Hyper DeepSORT: Elevating Precision in Multi-Object Tracking through HyperNMS and Adaptive Kalman Filtering Innovations

Zhiyang Wang, Lei Shan*, and Lei Feng

Abstract—**Multiple Object Tracking (MOT) aims to employ computer vision techniques for real-time tracking and recognition of multiple objects within video sequences. It encompasses the tasks of detection, tracking, and Re-identification (ReID) of objects to achieve continuous tracking of targets over both temporal and spatial domains. MOT makes up a significant challenge within the domain of computer vision. This paper proposes Hyper DeepSORT, an advanced MOT model integrating three significant innovations: HyperNMS, Hyper Kalman Filter, and MTRNet. HyperNMS, a novel Non-Maximum Suppression (NMS) technique, leverages parallel matrix operations to perform NMS in a single iteration, enhancing object recognition accuracy and system efficiency. The Hyper Kalman Filter, an adaptive variant of the traditional Kalman filter, dynamically adjusts noise covariance based on detection confidence, improving the tracker's adaptability and robustness. Additionally, MTRNet incorporates ReID technology to refine feature representation within the DeepSORT framework, encompassing attributes like colour, texture, shape, and motion parameters, bolstering tracking performance. Experimental evaluations on multiple MOT datasets show Hyper DeepSORT outperforms existing models. Specifically, it shows average improvements of 12.75%, 5.37%, 7.20%, 9.94%, 4.90%, and 12.25% over current mainstream models in mAP, MOTA, IDF1, IDSW, FP, and FN metrics, respectively. These results underscore Hyper DeepSORT's superior accuracy and efficiency in complex tracking scenarios.**

Index Terms—**Multi-Object Tracking, Deep learning, DeepSORT, Kalman Filter**

I. Introduction

THE expeditious progression of computer vision technology has engendered a plethora of prospects for applications spanning diverse domains. One of them is the field of Multiple Object Tracking (MOT). MOT tasks are pivotal for real-time surveillance, autonomous vehicular navigation, intelligent security systems, and allied domains. MOT endeavours to attain precision, real-time tracking, and localisation of multiple dynamic entities within video datasets, furnishing nuanced data support for various application scenarios.

MOT tasks confront challenges emanating from the diversity of video content, occlusion phenomena, variations in scale, and fluctuations in illumination[1]. To redress these challenges, investigators initially used statistical methodologies, such as Kalman filters[2] and particle filters[3]. In recent epochs, concomitant with the rapid evolution of deep learning methodologies, incorporating technologies such as Siamese Networks[4], Twin Networks[5], and Convolutional Neural Networks (CNN)[6] have markedly propelled the field of MOT. This paper introduces a novel deep learning-based MOT model, denoted as Hyper DeepSORT, distinguished by the ensuing innovations:

1) This paper introduces HyperNMS as a substitution for conventional Non-Maximum Suppression (NMS). HyperNMS achieves NMS in a singular iteration through parallel matrix operations, thereby augmenting the precision of object recognition without the necessity for iterative processes.

2) Aiming to improve the persistent challenge of a constant noise scale in traditional Kalman algorithms, this paper posits the Hyper Kalman Filter—an adaptive mechanism for computing noise covariance. This adaptation is contingent upon fluctuations in detection confidence, thereby enhancing the adaptability of the tracking mechanism.

3) This paper enhances the MultiTrack Reid Net by utilising uniformly larger convolutional kernels and introducing attention mechanisms in both directions of the image. Experimental results show the model's efficient ReID performance. Even under challenging conditions such as noise, lighting variations, and occlusion, the method achieves continuous and accurate tracking of targets in video sequences.

This paper aspires to furnish a more sophisticated and resilient tracking solution through these novel contributions, thereby surmounting the intricate challenges of MOT tasks.

II. Related work

A. MOT Task Methodologies

The two primary approaches to addressing the MOT task are Motion Feature Tracking (MFT)[7] and Tracking-by-Detection (TBD)[8].

MFT primarily relies on the motion features of objects between consecutive frames, such as optical flow and motion vectors, to achieve object tracking. The MFT method first extracts motion information from each frame using optical flow algorithms, such as the Lucas-Kanade optical flow and dense optical flow or other motion estimation methods. Then, the extracted motion information is used to predict the object's position in the next frame. Finally, the object's trajectory is updated based on these motion features, including position and velocity information.

