Multi-Object Tracking using Mamba and an Investigation into Data Association Strategies

by

Dheeraj Khanna

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2025

© Dheeraj Khanna 2025

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

All the sections in this thesis that incorporate the motivation, research methodology, and experimentation are based on the two papers where I was the lead author during my master's program. As the lead author, I was responsible for the conceptualization, algorithm development, methodology design, conducting experiments, and writing and submitting the manuscripts. Prof. John Zelek, Prof. Yuhao Chen, and Jerrin Bright provided guidance on the methodology and contributed to the drafting and revision of these papers.

Dheeraj Khanna, John S. Zelek, "Attention-Mamba for Multi-Object Tracking", Proceedings of the Conference on Robots and Vision, 2025. *(Under Review)*

Dheeraj Khanna, Jerrin Bright, Yuhao Chen, John S. Zelek, "SportMamba: Adaptive Non-Linear Multi-Object Tracking with State Space Models for Team Sports", IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), 2025. *(Under Review)*

Additionally, I also co-authored 1 research paper in the area of computer vision, object annotation, and multi-object tracking. As the secondary author, I contributed to the development of methodology, as well as writing manuscripts for submission with Adam Srebrnjak Yang.

Yang, Adam Srebrnjak, **Dheeraj Khanna**, and John S. Zelek. "POPCat: Propagation of Particles for Complex Annotation Tasks." Proceedings of the Conference on Robots and Vision. PubPub, 2024. *(Accepted)*

Abstract

Multi-Object Tracking (MOT) is a critical component of computer vision, with applications spanning autonomous driving, video surveillance, sports analytics, and more. Despite significant advancements in tracking algorithms and computational power, challenges such as maintaining long-term identity associations, handling dynamic object counts, managing irregular movements, and mitigating occlusions persist, particularly in complex and dynamic environments. This research addresses these challenges by proposing a learning-based motion model that leverages past trajectories to improve motion prediction and object re-identification, and we also investigate how to maximize the performance of trackers with data association.

Inspired by recent advancements in state-space models (SSMs), particularly Mamba [1], we propose a novel learning-based architecture for motion prediction that combines the strengths of Mamba and self-attention layers to effectively capture non-linear motion patterns within the Tracking-By-Detection (TBD) paradigm. Mamba's input-dependent sequence modeling capabilities enable efficient and robust handling of long-range temporal dependencies, making it well for complex motion prediction tasks. Building on this foundation, we explore hybrid data association strategies to improve object tracking robustness, particularly in scenarios with occlusions and identity switches. By integrating stronger cues such as Intersection over Union (IoU) for spatial consistency and Re-Identification (Re-ID) for appearance-based matching, we enhance the reliability of object associations across frames, reducing errors in long-term tracking. Fast motion and partial overlaps often lead to identity mismatches in object tracking. Traditionally, spatial association relies on IoU, which can struggle in such scenarios. To address this, we enhance the cost matrix by incorporating Height-based IoU to handle partial overlaps more effectively. Additionally, we extend the original bounding boxes with a buffer to account for fast motion, thereby improving the robustness and accuracy of the spatial association process. Additionally, we study the impact of dynamically updating the feature bank for Re-ID during the matching stage, culminating in a refined weighted cost matrix. To further address challenges in identity switching and trajectory consistency, we introduce the concept of virtual detections in overlapping scenarios and explore its effectiveness in mitigating ID switches.

Developing a robust and accurate MOT tracker demands a critical interplay between accurate motion modeling and a sophisticated combination of stronger and weaker cues in data association. Through extensive experimental evaluations on challenging benchmarks such as DanceTrack [2] and SportsMOT [3], the proposed approaches achieve significant performance gains, with HOTA scores of 63.16% and 77.26% respectively, surpassing multiple existing state-of-the-art methods. Notably, our approach outperforms DiffMOT by

0.9% on DanceTrack and 0.06% on SportsMOT, while achieving 3- 7% improvements over other learning-based motion models. This work contributes to advancing MOT systems capable of achieving high performance across diverse and demanding scenarios.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Prof. John Zelek for giving me the opportunity to join the Vision and Image Processing (VIP) Lab and for his invaluable guidance and support throughout my master's program. I was originally admitted to MEng program in ECE, and I had always dreamed to be a part of the Vision and Image Processing Lab. Transitioning to the MASc program and being able to associate with such an esteemed lab has already been a dream come true in my academic journey. I believe this opportunity will play a crucial role in shaping my career going forward. His encouragement to explore new techniques and his expertise in the field have been crucial to shaping this thesis.

I sincerely thank Stan Kleinikkink and Paul Dragan from ATS Automation for providing me with an enriching project that offered valuable exposure to real-world computer vision challenges in manufacturing, as well as for their continued support and guidance. I would also like to thank my readers Prof. Yuhao Chen and Prof. Sirisha Rambhatla for their time in providing thoughtful and constructive feedback on my work despite their busy schedules. Your thorough examination have been instrumental in refining my work.

I am also grateful to my friends and colleagues in the VIP Lab (Kevin, Yuxiang, Adrian, Jerrin, and Prof. Yuhao Chen) for making this journey memorable through engaging work discussions and enjoyable game nights. Special thanks to Adam Srebrnjak Yang for his constant feedback, collaboration on the ATS project, insightful ideas for our CRV paper, and for always being there to drive me wherever needed. His perspective and attention to detail continually pushed me to go above and beyond in everything I worked on.

To my dear friend Ursula Das, thank you for your steadfast encouragement and unwavering belief in me over the past two years. Lastly, I am profoundly grateful to my family for their immense sacrifices and unwavering support, both financial and emotional, throughout my international education. No words of acknowledgment can truly capture the depth of gratitude I feel for my parents and everything they have done for me. Their countless sacrifices and unconditional love have been the foundation of my journey, and this accomplishment would not have been possible without them.

Dedication

This thesis is dedicated to my parents. Thank you for everything.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements vi

Dedication vii

List of Figures xi

List of Tables xiii

List of Abbreviations xiv

1 Introduction 1

1.1 Overview 1

1.2 Problem Definition and Challenges in MOT 1

1.3 Motivation 2

1.4 Contributions 4

1.5 Thesis Outline 5

2 Background **6**
2.1 Overview . . . 6
2.2 Components of Multi-Object Tracking . . . 6
2.2.1 Object Detection . . . 7
2.2.2 Object Prediction . . . 7
2.2.3 Data Association . . . 8
2.3 Categorization of MOT and Related Work . . . 8
2.3.1 Tracking by Propagation Paradigm . . . 10
2.3.2 Tracking by Detection Paradigm . . . 10
2.3.3 Motion Models for Multi-Object Tracking . . . 11
2.4 Data Association Strategies for MOT . . . 13
2.5 State Space Models . . . 14
2.5.1 Preliminaries of SSMs and Mamba . . . 15
2.5.2 Introduction to Mamba . . . 16
2.5.3 Multi-Object Tracking using Mamba . . . 17
2.6 Datasets . . . 18
2.7 Chapter Summary . . . 19

3 Methodology **21**
3.1 Notation . . . 21
3.2 Motion Prediction with Mamba-Attention . . . 22
3.2.1 Model Architecture . . . 22
3.2.2 Data Augmentation . . . 23
3.2.3 Loss Functions . . . 25
3.3 Tracking and Data Association . . . 26
3.3.1 Data Association Pipeline . . . 27
3.3.2 Spatial matching . . . 27
3.3.3 Re-ID based matching . . . 29
3.3.4 Virtual Detection Generator . . . 33
3.4 Chapter Summary . . . 39

4 Experiments and Results **40**
4.1 Implementation Details . . . 40
4.1.1 Motion Predictor Training Details . . . 40
4.1.2 Data Association Details . . . 41
4.2 Evaluation Metrics . . . 41
4.2.1 CLEAR MOT . . . 41
4.2.2 Identification F1 (IDF1) Score . . . 42
4.2.3 Higher Order Tracking Accuracy (HOTA) . . . 43
4.3 Benchmark Results . . . 44
4.3.1 DanceTrack Results . . . 44
4.3.2 SportsMOT Results . . . 44
4.4 Ablation Study and Experimental Analysis . . . 46
4.4.1 Prediction Ablation . . . 53
4.4.2 Association Ablation . . . 55
4.5 Chapter Summary . . . 60

5 Conclusion and Future Work **61**
5.1 Summary . . . 61
5.2 Limitations and Future Work . . . 62

References **63**

List of Figures

1.1 Examples of occlusions and overlap in DanceTrack and SportsMOT dataset 3

2.1 Basic components of Multi-Object Tracking 7
2.2 An outline of Tracking by Propagation and Tracking by Detection paradigms in MOT 9
2.3 Mamba [1] basic building block 17
2.4 Examples of sequences of Danceterack and SportsMOT datasets 19

3.1 Motion Prediction Architecture 24
3.2 Visual representation of EIoU 28
3.3 Demonstration of Height Adapted IoU with example on DanceTrack sequence 29
3.4 Representation of extraction of diffusion-based feature embeddings of bounding boxes for Re-identification task 33
3.5 Overview of the data association pipeline 35

4.1 Results for the evaluation metrics for DanceTrack across all the sequences in the validation dataset 46
4.2 Qualitative results of DanceTrack sequences exhibiting lower tracking performance 47
4.3 Results for the evaluation metrics for SportsMOT across all the sequences in the validation dataset. 48
4.4 Qualitative results of SportsMOT sequence exhibiting lower performance . 48
4.5 Examples of motion prediction results on sequences of Danceterack and SportsMOT datasets 50

4.6 Qualitative Results for DanceTrack validation set for sequence "dancetrack0005" on different methods 51
4.7 Qualitative results on different methods on basketball sequence from SportsMOT 52
4.8 Heatmaps for ablation study of window size w and Mamba-Attention blocks M 54
4.9 Heatmaps for ablation study of window size w and Mamba-Attention blocks M 54
4.10 Heatmaps demonstrating the effect of extended buffer sizes b_1 and b_2 for SportsMOT dataset 58
4.11 Heatmaps demonstrating the effect of extended buffer sizes b_1 and b_2 for DanceTrack dataset 58

List of Tables

4.1 Tracking results for DanceTrack test set for filter-based methods and learning-based methods . . . 45

4.2 Tracking results for SportsMOT test set for filter-based methods and learning-based methods . . . 49

4.3 Performance comparison of different Data Augmentation strategies for SportsMOT and DanceTrack datasets . . . 56

4.4 Performance comparison of effect of Height Adaptation for SportsMOT and DanceTrack datasets . . . 57

4.5 Performance comparison of different appearance extraction models for SportsMOT and DanceTrack datasets . . . 59

4.6 Effect of creating virtual tracklets on DanceTrack and SportsMOT validation datasets . . . 59

List of Abbreviations

AssA Association Accuracy 59

CIoU Complete Intersection over Union 25, 26, 39

CLEAR Classification of Events, Activities, and Relationships 41, 43

DetA Detection Accuracy 59

EIoU Extended Intersection over Union xi, 22, 28, 55, 61

EMA Exponential Moving Average 14, 30, 39, 61

GIoU Generalized Intersection over Union 26

GNN Graphical Neural Networks 2

HA-EIoU Height Adaptive Extended Intersection over Union 28, 29, 36, 41

HOTA Higher Order Tracking Accuracy 41, 43, 59

IDF1 Identification F1 score 42, 43, 59

IoU Intersection over Union iv, 11, 13, 14, 19, 27–29, 31, 41, 53, 55

KF Kalman Filter 11–13

MA Mamba-Attention 22, 23

MHSA Multi-Head Self Attention 22

MOT Multi-Object Tracking 1, 3, 5–8, 10, 11, 13, 14, 17–20, 27, 29, 41, 43, 61, 62

MOTA Multi Object Tracking Accuracy 42, 59

NSA-KF Noise Scale Adaptive Kalman Filter 12, 14

SSMs State Space Models 14, 15, 17, 20

TBD Tracking By Detection 3, 4, 10, 11, 19–23, 39, 61, 62

TBP Tracking By Propagation 10, 18–20

Chapter 1

Introduction

1.1 Overview

Multi-Object Tracking (MOT) refers to continuously and accurately tracking multiple moving objects or individuals across video frames. It is a fundamental area of computer vision with a wide range of applications, including autonomous driving, video surveillance, robot navigation, and industrial manufacturing. MOT has evolved significantly over the last decade with the advancements of sophisticated tracking algorithms [4, 5, 6, 7] and faster processing due to higher computational power [8, 9, 10]. Despite these advancements, the field continues to face challenges in achieving tracking stability comparable to human vision, particularly in long-term identity maintenance in complex and dynamic scenes [2, 11, 3, 12, 13]. The progress in MOT has been fueled by improvements in object detection, re-identification, and long-range sequence modeling algorithms [14, 15, 16, 1].

1.2 Problem Definition and Challenges in MOT

Despite significant advancements, MOT continues to face numerous challenges that hinder performance in complex scenarios such as dynamic number of targets in the frame at each instance or irregular object and camera movements. Several benchmarks and real-world applications consistently highlight these challenges, particularly in crowded environments and computationally intensive settings.

- **Dynamic Number of Targets**: In applications like autonomous driving and video surveillance, the number of objects within a scene often varies dynamically. Scenarios

such as urban traffic or crowded public spaces involve individuals or vehicles entering and leaving the camera's field of view unpredictably. Benchmarks like MOT16 [12], MOT20 [13], Crowdhuman [17] and BDD100K [18] illustrate the difficulty of maintaining consistent identity associations under these conditions. Cameras capturing dense crowds further exacerbate the challenge with frequent occlusions and highly dynamic target numbers. In autonomous vehicle scenarios, the sudden appearance or disappearance of objects complicates long-term identity retention, demanding trackers capable of adaptive, real-time, and accurate association strategies.

- **Irregular Movements**: Irregular object trajectories present another critical challenge, especially in domains like sports tracking [3, 11]. For instance, players in sports such as soccer or volleyball frequently change direction while interacting with other players or objects like the ball. Traditional linear prediction models, such as the Kalman Filter, struggle in these settings due to their reliance on predictable motion patterns. The inability to handle such non-linear and dynamic behaviors results in degraded tracking accuracy and increased identity switching.

- **Occlusions and Overlaps**: Occlusions and overlaps remain the most pervasive challenges in MOT, leading to identity switches and fragmented trajectories. Re-identification (Re-ID) modules that utilize appearance-based features have shown promise in addressing short-term occlusions. However, they often fall short in scenarios involving long-term occlusions or when appearance similarity between objects is high. Datasets such as DanceTrack [2] provide examples of these challenges in intricate scenarios. In dance sequences, performers often overlap for extended durations, creating gaps in detections and fragmented trajectories. Additionally, dancers wearing similar outfits further challenge Re-ID modules, resulting in misidentification. While graph-based association techniques, such as those employing Graph Neural Networks (GNN) [19, 20, 21], have demonstrated success in offline methods for reducing fragmentation and identity switching, their application in real-time online tracking remains not feasible.

1.3 Motivation

A high-performing MOT system must integrate accurate motion prediction in both linear and non-linear scenarios with robust data association strategies to minimize identity switching and trajectory fragmentation. Additionally, for applications such as autonomous

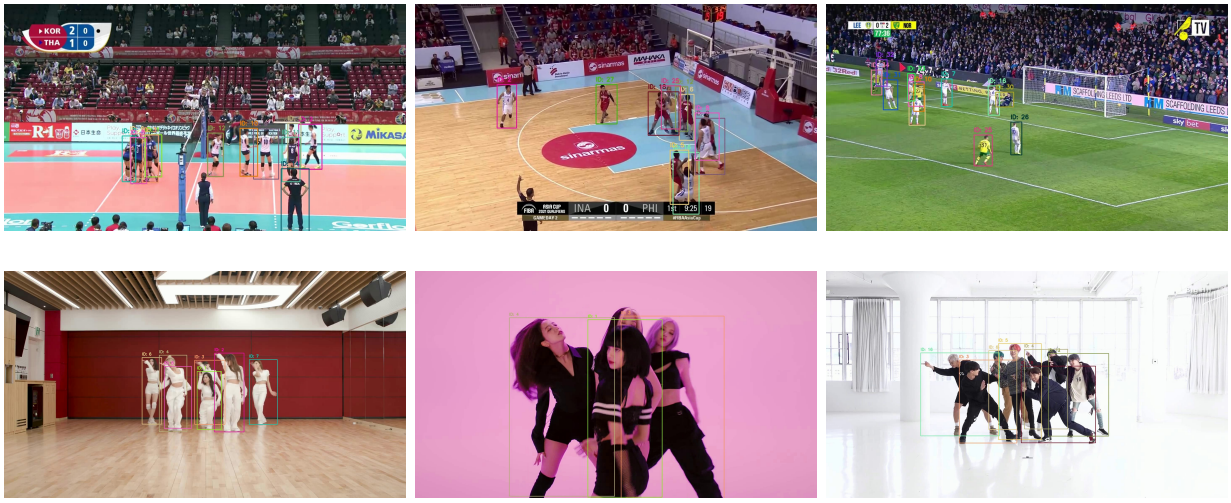


Figure 1.1: Examples of occlusions and overlap in DanceTrack and SportsMOT dataset

vehicles, sports analytics, and video surveillance, MOT systems should deliver reliable performance while meeting real-time processing requirements.

Recent advancements in Joint Detection and Tracking (JDT) methods, such as MOTR [8], MeMOTR [10], and MOTRv2 [22], have introduced end-to-end frameworks that integrate motion prediction and tracking into a unified approach (query-propagation). While promising, these methods face considerable challenges in real-time applications due to their heavy reliance on computationally expensive appearance features and intricate object interaction modules for data association.

The Tracking by Detection (TBD) paradigm has gained significant popularity in the field of MOT, particularly through the integration of heuristic-based motion models like the Kalman Filter [23], as demonstrated in methods such as OC-SORT, SORT, DeepSORT, and ByteTrack [7, 24, 25, 26]. In the TBD framework, objects are first detected in each frame and subsequently associated with existing trajectories using various data association strategies. However, methods adopting this paradigm often prioritize either enhancing data association strategies while relying on established motion models [25, 26] or improving motion models while using conventional association techniques [7, 27]. We argue that both motion modeling and data association are equally critical components of an effective tracker. A robust MOT system requires an optimal combination of accurate motion modeling and advanced data association strategies to achieve high performance across diverse and challenging scenarios.

To address this, we propose a learning-based motion model for improved prediction capabilities, complemented by an extensive experimental analysis of data association techniques. Additionally, we explore ideas to address persistent challenges such as occlusions and overlaps, aiming to enhance the system's ability to maintain track consistency and reduce identity switches.

1.4 Contributions

The primary contributions of this thesis are as follows:

1. **Novel Learning-Based Motion Model for Non-Linear Motion Prediction**: We introduce a novel learning-based motion modeling framework that effectively captures non-linear motion patterns in challenging datasets such as DanceTrack [2] and SportsMOT [3]. By integrating the state-of-the-art sequence model Mamba [1] with self-attention layers, our approach achieves a unique balance between high prediction accuracy and computational efficiency of $\approx$ 30 FPS in the Tracking-by-Detection (TBD) paradigm. Unlike traditional TBD methods that rely on handcrafted motion models, our approach enhances TBD by incorporating learned motion representations, allowing for more robust handling of complex and dynamic motion patterns.

