame rates. We also provide mean-IDF1 (mIDF1) in the main results for reference. Similarily, $\mathrm{mIDF1} = \frac{1}{|F_{all}|} \sum_{F' \in F_{all}} \mathrm{IDF1}_{F'}$.

To further investigate the robustness of the trackers against multiple frame rates, we further propose a new metric named ‘Vulnerable Ratio" (VR):

$$\mathrm{VR} = \frac{\mathrm{HOTA}_{\mathrm{highest}} - \mathrm{HOTA}_{\mathrm{lowest}}}{\mathrm{HOTA}_{\mathrm{highest}}}, \tag{5.9}$$

where $\mathrm{HOTA}_{\mathrm{highest}}$ and $\mathrm{HOTA}_{\mathrm{lowest}}$ denote the highest HOTA and lowest HOTA among all different frame rate settings. VR indicates the largest possible drop in extreme cases and will be helpful in reliability and validity evaluation.

5.2.2 Implementation Details

Joint extractor. We adopt YOLOX (Ge et al. 2021) detection framework with extra identity branch as the joint extractor model. The backbone type is yolox-x.

Training data. The joint extractor is first pre-trained on the COCO dataset without the identity branch and then trained on joint MOT17/20 (Milan et al. 2016; Dendorfer et al. 2021; Dendorfer et al. 2020), CrowdHuman (Shao et al. 2018), CityScapes (Cordts et al. 2016) and HIE (Lin et al. 2020) datasets. For MOT datasets, we split the datasets into two different parts without overlapping with each other. The first part is for training, occupying about 60% of the original datasets, and the second part is for evaluation and ablation studies. The training part is further divided into two sets, i.e., Set-A and Set-B, which share some frames but also have their own independent data. Set-A and other non-MOT datasets are for the joint extractor

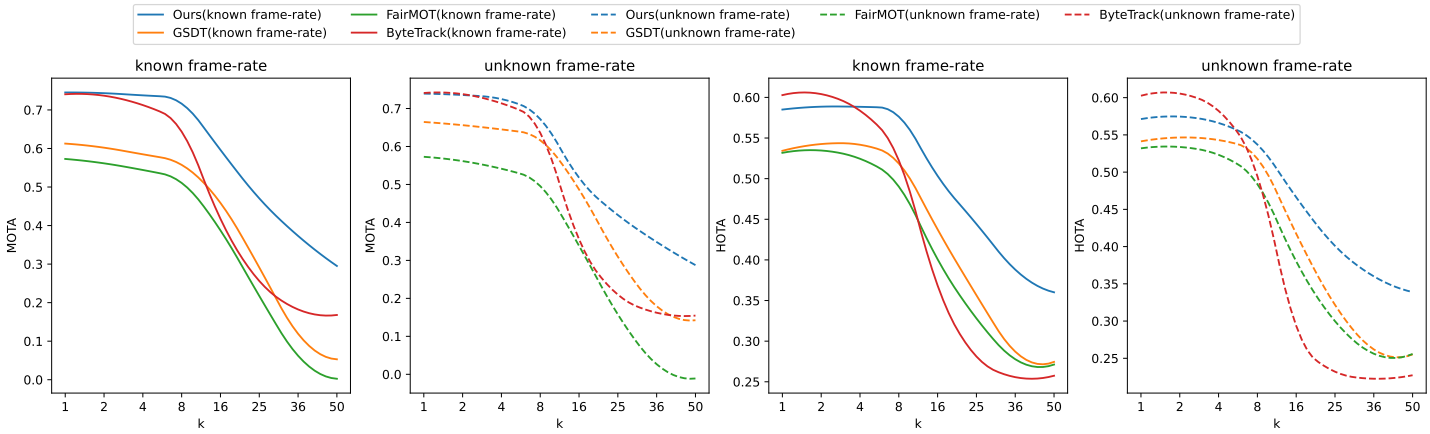


FIGURE 5.5. MOTA and HOTA performance curves of recent state-of-the-art methods and our method on the MOT20 testing set (FraMOT versions) with both known frame rate and unknown frame rate.

training, and Set-B is for joint extractor fine-tuning (smaller learning rate) and association module training. Set-B samples 300 images from each video, with 200 images shared with Set-A. The design of Set-A and Set-B can help avoid the association module from overfitting to the seen data.

Data augmentation. We follow the same data augmentation techniques as YOLOX on non-tracking data (identity branch and association module will not be optimized on these data). On tracking data, we only apply random flip and random resize and remove other augmentations in YOLOX.

Optimization. Stochastic Gradient Descent (SGD) is used as our optimizer. We first train the joint extractor without association module on Set-A and other non-MOT datasets for 60 epochs. In this stage, we apply a cosine learning rate strategy with an initial learning rate of 0.00001. The batch size is set to 32. Then, we fine-tune the joint extractor and train the association module following the proposed Periodic Training Scheme (PTS) on Set-B for 10 more epochs each period. The period number N_p is set to 3.

Other hyper-parameters. In the training stage, loss weights α_2 is set to 0.5 and β_2 is set to 1. During online tracking, λ_{high} is set to 0.6 and λ_{low} is set to 0.1. The drop interval λ_i is set to 30. A matching pair is formed only if the association score is greater than 0.1.

Dataset	Method	known frame rate				unknown frame rate			
		mHOTA↑	mMOTA↑	mIDF1↑	VR↓	mHOTA↑	mMOTA↑	mIDF1↑	VR↓
MOT17	ByteTrack(Zhang et al. 2022)	52.5	64.4	61.3	38.3	51.0	62.9	59.3	43.9
	CenterTrack(Zhou et al. 2020)	-	-	-	-	47.0	60.5	54.8	24.6
	FairMOT(Zhang et al. 2021)	51.0	60.1	56.9	32.3	49.7	58.0	56.0	35.5
	GSDT(Wang et al. 2021c)	48.6	52.3	56.8	32.1	46.8	50.2	54.7	37.1
	TraDeS(Wu et al. 2021)	-	-	-	-	38.6	58.3	44.4	49.0
	Ours	**57.0**	**70.0**	**67.3**	**14.6**	**53.8**	**65.6**	**62.1**	**22.4**
MOT20	ByteTrack(Zhang et al. 2022)	43.5	48.6	49.9	58.1	40.7	46.6	46.9	63.3
	FairMOT(Zhang et al. 2021)	42.2	36.6	48.4	49.9	41.0	34.2	47.1	53.1
	GSDT(Wang et al. 2021c)	44.0	41.9	51.7	50.1	42.9	46.9	51.9	54.0
	Ours	**50.8**	**59.3**	**58.3**	**38.8**	**48.0**	**56.1**	**53.5**	**41.0**

TABLE 5.2. Overall results of recent state-of-the-art MOT methods and our method on MOT17 and MOT20, FraMOT versions. ↑ means higer is better, ↓ means lower is better. **Bold** for best results. (mHOTA, mMOTA, and mIDF1 are presented in the form of percentages.) Our method achieves the best performance under FraMOT settings both with known frame rate mode and unknown frame rate mode.

Setting	Param	FAAM	PTS	known frame rate			unknown frame rate		
				mHOTA↑	mMOTA↑	VR↓	mHOTA↑	mMOTA↑	VR↓
Baseline (UM)	1x			-	-	-	56.5	74.6	34.6
Baseline (MM)	3x			58.9	77.0	29.5	-	-	-
UM + PTS	1x		✓	-	-	-	58.6	75.6	26.7
FAAM	1x	✓		59.3	77.4	28.5	59.1	75.8	29.1
FAAM + PTS	1x	✓	✓	61.1	77.6	24.8	61.0	77.3	24.9

TABLE 5.3. Ablation study about the effectiveness of the proposed approaches.

5.2.3 Results

Table 5.2 shows the overall results of the recent open-source state-of-the-art MOT methods and our method on the challenging MOT17 and MOT20 datasets, with the aforementioned multiple frame rate simulation. ByteTrack (Zhang et al. 2022), FairMOT (Zhang et al. 2021), and GSDT (Wang et al. 2021c) have design strategies using the provided frame rate to control cache length. CenterTrack and TraDeS do not design any frame-rate-relevant procedures so their known frame rate mode results are absent. Our method outperforms all other methods with all metrics in terms of frame rate agnostic tracking. In the known frame rate mode, we are 4.5% and 6.8% higher than the runner-up methods in MOT17 and MOT20 for mHOTA, respectively. For mMOTA, we are 5.6% and 10.7% higher. For mIDF1, we are 6.0% and 6.6%

higher. Besides, we have the least Vulnerable Rate (VR), which means our performance is the least influenced by the change of frame rate. In unknown frame rate mode, our method still outperforms other methods. The improvement in this mode is smaller than those of the known frame rate mode due to the difficulty of obtaining accurate frame rate information. We have 2.8% and 5.1% improvement in the term of mHOTA for MOT17 and MOT20, respectively. We also have higher mHOTA and mIDF1 scores in this mode, and the Vulnerable Rate is also lower than other methods. The results show that our proposed method is effective for frame rate agnostic tracking.