TBD employs a detect-then-track strategy. Initially, all objects in the video are identified using an object detector[9],

Manuscript received January 9, 2024; revised July 7, 2024.

Zhiyang Wang is an Undergraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (e-mail: wzy13323616619@outlook.com).

Lei Shan* is an Instructor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-187-4222-8155; fax: 0412-5939828; e-mail: Caaaaallisto@hotmail.com).

Lei Feng is an Undergraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (e-mail: 250360716@qq.com).

and these objects are then associated across different frames using an association algorithm to achieve tracking. The TBD model uses pre-trained object detection models, such as YOLO and Faster R-CNN, to detect all objects in each frame. Subsequently, matching algorithms are employed to associate the detection results. Standard matching algorithms include the Hungarian algorithm, the Kalman filter, and the Hungarian algorithm combined with appearance features. Ultimately, the trajectory of each object is established and maintained, including the initialisation of new objects, handling of object loss, and termination of trajectories[10].

The MFT approach is more suitable for scenarios with high real-time requirements and relatively stable objects, such as video stabilisation and motion analysis. In contrast, the TBD method is appropriate for scenarios with many objects requiring high detection accuracy, such as intelligent surveillance and autonomous driving. Currently, mainstream MOT models, such as YOLO, Faster R-CNN, and the improved DeepSORT model used in this paper, all employ the TBD approach.

B. DeepSORT

DeepSORT, an extension of the SORT framework, is a deep learning-based model crafted for MOT[11]. Its structure diagram is shown in Fig. 1. DeepSORT employs advanced techniques to elevate target tracking performance in intricate scenarios, emphasising the resolution of target association challenges and the acquisition of target feature representations.

The methodology of DeepSORT is initiated with a target detector that provides bounding boxes and category information for identified targets. Subsequently, deep learning techniques extract features characterising target appearance and motion attributes, enhancing the tracker's discriminative capabilities. Components for motion estimation and trajectory prediction, often involving Kalman filters, are then utilised to model and estimate target motion, addressing uncertainties and dynamic variations in video sequences[12].

To resolve the challenge of multiple target associations, DeepSORT employs a Hungarian algorithm-based method. This algorithm efficiently matches targets in the current frame with those in the preceding frame, constructing motion trajectories. DeepSORT introduces a Deep Association Metric, leveraging learned deep feature representations to enhance the accuracy of target associations. Furthermore, DeepSORT facilitates differentiation by assigning unique identification numbers (IDs) to targets, supporting prolonged tracking scenarios and multi-target environments. Despite its proven effectiveness in MOT tasks, DeepSORT exhibits potential shortcomings:

1) DeepSORT encounters challenges in scenes involving target occlusion or partial occlusion, as the appearance information of targets may be compromised, leading to tracking instability.

2) The robustness of DeepSORT is a challenge when dealing with rapidly moving targets. Swift motion can cause substantial positional variations between adjacent frames, thereby increasing tracking difficulty[13].

3) DeepSORT has experienced performance bottlenecks when dealing with large-scale target groups, entailing more complex target association and trajectory management challenges.

Addressing these challenges, researchers have proposed improved target association algorithms for enhanced performance in target-dense scenarios and introduced more sophisticated appearance models, leveraging deep learning methods to fortify tracking robustness.

C. Kalman Filtering

Kalman filtering, a recursive and dynamic state estimation methodology, is widely used in MOT tasks. Its primary function is to manage the state information about targets, encompassing parameters such as their positions and velocities within video sequences. The overarching goal is to elevate the precision and robustness of target tracking[14].

The Kalman filtering process involves estimating a target's state utilising observational data, such as the target's position, coupled with a system model. The target's state typically comprises position and velocity parameters in MOT tasks. Kalman filtering not only provides estimates of the current target state but also facilitates the prediction of the subsequent target state based on the underlying system model, thus enabling proactive anticipation of target motion[15].

A pivotal challenge in MOT tasks revolves around associating target positions across different frames to construct coherent target trajectories. By modelling target trajectories, Kalman filtering offers substantial support for data association. By weighing the disparities between predicted and measured target positions, Kalman filtering imparts a refined structure to data association, thereby augmenting its accuracy[16]. Another concern within MOTA tasks is the inherent uncertainty associated with target motion. Kalman filtering tackles this challenge by eliminating process noise terms within the model. This strategic inclusion enhances the model's ability to handle uncertainties in target motion, ultimately bolstering the robustness of the tracking system. Notably, Kalman filtering exhibits adaptability to variations in target speed, accommodating changes induced by acceleration or deceleration.