2. **Enhanced Data Association Framework**: We propose a data association framework that incorporates hybrid matching cues, enhancing prior approaches that utilized spatial and appearance cost matrices [28, 25]. By combining weaker cues with updated feature bank strategies inspired by [6, 29], our method achieves improved robustness in complex tracking scenarios. Additionally, we extend the concept of virtual tracklet creation to further reduce identity switches and improve trajectory consistency, especially in cases involving overlapping objects. Our enhanced framework demonstrates superior tracking performance, effectively reducing ID-switch rates and improving robustness under occlusion and partial overlaps.

In our work, we hope to gain a deeper understanding of how much contribution different motion models and data association strategies play in creating an effective and accurate tracker.

1.5 Thesis Outline

The thesis is structured as follows:

- **Chapter 2**: Provides a comprehensive background on the key components of MOT. It includes an in-depth literature review of various tracking paradigms, motion models, and data association strategies. Additionally, it covers the foundational concepts of state-space models, which form the basis of the proposed Mamba model.
- **Chapter 3**: Details the methodology, presenting the Mamba-Attention-based motion model for learning complex object dynamics. It also elaborates on the data association strategies integrated into the tracking pipeline.
- **Chapter 4**: Describes the experimental framework, including implementation details, evaluation metrics, and benchmarks conducted on various datasets. This chapter presents quantitative and qualitative results, ablation studies, and hyperparameter tuning for both motion prediction and data association components.
- **Chapter 5**: Concludes the thesis by summarizing the findings and discussing future directions for research in MOT, highlighting potential advancements in motion modeling and data association techniques.

Chapter 2

Background

2.1 Overview

The background chapter will provide a foundational understanding of Multi-Object Tracking (MOT), covering its core components and literature review. Section 2.2 will discuss the fundamental elements of MOT, followed by an overview of recent advancements in related work in Section 2.3. Lastly, Section 2.5 will introduce the State Space models and Mamba sequence model, originally a foundational approach in natural language and sequence modeling, which has recently been adapted for vision tasks, highlighting its concepts and related research.

2.2 Components of Multi-Object Tracking

Multi-Object Tracking primarily relies on three interconnected components: **Object Detection**, **Object Prediction**, and **Data Association** [24]. Object Detection serves as the foundation by identifying objects of interest within individual frames, providing bounding boxes and class labels. Object Prediction extends this by estimating the future locations of these objects based on their past trajectories, often using motion models or sequence-based predictions. Finally, data association ties everything together by matching detections across frames to maintain consistent object identities, ensuring seamless and accurate tracking throughout the sequence.

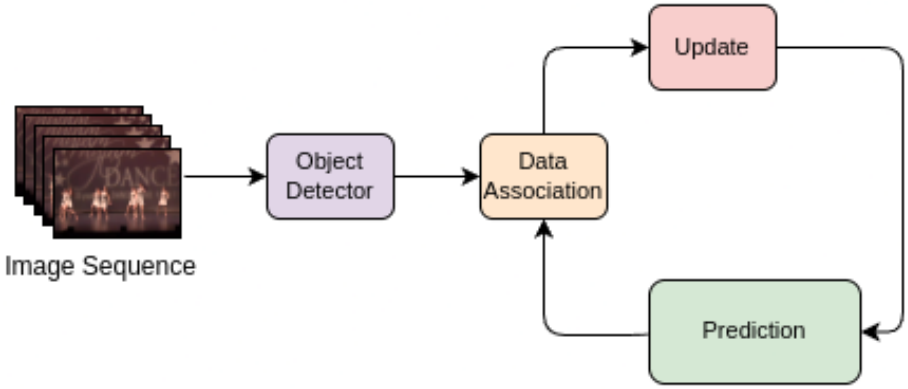


Figure 2.1: Basic components of Multi-Object Tracking

2.2.1 Object Detection

Object detection is the foundational pillar of Multi-Object Tracking, serving as the initial step where objects are localized and identified in individual frames. The quality and accuracy of detections significantly influence the overall tracking performance, as missed detections or false positives can propagate errors throughout the pipeline. Modern object detection techniques, using deep learning approaches, such as YOLO [14, 30], Faster R-CNN [31], and DETR [15, 32], have demonstrated remarkable performance in identifying objects with high precision and speed. These detectors generate bounding boxes and class labels, which serve as inputs for the subsequent tracking and association stages. Advances in object detection, including improved feature extraction, multi-scale processing, and robust handling of occlusions, have played a pivotal role in enhancing the reliability of MOT systems. Recent MOT approaches [26, 7, 27, 28, 33, 8, 10] adopt YOLO-X [14] and Deformable DETR [32, 34] as their primary detectors depending on the data association strategies followed in their approaches.

2.2.2 Object Prediction

Object prediction focuses on estimating the future states of tracked objects, leveraging motion models and temporal information where motion models capture patterns of object movement, and temporal information utilizes sequential data to account for changes over time. This component bridges the spatial gap between frames, enabling the tracker to maintain consistent identities even when detections are ambiguous or temporarily missing.

State-estimation models, such as Kalman filters [23], Particle filters [35], and deep-learning-based methods, forecast the position, velocity [24, 26], and sometimes appearance of objects [36, 8, 37, 38] in the next frame. By incorporating historical information, the detections and predictions are fused based on the accuracy of the two processes, helping the tracker handle abrupt motions and missed detections, and ensuring continuity and robustness in the tracking process [33, 39, 40]. This predictive capability is particularly crucial in online tracking scenarios, where decisions must be made frame-by-frame without access to future data.

2.2.3 Data Association

Data association links detections from consecutive frames to establish consistent object identities over time. This task involves matching new detections to existing tracklets based on spatial proximity, motion consistency, and appearance features. Effective data association strategies must address challenges such as occlusions [12, 13], appearance changes, and overlapping objects [2]. Techniques range from greedy algorithms, such as Hungarian matching, to global optimization approaches [19, 20] and attention-based mechanisms [41, 22, 10] in transformer models. Recent advancements incorporate deep embeddings and similarity metrics, further enhancing the accuracy and robustness of association [5, 28, 29]. The success of MOT systems often hinges on the effectiveness of their data association strategies, as errors at this stage can lead to identity switches and fragmented tracks, leading to lower overall tracking accuracy.

2.3 Categorization of MOT and Related Work

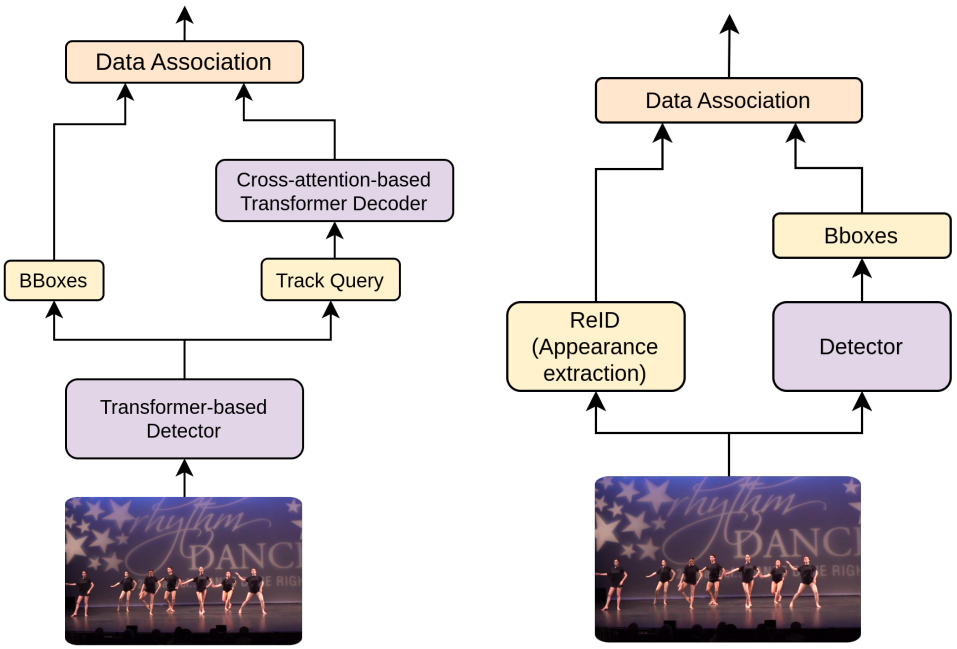
The field of Multi-Object Tracking (MOT) is extensive, with numerous approaches proposed to address the tracking problem. MOT methods are typically classified into several distinct categories, which can often overlap or combine in practice. These distinctions include:

1. **Online Tracking vs. Offline Tracking**: Online tracking methods perform data association incrementally, considering only past observations [24, 25, 26]. In contrast, offline tracking methods utilize a global perspective, forming data associations across the entire sequence by leveraging all the frames together [19].

2. **Tracking-by-Detection (TBD) vs. Joint Detection and Tracking (JDT)**: TBD methods rely on pre-detected objects and focus on associating these detections over time. JDT methods integrate detection and tracking into a single unified framework, reducing reliance on external detectors [42].

3. **Tracking-by-Regression**: These methods frame the tracking task as a regression problem, predicting the movement or position of objects across frames directly [43, 44, 45].

4. **Tracking-by-Propagation and Transformer-Based Tracking** These methods can also be included in the subcategory of joint detection and tracking paradigm. These methods utilize transformer query mechanisms to establish tracks and detections in an end-to-end manner [8, 37].



(a) Tracking by Propagation paradigm (b) Tracking by Detection paradigm

Figure 2.2: An outline of Tracking by Propagation and Tracking by Detection paradigms in MOT

2.3.1 Tracking by Propagation Paradigm

Following the success of Vision Transformers in image recognition tasks [46, 15, 32, 47], their application has expanded into the field of MOT [22, 8, 10]. Transformer-based tracking architectures typically adopt the tracking-by-propagation (TBP) paradigm. Unlike TBD methods, which rely on separate detectors and data association pipelines for each frame, TBP methods utilize track queries that are updated frame by frame. These queries are progressively used to decode and associate the locations of tracked objects. Approaches such as [8, 37] extend query-based object detection frameworks [15, 32] to develop end-to-end trainable tracking systems. By treating MOT as an autoregressive problem, these methods propagate track queries across subsequent frames to re-detect and track objects. [48, 49] use transformer architecture as a feature extractor network and pass track features to learn the aggregates embeddings of each object. Me-MOT [50] creates a memory-aggregator feature bank for short-term and long-term features using cross-attention modules. TrackFormer, P3AFormer[48, 51] utilize location-based Hungarian matching for tracklets and detections, rather than query-propagation techniques for data association. MOTR-v2 [22] incorporates YOLO-X object detector and uses its detections as anchor query in DAB-DETR [34] to further improve the detection performance and concatenates them with track query.

Transformer-based methods demonstrate strong performance on MOT benchmarks but are hindered by slower inference speeds and higher computational demands [8, 22]. For instance, MeMOTR [10] and MOTR-v3 [52] require extensive training resources, utilizing eight Tesla V100 GPUs for their TBD paradigm-based architectures. Similarly, MOTR [8] achieves an inference speed of just 7.5 FPS on a Tesla V100 GPU, significantly slower than most methods based on the TBD paradigm. MOTR-v2 [22] further highlights this computational burden, showing that the object detection step using YOLO-X [14] operates at 25 FPS, which is $\approx 3\times$ faster than their MOTR module. While transformer models provide robust tracking capabilities, their high computational requirements and slower inference speeds limit their applicability in real-time scenarios such as surveillance and autonomous driving, where efficient data association is crucial.

2.3.2 Tracking by Detection Paradigm

The Multi-Object Tracking (MOT) problem has seen significant progress through the tracking-by-detection (TBD) paradigm. In this approach, objects are first detected using off-the-shelf object detectors and subsequently associated with existing tracks through matching algorithms like the Hungarian algorithm. Recent advancements in object detection methods, such as YOLO-X [14], YOLO-v8 [30], and DETR [15], have bolstered

the effectiveness of the TBD paradigm. Many state-of-the-art methods [28, 29, 7, 53, 25] adopt this paradigm for MOT. Specifically, several leading approaches [28, 29, 7] employ YOLO-X as their primary object detector, citing its superior performance in accuracy and speed. A critical component of TBD methods is the prediction step, which uses a motion model to estimate an object's state in the current frame based on detections from previous frames. These motion models play a crucial role in maintaining object tracks, as incorrect predictions in challenging scenarios such as highly occluded scenes or non-linear trajectories can lead to misassociated tracks or unmatched predictions.

2.3.3 Motion Models for Multi-Object Tracking

Motion models are essential for predicting bounding boxes in the current frame by leveraging information from previous frames. Among these, the Kalman Filter (KF) [23] is most widely used in MOT for state estimation under a constant velocity assumption. SORT [24], for example, employs KF to perform MOT with a linear motion assumption. Several other methods [28, 25, 29, 54] enhance KF by integrating it with various association techniques, achieving improved performance on MOT benchmarks such as [12, 13]. These motion models are categorized as heuristic-based motion models.

However, the linear motion assumption often fails in scenarios with complex motion dynamics, such as tracking dancers or players on a field [2, 3]. To address this limitation, learning-based motion models [27, 39, 40, 33] have emerged, capable of capturing intricate motion trajectories to improve prediction accuracy.

Heuristic Based Motion Models

Motion models that rely on fixed motion priors and predefined hyperparameters for state estimation process are categorized as Heuristic-based motion models, Kalman Filter being a prominent example. SORT [24] employs Faster-RCNN [31] for the object detection phase and Kalman Filter for prediction, using Intersection over Union (IoU) for tracklet association across different frames. DeepSORT [25] extends this approach by extracting deep visual features for each detected bounding box, and employing a cascading approach for association using IoU as well as feature matching. Bytetrack [26] uses YOLO-X as the detector and proposes a data association strategy that incorporates both high as well as low confidence detections while still relying on IoU for matching.

These methods share a common reliance on the Kalman Filter with a constant linear velocity assumption, which makes them less effective in scenarios with complex motion

dynamics or occlusions. One critical factor in KF-based state estimation is the noise covariance matrix, which remains constant across predictions. This simplification can lead to inaccuracies in real-world scenarios where objects exhibit diverse velocities and trajectories [55]. Several methods work to improve the state estimation using Kalman Filter and tackle the errors caused by noise covariance matrix [55, 7, 54]. GIAOTracker [55] proposes Noise Scale Adaptive (NSA) Kalman Filter that updates the noise scale matrix for each detection based on its confidence score. Inspired by NSA-KF, [54] further enhances this approach by penalizing low-confidence detections. OC-SORT [7] identifies additional limitations in KF such as noise scale sensitivity and the magnification of temporal errors due the prediction being estimation-centric in nature. It proposes an Observation-Centric Re-Update (ORU) strategy, which adjusts the parameters of the tracklet when they are re-activated during data association stage. With this approach, OC-SORT improves performance in non-linear motion scenarios but still suffers in occluded environments. Several methods [29, 28, 5] highlight the added errors from moving cameras and hence append camera motion compensation (CMC) techniques in the Kalman Update stage of object prediction. StrongSORT [29] adopts the Enhanced Correlation Coefficient (ECC) model for parametric image alignment and estimates the global translation and rotation between adjacent frames. BoTSORT [28] utilizes the global motion compensation (GMC) technique provided by OpenCV [56] for video stabilization. Deep OC-SORT [5] updates the OC-SORT proposed Kalman filter with the scaled rotation and translation parameters. Despite these advancements, these methods remain heuristic in nature, as they rely on hyperparameter tuning and assumptions inherent to the Kalman Filter framework for object prediction.

Learning Based Motion Models

Learning-based motion models have emerged as a promising approach to address the limitations associated with the heuristic nature of Kalman Filter based approaches [41]. These approaches focus on data-driven motion analysis based on trajectory information for each tracklet and predict the bounding boxes in the current frame. Recent advancements in deep learning architectures, such as diffusion models [57], self-attention mechanisms [58], and sequence-based frameworks including RNNs, LSTMs [59] and Mamba [1], have positioned data-driven approaches as a compelling choice for autoregressive modeling of complex motion dynamics. These methodologies leverage their capacity to learn intricate temporal dependencies and nonlinear motion patterns, making them particularly effective for Multi-Object Tracking.

A few learning-based motion prediction models that do not depend on state estimation filters such as KF include Tracktor [60] and CenterTrack [38]. Tracktor [60] uses Faster-

RCNN with Resnet-101 as the detector, and repurposes the regression branch to update the trajectory position across frames, eliminating the need for Kalman Filters. ArTIST [61] proposes the object prediction task by creating a non-linear stochastic motion model that uses probability distribution to estimate the likelihood of the object at the next frame. It also models the interaction between the surrounding objects. CenterTrack [38] is built using CenterNet [62] object detector and focuses on predicting the displacement between the center of the objects and tracks them in an end-to-end manner. However, it only focuses on associating adjacent frames without re-initializing lost long-range tasks and hence, showcases a trade-off between speed and accuracy. MotionTrack [41] attempts to understand complex motion patterns by combining a vanilla-transformer and a dynamic multi-layer perception (Dynamic MLP) to predict the movement of the trajectory at the next moment. DiffMOT [27] treats motion prediction as a generative task and proposes a decoupled-diffusion architecture for non-linear motion modeling.

2.4 Data Association Strategies for MOT

Data association is a critical component of Multi-Object Tracking (MOT), where the primary objective is to establish consistent identities for objects as they move through a sequence of frames. This task involves associating detections from one frame with tracklet-representations of objects maintained across time while addressing challenges such as occlusions, appearance changes, and false positives. Over the years, a variety of data association strategies have been developed, ranging from simple heuristic-based methods to sophisticated optimization and deep learning approaches. These strategies often leverage spatial, temporal, and appearance features to resolve ambiguities and ensure robust tracking. This section explores the diverse approaches to data association, highlighting their underlying principles, key methodologies, and applications in different MOT paradigms.