Fig. 5.5 further shows the HOTA and MOTA curves of these methods w.r.t the sampling factor k (larger k means lower frame rate) on MOT20 FraMOT simulation datasets. Among these methods, ByteTrack is the first method making use of the YOLO-X detection baseline, and thus has more accurate detection results and outperforms all other trackers in normal frame rate setting ($k = 1$). To compare with ByteTrack fairly, we also develop our method based on YOLO-X baseline. However, ByteTrack does not use any appearance features for tracking, while our method has an extra tracking branch for exploiting appearance features. As can be observed in Fig. 5.5, our method is slightly lower than ByteTrack at $k \leq 4$ in terms of HOTA, which confirms that appearance features do not help improve and may harm the tracking performance with high frame rates. However, when the frame rates are low (e.g., with larger k), the performance of ByteTrack drops significantly due to less reliable motion. At the same time, the performance of our method also drops slowly but is much higher than all compared methods. It shows that the tracking becomes more difficult when the frame rate is lowered, but our method has better capability to handle these complicated scenarios. Nevertheless, the overall performance of our method is the best, which reveals our design of FAAM and PTS training is effective for FraMOT.

5.2.4 Ablation Study and Analysis

To better understand the proposed methods, we further conduct quantitative experiments on the MOT20 validation set (which is split from the original training set and is not included in the real training set).

5.2.4.1 Effects of the Proposed Methods

Table 5.3 shows how different components affect the overall performance. In this experiment, we first present two straightforward baselines for the FraMOT problem, i.e., Unified Model (UM) and Multiple Model (MM). Baseline (UM) is a simple model which is directly trained from sampled frame pairs of all different frame rates without any extra strategy proposed in this work. In contrast, Baseline (MM) trains 3 separate models for 3 different ranges of sampling factor k, i.e., high-frame-rate ($k < 6$), middle-frame-rate ($6 \leq k < 20$) and low-frame-rate ($20 \leq k$), and deploys each frame rate specific model independently for the corresponding frame rate during testing. The first baseline (UM) does not need a frame rate number during inference and thus is presented in the unknown frame rate mode. The second baseline (MM) requires a frame rate number for model switching, so its results are presented in known frame rate mode. All methods in this ablation study use the same detector framework and the same backbone checkpoints. The mean-HOTA, mean-MOTA, and Vulnerable Ratio of the UM baseline are the worst among all settings in the table, indicating traditional unified association models are not capable of performing frame rate agnostic tracking. The MM baseline is better than the unified model baseline due to the prior knowledge of frame rate division. However, the results are still not satisfying and the method is not feasible when the frame rate number is not given, which is not comparable with the way that humans track objects. Our proposed methods are presented by 'UM + PTS', 'FAAM', and 'FAAM + PTS'. 'UM + PTS' means applying the Periodic Training Scheme on the unified model baseline. With the help of the PTS strategy, the unified model baseline improves by 2.1% mHOTA and 1.0% mMOTA, as well as reduces the Vulnerable Ratio by 8.9%, showing that the PTS training successfully reduces the difficulty of frame rate agnostic tracking training. 'FAAM' means only applying the Frame Rate Agnostic Association Module, which can both work at known frame rate mode and unknown frame rate mode. This unified approach performs better than both baselines while using only 1/3 of the parameters compared with the MM baseline. More importantly, it can automatically infer the frame rate information. When we apply both the two proposed methods, presented as 'FAAM + PTS', we obtain the best results of 61.1% mHOTA and 77.6% mMOTA, with the lowest Vulnerable Ratio of 24.9% (known frame rate

	unseen frame rate		
Setting	mHOTA↑	mMOTA↑	VR↓
Baseline (UM)	54.9	72.1	36.7
Baseline (MM)	57.0	74.4	33.5
FAAM + PTS	59.6	75.0	29.6

TABLE 5.4. Performance on unseen frame rates.

mode). The results in the known frame rate mode are also improved and are quite competitive compared with the known frame rate mode. The results above show that both two approaches are effective, improve the overall frame rate agnostic tracking accuracy and make the tracker wiser.

5.2.4.2 Results on Unseen Frame Rates

We have selected 8 different sampling factors k in the training and testing dataset simulation. To better understand the behaviors of the proposed method about handling various frame rates, we conducted experiments on unseen frame rates, i.e., testing on frame rates that are different from the training set. Specifically, we select $k = 1, 3, 6, 12, 21, 30, 43, 62$ and rebuild the validation set based on these unseen k. We still include the normal frame rate, i.e., $k = 1$, in the unseen set because it is a baseline for analysis. Table 5.4 shows the results of two baselines and our FAAM equipped with PTS training. All models drop slightly when tested on unseen frame rates. However, our FAAM still outperforms the two baselines. Note that the FAAM result here is from unknown-frame-rate mode, which does not need extra frame rate information, while the MM baseline must be provided with a frame rate number.

5.2.4.3 Influence of Hyper-Settings

Table 5.5 shows how the hyper-settings influence the overall performance.

Period number N_p. In PTS training, we introduce the period number N_p. To understand its influence, we conduct experiments with different N_p from 0 to 4. 0 means PTS is not applied. We can see that when $N_p = 3$, we obtain the best performance, while the other N_p settings are also obtaining better performance than the not applied case. The results indicate that a

Group	N_p	threshold	criterion	mHOTA↑	mMOTA↑	VR↓	HOTA@k↑		
							$k=1$	$k=16$	$k=50$
Different N_p	0	0.9	dist.	59.1	75.8	29.1	65.5	60.0	46.4
	1	0.9	dist.	60.6	**77.4**	26.9	**65.6**	62.3	48.3
	2	0.9	dist.	60.8	77.2	25.8	65.2	62.6	48.8
	3	0.9	dist.	**61.0**	77.3	**24.9**	65.3	**62.7**	**49.4**
	4	0.9	dist.	60.7	77.2	27.2	65.3	**62.7**	47.9
Different thresholds	3	0.5	dist.	60.8	**77.4**	25.2	65.2	62.6	49.1
	3	0.7	dist.	60.9	77.3	25.0	65.3	62.6	49.3
	3	0.9	dist.	61.0	77.3	24.9	65.3	62.7	49.4
	3	1.1	dist.	61.0	77.2	24.8	65.3	62.7	49.5
	3	1.3	dist.	**61.1**	77.1	**24.7**	**65.4**	**62.8**	**49.6**
Different criteria	3	0.9	random	58.7	75.9	26.6	63.1	61.0	46.7
	3	0.9	sim.	60.8	**77.5**	26.8	**65.7**	**62.7**	48.3
	3	0.9	dist.	**61.0**	77.3	**24.9**	65.3	**62.7**	**49.4**

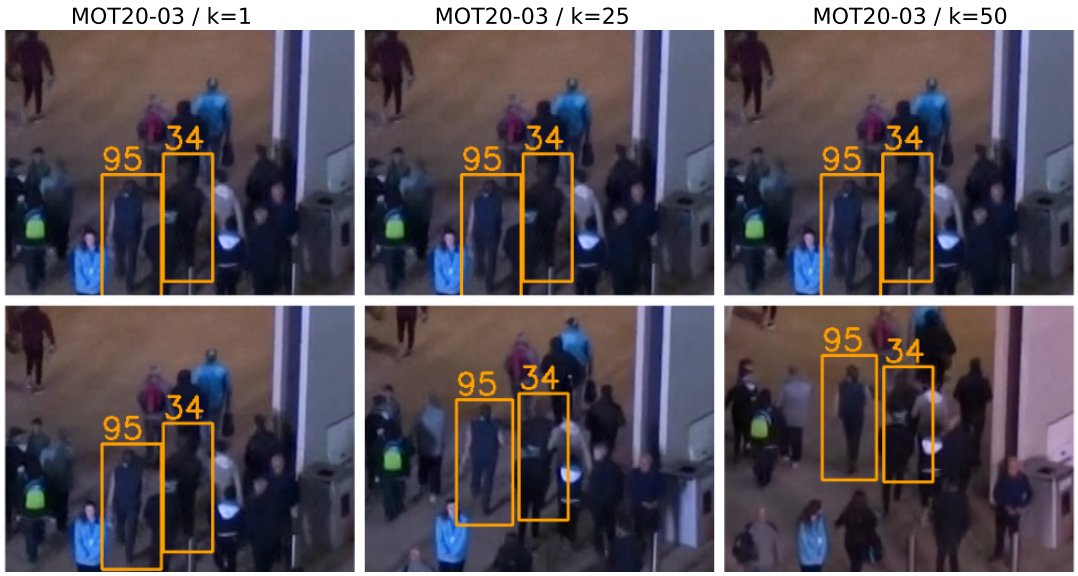
TABLE 5.5. Ablation studies about different hyper-parameters including period number N_p, matching threshold, and IBDV matching criteria.

large N_p is not bringing more gain. It also suggests different checkpoints at later periods do not change the inference environment a lot, the critical change is at the first period.