D. ReID

Re-identification (ReID) technology is employed for the cross-camera tracking of identical targets within a given scene. The primary objective of ReID technology is to mitigate the substantial variations in the appearance of the same target across diverse camera perspectives, locations, angles, and lighting conditions[17].

This technological focus on extracting features from targets aims to construct a robust representation for consistent target identification across varied scenes and viewpoints. These features predominantly encapsulate appearance characteristics such as colour, texture, and attire. In recent years, notable strides in ReID have emerged through integrating deep learning techniques, wherein CNNs play a pivotal role in learning discriminative target features. These models adeptly extract high-level semantic features from images, thereby elevating the precision of target identification.

Metric learning methods are frequently employed in ReID technology to acquire a distance metric capable of gauging the similarity between two targets. Loss functions, including

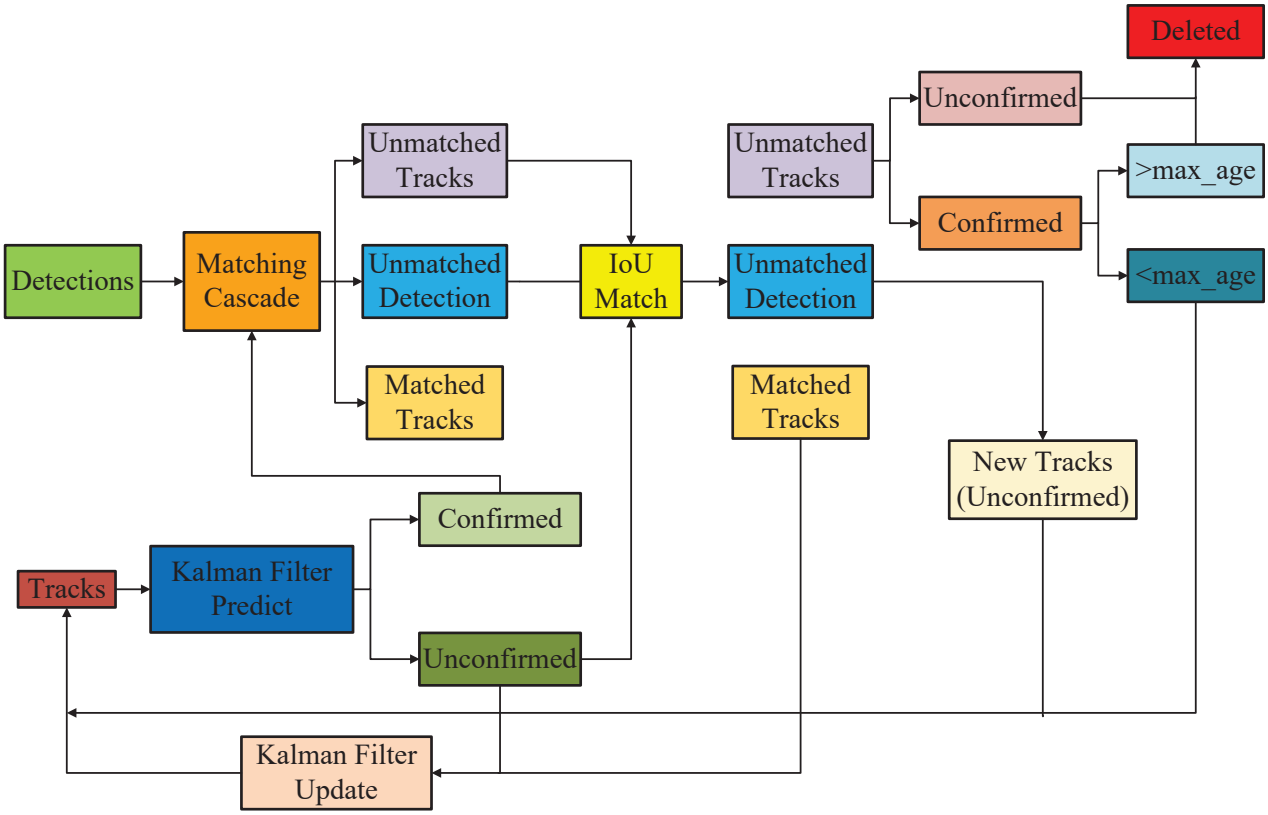

Fig. 1. Overview of DeepSORT

Triplet loss, help to drive the features of the same target closer. The calculation method for Triplet loss, expressed by Equation (1):