Spatial heuristics, such as Intersection over Union (IoU), remain the most widely used similarity metric in data association. SORT [24] is a traditional method that uses a standard Kalman Filter (KF) as the motion model and performs IoU-based matching after every frame. ByteTrack [26], while also relying on IoU as the similarity metric, introduces a two-stage approach by utilizing both low and high-confidence detections, outperforming single-stage IoU matching. Similarly, ConfTrack [54] divides tracks into confirmed and tentative categories and matches them separately with high and low-confidence detections in a cascaded manner. Re-identification (Re-ID) is another crucial cue for effective data association, particularly in long-term tracking scenarios where tracklets may be lost due to occlusion. DeepSORT [25] was one of the first methods to incorporate appearance features

by introducing an independent Re-ID backbone to extract these features and employing a cascaded matching approach. MOTDT [63] follows a two-stage strategy, using appearance similarity in the first stage and IoU to match unmatched tracklets in the second. QD-Track [44] transforms appearance similarity into a probability distribution via a softmax operation, followed by nearest neighbor matching. These methods are often benchmarked on datasets such as MOT17 [12], and MOT20 [13] which are crowd MOT datasets where linear assumption of motion works in multiple cases

Datasets such as DanceTrack [2] and SportsMOT [3], characterized by complex motion scenarios, long occlusions, and individuals wearing similar clothing or jerseys, present significant challenges for traditional approaches that rely solely on IoU and Re-ID cues. These video sequences demand more advanced similarity metrics and matching strategies. Recent methods address these challenges by combining multiple cues and leveraging motion analysis. StrongSORT [29] enhances the DeepSORT algorithm by replacing the Kalman Filter with NSA-KF, utilizing YOLO-X instead of Faster R-CNN for detection, integrating a stronger Re-ID module [29], and incorporating an Exponential Moving Average (EMA)-based feature bank to dynamically update appearance embeddings. OC-SORT [7] calculates the association cost matrix by combining IoU with changes in object velocity direction across frames. Building on OC-SORT, Deep OC-SORT [5] integrates deep appearance features and dynamically updates the EMA-based feature bank based on detection confidence scores. HybridSORT [6] addresses the limitations of relying solely on strong cues such as IoU and Re-ID by introducing additional "weaker" cues, such as the IoU of object heights and changes in directional velocities across frames. It also incorporates confidence scores for tracklet prediction and combines these cues to create a robust data association pipeline. Similarly, C-BIoU [64] and Deep-EIoU [4] tackle challenges associated with fast-moving objects by adding buffers to the IoU calculation for the association matrix. Both approaches use a Kalman Filter as the motion model and refine the data association pipeline accordingly.

These advancements highlight that adding buffers for IoU and dynamically updating Re-ID features using EMA can be highly effective for learning-based motion models as well on challenging datasets like DanceTrack and SportsMOT.

2.5 State Space Models

State Space Models (SSMs) have long been foundational in fields such as control theory and signal processing, providing an efficient framework for sequential data modeling. Recent advancements have extended their application to Natural Language Processing (NLP) and

Computer Vision. The introduction of frameworks like the HiPPO matrix [65], designed to optimize memory updates in long-range sequences, addressed the fundamental memory limitations in sequence models such as RNNs, LSTMs, and GRUs, catalyzing the development of modern SSMs. Subsequent works such as Linear State Space Layers (LSSLs) [66] and Structured State Space models (S4) [67] advanced the field by demonstrating robust performance on long-sequence modeling tasks through spectral parameterization and linear scalability. LSSL [66] proposes a simple sequence model that unifies the concept of recurrence, convolutions, and continuous time in a single model. However, it is still infeasible for long-range sequences because of its quadratic computation and high memory requirements. S4 [67] modified sequence modeling such that it scales linearly with the sequence length. This makes S4 an efficient alternative to the transformers. S5 [68] further streamlines the S4 layer utilization by integrating parallel scans and multi-input multi-output (MIMO) capabilities, leading to widespread adoption across various domains. SSM adoptions have also shown promise in various computer vision domains such as image classification, image and video generation, and segmentation.

2.5.1 Preliminaries of SSMs and Mamba

The foundational equations for SSMs are described by linear ordinary differential equations with the goal of mapping an input sequence $x(t) \in \mathbb{R}$ to an output response $y(t) \in \mathbb{R}$ through the latent space $h(t) \in \mathbb{R}^N$. This also involves parameterization coefficients $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, and $C \in \mathbb{R}^{N\times 1}$, as outlined in equation 2.1.

$$\begin{aligned} h'(t) =& Ah(t) + Bx(t) \\ y(t) =& Ch(t) \end{aligned} \tag{2.1}$$

In the context of deep learning, sequence models deal with discrete inputs, necessitating the conversion of continuous state transition parameters into their discrete counterparts. In any state space layer architecture, this discretization is done using Zero-Order Hold (ZOH) discretization formulation using a time-scaling parameter Δ to update A and B to $\overline{A}$ and $\overline{B}$. The formulations for these updated parameters are provided in Equation 2.2, while the updated discrete state transition equations are described in Equation 2.3. To emphasize the discrete nature of the variables, the hidden, input, and output states are represented at time instance k by h_k, x_k, and y_k respectively instead of their continuous-time counterpart in Equation 2.3.

$$\begin{aligned}\overline{A} &= \exp(\Delta A) \\ \overline{B} &= \overline{A}^{-1}(\exp(\Delta A) - I).\Delta B\end{aligned} \tag{2.2}$$

$$\begin{aligned}h'_k &= \overline{A}h_k + \overline{B}x_k \\ y_k &= Ch_k\end{aligned} \tag{2.3}$$

2.5.2 Introduction to Mamba

Building upon the advancements in state space models, Mamba [1] shows a significant improvement by introducing input-dependant parameterization. where the state transition ($\overline{A}$) and input transformation ($\overline{B}$) matrices evolve dynamically based on the input. By parameterizing these matrices the model enhances adaptability, allowing the state-space dynamics to shift in response to incoming data, leading to improved expressiveness and flexibility in sequence modeling. It also incorporates a hardware-aware approach in performing GPU operations over fast SRAM rather than performing frequent data transfer between SRAM and slow HBM (High Bandwidth Memory) to optimize the computational efficiency.

Unlike other SSM layers such as S4, [67] where the parameterization coefficients $\overline{A}$, $\overline{B}$, $\overline{C}$, and Δ are constant, the design of Mamba involves parameterization of $(\Delta, \overline{B}, \overline{C})$ using representations extracted from the input sequence data, enabling it to model time-variant systems effectively. Equations 2.4 refers to the input-driven algorithm used in Mamba [1]. In the given set of equations, $f_B(x)$, $f_c(x)$, $f_\Delta(x)$ represent the transformation functions to initialize the parameterization coefficients based on input data. τ_Δ represents softplus activation function and the discretize(Δ, A, B) function utilize the set of equations defined in 2.2.

$$\begin{aligned}B &= f_B(x) \\ C &= f_c(x) \\ \Delta &= \tau_\Delta(\Delta + f_\Delta(x)) \\ (\overline{A}, \overline{B}) &= \text{discretize}(\Delta, A, B)\end{aligned} \tag{2.4}$$

Each Mamba block integrates a selective scan module, 1D causal convolution, and normalization layers, enhancing its ability to process sequences with speed and accuracy. The basic building block of a Mamba module is shown in Fig. 2.3 Due to its computational

efficiency, it achieves linear scaling with sequence length during training and inference, outperforming Transformers, which require $\mathcal{O}(n^2)$ computation [58] while maintaining comparable accuracy. Its robustness in long-range modeling makes it a viable choice for complex tasks like natural language modeling and motion modeling. By bridging the gap between Transformers and traditional SSMs, Mamba shows the potential of model SSMs to redefine sequence modeling across diverse applications.

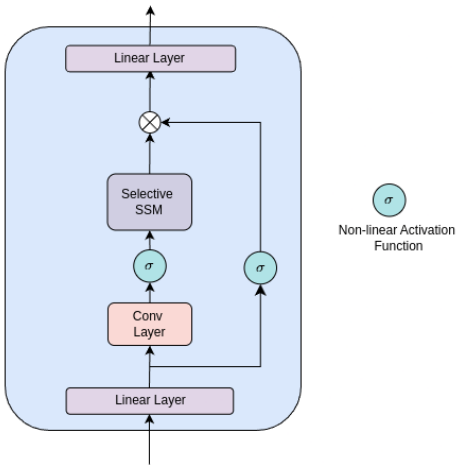


Figure 2.3: Mamba [1] basic building block

2.5.3 Multi-Object Tracking using Mamba

Mamba-based models have also been leveraged in the field of MOT[39, 33, 9, 69] to address the challenges of non-linear motion modeling [33, 39, 69] and to achieve efficient data association [9]. MambaTrack$^+$ [39] establishes the foundation of Mamba in MOT and introduces a vanilla-Mamba based architecture that learns the offsets between the bounding boxes between the previous frame and current frame using sequential trajectory information. It also makes use of the embeddings containing useful information from hidden state spaces and incorporates them in calculating L1 and L2 losses with previous frame embeddings during data association. MambaTrack [33] employs a bidirectional approach by concatenating 2 mamba blocks and using forward and backward trajectory information. It also introduces a "Tracklet Patching Module" that treats the predictions as actual observations in cases of lost tracks, enabling forward and continuous predictions of their states.

TrackSSM [69] adopts an encoder-decoder framework, with Mamba-based layers in the encoder for aggregating motion representations and a Flow-SSM module in the decoder to facilitate temporal autoregression of bounding boxes. Meanwhile, SambaMOTR [9] utilizes the Samba [32] framework for object detections and treats MOT under the TBP paradigm, inspired by MOTR [8] and TrackFormer [37]. It relies on individual SSMs to independently model each tracklet and then synchronize the memory representations across the set of tracklets to enable tracklet interaction. These diverse adaptations of Mamba illustrate its versatility and potential for advancing MOT methodologies across various applications.

2.6 Datasets

There have been multiple datasets introduced for multi-object tracking ranging for different use-cases. Datasets for pedestrian tracking use-case include MOT16 [12], MOT17 [12], and MOT20 [13] which include camera captured videos of people moving in different areas such as malls, roads, etc. The movement of people in these datasets is mostly linear in nature and hence, methods such as [24, 25, 26] which have a linear tracking assumption perform well on such datasets.

Other datasets include complex use-case and are typically difficult in nature due to their irregular and non-linear nature. These included datasets such as DanceTrack [2] and SportsMOT [3] tracking dataset.

- **DanceTrack**: DanceTrack [2] is a recently proposed and highly challenging dataset in the field of multi-object tracking (MOT), designed to evaluate the robustness of trackers in complex scenarios. It consists of 100 group dancing videos, divided into 40 training sequences, 25 validation sequences, and 35 testing sequences. DanceTrack is characterized by targets with highly similar appearances, frequent interactions, and severe occlusions, as well as diverse and non-linear motion patterns, including irregular directional changes. Unlike datasets such as MOT17 [12], which primarily feature pedestrians with linear motion, DanceTrack emphasizes non-linear motion due to the dynamic nature of various dance movements and rhythms, posing significant challenges to advanced tracking approaches based on appearance and motion cues. Notably, the detection task in DanceTrack is relatively straightforward, making it an ideal benchmark for assessing the association performance of trackers under demanding conditions.
- **SportsMOT**: SportsMOT [3] is a challenging dataset for MOT in sports, featuring 240 video sequences with over 150K frames and 1.6M bounding boxes from basketball,

soccer, and volleyball. The dataset includes 45 training and validation clips and 150 testing sequences, sourced from events like the Olympics and NBA, with videos at 720p and 25 FPS. Unlike certain DanceTrack sequences, where the camera is positioned at eye level with a straight, horizontal view, often leading to occlusions and overlapping dancers, SportsMOT adopts an overhead perspective. This top-down view minimizes such challenges of occlusion and provides a clearer visual of the scene. This dataset is distinguished by its unique motion patterns, including fast and variable-speed movements, abrupt directional changes, extensive acceleration and deceleration, and frequent camera movements, making it a significant benchmark for evaluating the association performance of MOT systems.

A few examples of both the datasets are given in Figure 2.4

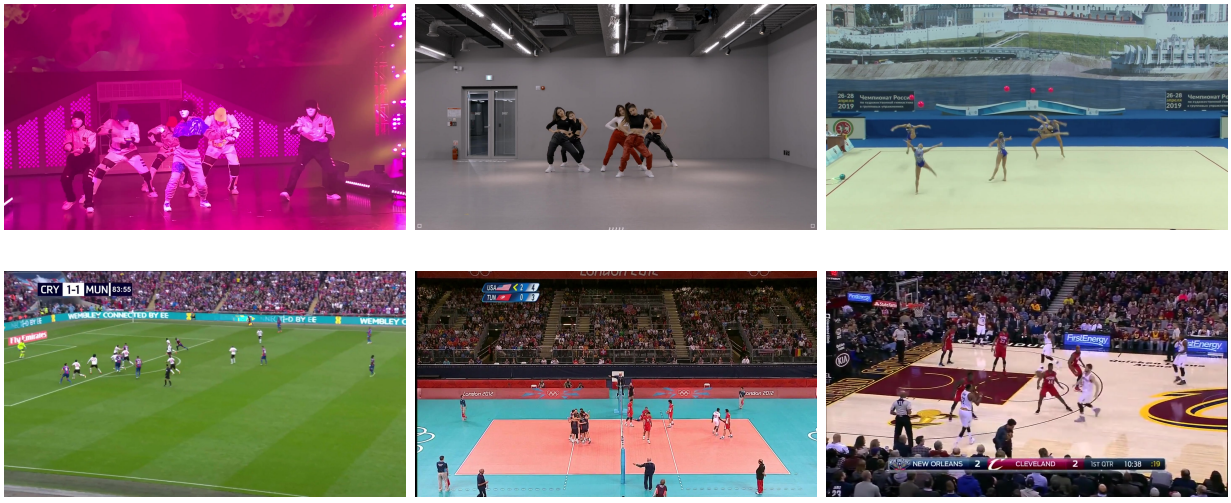


Figure 2.4: Examples of sequences of Danceterack and SportsMOT datasets

2.7 Chapter Summary

This chapter provides an overview of MOT, focusing on object detection, motion prediction, and data association in MOT, with related works in TBD and TBP paradigm. SORT [24] serves as the foundation model in the TBD paradigm that uses Kalman Filter for motion prediction and IoU for data association. Key developments in TBD are discussed

[7, 55, 28, 5] to improve upon the traditional methods in motion modeling. Additionally, we highlight learning-based motion models [33, 27, 39] that address complex motion patterns, particularly in datasets like Dancetrack [2] and SportsMOT [3]. We also discuss the methods that have primarily contributed to improving data association [4, 64, 53].

In the TBP paradigm, we explore approaches leveraging a query-propagation methodology [52, 8, 10], using Detection Transformers [15] as the baseline. While some of these methods outperform TBD-based approaches, their primary limitation lies in computational inefficiency, preventing real-time performance. We also examine SSMs and Mamba, highlighting their recent applications in MOT for both the TBD paradigm [33, 39] and the TBP paradigm [9].

In this discussion, notably, only a few methods [5, 53] combine accurate motion modeling with data association. However, these rely on updated Kalman filters rather than learning-based motion modeling. In our work, we propose a learning-based motion modeling approach using Mamba [1], complemented by a thorough analysis of effective matching strategies for efficient data association.

Chapter 3

Methodology

The proposed research is structured around 2 pivotol phases for TBD-based Multi-Object Tracking. Keeping the object detection phase constant, these phases include (i). Motion Prediction, (ii) Data Association.

This chapter comprises of 4 sections. Section 3.1 describes the notation used for the task of object prediction and data association in the upcoming sections. Section 3.2 describes the proposed motion prediction architecture that is built using a combination of Mamba and self-attention mechanisms. Section 3.3 showcases a combination of data association techniques in order to achieve an optimal performance on several benchmarks.

3.1 Notation

Model Training Notations

For the task of prediction, an input tracklet for an object over the sequence of l frames is defined as $T_{in} = \{B_{t-l}, B_{t-l+1}, ...B_{t-1}\}; T_{in} \in \mathbb{R}^{l\times 4}$, where each bounding box $B_i \in \mathbb{R}^{1\times 4}$ is represented as $\{x_i, y_i, w_i, h_i\}$ with (x_i, y_i) denoting the top-left corner coordinates, and (w_i, h_i) denoting the width and height, respectively. These bounding boxes are ground truth annotations for the object's state at each frame from $t-l$ to $t-1$.

We then define a motion prediction model that takes in tracklets T_{in} as input of length $l \in \{2, 3, 4, ..., w\}$ and predicts the state of the bounding box at time t, given by $P_t = Predictor(T_{in}); P_t \in \mathbb{R}^{1\times 4}$. Here P_t is the predicted bounding box for the object at the current frame t. The parameter w refers to the maximum tracklet length used for training

the model. The number of blocks used in the model, denoted by M, determine the depth of the model.

Tracking and Association Notations

The proposed method adheres to online tracking and follows the TBD paradigm. We use the widely accepted YOLO-X [14] as the off-the-shelf object detector. The output from the detector is defined as $D_t = [B_t, s_t)]; D_t \in \mathbb{R}^{1\times5}$ where B_t are the detected bounding boxes for each object given as $B_t = \{x_t, y_t, w_t, h_t\}$ and $s_t \in [0, 1]$ defines the confidence score for each detection. We use buffer sizes $b_1 \in \{0.25, 0.3, ..., 0.4\}$ and $b_2 \in \{0.25, 0.3, ..., 0.45\}$ for EIoU calculation in the data association step.

3.2 Motion Prediction with Mamba-Attention

3.2.1 Model Architecture

We propose to implement a motion model using the Mamba, paired with a self-attention block, that is used in predicting the next state P_t based on the input trajectory T_{in}. The model is divided into 3 segments, (a) input embedding, (b) Mamba-Attention Module, (c) Prediction Head.

- **Input Embedding**: The input trajectory T_{in} is first converted to token embeddings, denoted as E_{in} by passing them through a Linear Projection Layer. This transformation maps the bounding box tracklets to higher-dimensional embeddings. Equation 3.1 shows the transformation from input sequences to token embeddings.

$$\mathrm{E}_{in} = \mathrm{Linear}(T_{in}) \tag{3.1}$$

- **Mamba Attention Module**: After the input tracklets are converted to higher dimensional token embeddings, they are fed into the Mamba-Attention Module that consists of M Mamba-Attention blocks. Each block integrates the Mamba architecture with a multi-head self-attention (MHSA) mechanism, followed by normalization and feed-forward operations.

  Given the input passed through the l^{th} Mamba-Attention block (MA Block) be x, equation 3.2 shows the processing steps in the MA block, where x_{ssm}, x_{ln}, x_{att}, and

x_{ln2} represent the intermediate outputs of a single block. Further processing through the feed-forward network and layer normalization is shown in equations 3.3 and 3.4, denoted by x_{ff} and x_{out} respectively.

$$\begin{aligned} x_{ssm} &= \text{Mamba}(x) \\ x_{ln} &= \text{LayerNorm}(x + x_{ssm}) \\ x_{att} &= \text{MHSA}(x_{ln}) \\ x_{ln2} &= \text{LayerNorm}(x_{ln} + x_{att}) \end{aligned} \tag{3.2}$$

$$x_{ff} = \text{Linear}(\text{Gelu}(\text{Linear}(x_{ln2}) \tag{3.3}$$

$$x_{out} = \text{LayerNorm}(x_{ff}) \tag{3.4}$$

- **Prediction Head Module**: After processing the sequence token embeddings through the M MA blocks, the final stage consists of a prediction head. This head includes a Linear layer followed by a Sigmoid activation function, which normalizes the output embeddings to the range [0, 1]. The overall architecture of the combined 3 segments of the proposed model is shown in Figure 3.1.