Matching threshold. Thresholding is sometimes affecting the results a lot. To ensure the matching threshold is stable and the gain is not from the thresholding strategy, we conduct experiments with different thresholds ranging from 0.5 to 1.3. We can see that our method obtains the best mHOTA at the threshold of 1.3, and obtains the best mMOTA at the threshold of 0.5. Although the thresholds vary, the performances are still stable with less than 0.3% difference. We choose 0.9 as the final threshold because we can obtain a balance between mHOTA and mMOTA at the threshold of 0.9.

IBDV criteria. In Section 5.1.2.3 we introduce Inter-frame Best-matched Distance Vector (IBDV) as the feature of frame rate embedding. We further design more different matching rules to understand how the rules affect the overall results. We present three different matching rules in the table. 'random' means we randomly sample pairs as matched pairs. 'sim.' means we choose the pairs with the highest appearance similarity as the matched pairs. 'dist.' means we choose the pairs with closest positional distance as the matched pairs. The results show that both 'sim.' and 'dist.' rules perform better than the 'random' rule, while the 'dist.' rule is

(A) Sequence MOT17-13 with sampling factor $k = 1, 25, 50$. ID 133 is emphasized.



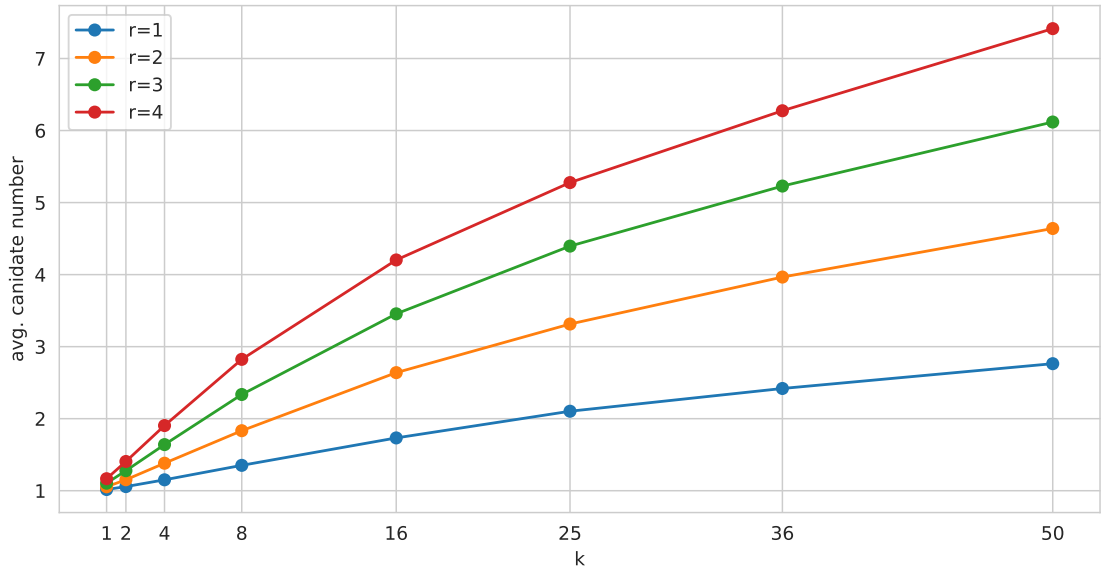
(B) Sequence MOT20-03 with sampling factor $k = 1, 25, 50$. ID 34 and 95 are emphasized.

FIGURE 5.6. Visualization of the multi-frame-rate simulation videos. The first row and the second row are adjacent frames. When the sampling factor k is increased, the task becomes more challenging.

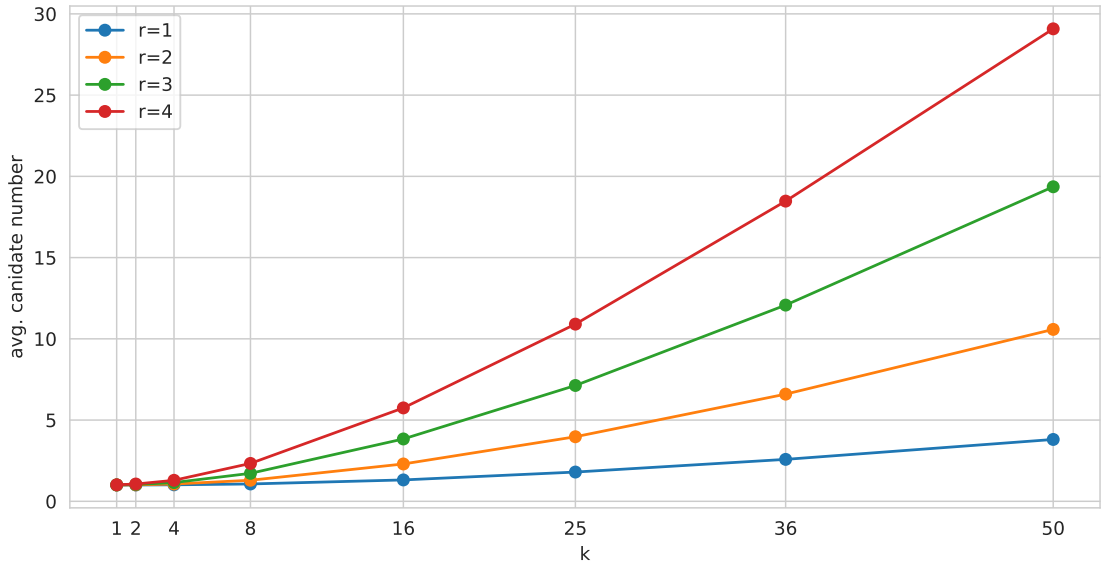
slightly better than the 'sim.' rule. It might be because positional information is more reliable in general. It also reminds us that more complicated rules might be more effective.

5.2.4.4 Visualization and Analysis

In this section, we demonstrate the data simulation and our proposed method via visualization and analysis.



(A) MOT17 training set.



(B) MOT20 training set.

FIGURE 5.7. Curves of candidate numbers w.r.t sampling factor k and thresholding factor r in MOT17 and MOT20 training set.

Visualization of FraMOT simulation data. Fig. 5.6 shows some selected simulation image data from sequence MOT17-13 and sequence MOT20-03. In MOT17-13, the concerned target (ID 133) has a small movement between adjacent frames when $k = 1$, while the movement becomes much larger at $k = 50$. In MOT20-03, the movements (ID 34 and 95) are moderated, but the numbers of matching candidates are still increased a lot in this crowded scenario.

Fig. 5.7 illustrates curves of candidate numbers during matching with respect to sampling factor k and thresholding factor r, i.e., how many possible candidates that are having an equal or closer distance to the concerned object, compared with the r times distance to the correct ground-truth object. In other words, the curves show the average numbers of objects we must compare with during matching, under different k and different thresholding strategies. Usually, r is set to a number larger than 1 in order to make the correct object with the same identity number being compared for sure, or we are not able to recall the previously tracked object. If the candidate number is large, that means we have more chances to make mistakes and thus the task is more challenging. From the two figures we can see that when k is increased, the candidate numbers increase significantly, especially in MOT20 dataset. In fact, in normal frame rate scenarios, the candidate number is slightly larger than 1, which means in most cases, finding the object with the closest distance will hit the ground-truth. It also well explained why introducing Re-ID branch in YOLO-X baseline did not improve the results at the normal frame rate, it is because positional information is so reliable to perform matching. Fig. 5.7 shows that the FraMOT task is much more difficult due to various candidate numbers.

Visualization of the PTS data. Fig. 5.8 visualizes the affinity features used in this paper. We both show PTS and non-PTS affinity features at sampling rates of $k = 1$ and $k = 50$. The four sub-figures show that after applying PTS training, the boundary between the same and different identity pairs is clearer, which means the task is easier. At both sampling rates, there are more difficult data points in non-PTS counterparts. We can further find that at $k = 50$, the gap between PTS and non-PTS is enlarged compared with $k = 1$, which indicates PTS is more effective at larger sampling rates.