$$L = \max\left(0, d(a,p) - d(a,n) + \alpha\right) \tag{1}$$

In this Equation, L represents the Triplet loss result, $d(a,p)$ signifies the distance between the anchor a and positive p examples (typically Euclidean or cosine distance), and $d(a,n)$ denotes the distance between the anchor a and negative n examples. α serves as the margin, a predefined constant ensuring that the distance between features of the same target is smaller than between features of different targets.

Within MOT tasks, ReID technology assumes a pivotal role in associating targets originating from distinct cameras. The system can recognise identical targets through the comparative analysis of ReID features across different cameras, thereby establishing cross-camera trajectories for these targets[18].

III. Hyper DeepSORT Model

A. Hyper Kalman Filter

To employ the Kalman filter for estimating the internal state of an observed process from a series of noisy observational data, it is crucial to formulate the process within the Kalman filter framework. At each step k, matrices F_K, H_K,Q_K, R_K, and B_K are defined. The Kalman filter assumes that the true state x_k at time k and its corresponding measurement z_k evolve from the state at time $k-1$ according to the following Equations (2) and (3):

$$x_k = F_k x_{k-1} + B_k u_k + w_k \tag{2}$$

$$z_k = H_k x_k + v_k \tag{3}$$

In the above Equations, F_K represents the state transition model acting on x_{k-1}, B_K represents the input-control model acting on the control vector u_k, w_k represents process noise assumed to follow a multivariate normal distribution with zero mean and covariance matrix Q_K. Additionally, H_K represents the observation model mapping the actual state space to the observation space, v_k represents the observation noise with zero mean, covariance matrix R_K, and it follows a normal distribution.

The Kalman filter operates as a recursive estimation, enabling the computation of the current state estimate with knowledge of the previous moment's estimated state and the current state's observation. Thus, there is no need to keep a history of observed or estimated information. The state of the Kalman filter comprises the estimate of the state at time k, denoted as $\hat{x}_{k|k}$, and the posterior estimate error covariance matrix $P_{k|k}$, reflecting the accuracy of the measurement estimate.

The Kalman filter's operation comprises two stages: prediction and update. In the prediction phase, the filter utilizes the estimate from the previous state to project an estimate for the current state. In the update phase, the filter refines the predicted value obtained in the prediction phase using the observed value of the current state. Equations (4) and (5) are employed in the prediction phase to update the predicted state $\hat{x}_{k|k-1}$ and the predicted estimate covariance matrix $P_{k|k-1}$.

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{4}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{5}$$

In the update phase, the measurement residual $\widetilde{y}_k$, measurement residual covariance S_k, and optimal Kalman ac-

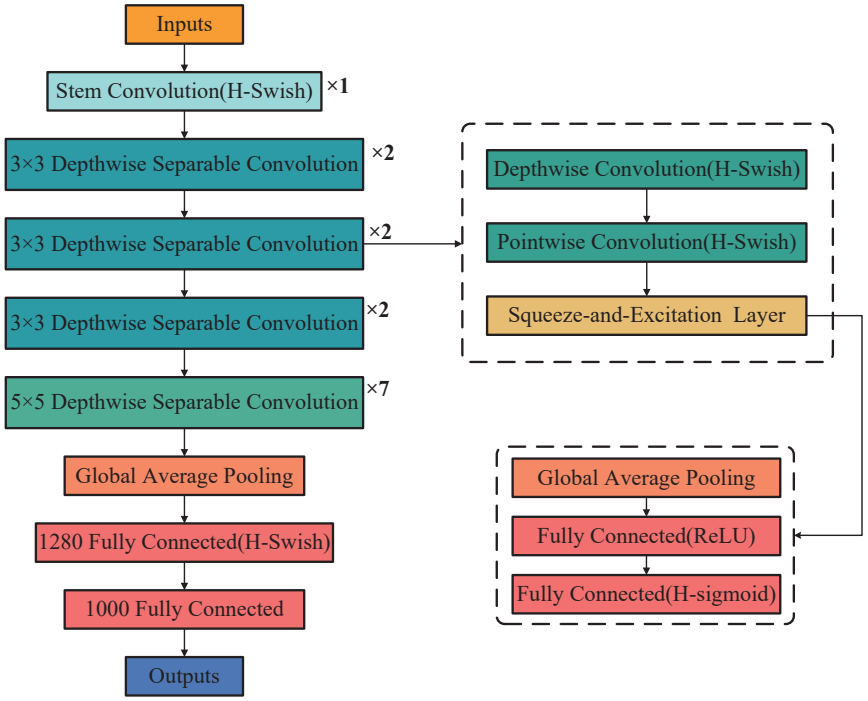

Fig. 2. Overview of MTRNet

quire K_k are calculated through Equations (6) to (8):