3.2.2 Data Augmentation

As with any learning-based model, data augmentation plays a crucial role in enhancing the accuracy of the predicted bounding boxes, especially when dealing with sequences of tracklet information. In the data association stage of TBD paradigm, tracklets are constructed based on detected bounding boxes, which themselves are prone to errors. This introduces a necessity to inject various conditional noises into the training data to enable the model to handle the imperfections present in real-time detections more robustly.

The data augmentation techniques employed during training are outlined below:

- ***Random Length Sequences***:

  During the inference stage, each tracklet can vary in length due to objects entering or leaving the frames. Therefore, it is essential to train the model with tracklets of varying lengths to improve its prediction accuracy. In our setup, the tracklet lengths range between 2 and w, where w denotes the maximum window size of a tracked sequence during training.

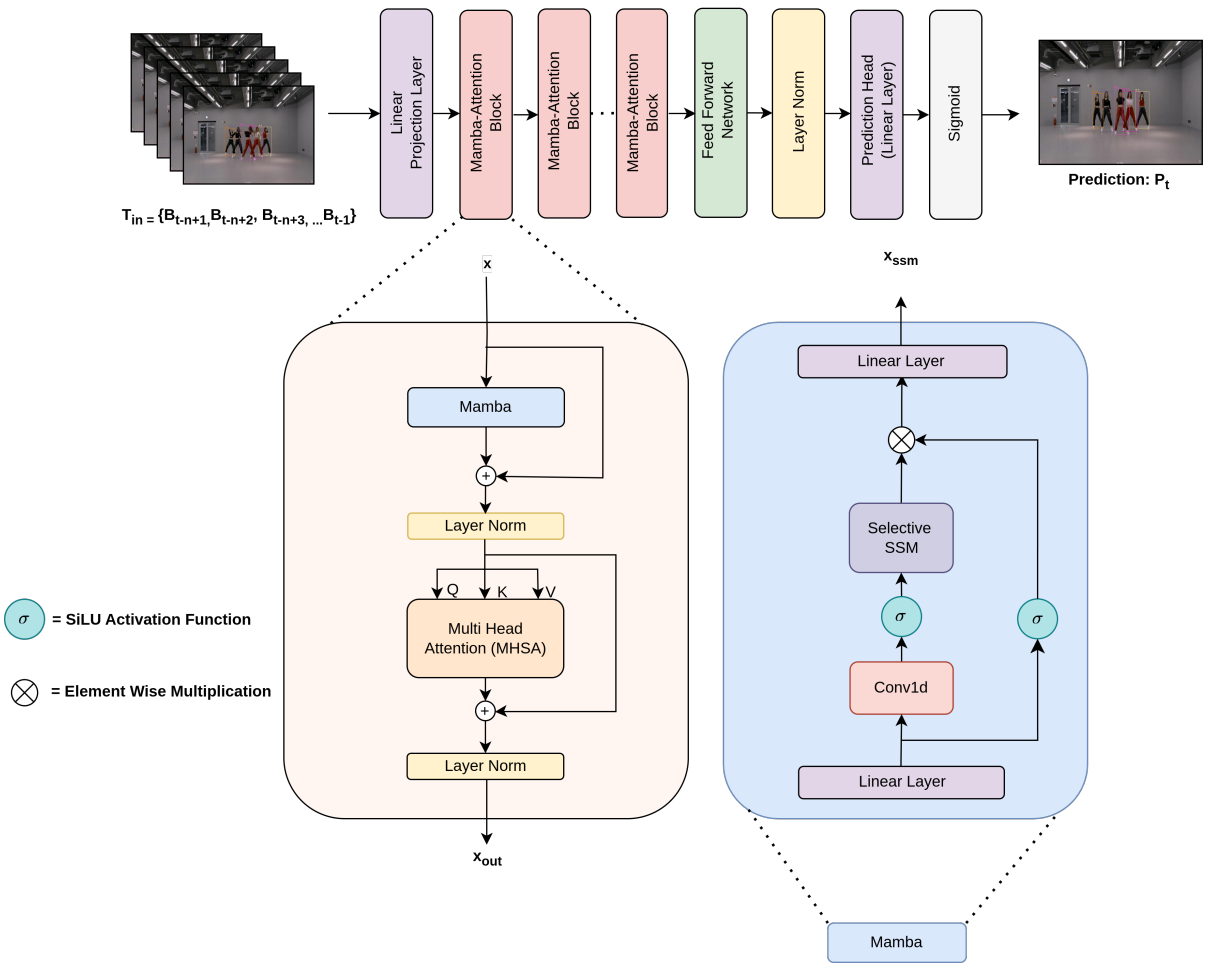


Figure 3.1: Motion Prediction Architecture

- ***Spatial Jitter***:

  Detections in individual frames may deviate from the ground truth bounding boxes due to factors such as motion blur, partial occlusion, and camera noise. To address these deviations, spatial jitter is incorporated into the ground truth bounding boxes during the training phase. This augmentation simulates real-world detection imperfections and includes the following sub-techniques:

  – **Random Scaling:** In random scaling, the width and height of the bounding boxes are scaled up or down by a factor α. This simulates variations in object

size.

- **Random Translation:** In random translation, the x_t and y_t coordinates of the bounding boxes are shifted by a random factor, introducing slight positional variations.
- **Noise Injection:** Gaussian noise is added to the elements of the bounding box coordinates D_t. This ensures that the bounding box deviations reflect typical sensor and detection inaccuracies encountered in real-world scenarios.

Random Length Sequencing is applied to every tracklet during training to ensure the model can handle variations in tracklet lengths. Meanwhile, different types of spatial jitters are applied only to a small percentage of the dataset. It is ensured that no single tracklet undergoes all three spatial jitter transformations simultaneously, as such a combination would result in significant deviations from the ground truth's position. These deviations can hinder the model's ability to accurately predict bounding boxes by introducing unrealistic spatial distortions, which in turn compromise the spatial alignment and object localization accuracy.

3.2.3 Loss Functions

The loss functions used for training the motion predictor was a combination of Smooth L1 Loss ($\mathcal{L}_{\text{SmoothL1}}$) [70] and CIoU Loss ($\mathcal{L}_{\text{CIoU}}$). Let $G_t \in \mathbb{R}^{1\times 4}$ be the ground truth bounding box and $P_t \in \mathbb{R}^{1\times 4}$ be the predicted bounding box at time t. Smooth L1 Loss is defined in equation 3.5.

$$\mathcal{L}_{\text{SmoothL1}} = \begin{cases} \frac{1}{2}(P_t - G_t)^2 & \text{if } |P_t - G_t| < 1, \\ |P_t - G_t| - \frac{1}{2} & \text{otherwise} \end{cases} \tag{3.5}$$

The CIoU Loss is defined in equation 3.6.

$$\mathcal{L}_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^g)}{c^2} + \alpha v, \tag{3.6}$$

where:

- IoU is the Intersection over Union.

- $\rho^2(\mathbf{b}, \mathbf{b}^g)$ is the squared Euclidean distance between the centers of the predicted and ground truth bounding boxes.
- c is the diagonal length of the smallest enclosing box covering both bounding boxes.
- $v = \frac{4}{\pi^2}\left(\arctan\frac{w^g}{h^g} - \arctan\frac{w}{h}\right)^2$ measures the aspect ratio difference.
- $\alpha = \frac{v}{(1-\text{IoU})+v}$ balances the aspect ratio term.

The overall loss function is defined in Equation 3.7, with weighting coefficients $\lambda_2 = 1$ and $\lambda_1 = 50$. Smooth L1 Loss measures the coordinate differences between the predicted and ground truth bounding boxes, while CIoU Loss evaluates the overlap, center alignment, and aspect ratio consistency of the bounding boxes. During training, we observed that Smooth L1 Loss decreased much faster than CIoU Loss due to their different scales, causing the optimization process to prioritize minimizing CIoU Loss. This imbalance resulted in slower or incomplete convergence of the model.

$$\mathcal{L} = \lambda_1 * \mathcal{L}_{\text{SmoothL1}} + \lambda_2 * \mathcal{L}_{\text{CIoU}} \tag{3.7}$$

To address this issue, a higher weight $\lambda_1 = 50$ was assigned to Smooth L1 Loss. This adjustment balances the contributions of the two loss terms, ensuring that the model optimizes both localization accuracy and bounding box consistency. By scaling Smooth L1 Loss appropriately, the model achieved better generalization and faster convergence.

Several loss functions were considered to supervise the training process, including Mean Squared Error (MSE) Loss and Generalized Intersection over Union (GIoU) Loss. However, CIoU Loss was ultimately chosen over GIoU. This choice was made because CIoU explicitly optimizes the alignment between predicted and ground truth bounding boxes by incorporating the minimization of the Euclidean distance between their centers, in addition to preserving aspect ratio consistency. In contrast, while GIoU focuses on enhancing the overlap between bounding boxes, it does not account for the spatial alignment and shape similarity as comprehensively as CIoU.

3.3 Tracking and Data Association

After trajectory prediction, data association is the most important step that determines which prediction will match with the detected bounding boxes in the current frame by matching the predictions with detections at each frame.

3.3.1 Data Association Pipeline

ByteTrack [26] is one of the most popular data association strategies used in recent Multi-Object Tracking (MOT) methods, often combined with various motion models for prediction. We adopt the ByteTrack strategy in our pipeline, introducing several modifications at different stages of the matching process.

The detections are generated by the YOLO-X detector and are used for matching, including both high-confidence and low-confidence detections [26]. This ensures that as much information as possible is utilized during association. At each stage, detections are matched to tracklets by solving a linear assignment problem using the Hungarian Algorithm. Matching is done by using cost embeddings that are obtained by using several types of cues. We use a combination of cues for different stages of association and they are as follows:

3.3.2 Spatial matching

Extended Intersection over Union (EIoU)

In complex motion scenarios, such as team sports [3, 11] and dance sequences [2], players and dancers often exhibit accelerated movements and sudden changes in direction. These rapid and unpredictable motions can make it challenging, even for learnable motion models, to predict the next bounding box accurately. As a result, when calculating the Intersection over Union (IoU) in such sequences, there may be insufficient overlap between the tracklet and the detection, leading to potential mismatches.

Two approaches can help address this issue. The first is to lower the IoU matching threshold, making it more forgiving during association. The second approach involves enlarging the bounding boxes by a small fraction to increase the likelihood of overlap between the predicted and detected boxes. This adjustment effectively adds a buffer to the search area, improving the chances of successful matching during IoU-based association.

As described in 3.1, a detected bounding box at frame t is given as $B_t = \{x_t, y_t, w_t, h_t\}$. Similarly, the predicted bounding box at frame t from the previous tracklet information is given as $P_t = \{\hat{x}_t, \hat{y}_t, \hat{w}_t, \hat{h}_t\}$. Buffer sizes are represented as $b \in \{0.25, 0.3, ...0.45\}$. We use 2 different buffer sizes b_1 and b_2 at different stages of matching.

The updated bounding boxes are then given as $B_t^{upd} = \{x_t - 0.5bw_t, y_t - 0.5bh_t, w_t + bw_t, h_t + bh_t$ for detection, and $P_t^{upd} = \{\hat{x}_t - 0.5b\hat{w}_t, \hat{y}_t - 0.5b\hat{h}_t, \hat{w}_t + b\hat{w}_t, \hat{h}_t + b\hat{h}_t$ for

prediction. The Extended IoU (EIoU) is defined in equation 3.8 and an example diagram is shown in Figure 3.2.

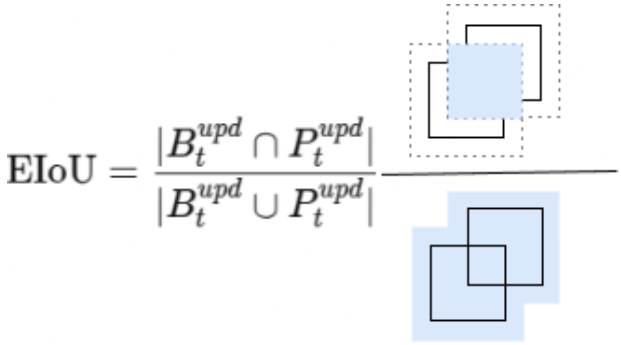


Figure 3.2: Visual representation of EIoU. Dotted lines represent the extended BBoxes. Blue shaded region represents the intersection in the 2 BBoxes in the numerator and union of the 2 BBoxes in the denominator.

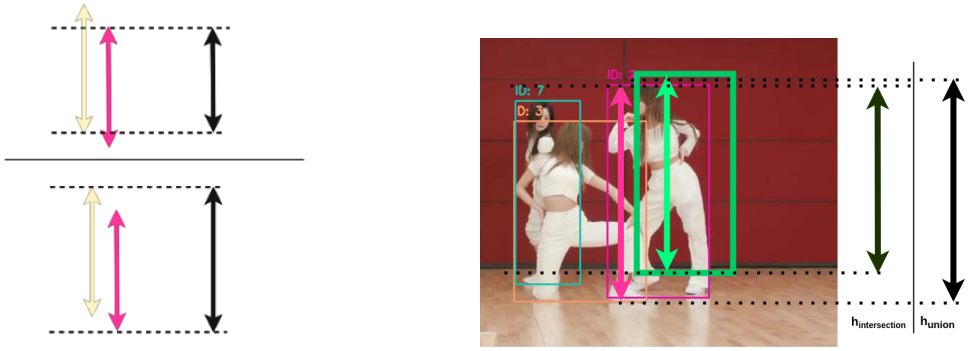
$$\text{EIoU} = \frac{\text{Area of Intersection of Extended Boxes}}{\text{Area of Union of Extended Boxes}} \tag{3.8}$$

Height Adapted EIoU

As highlighted in [6], weaker cues also play a role in improving the performance of a tracker. In sequences such as DanceTrack, height is independent of the poses of the dancers and can give a better estimation of the depth of the dancers with respect to the camera. This makes height as an effective cue in order to improve the performance of IoU matching. An illustration of HIoU computation and the role of height perception in tracking is presented in Figure 3.3. We utilize a combination of both EIoU, as well as the height of actual bounding boxes to create Height Adapted EIoU (HA-EIoU).

Given a detection at a frame t, it is represented as $B_t = (x_t^1, y_t^1, x_t^2, y_t^2)$, where (x_t^1, y_t^1) and (x_t^2, y_t^2) denote the top-left and bottom-right corners of the bounding box, respectively. Similarly, let the prediction at time t be represented as $P = (\hat{x}_t^1, \hat{y}_t^1, \hat{x}_t^2, \hat{y}_t^2)$.

$$\begin{aligned} \text{HIoU} &= \frac{\min(y_t^2, \hat{y}_t^2) - \max(y_t^1, \hat{y}_t^1)}{\max(y_t^2, \hat{y}_t^2) - \min(y_t^1, \hat{y}_t^1)} \\ \text{HA-EIoU} &= \text{HIoU.EIoU} \end{aligned} \tag{3.9}$$



(a) Visual representation of HIoU

(b) Visual example of perception of heights on DanceTrack

Figure 3.3: Demonstration of Height Adapted IoU with example on DanceTrack sequence [Demonstration of Height Adapted IoU with example on DanceTrack sequence.]

3.3.3 Re-ID based matching

IoU serves as the primary matching metric in MOT due to its reliance solely on spatial information, making it highly effective in scenarios without occlusions. However, challenges arise in cases involving significant occlusions. For instance, dancers frequently move behind one another in complex dance sequences [2], while players engage in tackles and close physical interactions in sports such as volleyball, basketball, and soccer [3, 11]. In such scenarios, multiple detections may overlap with predicted bounding boxes, leading to identity switches and incorrect associations. These limitations underscore the necessity for an additional robust matching cue. Appearance-based feature matching, particularly through re-identification across frames, is widely adopted in MOT as it enhances association accuracy in occluded and visually complex environments.

In addition to HA-EIoU, we also use appearance features for tracklet matching during 1st stage of association. We use the SBS-S50 model from FastReID library as the Re-ID feature extractor. This model is pre-trained on bag-of-tricks dataset, and we use the fine-tuned weights on DanceTrack and SportsMOT datasets. Appearance embeddings are only calculated for the latest detections and not for the predictions.

In Re-ID matching, the feature embeddings are calculated from the detections at current timestep, ie. at frame t, and are then matched with the feature embeddings of tracklets that were obtained at frame $t-1$.

Dynamic Exponential Moving Average

The feature embeddings of tracklets obtained at time $t-1$, when matched, are updated using the **Exponential Moving Average (EMA)** method, as initially proposed by [71] and subsequently adopted by [28, 29]. This approach eliminates the need to compute feature embeddings separately for both detections and predictions in each frame, thereby reducing computational overhead. The feature embedding of a detection at frame $t-1$, denoted as $f_i^{t-1}; f_i \in \mathbb{R}^{L\times 1024}$, matches tracklet embedding $e_i^{t-2}; e_i \in \mathbb{R}^{M\times 1024}$ from the previous timestep. Here, L and M are the number of tracklets and detections respectively, and 1024 is the dimension of the feature vector for each detection. The updated embedding e_i^{t-1} is computed as shown in Equation 3.10.

$$e_i^{t-1} = \alpha e_i^{t-2} + (1-\alpha) f_i^{t-1} \tag{3.10}$$

where α is a smoothing factor that determines the contribution of the historical embedding e_i^{t-1} relative to the new detection embedding f_i^{t-1}. The updated embedding e_i^{t-1} is then used to match the detections f_i^t at the current timestep t.

The EMA process enables the feature embedding of each tracklet to evolve smoothly over time by incrementally incorporating the appearance of newly matched detections while preserving historical consistency. In conventional EMA, α is a fixed constant; however, in challenging scenarios, such as when motion blur reduces detection confidence, the feature embeddings f_i^{t-1} may not accurately represent the object's appearance. Directly updating the tracklet embedding with such noisy features can degrade association performance.

To address this issue, a dynamic appearance update technique is adopted from [5], which modifies the smoothing factor α based on the confidence score $s_t \in [0,1]$ of the detection. Specifically, the updated smoothing factor α_{new} is given in Equation 3.11

$$\alpha_{new} = \alpha_{fixed} + (1-\alpha_{fixed})(1 - \frac{s_t - \sigma}{1-\sigma}) \tag{3.11}$$

where α_{fixed} is the default smoothing factor, s_t is the confidence score of the detection at time t, and σ is the minimum confidence threshold. The tracklet embedding update incorporating this dynamic α_{new} is then expressed in Equation 3.12

$$e_i^{t-1} = \alpha e_i^{t-2} + (1-\alpha_{new}) f_i^{t-1} \tag{3.12}$$

In cases where the detection confidence s_t is high (in an ideal scenario, $s_t = 1.0$), the dynamic smoothing factor α_{new} reduces to α_{fixed}, and the EMA behaves as the standard

update rule. Conversely, when the detection confidence s_t approaches the threshold σ, the dynamic smoothing factor α_{new} approaches to 1.0, effectively nullifying the contribution of the current detection embedding f_i^{t-1} to the tracklet embedding.