Visualization of the attention embedding of Frame Rate Agnostic Association Module. Fig. 5.9 illustrates the attention maps learned by the proposed Frame Rate Agnostic Association Module (FAAM) at different sampling rates (different k-s). The channels of the attention map at $k = 1$ are sorted in increasing order. Other attention maps are aligned with the $k = 1$ attention map. When k is small, e.g., $k < 16$, the right part of the affinity features are emphasized. Then, the right part becomes less important when k goes large. Meanwhile, some critical points on the left part are focused on. These observations reflect that the module

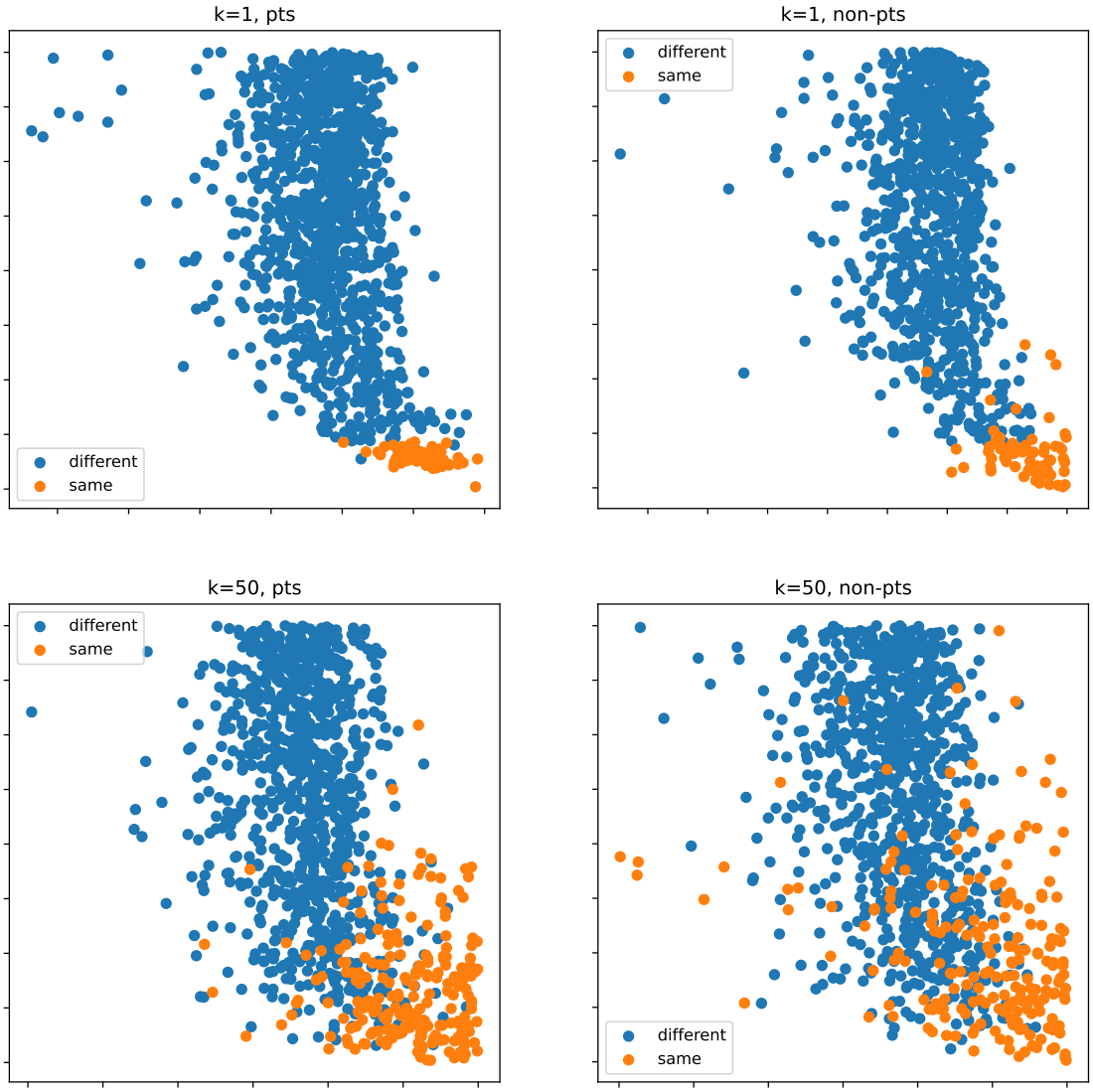


FIGURE 5.8. Visualization of affinity features w.r.t positional cues and appearance cues. Applying the PTS strategy leads to a clearer boundary between the same and different identity pairs.

tends to focus on different dimensions of the affinity features at different sampling rates. We hypothesize that most of the right part is related to motion features, which well explains motion becomes less reliable when the frame rate is lowered. More interestingly, when the frame rate is slightly lower (e.g., $k = 8$), the attention map focuses on more different channels, which indicates motion cues are still reliable but it is probably not reliable using the single feature of motion (e.g., IoU).

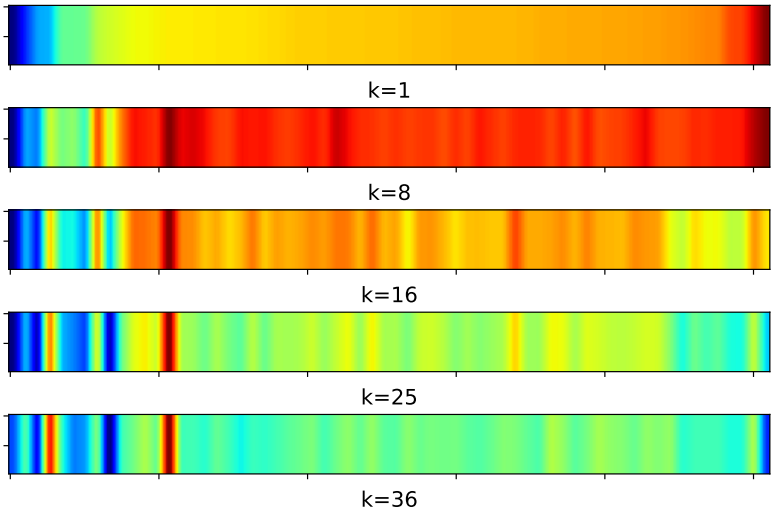


FIGURE 5.9. Attention map of the Frame Rate Agnostic Association Module (FAAM).

CHAPTER 6

Unsupervised Pretraining with Foreground Trajectory Detection

6.1 Methodology

In this section, we introduce our study on unsupervised pre-training for joint detection and tracking models. Fig. 6.1 illustrates the overview of the proposed method, which consists of two steps, i.e., video selection and pseudo-label generation. In the video selection stage, videos with large camera motions will be filtered out since it is more difficult to determine foreground trajectories in these videos. In the pseudo-label generation stage, the remaining video frames are used to generate trajectory-level labels using the foreground object discovery method and multi-frame consistency constraint.

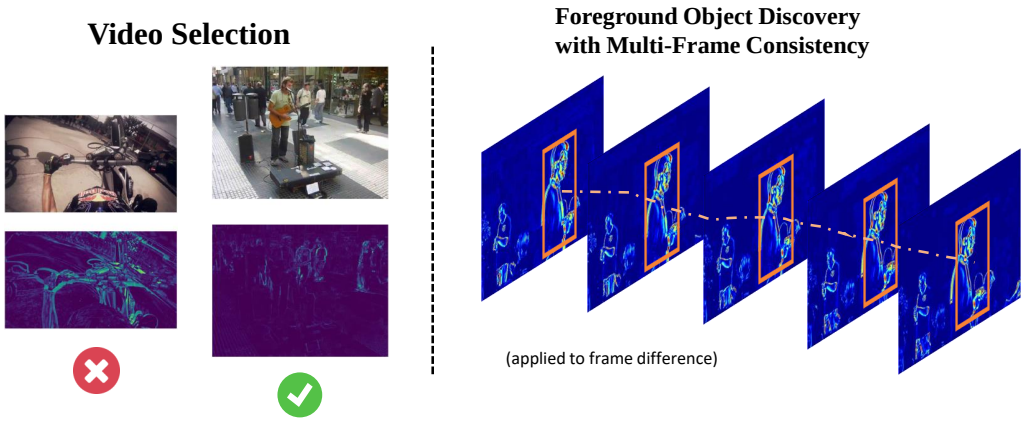


FIGURE 6.1. Overview of the proposed method.

6.1.1 Video Selection

In order to filter out video clips with large camera motions, an indicator of camera motion is needed. We simply use the difference ratio (DR) of adjacent image pairs as the indicator:

$$DR = \frac{\sum |I_t - I_{t-1}|}{size(I_t)} \tag{6.1}$$

where I_t is the t-th frame and $size(I)$ is the tensor size of the image I.

The DR is straightforward because, in static or low-motion scenarios, the backgrounds remain almost the same, and the only differences in the adjacent frames mainly come from the foreground moving objects. Fig. 6.2 demonstrates several examples of different DRs. It shows that with a DR ranging from 0.05 - 0.2, the camera motions are small and there are meaningful foreground objects. When DR is lower than 0.05, it is usually a static title page or blackout transactions. When DR is higher than 0.3, it is usually with large camera motions or fast-moving objects.