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{6}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{7}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{8}$$

Subsequently, these values are utilized to update the filter variables $\hat{x}_{k|k}$ and $P_{k|k}$ in the update step, as outlined in Equations (9) and (10):

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{9}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{10}$$

Within the update step, the measurement noise covariance R_k is employed to represent the noise scale of the measurement. A larger noise scale implies a smaller weight for the measurement in the state update step, reflecting higher uncertainty. In traditional Kalman algorithms, the noise scale is a constant matrix. However, different measurements may entail noise of varying scales, which should adapt to changes in detection confidence. Hence, this paper proposes an adaptive equation for computing the noise covariance, termed NSA noise covariance $\tilde{R}_k$, as illustrated in Equation (11):

$$\tilde{R}_k = (1 - c_k) R_k \tag{11}$$

This equation calculates the noise covariance by multiplying a preset constant measurement noise covariance R_k with the detection confidence score c_k and λ. The Kalman filter in this paper transforms Equation (7) into (12). Subsequent experimental results indicate that, despite the simplicity of this Hyper Kalman approach, it significantly enhances tracking performance.

$$S_k = H_k P_{k|k-1} H_k^T + \tilde{R}_k \tag{12}$$

B. MultiTrack Reid Net

The MultiTrack Reid Net (MTRNet) is designed in this paper to enhance ReID tasks. The process is shown in Fig. 2, and the implementation steps are as follows.

1) H-Swish Activation Function: The H-Swish function is introduced as an activation function, optimizing based on the Swish function[19]. Compared to the Swish function, H-Swish incurs lower computational costs. Swish involves sigmoid functions in its computation, while H-Swish requires simple numerical comparisons and multiplication operations, making it more efficient under limited computational resources. Furthermore, H-Swish retains the non-linear characteristics of ReLU, enabling better learning of non-linear relationships by neural networks. Even when activation values are small, H-Swish maintains higher gradients than the Swish function, avoiding gradient vanishing issues and effectively activating neurons. Lastly, H-Swish exhibits improved gradient propagation, addressing the problem of small gradients at the negative half-axis in the Swish function, thereby aiding better gradient propagation and enhancing the convergence speed of neural networks.

2) SE Layer: A novel efficient attention mechanism, the Squeeze-and-Excitation (SE) Layer[20], is introduced at the end of the backbone network to ease the loss of positional information caused by 2D global pooling. Channel attention is decomposed into x and y directions, two parallel one-dimensional feature encoding processes that effectively integrate spatial coordinate information into the generated attention map. Subsequently, the feature maps encoding the information in these two embedding directions are transformed into two attention maps, each capturing distant dependencies along the spatial direction of the input feature map. Consequently, positional information is preserved in the generated attention map, multiplied by the input feature map to enhance its representational capacity. This attention operation distinguishes spatial directions and

generates coordinate-aware feature maps. Introducing this efficient attention mechanism enables the backbone network to preserve positional information better, thereby improving the model's perception of spatial features. This is crucial for tasks requiring accurate positional information, such as object detection and image segmentation[21].

3) Larger Convolutional Kernel: The size of convolutional kernels often influences network performance. The MixNet[22] analyzed the impact of different kernel sizes on network performance and ultimately mixed different-sized convolutional kernels in the same layer[23]. However, this mixing approach may decrease the model's inference speed. Therefore, this paper attempts to use only one size of the convolutional kernel in a single layer and ensures the use of larger convolutional kernels for improved accuracy under low latency conditions.

4) Adding Larger-Dimension 1×1 Convolutional Layer After Global Average Pooling (GAP): In MTRNet, the output dimension after GAP is relatively tiny. Directly appending the final classification layer may lead to inadequate capture of feature combinations. To enhance the network's fitting capability, a 1×1 convolutional layer with a dimension of 1280 is added after the GAP layer, equivalent to a fully connected (FC) layer. This design allows the network to store more model information while only marginally increasing inference time.