This dynamic adjustment ensures that unreliable feature embeddings from low-confidence detections have minimal impact on tracklet updates, while high-confidence detections are smoothly integrated. As a result, the tracklet embeddings remain robust, adapting effectively to gradual changes in object appearance over time while mitigating the adverse effects of noisy or inaccurate detections.

Adaptive Appearance cost embeddings and fusion of cost matrix

One of the challenges imposed by DanceTrack [2] and SportsMOT [3] datasets is the similar appearance of individuals due to wearing similar costumes and jerseys. This makes it challenging for Re-ID architectures to create discriminative appearance embeddings for each object.

Given the number of predictions at frame t is L and the number of detections is M. The typical way of fusing different cues together is to create a weighted cost matrix using a combination of spatial (IoU-based) and Re-ID-based cost embeddings. The formula for the same is given in Equation 3.13. Due to similar features, the Re-ID cost embeddings do not show a lot of discriminativeness for appropriate matching, and sometimes inversely affect $C_{final} \in \mathbb{R}^{L \times M}$, minimizing the effect of $C_{spatial} \in \mathbb{R}^{L \times M}$, the cost matrix obtained using spatial cues. Here, we use $C_{spatial}$ as a generalized term for cost matrix calculation using spatial cues such as IoU. In our case, $C_{spatial}$ is calculated using height modulation and extended IoU (HA-EIoU) in the first stage of matching.

$$C_{fused} = \alpha_{spatial} C_{spatial} + \alpha_{reid} C_{reid} \tag{3.13}$$

Inspired by the work in [5], we investigate the use of incorporating adaptive appearance cost embeddings in our hybrid data association setup. The idea is to add a weighing factor w_{reid} to the overall weight α_{reid} to increase the discriminativeness between the appearance based scores.

As defined previously, the number of predictions at frame t is L and the number of detections is M. We obtain the overall cost matrix of $L \times M$, where each element in a row is matched with exactly one element in the columns. Hungarian algorithm is used

to match the predictions and detections in each frame, minimizing the final cost over $-C_{fused}$. $C_{reid}[l, m]$ represents the l_{th} and m_{th} column of the cost matrix and containers the appearance similarity scores between l_{th} prediction and m_{th} detection in the matrix $L \times M$. Let $\mathcal{P}_l$ be the l_{th} tracklet and $\mathcal{D}_m$ be m_{th} detection. If $\mathcal{P}_l$ should be matched with $\mathcal{D}_m$, $C_{reid}[l, m]$ should be minimum across l_{th} row and m_{th} column. For this, we increase the weight $w_{reid}[l, m]$ combination to make a discriminative association. For this, we calculate the difference between the highest and the second-highest association scores across each row and column. The set of Equations: 3.14, 3.15 and 3.16 show adaptive appearance cost embedding updates and consecutively the final fused cost matrix C_{final} that is used for the first stage of association.

$$\begin{aligned} z^{det}_{diff}(C_{reid}, m) &= \min(\max_i C_{reid}[i, m] - \max_{j \neq i} C_{reid}[j, m], \epsilon) \\ z^{track}_{diff}(C_{reid}, l) &= \min(\max_i C_{reid}[l, i] - \max_{j \neq i} C_{reid}[l, j], \epsilon) \end{aligned} \tag{3.14}$$

$$w_{reid}(l, m) = [z^{track}_{diff}(C_{reid}, l) + z^{det}_{diff}(C_{reid}, m)]/2 \tag{3.15}$$

$$\begin{aligned} C^{upd}_{reid}[l, m] &= [\alpha_{reid} + w_{reid}(l, m)]C_{reid}[l, m] \\ C_{final}[l, m] &= C_{spatial}[l, m] + C^{upd}_{reid}[l, m] \end{aligned} \tag{3.16}$$

where:

- $z^{det}_{diff}(C_{reid}, m)$ is a vector showing the difference between highest costs and second-highest costs in C_{reid} for m_{th} detection, ie. m_{th} column of the cost matrix, with respect to all the tracklets.
- $z^{track}_{diff}(C_{reid}, l)$ is a vector showing the difference between highest costs and second-highest costs in C_{reid} for l_{th} tracklet, ie. l_{th} row of the cost matrix, with respect to all the detections.
- $w_{reid}(l, m)$ is the modified weight for each element in C_{reid} computed as the average of z^{track}_{diff} and z^{det}_{diff} for the corresponding element across l_{th} row and m_{th} column.
- $C^{upd}_{reid}[l, m]$ is the updated weight for element [l, m] in the cost matrix $C_{reid} \in \mathbb{R}^{L \times M}$.
- $C_{final}[l, m]$ are the fused costs obtained from modified appearance scores C^{upd}_{reid} and spatial scores $C_{spatial}$ for element $[l, m] \in [L \times M]$.

Diffusion Features for feature extraction - Experimental Study

Re-identification (Re-ID) is commonly employed to extract appearance embeddings from detected bounding boxes and to compute cosine similarity between tracklet embeddings at time $t-1$ and detections at time t. This study explores how the nature of appearance extraction changes when employing a more generic feature extractor compared to a dedicated Re-ID network for generating appearance embeddings.

To this end, we integrate the Diffusion Features (DIFT) network [72], treating Re-ID as a geometric correspondence problem for establishing the cost matrix between tracklets at $t-1$ and detections at t. Diffusion features are derived from the diffusion process, which iteratively refines noise-added inputs back into coherent structures, revealing robust representations during the denoising steps. Specifically, we utilize the Stable Diffusion model [73] to generate images and extract feature maps from its intermediate layers at a particular timestep n during the backward diffusion process. These feature maps are then used to perform spatial correspondence between tracklets and detections. Figure 3.4 showcases how features are first extracted from the whole image, and then are filtered into different bounding box features based on the detection coordinates.

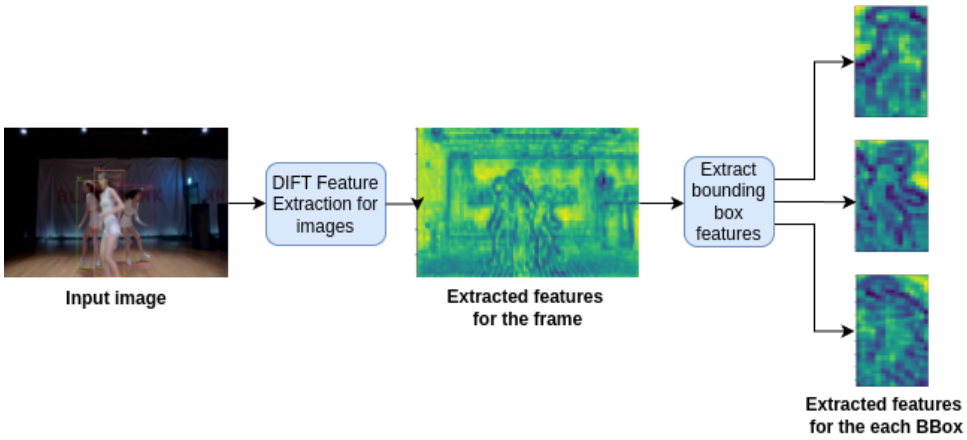


Figure 3.4: Representation of extraction of diffusion-based feature embeddings of bounding boxes for Re-identification task

3.3.4 Virtual Detection Generator

In several sequences from [2], it is observed that dancers often pass by each other, resulting in complete occlusion of one dancer by another. In such scenarios of total occlusion, the

detector—in our case, YOLO-X, fails to generate a detection for the occluded object. This limitation frequently leads to ID switching during tracking. This issue is particularly prominent in situations involving more than two objects undergoing complete overlap. In such cases, two predictions with high spatial association scores ($C_{ha-eiou}$) may correspond to a single detection. Consequently, one of the tracklets, if unmatched in any round of association, is transferred to the list of lost tracklets, leading to a loss of tracking continuity.

To address this challenge, we introduce a virtual tracklet mechanism for objects that exhibit significant overlap during the first round of association. By creating a virtual detection for the occluded objects, we aim to mitigate the occurrence of ID switches and improve tracking consistency. This is done to give a push to trackers to not lose their current associations. However, creating virtual detections and updating them into a tracklet can also create multiple phantom bounding boxes without context, which maybe harmful to the algorithm. Hence, we only take this step after checking if the overlapped tracklet is not matched in any of the prior association stages. We also set the overlap threshold > 0.9 so that unnecessary unmatched tracklets do not get virtual detections. We also set a virtual track existence threshold to $\mathcal{N}_{virtual} = 5$ frames after which we remove the virtual track if it is not matched in the next $\mathcal{N}_{virtual}$ frames. With the detections belonging to the virtual tracks, we reduce the associated detection confidence after each frame from s_t to $0.95s_t$ to minimize its influence on matching after each frame. This is done to minimize the creation of phantom detections.

Algorithm 1 Potential matches Algorithm

Input: $D_t = [B_t, s_t]; D_t \in \mathbb{R}^{1\times5}; B_t \in \mathbb{R}^{1\times4}; s_t \in [0,1]$: YOLO-X Detections at time t

Output: $\mathcal{T}_{potential-match}; \mathcal{T}_{potential-match} \leftarrow \{T[i], B_t^i, s_t^i\}$: Potentially matched tracklets with the associated detection and confidence

$\mathcal{T}_{potential-match} = \phi$

for *col in* $C_{ha-eiou}$ **do**

  matched_indices $\leftarrow \{row \mid C_{ha-eiou}[row][col] < 0.1\}$

  if $len(matched_indices) > 1$ **then**

    for *idx in matched_indices* **do**

      $\mathcal{T}_{potential-match} = \mathcal{T}_{potential-match} \cup \{T[idx], B[col], s[col]\}$

return T_p

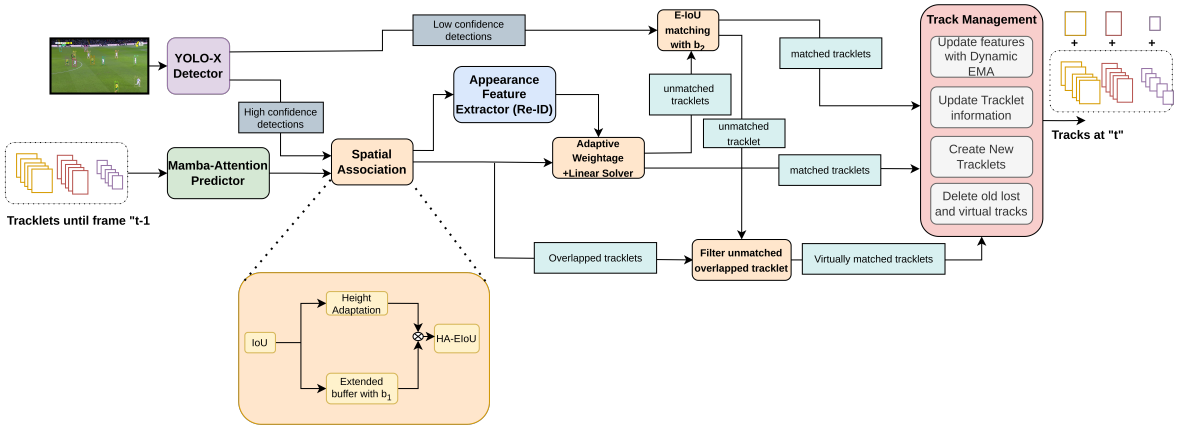


Figure 3.5: Overview of the data association pipeline

Overall Pipeline

We follow the BYTE [26] association strategy and use both low and high confidence detections with modifications to accommodate virtual tracklets as well. The overall pipeline is described in Figure 3.5, with the corresponding algorithms detailed in Alg 2 and Alg 3. The following steps provide an in-depth explanation of the different stages of the association process.

- **First stage of association**: The first stage includes matching tracklets with high confidence detections. This primarily focuses on matching based on spatial association with cost matrix $C_{ha-eiou}$ and updated appearance cost matrix C^{upd}_{reid}. It also includes determining those predictions that are overlapping with a single detection and have a potential match with that detection. The tracklets belonging to these pairs of predictions are defined as $\mathcal{T}_{potential-match}$. To determine these, the difference between the highest and second-highest $C_{ha-eiou}$ scores for each detection is calculated. If the difference between these scores is less than 0.1, the corresponding tracklets are considered potential matches. The algorithm for identifying potential matches is described in Alg 1. In this stage, the matching is performed using the fusion of $C_{ha-eiou}$ and C^{upd}_{reid} according to Equation 3.16. The Hungarian algorithm is then applied to derive matched pairs of predictions and detections ($\mathcal{M}$), unmatched tracklets ($\mathcal{T}_u$), and unmatched detections ($\mathcal{D}_u$).

- **Second stage of association**: In this stage, the unmatched tracklets from the first stage, $\mathcal{T}_u$, are matched to low-confidence bounding boxes, B_{low}, using the buffer b_2.

This process also involves processing the bounding boxes with HA-EIoU. The final results of this stage include matched tracklets ($\mathcal{M}$), unmatched tracklets ($\mathcal{T}_{left}$), and unmatched detections ($\mathcal{D}_{left}$).

- **Matching of virtual detections**: The matching of virtual detections involves checking whether any tracklet from $\mathcal{T}_{potential-match}$, which went untracked in the first stage due to overlapping, is matched in the second stage as outlined in Line 12 of Algorithm 2. If no match is found, the tracklet is updated with the same detection it was previously matched to, but with a lower confidence score, as indicated in Line 13. These updated tracklets are then appended to the list of virtual tracklets, denoted by $\mathcal{T}_{virtual}$.

- **Tracklet Updates**: The process of tracklet updates, detailed in Lines 14 to 22 of Algorithm 3, outlines the steps for updating tracklet information for the next frame. This involves managing lost tracklets, $\mathcal{T}_{lost}$, and virtual tracklets, $\mathcal{T}_{virtual}$, by evaluating their existence against predefined thresholds, $\mathcal{N}_{lost}$ and $\mathcal{N}_{virtual}$.

Algorithm 2 Mamba-Attention Inference at frame t (Part 1)

Input: $D_t = [B_t, s_t]; D_t \in \mathbb{R}^{1\times 5}; B_t \in \mathbb{R}^{1\times 4}; s_t \in [0,1]$: YOLO-X detections at time t
$\mathcal{T}_{t-1} = \{\mathcal{T}_j\}_{j=1}^{t-1}$: Tracklets until frame $t-1$
s_{high}, s_{low} : High confidence threshold, low confidence threshold
Output: T : Updated tracklets at frame t

`/* Detection division using confidence */`

$D_{high} \leftarrow D_t^i$ where $s_t^i > s_{high}$
$D_{low} \leftarrow D_t^i$ where $s_{low} < s_t^i < s_{high}$

`/* Merging Tracklets */`

$\mathcal{T} \leftarrow \mathcal{T}_{active} \cup \mathcal{T}_{lost} \cup \mathcal{T}_{virtual}$

`/* Motion Prediction */`

$P_t \leftarrow$ Mamba-Prediction-Model(T_{t-1}) $\quad P_t \in \mathbb{R}^{L\times 4}$: L predictions

`/* First stage of association */`

$C_{reid} \leftarrow$ Re-ID-Cost$(T_{t-1}^{emb}, D_t^{emb})$; $C_{reid} \in \mathbb{R}^{L\times M}$
$C_{ha-eiou}, \mathcal{T}_{potential-match} \leftarrow$ IoU-distance(P_t, B_{high}, b_1) ; $C_{ha-iou} \in \mathbb{R}^{L\times M}$
$C_{reid}^{upd} \leftarrow$ Adaptive-Weightage(C_{reid}) ; $C_{reid}^{upd} \in \mathbb{R}^{L\times M}$
$C_{final} \leftarrow C_{reid}^{upd} + 1 - C_{ha-eiou}$; $C_{final} \in \mathbb{R}^{L\times M}$
$\mathcal{M}, \mathcal{T}_u, \mathcal{D}_u \leftarrow$ Hungarian(C_{final})

Continue on next page...

Algorithm 3 Mamba-Attention Inference at frame t (Part 2)

/* Filtering the potential tracklets that were matched in Stage 1 */
$\mathcal{T}_{potential-match} \leftarrow \mathcal{M} - \mathcal{T}_{potential-match}$

/* Second Stage of Association */
$C_{ha-eiou} \leftarrow$ IoU-distance$(\mathcal{T}_u, B_{low}, b_2)$
$\mathcal{M}, \mathcal{T}_{left}, \mathcal{D}_{left} \leftarrow$ Hungarian$(C_{ha-eiou})$

/* Filtering Virtual Detections */
if $\mathcal{T}_{potential-match} \in \mathcal{T}_{left}$ **then**
    $\mathcal{T}_{left} \leftarrow \mathcal{T}_{left} - \mathcal{T}_{potential-match}$
    $\mathcal{T}_{virtual} \leftarrow$ Hungarian$(\mathcal{T}_{left}, \mathcal{D}_{common})$
/* Deleting unmatched tracklets from the tracklet list */
$\mathcal{T} \leftarrow \mathcal{T} - \mathcal{T}_{left}$

/* Initializing new tracklets from unmatched detections */
for $d \in \mathcal{D}_{left}$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} \cup d$
/* Removing lost tracks that exceed the lost time threshold */
for $t \in \mathcal{T}_{lost}$ **do**
    if $len(t) > \mathcal{N}_{lost}$ **then**
        $\mathcal{T}_{lost} \leftarrow \mathcal{T}_{lost} - t$
/* Removing virtual tracks that exceed the virtual existence threshold */
for $t \in \mathcal{T}_{virtual}$ **do**
    if $len(t) > \mathcal{N}_{virtual}$ **then**
        $\mathcal{T}_{virtual} \leftarrow \mathcal{T}_{virtual} - t$
return $\mathcal{T}$

3.4 Chapter Summary

We propose a learning-based motion model for the prediction task within the TBD paradigm. This motion model combines Mamba [1] with self-attention layers [58], and has been trained on the DanceTrack [2] and SportsMOT [3] datasets using data augmentation techniques. To ensure efficient convergence, we use a combination of CIoU and Smooth L1 loss functions.

In addition to the motion model, we conduct an extensive analysis of data association strategies for matching tracklets across frames. These strategies include modifying the spatial cost matrix with techniques such as height adaptation and expanding the bounding boxes by adding a buffer zone. To improve appearance-based association, we integrate dynamic EMA to enhance the feature bank of appearance embeddings. We further refine the appearance cost matrix by boosting the discriminative power of appearance embeddings, incorporating adaptive appearance embeddings. An experimental evaluation is also performed to explore replacing the default SBS-50 [29] Re-ID network used in other methods with a diffusion-based network for feature extraction. To address the issue of ID switching, we introduce a virtual detection generator that creates virtual tracks for occluded tracklets that overlap completely with others. This helps maintain accurate identity tracking even during occlusions.

In conclusion, we propose a novel TBD pipeline that integrates our learning-based motion model with advanced data association techniques to enhance tracking performance.