6.1.2 Pseudo-label Generation

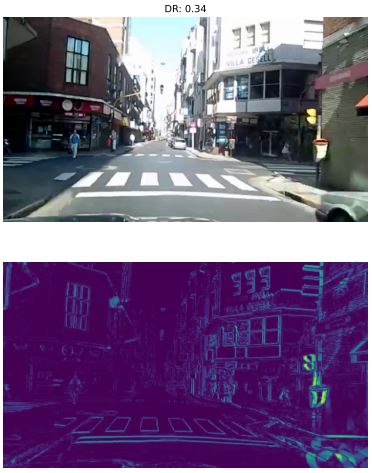
Foreground object discovery. We then apply the foreground object discovery method to the adjacent frame differences of the selective video frames. We choose Selective Search(Uijlings et al. 2013) as the foreground discovery method in our study, which other discovery methods could replace. For frame I_t, the foreground proposal boxes are denoted by $B_t = \{b_t^i, i = 1, 2, \ldots, M_t\}$.

Multi-frame consistency. To remove false positives, we only keep the foreground proposal boxes which also appear in nearby frames, i.e., having another k_u proposals in nearby frames with Intersection over Union (IoU) larger than a threshold α_3. Formally, define $c(b_t^i)$ as the number of nearby consistent appearances of b_t^i, then

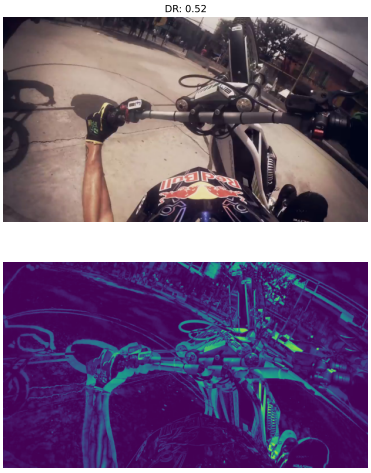
$$c(b_t^i) = \sum_{\hat{t}=t-\tau_r}^{t+\tau_r} [\max_j (IoU(b_{\hat{t}}^j, b_t^i)) > \alpha_3], \tag{6.2}$$

(A) DR=0.02

(B) DR=0.18



(C) DR=0.34



(D) DR=0.52

(E) DR=0.70

FIGURE 6.2. Video examples with DRs ranging from 0.00 to 0.70. When DR is lower than 0.05, it is almost static, e.g., a static title page in (A). When DR is around 0.15, the scenarios are more meaningful, e.g., (B). With larger DRs, the videos are probably with larger camera motions, e.g., (C) and (D), or with fast-moving objects, e.g., (E).

and the confident boxes that will be kept are $\hat{B}_t = \{b|b \in B_t, c(b) > k_u\}$.

These proposals naturally form a tracklet. Since then, we have obtained trajectory-level pseudo labels. To make the trajectory smooth, we further modify these boxes using the moving average.

6.1.3 Pre-training and Finetuning

Once we obtained the pseudo-labels, model pre-training can be performed. Since the pseudo-labels did not include any hints of the categories of interest, a finetuning procedure is required to produce the final model. The finetuning was done on a small dataset and with only a few epochs.

6.2 Experiments

6.2.1 Implementation Details

Joint Framework. In this study, we used ByteTrack (Zhang et al. 2022), one of the most recent advanced joint detection and tracking frameworks, as our experimental baseline. We follow the full pipeline of training and tracking as Yolo-X (Ge et al. 2021) and ByteTrack. For efficiency, we choose yolox-l as the backbone architecture.

Pseudo-label Generation. In video selection, we choose videos with DR ranging from 0.05 to 0.2. We search ± 2 frames ($\tau_r = 2$) nearby and set the IoU threshold $\alpha_3 = 0.4$, consistency threshold $k_u = 3$.

Datasets. We used the unlabeled videos from LUPerson(Fu et al. 2021) as the search library. Finally, we filtered out 16781 video clips with 60 frames in each as our unsupervised training set. For fine-tuning, we combined the MOT20(Dendorfer et al. 2020) and SOMPT22(Simsek et al. 2022) datasets into one larger dataset, named MIX20-22. We split this combined dataset into three parts, i.e., the training set, the validation set, and the testing set. For the training set, we choose MOT20-01 and MOT20-02 from the MOT20 training set, and SOMPT22-02/04/05/07/08/10 from the SOMPT22 training set. For the validation set, we

Pre-train	MOTA↑	IDF1↑	Precision↑	Recall↑	mAP@0.5↑
Random	37.7	39.9	75.6	56.4	43.4
Supervised COCO	51.7	55.5	77.6	73.2	46.6
Unsupervised COCO	42.7	47.2	75.4	64.2	45.6
LUPerson-NL(Fu et al. 2022)	46.9	47.4	81.4	61.3	46.3
SwAV(Caron et al. 2020)	52.0	52.3	77.4	74.2	46.6
non-video LUP	49.4	49.1	78.6	68.4	46.0
video-based LUP (ours)	49.8	49.1	81.0	65.5	46.4
SwAV + video-based LUP (ours+)	54.5	53.9	79.6	73.8	46.9

TABLE 6.1. Testing results on MIX20-22 dataset, ↑ indicates higher is better. All metrics are in percentage.

choose MOT20-03 from the MOT20 training set. For the testing set, we choose MOT20-05, and SOMPT22-11/12/13. All the videos in the testing set of MIX20-22 have different scenarios that are different from those in the fine-tuning training set, which could fairly show the performance of the trained model in unseen scenarios.

Training. We first trained the model using the selective LUP videos for 20 epochs, with a learning rate of 0.001 and a batch size of 48. Then we fine-tune the model on the MIX20-22 training set for 60 epochs, with a learning rate of 0.005 and a batch size of 24. Both procedures used SGD as the optimizer, with a cosine learning rate schedule, and the warm-up epoch number was set to 1.

Evaluation Metrics. We followed the most popular evaluation metrics for MOT testing, i.e., the CLEAR MOT metrics (Bernardin and Stiefelhagen 2008) and the ID metrics (Ristani et al. 2016). Among these metrics, MOTA and IDF1 are the main indicators. MOTA mainly reflects the detection performance and the IDF1 mainly reflects the trajectory integrity. We also provide COCO-style mean average precision with IoU threshold=0.5 (mAP@0.5) for reference.

6.2.2 Results

Table 6.1 shows the testing results of several baselines and our method on the MIX20-22 testing set. Unsupervised COCO only used pseudo-labels from Selective Search results on the COCO dataset. LUPersion-NL used model-predicted detection boxes of the LUP dataset

and used simple MOT algorithms to generate trajectories, and used contrastive learning for model pre-training (ResNet-50 backbone). The non-video model used only Selective Search without multi-frame consistency on our selective dataset. Comparing non-video LUP with Unsupervised COCO and LUPerson-NL shows that the video selection leads to video data of higher quality. Comparing video-based LUP and non-video LUP, the multi-frame consistency improved the MOTA by 0.4% and mAP by 0.4%. When combing SwAV and our method, i.e., the SwAV + video-based LUP, the model performance reached 54.5% MOTA, 53.9% IDF1, and 46.9% mAP, which is better than SwAV (yolox-l backbone), and the supervised COCO pre-training in terms of MOTA and mAP. At the same time, the model achieved the best IDF1 score among all unsupervised baselines. The results show that our method is effective.

6.2.3 Pseudo-Label Quality Analysis

The proposed method's ability to generate high-quality trajectories is demonstrated in Fig. 6.3, which displays a set of pseudo trajectories. These trajectories exhibit the capacity to track various moving objects with diverse positions and shapes (identified as 105, 197, and 298). However, it should be noted that certain trajectories may experience drift (identified as 201), and in some instances, background objects are detected as moving objects. Despite these limitations, the overall performance of the proposed method is promising, suggesting its potential for applications in diverse fields.

FIGURE 6.3. Illustration of some generated pseudo trajectories.

CHAPTER 7

Conclusion and Future Outlook

In conclusion, this thesis presented four different approaches in an attempt to explore the next-generation joint detection and tracking (JDT) system. The first approach proposed a switcher-aware MOT framework that incorporates multiple cues and potential switchers, and the effectiveness of this approach was demonstrated through comprehensive experiments. The second approach introduced a new unified relation learning method for joint detection and tracking using Similarity- and Quality-Guided Attention (SQGA) and showed promising results on the MOT benchmark. The third approach addressed the challenge of frame rate agnostic MOT and proposed a Frame Rate Agnostic MOT Framework with a Periodic training Scheme (FAPS) to improve tracking stability against various frame rates. The fourth approach introduced a video-based pseudo-label generation method called Foreground Trajectory Detection (FTD) to pre-train any joint detection and tracking framework.

The potential avenues for future research stemming from this thesis are both diverse and promising.