C. Hyper Non-Maximum Suppression

The conventional Non-Maximum Suppression (NMS) algorithm initially arranges all bounding boxes in a descending order based on their classification scores, followed by iterative processing. During each iteration, the bounding box with the highest classification score is preserved, and the Intersection over Union (IoU) is computed for this box concerning other bounding boxes. The IoU, a metric quantifying the overlap between two bounding boxes, is defined by the Equation (13):

$$IoU = \frac{Intersection(A, B)}{Union(A, B)} \tag{13}$$

In Fig. 3, $Intersection(A, B)$ represents the area of the intersection region between two bounding boxes A and B, and $Union(A, B)$ denotes the area of their union region. The IoU value ranges from 0 to 1, with higher values indicating increased overlap between the two bounding boxes.

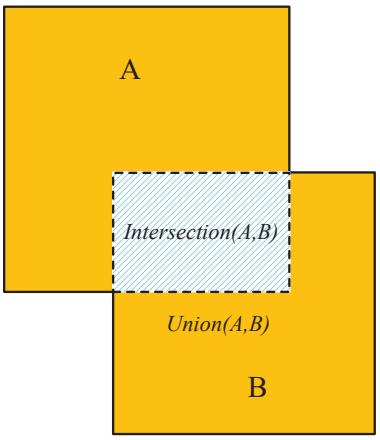

Fig. 3. Illustration of IoU

Assuming the existence of initial detection boxes, denoted as $B = \{b_1, b_2, b_3, \dots b_N\}$, where b_i represents any detection box, and each detection box corresponds to scores $S = \{s_1, s_2, s_3, \dots s_N\}$, each b_i entails the retention of the box with the highest classification score. Subsequently, the IoU is computed concerning this box and other boxes. For boxes with IoU exceeding or equaling the threshold N_t, they undergo removal, and the process iterates until no candidate boxes persist. The mathematical articulation of this process is shown in Equation (14). This methodology confronts triple challenges: heightened computational intricacy, sluggish processing velocity, and the intricate determination of NMS thresholds, thereby predisposing the algorithm to potential false positives or false negatives.

$$s_i = \begin{cases} s_i, & IoU(\mathcal{M}, b_i) < N_t \\ 0, & IoU(\mathcal{M}, b_i) \geq N_t \end{cases} \tag{14}$$

To address these issues, this paper amalgamates the concepts of Soft NMS and Fast NMS, proposing an enhanced NMS method. Specifically, the iterative calculation approach is replaced with a one-time matrix computation. Initially, all bounding boxes are sorted in descending order of classification scores, and then the IoU between all pairs of bounding boxes is calculated, forming a symmetric matrix. Subsequently, the matrix is upper-triangularized, and diagonal elements are set to 0, indicating the IoU of each bounding box with itself. The maximum IoU along dimension 0 is determined, and each IoU is compared against the filtering threshold. For detection boxes with IoU surpassing the threshold, the process involves not a simple filter but a reduction in their confidence score, as outlined in Equation (15).

$$s_i = \begin{cases} s_i, & IoU(M, b_i) < N_t \\ s_i\big(1 - IoU(M, b_i)\big) & IoU(M, b_i) \geq N_t \end{cases} \tag{15}$$

D. Hyper DeepSORT

This paper introduces a comprehensive reconfiguration that integrates NMS, Kalman filtering, and MTRNet, proposing an innovative Hyper DeepSORT framework for MOT tasks. Primarily, the traditional NMS is replaced with HyperNMS, which executes NMS through parallel matrix operations in a single step, obviating the necessity for multiple iterations and substantially augmenting recognition accuracy. Its merits encompass heightened computational efficiency and remarkable accuracy enhancement.

Subsequently, conventional Kalman filtering techniques typically utilize a constant matrix to represent the noise scale. Nevertheless, distinct measurements may exhibit noise of diverse scales. This paper introduces an inventive method for adaptively computing the noise covariance within the Kalman filtering process—the Hyper Kalman Filter to adapt more effectively to variations in detection confidence. This design innovation possesses the potential to elevate performance in tracking tasks by accommodating the dynamic alterations in measurement uncertainty.