Chapter 4

Experiments and Results

4.1 Implementation Details

4.1.1 Motion Predictor Training Details

Dataloader Creation:

Data augmentation techniques, as described in Section 3.2.2, were applied to 30% of the training data to enhance the diversity of the dataset. This resulted in a total of $\approx$ 7050 tracklet sequences for DanceTrack and $\approx$ 6200 tracklet sequences for SportsMOT. The maximum window size w was set to 15 for DanceTrack and 10 for SportsMOT. Each tracklet sequence was randomly selected to have a length between 2 and w necessitating padding to w to ensure uniform sequence lengths for batch processing during training. Among the three data augmentation techniques employed, each had a 50% probability of being applied, with the constraint that not all techniques were applied simultaneously to a single tracklet.

Training Details:

The model was trained using the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, a learning rate of 0.0001, and a weight decay of 10^{-3} with a batch size of 64. A custom warmup scheduler, implemented with details consistent with [33], was employed during training. All experiments were conducted on a single NVIDIA RTX 4090 GPU with 24 GB of RAM, ensuring sufficient computational resources for efficient model training. Each model

was trained for 60 epochs, while saving only the best weights with minimum loss during validation. The number of blocks for the Mamba-Attention-Module $M = 4$ for both DanceTrack and SportsMOT, with the input token embedding dimension $d_m = 128$.

4.1.2 Data Association Details

Object Detection Details:

For a fair comparison with state-of-the-art methods [27, 69, 5, 26, 7] on the same benchmarks [2, 3], we adopt YOLO-X [14] as the off-the-shelf detector. YOLO-X, pre-trained on the COCO [74] dataset, is fine-tuned on the DanceTrack and SportsMOT benchmarks to ensure optimal performance. Detection weights for DanceTrack were sourced from the [7] repository, while those for SportsMOT were obtained from [27]. During post-processing in the detection stage, a confidence threshold of 0.1 was applied for detections, and a non-maximum suppression (NMS) threshold of 0.7 was used to handle redundant and overlapping detections effectively.

Matching Details

The buffer sizes $(b_1, b_2) = (0.4, 0.3)$ were used for SportsMOT and $(b_1, b_2) = (0.4, 0.25)$ for DanceTrack for HA-EIoU matching during first and second stages of association. IoU matching threshold of 0.5 was used in the second stage of association. Lost tracks buffer $\mathcal{N}_{lost}$ was kept at 30 frames. This means that if any tracklet is in the list of lost tracklets, it would be removed after 30 frames of being lost.

4.2 Evaluation Metrics

The evaluation of MOT systems is conducted using three principal metrics: the CLEAR MOT [75] metrics, Identification F1 Score (IDF1) [76], and Higher Order Tracking Accuracy (HOTA) [77]. CLEAR MOT and HOTA encompass several sub-metrics that provide comprehensive evaluations of detection and association performance.

4.2.1 CLEAR MOT

CLEAR stands for "Classification of Events, Activities, and Relationships" [75]. It includes two primary metrics: Multi-Object Tracking Accuracy (MOTA) and Multi-Object

Tracking Precision (MOTP).

Multi-Object Tracking Accuracy (MOTA) In MOTA, evaluation is done based on the number of false negatives (FN), false positives (FP), and identity switches that are accumulated during the processing of the sequences. For detection, it calculates True Positives (TPs), False Positives (FPs), and False Negatives (FNs) based on IoU similarity between the ground truth bounding boxes, and the detected bounding boxes coming from the YOLO-X detector for which a minimum IoU matching threshold is defined. For association, it calculates the number of identity switches (IDSWs) happening during the matching stage. IDSW only measures the switching between the previous frame TP to the current frame ID, rather than accumulating the switch in the frames going forward. MOTA is defined in equation 4.1.

$$\text{MOTA} = 1 - \frac{\text{FN} + \text{FP} + \text{IDSW}}{\text{GT}_{BBox}} \tag{4.1}$$

Multi-Object Tracking Precision (MOTP) MOTP is a metric that focuses on the spatial alignment between detected and ground truth bounding boxes. It measures the average similarity (e.g., IoU) across all true positive detections, providing an indication of the localization accuracy of the detected objects. The formula for MOTP is expressed in equation 4.2

$$\text{MOTP} = \frac{1}{\text{TP}} \sum_{\text{TP}} \text{S} \tag{4.2}$$

4.2.2 Identification F1 (IDF1) Score

IDF1 [76] calculates the consistency of ID assignments over a sequence. It measures how well the IDs are assigned using the Hungarian Algorithm to match with the ground truth object IDs by computing the harmonic mean between the precision and recall. Precision (Pr) measures the proportion of correctly matched identities out of all predicted IDs, while recall measures the proportion of correctly matched IDs out of all GT IDs. The set of equations for IDF1 evaluation is defined in 4.3.

$$\begin{aligned} \text{ID-Recall} &= \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}} \\ \text{ID-Precision} &= \frac{\text{IDTP}}{\text{IDTP} + \text{IDFP}} \end{aligned} \tag{4.3}$$

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFN} + \text{IDFP}}$$

4.2.3 Higher Order Tracking Accuracy (HOTA)

HOTA metric provides a unified framework to jointly evaluate both detection and association quality in MOT benchmarks unlike CLEAR MOT and IDF1. It introduces a balanced and interpretable way to measure tracking performance by considering the higher-order temporal dynamics of tracking systems. It brings new terms such as True Positive Associations (TPAs), False Positive Associations (FPAs), and False Negative Associations (FNAs) for each True Positive Detection. More details about TPA, FPA, and FNAs should be referred from the original paper [77]. With the actual TP, FP, FN, as described in 4.2.1, and the newly defined TPA, FPA, and FNA, equation 4.4 outlines the derivation of the HOTA evaluation metric.

$$\begin{aligned} \text{A(c)} &= \frac{\text{TPA(c)}}{\text{TPA(c)} + \text{FNA(c)} + \text{FPA(c)}} \\ \text{DetA}_\alpha &= \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \\ \text{AssA}_\alpha &= \frac{1}{\text{TP}} \sum_{\text{c} \in \text{TP}} \text{A(c)} \\ \text{HOTA}_\alpha &= \sqrt{\text{DetA}_\alpha * \text{AssA}_\alpha} \\ \text{HOTA} &= \int_0^1 \text{HOTA}_\alpha d\alpha \sim \frac{1}{19} \Sigma_{\alpha \in \{0.05, 0.1, \ldots, 0.9, 0.95\}} \text{HOTA}_\alpha \end{aligned} \tag{4.4}$$

Here, α defines different IoU similarity score thresholds. $A(c)$ refers to the association score for a particular set of True Positives. DetA_α and AssA_α are Detection Accuracy and Association Accuracy over α.

4.3 Benchmark Results

We compare our methods with filter-based motion models and learning-based motion models for both DanceTrack [2] and SportsMOT [3] dataset. All the results in Table 4.1 and Table 4.2 are the best results mentioned in the methods with the combination of motion models as well as data association strategies.

4.3.1 DanceTrack Results

- **Test set results**:

  Table 4.1 shows the test set results for several recent benchmarks for DanceTrack. The proposed method performs consistently well in all the metrics demonstrating promising results with **63.16** HOTA, **64.9** IDF1, **48.7**, AssA, **82.12** DetA, and **92.23** MOTA score with the use of a combination of height adaptation and extended IoU as well as Re-ID features for data association. It competes with previous learning-based DiffMOT [27] outperforming it in key metrics such as HOTA by 0.86, IDF1 by 1.9, and AssA by 1.5. It also outperforms recent Mamba-based motion models [33, 39, 69] by $\approx$ 5% HOTA, $\approx$ 7-10% in IDF1 and AssA.

- **Validation set results**:

  Figure 4.1 presents a line plot illustrating the performance of DanceTrack across all evaluation metrics for validation set sequences. The plot highlights that sequences 19, 41, and 63 exhibit lower performance, achieving HOTA scores between 35% and 45% and association accuracy (AssA) ranging from 20% to 30%. These sequences pose significant challenges, including high occlusion and a dynamic number of individuals entering and exiting the frame. The examples of sequence 19 and 41 are shown in Figure 4.2.

4.3.2 SportsMOT Results

- **Test set Results**:

  Table 4.2 shows the test set results for SportsMOT and their comparisons with recent methods. Like other methods [27], we use the detector model that is trained on both training and validation data. We achieve the best performance across all learning-based methods, outperforming DiffMOT [27] by 1.06 HOTA, 1.6 IDF1 and 1.7 AssA.

Method	**HOTA↑**	**IDF1↑**	**AssA↑**	**DetA↑**	**MOTA↑**
Filter-based					
FairMOT [36]	39.7	40.8	23.8	66.7	82.2
CenterTrack [38]	41.8	35.7	22.6	78.1	86.8
ByteTrack [26]	47.3	52.5	31.4	71.6	89.5
OC-SORT [7]	54.6	54.6	40.2	80.4	89.6
Deep OC-SORT [5]	61.3	61.5	45.8	**82.2**	92.3
C-BIoU Tracker [64]	60.6	61.6	45.4	81.3	91.6
MoveSORT-NODEFilter [78]	56.1	56.0	38.7	81.6	91.8
StrongSORT++ [29]	55.6	55.2	38.6	80.7	91.1
DeepMoveSORT-TransFilter [79]	**63.0**	**65.0**	**48.6**	**82.2**	**92.5**
Learning-based					
QDTrack [44]	45.7	44.8	29.2	72.1	83.0
DiffusionTrack [40]	52.4	47.5	33.5	82.2	89.3
ETTrack [80]	56.4	57.5	39.1	81.7	92.2
ByteSSM [69]	57.7	57.5	41.0	81.5	92.2
DiffMOT [27]	62.3	63.0	47.2	**82.5**	**92.8**
MotionTrack [41]	58.2	58.6	41.7	81.4	91.3
MambaTrack [33]	56.8	57.8	39.8	80.1	90.1
MambaTrack+ [39]	56.1	54.9	39.0	80.8	90.3
Ours	**63.16**	**64.9**	**48.7**	82.12	92.23

Table 4.1: Tracking results for DanceTrack test set. ↑ indicates that higher values are better. Best results are in **bold** each for filter-based methods and learning-based methods

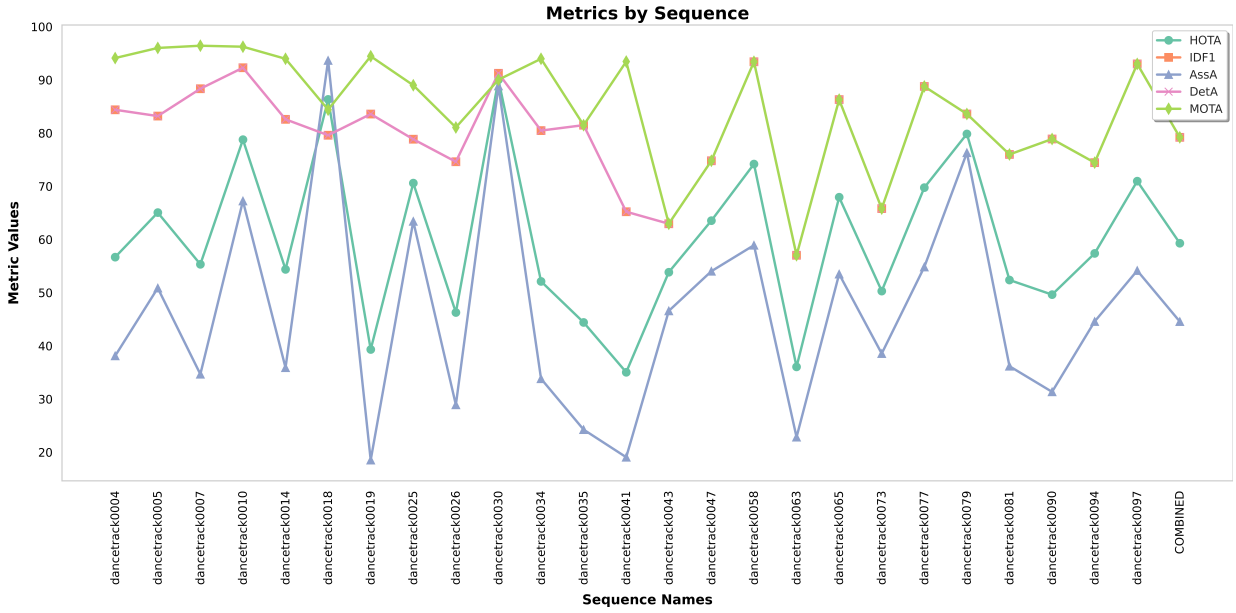


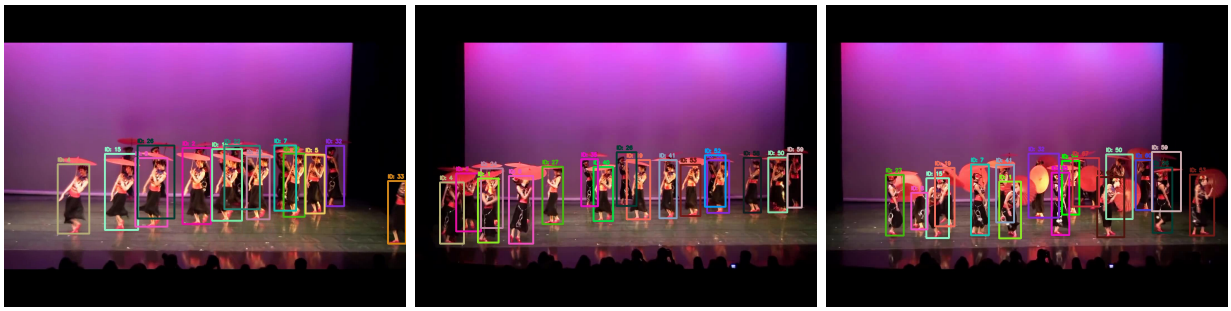
Figure 4.1: Results for the evaluation metrics for DanceTrack across all the sequences in the validation dataset. Higher values in metrics are better.

Overall, our pipeline outperforms all the methods including filter-based and learning-based motion models except [4].

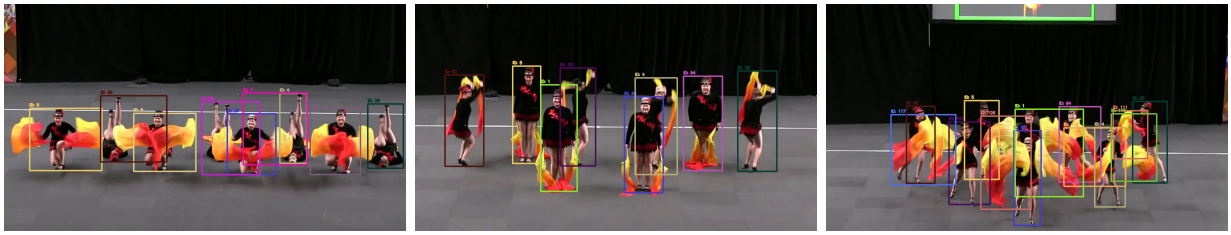
- **Validation set Results**: Figure 4.3 shows the line plot showing the performance of the proposed method on SportsMOT validation dataset across all the sequences. From the figure, we can see that the object detection in this dataset performs consistently well achieving $> 90\%$ DetA across all the sequences. The most challenging sequence in the SportsMOT is "v_i2_L4qquVg0_c010", ie. one of the sequences of soccer that has AssA of $< 50\%$. The sequence shows a fast camera motion, creating a lot of motion blur and multiple switched IDs over the duration of a few seconds. An example of this sequence is shown in Figure 4.4.

4.4 Ablation Study and Experimental Analysis

This section presents an ablation study to analyze the impact of key hyperparameters, training techniques, and data association components on the performance of the proposed model. Parameters to be focused on are as follows:



(a) dancetrack0041: Frame 35 (b) dancetrack0041: Frame 66 (c) dancetrack0041: Frame 84



(d) dancetrack0063: Frame 34 (e) dancetrack0063: Frame 44 (f) dancetrack0063: Frame 57

Figure 4.2: Qualitative results of DanceTrack validation set on Sequences "dancetrack0041" and "dancetrack0063" exhibiting lower tracking performance. Best realized when zoomed in.

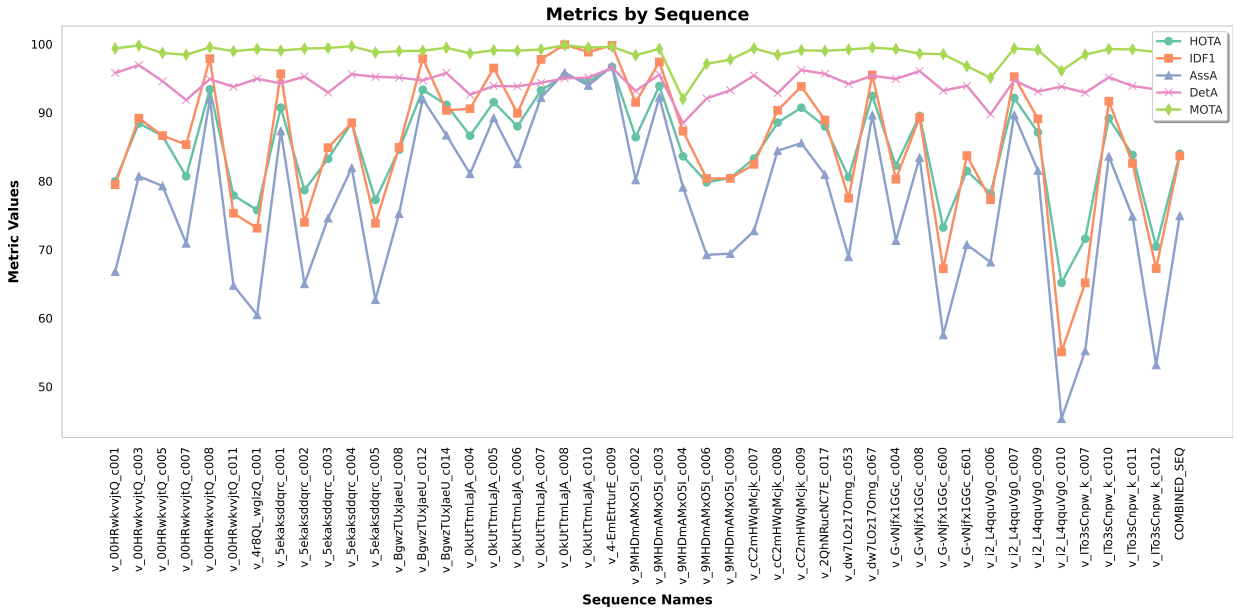
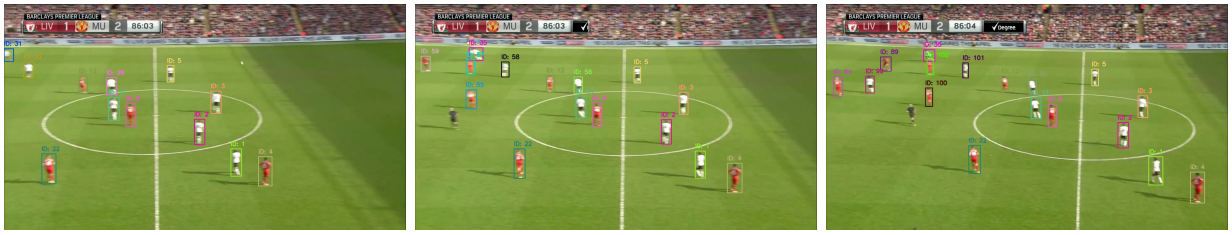


Figure 4.3: Results for the evaluation metrics for SportsMOT across all the sequences in the validation dataset. Higher values in metrics are better.