The exploration of data-related aspects includes delving into progressive learning techniques applied to unannotated video data. Additionally, the investigation of methodologies for learning from synthetic video data poses intriguing questions, such as the equivalence between generative AI and perceptive AI.

In terms of model development, an essential direction for future research involves addressing the challenges posed by distractors and identity-switch in the era of transformers. Further research efforts should be directed towards devising effective strategies to handle these issues and enhance the robustness of models.

On the learning front, the exploration of a supervised attention mechanism in specific tasks presents an intriguing area for future study. Investigating the circumstances where supervised attention surpasses implicit attention will contribute valuable insights into the nuanced applications of these attention mechanisms. Moreover, a critical question that warrants exploration is the generalizability of the SQGA (Similarity- and Quality-Guided Attention) mechanism to other joint-task model learning scenarios.

Regarding scenarios, the quest for novel designs for frame-rate agnostic Multiple Object Tracking (MOT) systems is essential. Future research should also consider the complexities introduced by multi-view scenarios, exploring the development of frame-rate agnostic multi-view MOT systems.

These outlined future research directions underscore the potential for further advancements in the fields of data processing, model development, learning mechanisms, and scenario-based applications, contributing to the continual evolution of joint detection and tracking systems.

Bibliography

Babaee, Maryam, Zimu Li and Gerhard Rigoll (2019). 'A dual CNN–RNN for multiple people tracking'. In: *Neurocomputing* 368, pp. 69–83.

Bae, Seung-Hwan and Kuk-Jin Yoon (2014). 'Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning'. In: *CVPR*.

— (2018). 'Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking'. In: *TPAMI* 40.3, pp. 595–610.

Belongie, Serge, Jitendra Malik and Jan Puzicha (2000). 'Shape context: A new descriptor for shape matching and object recognition'. In: *Advances in neural information processing systems* 13.

Bergmann, Philipp, Tim Meinhardt and Laura Leal-Taixe (Oct. 2019). 'Tracking Without Bells and Whistles'. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Bernardin, Keni and Rainer Stiefelhagen (2008). 'Evaluating multiple object tracking performance: the CLEAR MOT metrics'. In: *Journal on Image and Video Processing* 2008, p. 1.

Bertasius, Gedas, Lorenzo Torresani and Jianbo Shi (2018). 'Object detection in video with spatiotemporal sampling networks'. In: *ECCV*.

Bertinetto, Luca et al. (2016). 'Fully-convolutional siamese networks for object tracking'. In: *ECCV*.

Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao (2020). 'Yolov4: Optimal speed and accuracy of object detection'. In: *arXiv preprint arXiv:2004.10934*.

Bolme, David S et al. (2010). 'Visual object tracking using adaptive correlation filters'. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, pp. 2544–2550.

Brasó, Guillem and Laura Leal-Taixé (2020). 'Learning a neural solver for multiple object tracking'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6247–6257.

Calonder, Michael et al. (2010). 'Brief: Binary robust independent elementary features'. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, pp. 778–792.

Carion, Nicolas et al. (2020). 'End-to-end object detection with transformers'. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, pp. 213–229.

Caron, Mathilde et al. (2020). 'Unsupervised learning of visual features by contrasting cluster assignments'. In: *Advances in neural information processing systems* 33, pp. 9912–9924.

Chen, Kai et al. (2019). 'Mmdetection: Open mmlab detection toolbox and benchmark'. In: *arXiv preprint arXiv:1906.07155*.

Chen, Ting et al. (2020a). 'A simple framework for contrastive learning of visual representations'. In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Chen, Weihua et al. (2017a). 'An Equalized Global Graph Model-Based Approach for Multicamera Object Tracking'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.11, pp. 2367–2381. DOI: 10.1109/TCSVT.2016.2589619.

Chen, Weihua et al. (2017b). 'Beyond triplet loss: a deep quadruplet network for person re-identification'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412.

Chen, Yihong et al. (2020b). 'Memory Enhanced Global-Local Aggregation for Video Object Detection'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10337–10346.

Cho, Kyunghyun et al. (2014). 'On the properties of neural machine translation: Encoder-decoder approaches'. In: *arXiv preprint arXiv:1409.1259*.

Choi, Wongun (2015). 'Near-online multi-target tracking with aggregated local flow descriptor'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 3029–3037.

Chu, Peng and Haibin Ling (Oct. 2019). 'FAMNet: Joint Learning of Feature, Affinity and Multi-Dimensional Assignment for Online Multiple Object Tracking'. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Chu, Qi et al. (2017). 'Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism'. In: *ICCV*.

Cordts, Marius et al. (2016). 'The Cityscapes Dataset for Semantic Urban Scene Understanding'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, Zhigang et al. (2021). 'Up-detr: Unsupervised pre-training for object detection with transformers'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1601–1610.

Dalal, Navneet and Bill Triggs (2005). 'Histograms of oriented gradients for human detection'. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee, pp. 886–893.

Dendorfer, Patrick et al. (2020). *MOT20: A benchmark for multi object tracking in crowded scenes*. arXiv: 2003.09003 [cs.CV].

Dendorfer, Patrick et al. (2021). 'Motchallenge: A benchmark for single-camera multiple target tracking'. In: *International Journal of Computer Vision* 129.4, pp. 845–881.

Deng, Jiajun et al. (2019). 'Relation distillation networks for video object detection'. In: *ICCV*.

Dosovitskiy, Alexey et al. (2020). 'An image is worth 16x16 words: Transformers for image recognition at scale'. In: *arXiv preprint arXiv:2010.11929*.

Duan, Kaiwen et al. (2019). 'Centernet: Keypoint triplets for object detection'. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578.

Fang, Jianwu et al. (2022). 'Traffic Accident Detection via Self-Supervised Consistency Learning in Driving Scenarios'. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7, pp. 9601–9614. DOI: 10.1109/TITS.2022.3157254.

Fang, Kuan et al. (2018). 'Recurrent autoregressive networks for online multi-object tracking'. In: *WACV*.

Feichtenhofer, Christoph, Axel Pinz and Andrew Zisserman (2017). 'Detect to track and track to detect'. In: *ICCV*.

Feng, Weitao, Baopu Li and Wanli Ouyang (2022a). 'Multi-Object Tracking with Multiple Cues and Switcher-Aware Classification'. In: *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–10. DOI: 10.1109/DICTA56598.2022.10034575.

Feng, Weitao et al. (2022b). *Towards Frame Rate Agnostic Multi-Object Tracking*. DOI: 10.48550/ARXIV.2209.11404. URL: https://arxiv.org/abs/2209.11404.

— (2023). 'Towards frame rate agnostic multi-object tracking'. In: *International Journal of Computer Vision*, pp. 1–20.

Feng, Weitao et al. (2024). 'Similarity- and Quality-Guided Relation Learning for Joint Detection and Tracking'. In: *IEEE Transactions on Multimedia* 26, pp. 1267–1280. DOI: 10.1109/TMM.2023.3279670.

Freeman, William T and Michal Roth (1995). 'Orientation histograms for hand gesture recognition'. In: *International workshop on automatic face and gesture recognition*. Vol. 12. Citeseer, pp. 296–301.

Fu, Dengpan et al. (2021). 'Unsupervised pre-training for person re-identification'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14750–14759.

Fu, Dengpan et al. (2022). 'Large-scale pre-training for person re-identification with noisy labels'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2476–2486.

Fu, Zeyu et al. (2019). 'Multi-Level Cooperative Fusion of GM-PHD Filters for Online Multiple Human Tracking'. In: *IEEE Transactions on Multimedia*.

Ge, Zheng et al. (2021). 'YOLOX: Exceeding YOLO Series in 2021'. In: *CoRR* abs/2107.08430. arXiv: 2107.08430. URL: https://arxiv.org/abs/2107.08430.

Geiger, Andreas et al. (2013). 'Vision meets robotics: The kitti dataset'. In: *The International Journal of Robotics Research* 32.11, pp. 1231–1237.

Girshick, Ross (2015). 'Fast r-cnn'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Girshick, Ross et al. (2014). 'Rich feature hierarchies for accurate object detection and semantic segmentation'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Han, Tao et al. (2022). 'DR. VIC: Decomposition and Reasoning for Video Individual Counting'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3083–3092.

He, Kaiming et al. (2015). 'Spatial pyramid pooling in deep convolutional networks for visual recognition'. In: *IEEE transactions on pattern analysis and machine intelligence* 37.9, pp. 1904–1916.

— (2016). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Kaiming et al. (2020). 'Momentum contrast for unsupervised visual representation learning'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738.

He, Zhen et al. (2019). 'Tracking by animation: Unsupervised learning of multi-object attentive trackers'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1318–1327.

Hearst, Marti A. et al. (1998). 'Support vector machines'. In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.

Henriques, João F et al. (2014). 'High-speed tracking with kernelized correlation filters'. In: *IEEE transactions on pattern analysis and machine intelligence* 37.3, pp. 583–596.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780.

Hornakova, Andrea et al. (2020). 'Lifted disjoint paths with application in multiple object tracking'. In: *International conference on machine learning*. PMLR, pp. 4364–4375.

Huang, Chang, Bo Wu and Ramakant Nevatia (2008). 'Robust object tracking by hierarchical association of detection responses'. In: *ECCV*.

Karthik, Shyamgopal, Ameya Prabhu and Vineet Gandhi (2020). 'Simple unsupervised multi-object tracking'. In: *arXiv preprint arXiv:2006.02609*.

Kieritz, Hilke, Wolfgang Hubner and Michael Arens (2018). 'Joint detection and online multi-object tracking'. In: *CVPRW*.

Kim, Chanho, Fuxin Li and James M Rehg (2018). 'Multi-object Tracking with Neural Gating Using Bilinear LSTM'. In: *ECCV*.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2017). 'Imagenet classification with deep convolutional neural networks'. In: *Communications of the ACM* 60.6, pp. 84–90.

Law, Hei and Jia Deng (2018). 'Cornernet: Detecting objects as paired keypoints'. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750.

LeCun, Yann et al. (1998). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Byungjae et al. (2016). 'Multi-class multi-object tracking using changing point detection'. In: *ECCV*.

Li, Bo et al. (2018a). 'High Performance Visual Tracking With Siamese Region Proposal Network'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980.

Li, Bo et al. (2018b). 'SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks'. In: *arXiv preprint arXiv:1812.11703*.

Li, Wei and Xiaogang Wang (2013). 'Locally Aligned Feature Transforms across Views'. In: *CVPR*.

Li, Wei, Rui Zhao and Xiaogang Wang (2012). 'Human Reidentification with Transferred Metric Learning'. In: *ACCV*.

Li, Wei et al. (2014). 'DeepReID: Deep Filter Pairing Neural Network for Person Re-identification'. In: *CVPR*.

Lin, Tsung-Yi et al. (2014). 'Microsoft coco: Common objects in context'. In: *European conference on computer vision*. Springer, pp. 740–755.

Lin, Tsung-Yi et al. (2017). 'Feature pyramid networks for object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.

Lin, Weiyao et al. (2020). *Human in Events: A Large-Scale Benchmark for Human-centric Video Analysis in Complex Events*. arXiv: 2005.04490 [cs.CV].

Liu, Qiankun et al. (2020). 'GSM: Graph Similarity Model for Multi-Object Tracking.' In: *IJCAI*, pp. 530–536.

Long, C et al. (2018). 'Real-time multiple people tracking with deeply learned candidate selection and person re-identification'. In: ICME.

Lowe, David G (1999). 'Object recognition from local scale-invariant features'. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, pp. 1150–1157.

Luiten, Jonathon et al. (2021). 'Hota: A higher order metric for evaluating multi-object tracking'. In: *International journal of computer vision* 129.2, pp. 548–578.

Mahmoudi, Nima, Seyed Mohammad Ahadi and Mohammad Rahmati (2019). 'Multi-target tracking using CNN-based features: CNNMTT'. In: *Multimedia Tools and Applications* 78.6, pp. 7077–7096.

Milan, A. et al. (Mar. 2016). 'MOT16: A Benchmark for Multi-Object Tracking'. In: *arXiv:1603.00831 [cs]*. arXiv: 1603.00831. URL: http://arxiv.org/abs/1603.00831.

Milan, Anton et al. (2017). 'Online Multi-Target Tracking Using Recurrent Neural Networks.' In: *AAAI*.

Munkres, James (1957). 'Algorithms for the Assignment and Transportation Problems'. In: *Journal of the Society for Industrial & Applied Mathematics* 5.1, pp. 32–38.

Nair, Vinod and Geoffrey E Hinton (2010). 'Rectified linear units improve restricted boltzmann machines'. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Pang, Bo et al. (2020). 'TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model'. In: *CVPR*.

Pang, Jiangmiao et al. (2021). 'Quasi-dense similarity learning for multiple object tracking'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 164–173.

Peng, Jinlong et al. (2020). 'Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking'. In: *ECCV*.

Pirsiavash, Hamed, Deva Ramanan and Charless C Fowlkes (2011). 'Globally-optimal greedy algorithms for tracking a variable number of objects'. In: *CVPR*.

Qiu, Heqian et al. (2020). 'Hierarchical context features embedding for object detection'. In: *IEEE Transactions on Multimedia* 22.12, pp. 3039–3050.

Real, Esteban et al. (2017). 'Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5296–5305.

Redmon, Joseph and Ali Farhadi (2017). 'YOLO9000: better, faster, stronger'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

— (2018). 'Yolov3: An incremental improvement'. In: *arXiv preprint arXiv:1804.02767*.

Redmon, Joseph et al. (2016). 'You only look once: Unified, real-time object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Ren, Shaoqing et al. (2015). 'Faster r-cnn: Towards real-time object detection with region proposal networks'. In: *Advances in neural information processing systems*, pp. 91–99.

Ristani, Ergys et al. (2016). 'Performance measures and a data set for multi-target, multi-camera tracking'. In: *European Conference on Computer Vision*. Springer, pp. 17–35.

Rublee, Ethan et al. (2011). 'ORB: An efficient alternative to SIFT or SURF'. In: *2011 International conference on computer vision*. Ieee, pp. 2564–2571.

Rumelhart, David E, Geoffrey E Hinton and Ronald J Williams (1986). 'Learning representations by back-propagating errors'. In: *nature* 323.6088, pp. 533–536.

Russakovsky, Olga et al. (2015). 'ImageNet Large Scale Visual Recognition Challenge'. In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Sadeghian, Amir, Alexandre Alahi and Silvio Savarese (2017). 'Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies'. In: *ICCV*.

Sanchez-Matilla, Ricardo and Andrea Cavallaro (2019). 'A predictor of moving objects for first-person vision'. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 2189–2193.

Shao, Shuai et al. (2018). 'CrowdHuman: A Benchmark for Detecting Human in a Crowd'. In: *arXiv preprint arXiv:1805.00123*.

Sheng, Hao et al. (2018). 'Heterogeneous association graph fusion for target association in multiple object tracking'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.11, pp. 3269–3280.

Sheng, Hao et al. (2020). 'Hypothesis testing based tracking with spatio-temporal joint interaction modeling'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.9, pp. 2971–2983.

Shvets, Mykhailo, Wei Liu and Alexander C Berg (2019). 'Leveraging long-range temporal relationships between proposals for video object detection'. In: *ICCV*.

Simonyan, Karen and Andrew Zisserman (2014). 'Very deep convolutional networks for large-scale image recognition'. In: *arXiv preprint arXiv:1409.1556*.

Simsek, Fatih Emre, Cevahir Cigla and Koray Kayabol (2022). 'SOMPT22: A Surveillance Oriented Multi-Pedestrian Tracking Dataset'. In: *arXiv preprint arXiv:2208.02580*.

Son, Jeany et al. (2017). 'Multi-object tracking with quadruplet convolutional neural networks'. In: *CVPR*.

Stadler, Daniel and Jurgen Beyerer (2021a). 'Improving multiple pedestrian tracking by track management and occlusion handling'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10958–10967.

— (2021b). 'On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking'. In: *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, pp. 1–12.

Su, Chi et al. (2017). 'Pose-driven deep convolutional model for person re-identification'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 3960–3969.

Sun, ShiJie et al. (2020). 'Simultaneous Detection and Tracking with Motion Modelling for Multiple Object Tracking'. In: *ECCV*.

Sun, ShiJie et al. (2021). 'Deep Affinity Network for Multiple Object Tracking'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1, pp. 104–119. DOI: 10.1109/TPAMI.2019.2929520.

Sun, Yifan et al. (2018). 'Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)'. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 480–496.

Szegedy, Christian et al. (2015). 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tang, Siyu et al. (2015). 'Subgraph decomposition for multi-target tracking'. In: *CVPR*.

Tang, Siyu et al. (2016). 'Multi-person tracking by multicut and deep matching'. In: *ECCV*.

Tang, Siyu et al. (2017). 'Multiple people tracking by lifted multicut and person reidentification'. In: *CVPR*.

Tokmakov, Pavel et al. (2021). 'Learning to track with object permanence'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10860–10869.

Uijlings, Jasper RR et al. (2013). 'Selective search for object recognition'. In: *International journal of computer vision* 104, pp. 154–171.

Vaswani, Ashish et al. (2017). 'Attention is all you need'. In: *Advances in neural information processing systems*, pp. 5998–6008.

Wan, Xingyu et al. (2018). 'Multi-object tracking using online metric learning with long short-term memory'. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 788–792.

Wang, Gaoang et al. (2019). 'Exploit the connectivity: Multi-object tracking with trackletnet'. In: *ACM MM*.