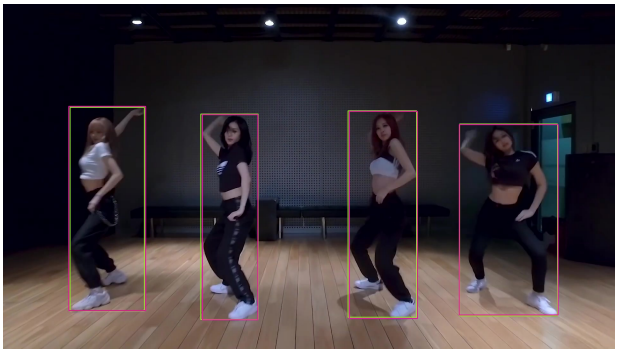


(a) Frame 79 (b) Frame 86 (c) Frame 94

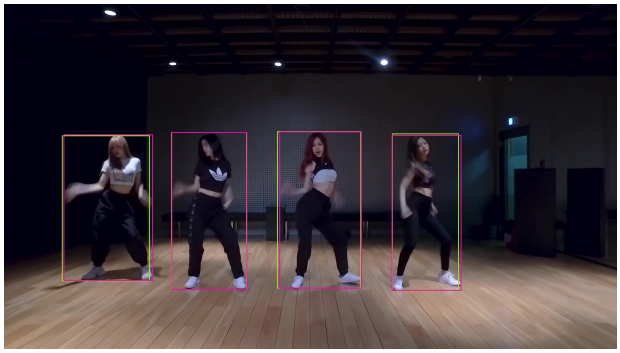
Figure 4.4: Qualitative results of SportsMOT sequence "v_i2_L4qquVg0_c010" exhibiting lower tracking performance due to rapid motion. Best realized when zoomed in.

Method	**HOTA↑**	**IDF1↑**	**AssA↑**	**MOTA↑**	**DetA↑**
Filter-based					
FairMOT [36]	49.3	53.5	34.7	86.4	70.2
CenterTrack [38]	62.7	60.0	48.0	90.8	82.1
ByteTrack [26]	62.8	69.8	51.2	94.1	77.1
*ByteTrack [26]	64.1	71.4	52.3	95.9	78.5
BoT-SORT [28]	68.7	70.0	55.9	94.5	84.4
OC-SORT [7]	71.9	72.2	59.8	94.5	86.4
*OC-SORT [7]	73.7	74.0	61.5	**96.5**	**88.5**
Deep-EIoU [4]	**77.2**	**79.8**	**67.7**	96.3	88.2
Learning-based					
QDTrack [44]	60.4	62.3	47.2	90.1	77.5
GTR [81]	54.5	55.8	45.9	67.9	64.8
TransTrack [48]	68.9	71.5	57.5	92.6	82.7
*MixSort-Byte [3]	65.7	74.1	54.8	96.2	78.8
*MixSort-OC [3]	74.1	74.4	62.0	96.5	88.5
MotionTrack [41]	74.0	74.0	61.7	96.6	88.8
DiffMOT [27]	72.1	72.8	60.5	94.5	86.0
MambaTrack+ [39]	71.3	71.1	58.6	94.9	86.7
MambaTrack [33]	72.6	72.8	60.3	95.3	87.6
*DiffMOT [27]	76.2	76.1	65.1	**97.1**	89.3
*ByteSSM [69]	74.4	74.5	62.4	96.8	88.8
Ours	**77.26**	**77.7**	**66.8**	96.9	**89.45**

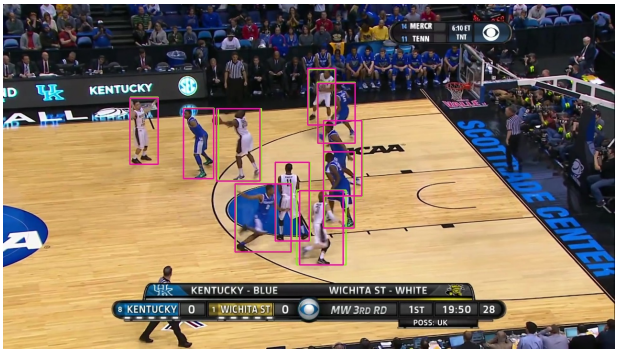
Table 4.2: Tracking results for SportsMOT test set. The results in * use both training and validation data to train the object detector. ↑ indicates that higher values are better. Best results are in **bold** each for filter-based methods and learning-based methods



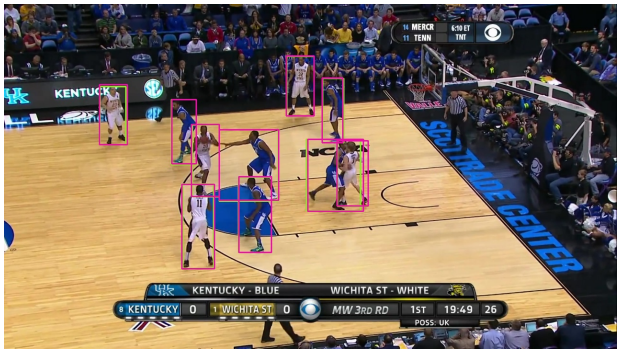
(a) DanceTrack: Frame 283



(b) DanceTrack: Frame 354

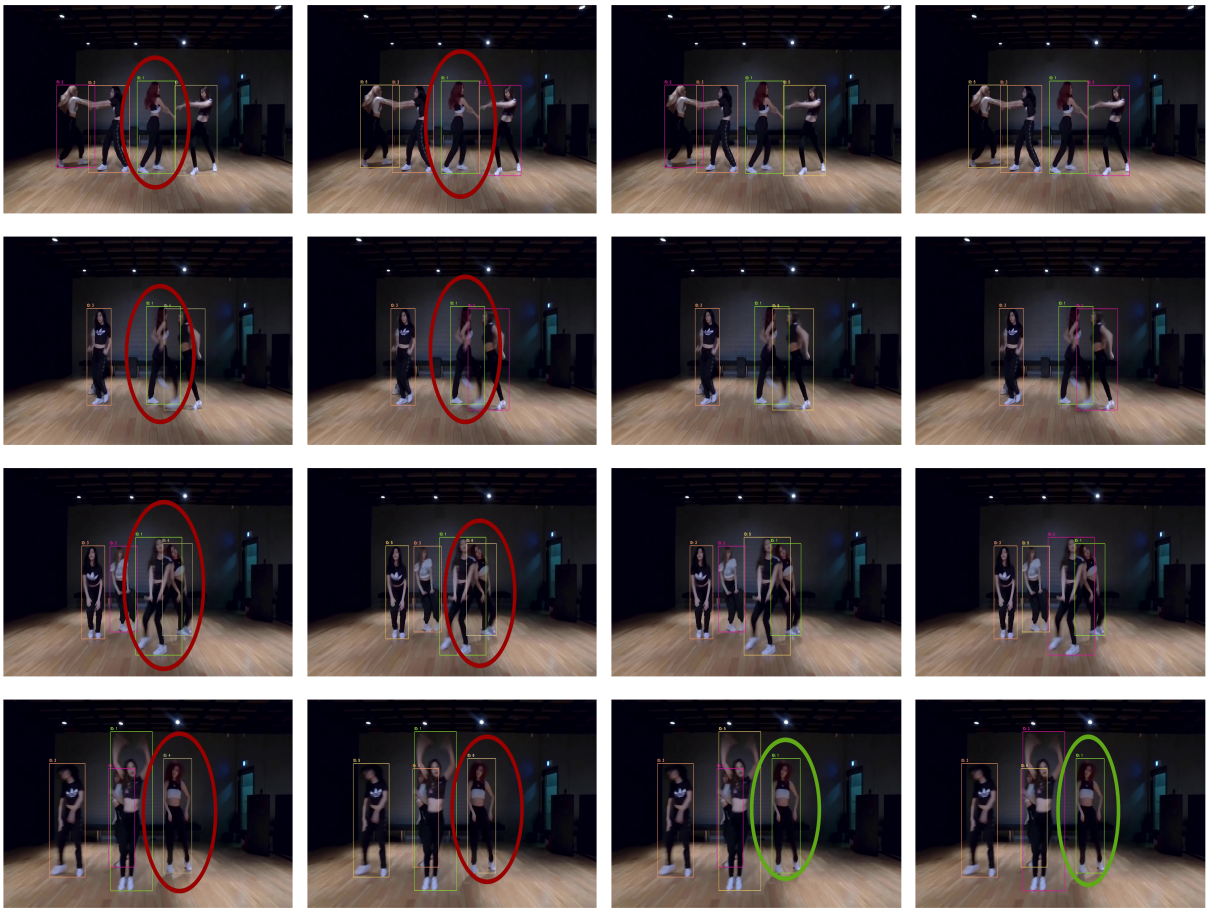


(c) SportsMOT: Frame 236



(d) SportsMOT: Frame 285

Figure 4.5: Examples of motion prediction results on sequences of Danceterack (dancetrack0005) and SportsMOT (v_00HRwkvvjtQ_c001) datasets. Bounding boxes in magenta represent detections from YOLO-X model and bounding boxes in green represent the predictions from the motion prediction. Best realized when zoomed in.



(a) Deep OC-SORT (b) DiffMOT (c) TrackSSM (d) Ours

Figure 4.6: Qualitative Results for DanceTrack validation set for sequence "dancetrack0005" on different methods. Sequences marked with red circle show ID switching. In (a) and (b), ID 5 is switched with ID 1 at the last frame due to individuals crossing each other. While in (c)TrackSSM and (d)Ours, the green circle shows that ID 1 and ID 5 stay consistent on the individuals after overlap and occlusion. Best realized when zoomed in.



(a) DiffMOT (b) TrackSSM (c) Ours

Figure 4.7: Qualitative results on different methods on basketball sequence from SportsMOT. Sequences marked with red circle show ID switching. ID 7 marked in red circles goes through complete occlusion in the second frame shown in each sequence. While (a) and (b) change the ID no. from 7 to 11 in the last shown frame, (c) Ours in green circle showcases the retention of track ID after the overlap. Best realized when zoomed in.

- Choice of Mamba-Attention Blocks M and window size w
- Effect of data augmentation
- Effect of height adaptation in association cost matrix calculation
- Choice of buffer sizes b_1 and b_2 for Extended IoU
- Comparison of different Re-ID techniques and effect of the presence of Virtual Tracklet component

For all experiments, training hyperparameters such as the number of epochs, learning rate, and optimization parameters remain unchanged to isolate the effects of the variables under study.

4.4.1 Prediction Ablation

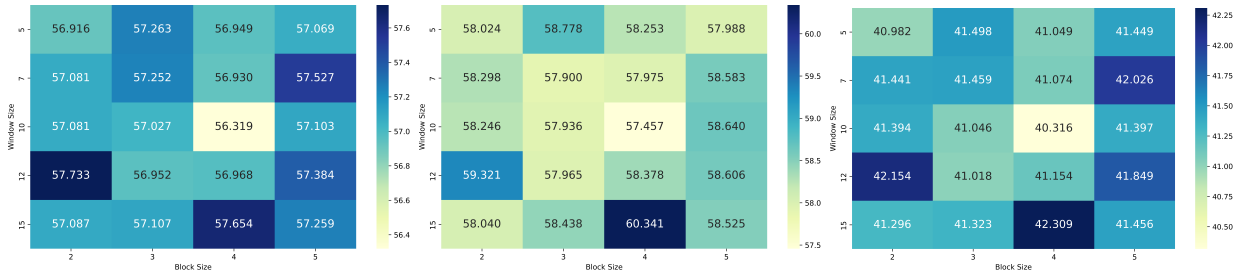
Maximum Window Size and Number of Blocks

To investigate the impact of varying the number of Mamba-Attention blocks M and the maximum window size w for random sequence length selection, we conduct experiments with $M \in \{2, 3, 4, 5\}$ and $w \in \{5, 7, 10, 12, 15\}$. During this study, data augmentation was enabled, and the number of attention heads was set to 4. We do this experimentation with standard IoU, ie. we do not incorporate height adaption or buffer sizes b_1, b_2 for updating the cost matrix.

Figure 4.8 and Figure 4.9 show the heatmaps for DanceTrack and SportsMOT respectively for key metrics HOTA, IDF1 and AssA. The results indicate that the model performs consistently well with $M = 4$ and $M = 5$ for SportsMOT, and $M = 5$ for DanceTrack. It is also seen that increasing w also improved the association, which should be the cause of better prediction. DanceTrack achieves its best average performance with $(M = 4, w = 15)$ with 57.65 HOTA, 60.34 IDF1 and 42.8 AssA. SportsMOT performs the best with $(M = 4, w = 10)$ with 82.93 HOTA, 82.44 IDF1, and 73.14 AssA.

Effect of Data Augmentation

To evaluate the impact of data augmentation, we train the model under consistent conditions, setting the number of attention heads to 4. The absence of data augmentation



(a) HOTA (b) IDF1 (c) AssA

Figure 4.8: Heatmaps for ablation study of window size w and Mamba-Attention blocks M (a), (b), and (c) represent HOTA, IDF1 and AssA metrics for DanceTrack



(a) HOTA (b) IDF1 (c) AssA

Figure 4.9: Heatmaps for ablation study of window size w and Mamba-Attention blocks M. (a), (b), and (c) represent HOTA, IDF1, and AssA metrics for SportsMOT

is analyzed in two scenarios. The first scenario excludes techniques such as spatial jitter, random scaling, noise injection, and random translations while still training the model with variable sequence lengths between $(2, w)$. The second scenario removes all augmentation techniques entirely, training the model with combinations of window size and block size $(M = 4, w = 10)$ for SportsMOT and $(M = 4, w = 15)$ for DanceTrack. We validate these results without using height adaptation or extended buffer sizes, ie. using standard IoU for data association.

- **Impact of constant sequence lengths:** Training the model with constant sequence lengths significantly degrades the performance of the model. On SportsMOT, this results in only 6.82 HOTA and 0.96 IDF1, while on DanceTrack, the scores drop to 20.345 HOTA and 9.97 IDF1. This substantial decline is expected as it highlights the importance of variable sequence lengths in adapting to the dynamic nature of data association in an online setting.
- **Impact of reduced spatial augmentation:** In this experiment, the model is trained with variable sequence lengths, but spatial jitter (used to introduce noise in bounding boxes and simulate detection imperfections) is excluded, removing 30% of the augmented data. On DanceTrack, this leads to a decline of 0.34 HOTA and 1.19 IDF1. On SportsMOT, the performance decreases by 0.96 HOTA and 0.05 IDF1. These results demonstrate that spatial jitter contributes to improved track association, though its impact is less significant compared to other parameters, such as block size M and the maximum window size w when performing matching with standard IoU. The results for data augmentation for different scenarios are in Table 4.3.

4.4.2 Association Ablation

For the association phase, the focus here is on analyzing the effect of varying buffer sizes for EIoU during the first and second stages of matching. Specifically, we experiment with buffer sizes b_1 and b_2 in the range $[0.25, 0.3, 0.35, 0.4, 0.45]$. We also perform ablation on the presence of different components in the association stage, such as height adaptation, adaptive weightage, and virtual detections.

Dataset	Data Augmentation	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
DanceTrack	No Augmentation	20.345	**79.796**	5.2364	34.709	9.9651
	Random Sequences only	56.311	79.432	**42.371**	90.048	59.512
	All Augmentations	**57.654**	78.873	42.309	**90.075**	**60.341**
SportsMOT	No Augmentation	6.820	**94.370**	0.49593	46.825	0.96234
	Random Sequences only	81.970	94.153	73.137	**98.530**	82.378
	All Augmentations	**82.934**	94.067	**73.141**	98.511	**82.437**

Table 4.3: Performance comparison of different Data Augmentation strategies for SportsMOT and DanceTrack datasets. ↑ indicates that higher values are better. Best results are in **bold**.

Effect of Height Adaptation in IoU

In this study, we evaluate the impact of height adaptation in the IoU-based cost association step during the first and second stages of data association. The prediction model configurations are set as $(M = 4, w = 10)$ for SportsMOT and $(M = 4, w = 15)$ for DanceTrack, with data augmentation enabled. No additional buffers are applied in this configuration.

Incorporating height adaptation into the IoU cost matrix demonstrates a noticeable improvement in data association performance. As shown in Table 4.4, this modification leads to measurable gains across both datasets. For DanceTrack, HOTA increases by 0.63, and IDF1 improves by 0.11. Similarly, for SportsMOT, HOTA rises by 0.49, while IDF1 achieves an increase of 0.6. These results underscore the effectiveness of height adaptation in enhancing the robustness of IoU-based association techniques. During additional study, it is also seen that incorporating height adaptation further increases the difference between the association costs between overlapping bounding boxes. This further ensures its effectiveness and its ability to reduce ID switching and ensuring correct association.

Effect of varying buffer sizes

- **SportsMOT**: For SportsMOT, we choose $w = 10, M = 4$, and vary the buffer $b1 \in \{0.25, 0.3, 0.35, 0.4, 0.45\}$, and $b2 \in \{0.25, 0.3, 0.35, 0.4\}$. For this ablation study, the results show that the configuration of $b_1 = 0.4$, $b_2 = 0.3$ yield the highest HOTA

Dataset	Height Adaptation	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
DanceTrack	✗	57.605	78.924	42.21	90.079	59.47
	✓	**58.235**	**79.6**	**42.762**	**90.147**	**59.585**
SportsMOT	✗	82.933	94.073	73.134	**98.499**	82.429
	✓	**83.421**	**94.113**	**73.964**	98.466	**83.024**

Table 4.4: Performance comparison of effect of Height Adaptation for SportsMOT and DanceTrack datasets. ↑ indicates that higher values are better. Best results are in **bold**.

(83.974), IDF1 (83.677) on the validation set, indicating optimum balance for association. We also see that while all the values in HOTA are above 83, the least values are with $b_1 = 0.25$ and $b_2 = 0.45$ as compared to other buffer values.