Wang, Gaoang et al. (2022). 'Split and Connect: A Universal Tracklet Booster for Multi-Object Tracking'. In: *IEEE Transactions on Multimedia*, pp. 1–1. DOI: 10.1109/TMM.2022.3140919.

Wang, Qiang et al. (2021a). 'Multiple object tracking with correlation learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3876–3886.

Wang, Shiyao et al. (Sept. 2018a). 'Fully Motion-Aware Network for Video Object Detection'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Wang, Shuai et al. (2021b). 'A General Recurrent Tracking Framework Without Real Data'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13219–13228.

Wang, Xiaolong and Abhinav Gupta (2018). 'Videos as space-time region graphs'. In: *ECCV*.

Wang, Xiaolong et al. (2018b). 'Non-local neural networks'. In: *CVPR*.

Wang, Yizhou et al. (n.d.). 'Unsupervised Object Detection Pretraining with Joint Object Priors Generation and Detector Learning'. In: *Advances in Neural Information Processing Systems*.

Wang, Yongxin, Kris Kitani and Xinshuo Weng (2021c). 'Joint object detection and multi-object tracking with graph neural networks'. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 13708–13715.

Wang, Yongxin, Xinshuo Weng and Kris Kitani (2020). 'Joint Detection and Multi-Object Tracking with Graph Neural Networks'. In: *BMVC*.

Wu, Jialian et al. (2021). 'Track to detect and segment: An online multi-object tracker'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12352–12361.

Xiang, Jun et al. (2020). 'End-to-End Learning Deep CRF Models for Multi-Object Tracking Deep CRF Models'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.1, pp. 275–288.

Xiao, Fanyi and Yong Jae Lee (2018). 'Video object detection with an aligned spatial-temporal memory'. In: *ECCV*.

Xie, Enze et al. (2021). 'Detco: Unsupervised contrastive learning for object detection'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8392–8401.

Xu, Jiarui et al. (Oct. 2019a). 'Spatial-Temporal Relation Networks for Multi-Object Tracking'. In: *The IEEE International Conference on Computer Vision (ICCV)*.

— (2019b). 'Spatial-temporal relation networks for multi-object tracking'. In: *ICCV*.

Yoon, Young-chul et al. (2018). 'Online Multi-Object Tracking with Historical Appearance Matching and Scene Adaptive Detection Filtering'. In: *arXiv preprint arXiv:1805.10916*.

Yu, En et al. (2022). 'RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation'. In: *IEEE Transactions on Multimedia*, pp. 1–1. DOI: 10.1109/TMM.2022.3150169.

Yu, Fengwei et al. (2016). 'Poi: Multiple object tracking with high performance detection and appearance feature'. In: *ECCV*.

Zhang, Li, Yuan Li and Ramakant Nevatia (2008). 'Global data association for multi-object tracking using network flows'. In: *CVPr*.

Zhang, Xiaoqin et al. (2010). 'Multiple object tracking via species-based particle swarm optimization'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 20.11, pp. 1590–1602.

Zhang, Yifu et al. (2021). 'Fairmot: On the fairness of detection and re-identification in multiple object tracking'. In: *International Journal of Computer Vision* 129.11, pp. 3069–3087.

Zhang, Yifu et al. (2022). 'ByteTrack: Multi-Object Tracking by Associating Every Detection Box'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhao, Liming et al. (2017). 'Deeply-learned part-aligned representations for person re-identification'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 3219–3228.

Zheng, Liang et al. (2015). 'Scalable Person Re-identification: A Benchmark'. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Zhou, Hui et al. (2019). 'Deep Continuous Conditional Random Fields With Asymmetric Inter-Object Constraints for Online Multi-Object Tracking'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.4, pp. 1011–1022. DOI: 10.1109/TCSVT.2018.2825679.

Zhou, Xingyi, Vladlen Koltun and Philipp Krähenbühl (2020). 'Tracking Objects as Points'. In: *ECCV*.

Zhou, Zongwei et al. (2018). 'Online multi-target tracking with tensor-based high-order graph matching'. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 1809–1814.

Zhu, Ji et al. (2018). 'Online Multi-Object Tracking with Dual Matching Attention Networks'. In: *ECCV*.

Zhu, Xizhou et al. (2017). 'Flow-guided feature aggregation for video object detection'. In: *ICCV*.

1 Appendix A

1.1 Optimal Subgraph Solving and Complexity Analysis

The graph built in the study 'Multi-Object Tracking with Multiple Cues and Switcher-Aware Classification' has sparse edges and is composed of many connected components. A connected component is a subgraph in which a path exists between any pair of nodes in the subgraph. The optimal subgraph of the conflict graph is the union of optimal subgraphs in the connected components. Therefore, we first enumerate all possible subgraphs in each connected component of the conflict graph and find out the optimal subgraphs of each connected component. Then, the overall optimal subgraph is the union of such subgraphs. Due to the low density of detection results, the average detection number in each connected component is expected to be a constant not larger than C ($C \leq 8$ in MOT17). Therefore, the time complexity of enumeration to deal with N detections is $O(\frac{N}{C} \cdot 2^C)$, when $C \leq 8$, $O(\frac{N}{C} \cdot 2^C) = O(32N) \approx O(N)$.

1.2 Proof about Binary Cross Entropy Loss and Smooth L1 Loss Satisfy Equation (4.8)

The most widely used loss functions including binary cross entropy (BCE) loss and smooth L1 loss satisfy Eq. (4.8).

Lemma For any function f, if $f' > 0$ or $f' < 0$, for any valid $x_1 \neq x_2$, we have

$$\begin{aligned} &f((1-\alpha)x_1 + \alpha x_2) < \max(f(x_1), f(x_2)) \\ &\text{for } \alpha \in (0,1). \end{aligned} \tag{A1}$$

Proof Since $\alpha \in (0,1)$, $x_1 \neq x_2$ we have

$$\min(x_1, x_2) < (1-\alpha)x_1 + \alpha x_2 < \max(x_1, x_2). \tag{A2}$$

If $f' > 0$, then

$$\begin{aligned} f((1-\alpha)x_1 + \alpha x_2) &< f(\max(x_1, x_2)) \\ &= \max(f(x_1), f(x_2)); \end{aligned} \tag{A3}$$

If $f' < 0$, then

$$\begin{aligned} f((1-\alpha)x_1 + \alpha x_2) &< f(\min(x_1, x_2)) \\ &= \max(f(x_1), f(x_2)). \end{aligned} \tag{A4}$$

BCE Loss. The BCE loss is defined as follows:

$$L_{bce}(x, y) = -(1-y)\log(1-x) - y\log(x) \tag{A5}$$

where $x \in [0, 1]$ is the output (after the sigmoid layer) and y is the class label.

If the label $y = 0$, then $\frac{\partial L_{bce}}{\partial x} < 0$; if $y = 1$, then $\frac{\partial L_{bce}}{\partial x} > 0$. According to Lemma A1, L_{bce} satisfies $L_{bce}((1-\alpha)Y_a + \alpha Y_b, \gamma_z) < \max(L_{bce}(Y_a, \gamma_z), L_{bce}(Y_b, \gamma_z))$.

Smooth L1 Loss. The Smooth L1 loss is defined as follows:

$$L_{sl1}(x, y) = \begin{cases} \frac{(x-y)^2}{2k}, \text{ if } |x-y| < k, \\ |x-y| - \frac{1}{2}k, \text{ otherwise.} \end{cases} \tag{A6}$$

where k is a constant scaling factor, x is the output and y is the regression target.

When $x < y$, we have $\frac{\partial L_{sl1}}{\partial x} < 0$; when $x > y$, we have $\frac{\partial L_{sl1}}{\partial x} > 0$. According Lemma A1, if $Y_a < \gamma_z$ and $Y_b < \gamma_z$ or $Y_a > \gamma_z$ and $Y_b > \gamma_z$, L_{sl1} satisfys Eq. (4.8).

For $\min(Y_a, Y_b) < \gamma_z < \max(Y_a, Y_b)$,

if $\min(Y_a, Y_b) < (1-\alpha)Y_a + \alpha Y_b \leq \gamma_z$, then $L_{sl1}((1-\alpha)Y_a + \alpha Y_b, \gamma_z) < L_{sl1}(\min(Y_a, Y_b), \gamma_z) \leq \max(L_{sl1}(Y_a, \gamma_z), L_{sl1}(Y_b, \gamma_z))$;

if $\max(Y_a, Y_b) > (1-\alpha)Y_a+\alpha Y_b > \gamma_z$, then $L_{sl1}((1-\alpha)Y_a+\alpha Y_b, \gamma_z) < L_{sl1}(\max(Y_a, Y_b), \gamma_z) \leq \max(L_{sl1}(Y_a, \gamma_z), L_{sl1}(Y_b, \gamma_z))$.

Therefore, BCE loss and Smooth L1 loss satisfy Eq. (4.8).