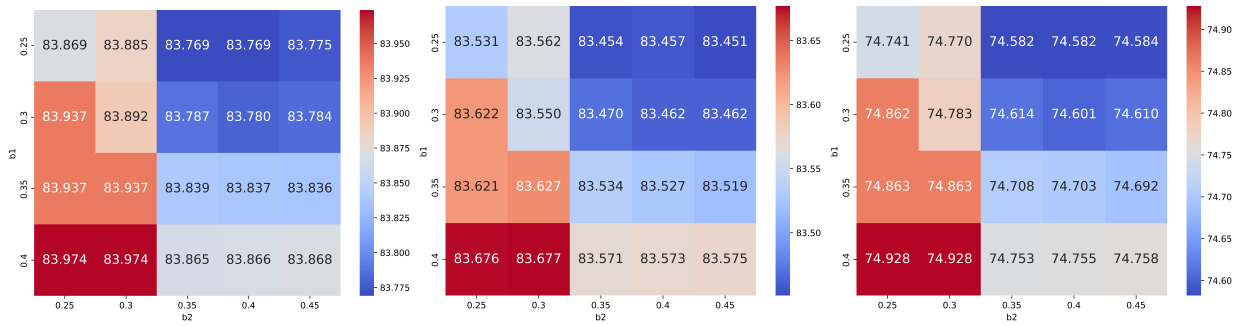
- **DanceTrack**: For DanceTrack, we choose the maximum window $w = 15, M = 4$ since it performs the best as shown in the previous experiment while varying the buffer sizes in the same way as SportsMOT. We see that the data association gives consistently high values with $(b_1, b_2) = (0.4, 0.25)$, achieving the best performance with 59.299 HOTA, 62.226 IDF1 score, and 44.465 AssA.

  The trend in both the datasets shows that having lower buffer values for first stage of association with $b_1 = 0.25$ does not perform well with any combination of b_2.

Figure 4.10 and Figure 4.11 showcase the variations in results with changing buffer sizes in SportsMOT and DanceTrack datasets.

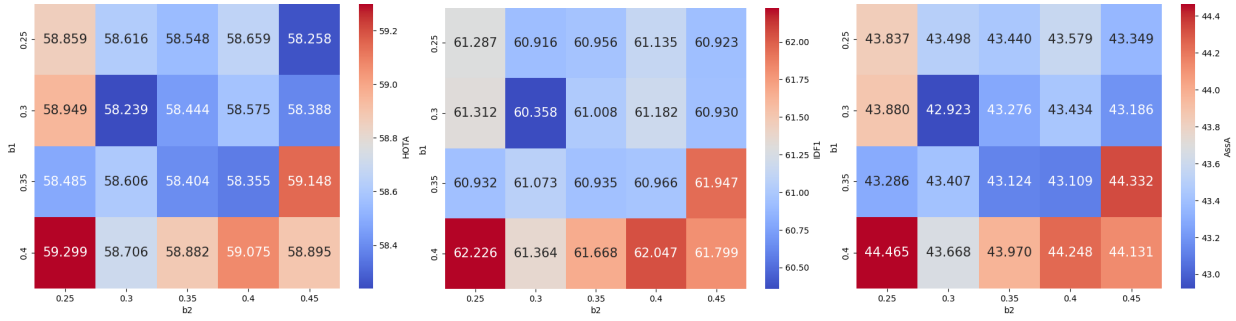
Comparison of Re-ID methods

By default, we perform our experiments with SBS-50 Re-ID model [16] for appearance feature extraction in first stage of association. In this section we explore the use of Diffusion Features network [72] as a replacement for extracting appearance embedding. For this, we use stable diffusion feature extraction network, with $(t = 301, n = 2)$. Here t is the diffusion timestep. and n is the block index as described in the original method [72]. Table 4.5 shows that the default Re-ID method [29] performs much better than features extracted from the diffusion-based method.



(a) HOTA (b) IDF1 (c) AssA

Figure 4.10: Heatmaps demonstrating the effect of extended buffer sizes b_1 and b_2 for SportsMOT dataset



(a) HOTA (b) IDF1 (c) AssA

Figure 4.11: Heatmaps demonstrating the effect of extended buffer sizes b_1 and b_2 for DanceTrack dataset

Dataset	Re-ID technique	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
DanceTrack	Diffusion Features [72]	56.414	78.847	40.520	90.159	58.296
	SBS-50 [29]	**59.299**	**79.198**	**44.101**	**90.210**	**61.820**
SportsMOT	Diffusion Features [72]	83.703	93.997	74.557	98.655	83.506
	SBS-50 [29]	**83.974**	**94.137**	**74.923**	**98.684**	**83.677**

Table 4.5: Performance comparison of different appearance extraction models for SportsMOT and DanceTrack datasets. ↑ indicates that higher values are better. Best results are in **bold**.

Effect of Virtual Tracklets

In this section, we analyze the impact of incorporating the virtual tracklet generation method on the performance of data association. Table 4.6 presents the results of three different combinations of these methods for each dataset. For this analysis, height adaptation remains activated, with buffer values b_1 and b_2 retained from previous experiments to ensure consistency. The results show that with the given combination, virtual tracklets slightly improve the performance of SportsMOT [3] by enhancing IDF1 and AssA scores, but have minimal effect on DetA and MOTA scores. For DanceTrack [2], virtual tracklets show small improvements in IDF1 and AssA but slightly decrease HOTA. Combining adaptive appearance embeddings with virtual tracklets does not offer significant benefits but may cause slight degradation, particularly in DanceTrack.

Dataset	Adaptive Appearance Embeddings	Virtual Tracklets	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
SportsMOT	✓	✗	83.974	94.137	74.923	98.684	83.677
	✗	✓	84.219	94.165	75.343	98.688	83.902
	✓	✓	83.974	94.137	74.928	98.685	83.676
DanceTrack	✓	✗	59.299	79.198	44.101	90.210	61.820
	✗	✓	59.586	78.873	45.179	90.207	62.479
	✓	✓	58.985	79.197	44.100	90.203	61.820

Table 4.6: Effect of creating virtual tracklets on DanceTrack and SportsMOT validation datasets. ↑ indicates that higher values are better.

4.5 Chapter Summary

This chapter focuses on the implementation and experimental analysis of various components in motion modeling and data association strategies. Spatial modifications, such as height adaptation and extended bounding box buffers, significantly improved the performance of the proposed method across both datasets. We also examined the impact of adaptive appearance embeddings and virtual tracklet generation. The results show that combining spatial and appearance cues effectively reduces the need for virtual tracklets by ensuring overlapping bounding boxes are matched early in the association process, minimizing lost tracklets. The absence of adaptive appearance embeddings, however, increases the reliance on virtual tracklet generation with improved data association performance on validation sets. We also explored using a general feature extraction method with diffusion features [72]. The results indicate that a dedicated Re-ID network [29] performs better than a generic feature extractor for appearance embeddings and data association.

In conclusion, this chapter outlines a comprehensive set of experiments to identify the best configuration for motion modeling and data association strategies to improve tracking performance. The results demonstrate that incorporating height adaptation and an extended buffer during the spatial cost matrix update stage yields the most effective improvements.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis, we presented a learning-based motion model that integrates the Mamba architecture with self-attention layers to enhance motion prediction within the Tracking-By-Detection (TBD) paradigm. Additionally, we conducted a comprehensive investigation into various matching cues and their combinations for data association in Multi-Object Tracking (MOT). These components include Height Adaptation and Extended IoU (EIoU) for improved spatial matching, dynamic Exponential Moving Average (EMA) updates for more robust feature-based matching, and the introduction of a virtual detection stage to mitigate ID switching and handle overlapping predictions effectively.

Experimental results highlight the effectiveness of the proposed pipeline in delivering accurate motion prediction and reducing ID switches, particularly in challenging scenarios involving overlaps and partial occlusions. Benchmarking the motion model alongside different data association components reveals that in the TBD paradigm, precise motion modeling and robust data association strategies are essential and complementary for building high-performing trackers. We see that updating the appearance feature bank using dynamic EMA method significantly enhances the performance of the tracker. Incorporating the combination of weaker cues such as height adaptation and E-IoU in place of standard IoU also helps in improved cost association matrix during the first stage of association. Particularly, we see that it helps in ensuring higher cost differences between overlapping predictions, making a cascading benefit to the final association cost matrix, thereby, mitigating the need for virtual detections as the third stage of association in several sequences.

Our approach achieves state-of-the-art performance on the challenging DanceTrack and SportsMOT datasets, outperforming several heuristic-based and learning-based motion models while maintaining real-time inference speeds of ≈ 30 FPS on RTX 4090.

5.2 Limitations and Future Work

Certain sequences in the DanceTrack dataset, such as Seq 41, Seq 63, and Seq 35, present particularly challenging real-world scenarios. Seq 41 involves multiple individuals entering and exiting the stage, testing the efficacy of data association strategies and the robustness of the appearance feature bank mechanism. Seq 63 features individuals dancing with props, highlighting the limitations and challenges faced by object detectors in such dynamic and complex scenes. While sequences in the SportsMOT dataset are comparatively easier to detect and track, some sequences are complicated by extreme camera motion, resulting in significant motion blur and association errors. These challenges underscore the complexities of MOT, particularly in sports-related scenarios where rapid motion and dynamic interactions are prevalent.

Looking ahead, there are several promising directions for future research. With advancements in semantic correspondence in recent literature, one potential avenue is the integration of semantic features into data association, which could further enhance robustness and accuracy. While this work addresses challenges such as crossing individuals, partial overlaps, and the modeling of non-linear complex motion patterns, long-term occlusions remain a critical obstacle in the data association stage of MOT. These occlusions often lead to the creation of new tracks and increased ID switching.

Although some approaches leverage transformer-based architectures to update feature memory banks for improved associations, they typically suffer from limitations in real-time performance. This highlights the need for developing efficient methods to incorporate stronger and more dynamic feature banks into the TBD paradigm, balancing accuracy with real-time processing to create a robust and scalable tracking solution.

References

[1] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *International Conference on Machine Learning (ICML)*, 2024.

[2] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20993–21002, 2022.

[3] Y. Cui, C. Zeng, X. Zhao, Y. Yang, G. Wu, and L. Wang, "Sportsmot: A large multi-object tracking dataset in multiple sports scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9921–9931, 2023.

[4] H.-W. Huang, C.-Y. Yang, J. Sun, P.-K. Kim, K.-J. Kim, K. Lee, C.-I. Huang, and J.-N. Hwang, "Iterative scale-up expansioniou and deep features association for multi-object tracking in sports," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 163–172, 2024.

[5] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," in *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 3025–3029, IEEE, 2023.

[6] M. Yang, G. Han, B. Yan, W. Zhang, J. Qi, H. Lu, and D. Wang, "Hybrid-sort: Weak cues matter for online multi-object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 6504–6512, 2024.

[7] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9686–9696, 2023.

[8] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *European Conference on Computer Vision*, pp. 659–675, Springer, 2022.

[9] M. Segu, L. Piccinelli, S. Li, Y.-H. Yang, B. Schiele, and L. Van Gool, "Samba: Synchronized Set-of-Sequences Modeling for Multiple Object Tracking," Oct. 2024.

[10] R. Gao and L. Wang, "Memotr: Long-term memory-augmented transformer for multi-object tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9901–9910, October 2023.

[11] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1711–1721, 2018.

[12] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," May 2016. arXiv:1603.00831 version: 2.

[13] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," Mar. 2020. arXiv:2003.09003.

[14] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," Aug. 2021. arXiv:2107.08430 [cs].

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[16] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "FastReID: A Pytorch Toolbox for General Instance Re-identification," July 2020. arXiv:2006.02631 [cs].

[17] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A Benchmark for Detecting Human in a Crowd," Apr. 2018. arXiv:1805.00123 [cs].

[18] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.

[19] L. Jianfeng, Y. Zhongliang, L. Yifan, and S. Guanghui, "Gtan: graph-based tracklet association network for multi-object tracking," *Neural Computing and Applications*, vol. 36, no. 8, pp. 3889–3902, 2024.

[20] J. Sun, H.-W. Huang, C.-Y. Yang, Z. Jiang, and J.-N. Hwang, "Gta: Global tracklet association for multi-object tracking in sports," in *Proceedings of the Asian Conference on Computer Vision*, pp. 421–434, 2024.

[21] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6247–6257, 2020.

[22] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22056–22065, 2023.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[24] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, 2016.

[25] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[26] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European conference on computer vision*, pp. 1–21, Springer, 2022.

[27] W. Lv, Y. Huang, N. Zhang, R.-S. Lin, M. Han, and D. Zeng, "Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19321–19330, 2024.

[28] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust Associations Multi-Pedestrian Tracking," July 2022. arXiv:2206.14651.

[29] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023.

[30] R. Varghese and S. M., "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pp. 1–6, 2024.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[32] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*.

[33] C. Xiao, Q. Cao, Z. Luo, and L. Lan, "Mambatrack: a simple baseline for multiple object tracking with state space model," in *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4082–4091, 2024.

[34] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *International Conference on Learning Representations*, 2022.

[35] P. Del Moral, "Non linear filtering: Interacting particle solution," *Markov Processes and Related Fields*, vol. 2, pp. 555–580, 03 1996.

[36] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International journal of computer vision*, vol. 129, pp. 3069–3087, 2021.

[37] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8844–8854, 2022.

[38] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European conference on computer vision*, pp. 474–490, Springer, 2020.

[39] H.-W. Huang, C.-Y. Yang, W. Chai, Z. Jiang, and J.-N. Hwang, "Mambamot: State-space model as motion predictor for multi-object tracking," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2025.

[40] R. Luo, Z. Song, L. Ma, J. Wei, W. Yang, and M. Yang, "Diffusiontrack: Diffusion model for multi-object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 3991–3999, 2024.

[41] C. Xiao, Q. Cao, Y. Zhong, L. Lan, X. Zhang, Z. Luo, and D. Tao, "Motiontrack: Learning motion predictor for multiple object tracking," *Neural Networks*, vol. 179, p. 106539, 2024.

[42] C. Du, C. Lin, R. Jin, B. Chai, Y. Yao, and S. Su, "Exploring the state-of-the-art in multi-object tracking: a comprehensive survey, evaluation, challenges, and future directions," *Multimedia tools and applications*, vol. 83, no. 29, pp. 73151–73189, 2024.

[43] L. F. Ellis, N. D. H. Dowson, J. Matas, and R. Bowden, "Linear regression and adaptive appearance models for fast simultaneous modelling and tracking," *International Journal of Computer Vision*, vol. 95, pp. 154–179, 2010.

[44] T. Fischer, T. E. Huang, J. Pang, L. Qiu, H. Chen, T. Darrell, and F. Yu, "Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[45] P. Chu and H. Ling, "Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6172–6181, 2019.

[46] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[48] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "TransTrack: Multiple Object Tracking with Transformer," May 2021. arXiv:2012.15460.

[49] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense representations for multiple-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7820–7835, 2023.

[50] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, "Memot: Multi-object tracking with memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8090–8100, 2022.

[51] Z. Zhao, Z. Wu, Y. Zhuang, B. Li, and J. Jia, "Tracking objects as pixel-wise distributions," in *European Conference on Computer Vision*, pp. 76–94, Springer, 2022.

[52] E. Yu, T. Wang, Z. Li, Y. Zhang, X. Zhang, and W. Tao, "MOTRv3: Release-Fetch Supervision for End-to-End Multi-Object Tracking," May 2023. arXiv:2305.14298 [cs].

[53] M. Gran-Henriksen, H. A. Lindgaard, G. Kiss, and F. Lindseth, "Deep HM-SORT: Enhancing Multi-Object Tracking in Sports with Deep Features, Harmonic Mean, and Expansion IOU," June 2024. arXiv:2406.12081 [cs].

[54] H. Jung, S. Kang, T. Kim, and H. Kim, "Conftrack: Kalman filter-based multi-person tracking by utilizing confidence score of detection box," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6583–6592, January 2024.

[55] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021," in *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 2809–2819, 2021.

[56] Itseez, "Open source computer vision library." https://github.com/itseez/opencv, 2015. Accessed: Dec. 1, 2024.

[57] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[58] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[59] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview," Nov. 2019. arXiv:1912.05911 [cs].

[60] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 941–951, 2019.

[61] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14329–14339, 2021.

[62] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as Points," Apr. 2019. arXiv:1904.07850 [cs].

[63] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE international conference on multimedia and expo (ICME)*, pp. 1–6, IEEE, 2018.

[64] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 4799–4808, 2023.

[65] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, vol. 33, pp. 1474–1487, 2020.

[66] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, "Combining recurrent, convolutional, and continuous-time models with linear state space layers," *Advances in neural information processing systems*, vol. 34, pp. 572–585, 2021.

[67] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2022.

[68] J. T. Smith, A. Warrington, and S. Linderman, "Simplified state space layers for sequence modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[69] B. Hu, R. Luo, Z. Liu, C. Wang, and W. Liu, "TrackSSM: A General Motion Predictor by State-Space Model," Sept. 2024. arXiv:2409.00487 [cs].

[70] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[71] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *European conference on computer vision*, pp. 107–122, Springer, 2020.

[72] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1363–1389, 2023.

[73] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[74] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[75] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[76] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*, pp. 17–35, Springer, 2016.

[77] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[78] M. Adžemović, P. Tadić, A. Petrović, and M. Nikolić, "Beyond kalman filters: deep learning-based filters for improved object tracking," *Machine Vision and Applications*, vol. 36, no. 1, pp. 1–22, 2025.

[79] M. Adžemović, P. Tadić, A. Petrović, and M. Nikolić, "Engineering an Efficient Object Tracker for Non-Linear Motion," June 2024. arXiv:2407.00738 [cs].

[80] X. Han, N. Oishi, Y. Tian, E. Ucurum, R. Young, C. Chatwin, and P. Birch, "Ettrack: enhanced temporal motion predictor for multi-object tracking," *Applied Intelligence*, vol. 55, no. 1, pp. 1–17, 2025.

[81] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, "Global tracking transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8771–8780, June 2022.

[82] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Transmot: Spatial-temporal graph transformer for multiple object tracking," in *Proceedings of the IEEE/CVF Winter Conference on applications of computer vision*, pp. 4870–4880, 2023.

[83] A. Rangesh, P. Maheshwari, M. Gebre, S. Mhatre, V. Ramezani, and M. M. Trivedi, "TrackMPNN: A Message Passing Graph Neural Architecture for Multi-Object Tracking," May 2021. arXiv:2101.04206.

[84] C. Huang, S. Han, M. He, W. Zheng, and Y. Wei, "Deconfusetrack: Dealing with confusion for multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19290–19299, 2024.

[85] J. Gao, Y. Wang, K.-H. Yap, K. Garg, and B. S. Han, "OccluTrack: Rethinking Awareness of Occlusion for Enhancing Multiple Pedestrian Tracking," Sept. 2023. arXiv:2309.10360 [cs].

[86] Z. Qin, L. Wang, S. Zhou, P. Fu, G. Hua, and W. Tang, "Towards generalizable multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19004, June 2024.

[87] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 31, 2017.

[88] X. Wan, J. Wang, and S. Zhou, "An online and flexible multi-object tracking framework using long short-term memory," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1311–13118, 2018.

[89] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 466–475, IEEE, 2018.

[90] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proceedings of the IEEE international conference on computer vision*, pp. 300–311, 2017.

[91] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13813–13823, 2023.

[92] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. (Accessed: Nov. 1, 2024).

[93] M. Jaward, L. Mihaylova, N. Canagarajah, and D. Bull, "Multiple object tracking using particle filters," in *2006 IEEE Aerospace Conference*, pp. 8 pp.–, 2006.