Lastly, in terms of feature sets, this paper adopts the feature set from MTRNet for target description and differentiation. The feature set encompasses an array of numerical

values or vectors that depict targets' appearance and motion characteristics, including colour, texture, shape, velocity, acceleration, and other pertinent information. In contrast to conventional methodologies, a novel neural network, MTR-Net, is devised in this research, offering robust support for target identification and tracking tasks.

IV. Experiment Setting

A. Datasets

Several mainstream MOT datasets have been selected for testing to evaluate the performance of the proposed Hyper DeepSORT and its various components. These datasets include MOT-16, ETH, KITTI Tracking, and UA-DETRAC. Below is a brief introduction to each of these four datasets.

1) MOT-16: The MOT-16 dataset makes up a widely utilized benchmark dataset within the MOT domain, designed to facilitate advancing and evaluating tracking algorithms. Comprising 14 high-resolution video sequences, MOT-16 incorporates 7 sequences for training and 7 for testing. These video sequences encapsulate diverse and complex scenarios, including varying weather, fluctuations in illumination, and camera movements. Each video sequence is meticulously annotated, furnishing detailed information such as object bounding boxes, object IDs, frame numbers, and visibility attributes. These annotation data serve as the basis for evaluating the performance of tracking algorithms. Primarily focusing on pedestrians, the dataset features a dense population of pedestrians within the video sequences, thereby increasing the challenge of tracking. Precise annotations of the positions and IDs of each pedestrian across different frames ensure consistency and accuracy in evaluation. The comprehensiveness and high-quality annotations of the MOT-16 dataset render it a pivotal benchmark within the realm of MOT, facilitating the advancement and progression of tracking algorithms.

2) ETH: The ETH dataset is a benchmark MOT and pedestrian detection dataset. This dataset primarily focuses on densely populated urban streets featuring numerous pedestrians, including multiple video sequences captured across various city blocks in Zurich, Switzerland. Consequently, it provides a suitable environment for evaluating the performance of pedestrian detection and tracking algorithms in high-density scenarios. As the video sequences are filmed in authentic urban environments, they exhibit high naturalism and realism, facilitating the assessment of algorithms in practical applications. The video sequences offer various urban environmental scenarios, including diverse weather, illumination variations, and background complexities. Each video sequence is meticulously annotated, including object-binding boxes and object IDs. The ETH dataset has made significant contributions and impacts within pedestrian detection and MOT. Since its highly authentic urban setting and comprehensive annotation information, the ETH dataset has become a crucial benchmark for evaluating and comparing the performance of different algorithms, thus fostering research and development in this domain.

3) KITTI Tracking: The KITTI Tracking dataset serves as a benchmark dataset for research in autonomous driving, primarily focusing on the detection and MOT of vehicles and pedestrians. This dataset includes 21 training and 29 testing video sequence scenes from roads in Karlsruhe, Germany, encompassing diverse driving environments such as urban, rural, and highway settings. Each target in every frame of the video sequence, including vehicles, pedestrians, and cyclists, is meticulously annotated, providing information regarding object categories, bounding boxes, 3D positions, and pose attributes.

4) UA-DETRAC: The UA-DETRAC dataset is a benchmark for vehicle detection and MOT within traffic scenarios. This dataset includes over 100 video sequences from various Beijing and Tianjin locations in China, depicting intricate traffic scenarios. The video sequences span different periods, including daytime and nighttime, and feature diverse weather such as clear skies, rain, and fog. Environments include highways, urban streets, and intersections, among others, showcasing high diversity and complexity. Each vehicle in every frame of the video sequence is meticulously annotated, including object IDs, bounding boxes, and category information. Detailed annotations for each target ensure the accuracy and consistency of the annotation data. The UA-DETRAC dataset has made significant contributions and impacts within vehicle detection and MOT. Its extensive and high-quality annotation data and its focus on traffic scenarios provide valuable resources for developing and evaluating relevant algorithms, thereby driving research and technological advancements in this domain.

Those four datasets mentioned above present unique challenges and complexities in MOT tasks. The pedestrian-dense scenarios in MOT-16 and ETH, the fast-moving vehicles in KITTI Tracking, and the intricate traffic environments in UA-DETRAC comprehensively test the performance of tracking algorithms across various scenarios. In summary, comparing the performance of MOT tasks using the MOT-16, ETH, KITTI Tracking, and UA-DETRAC datasets ensures algorithms' robustness, generality, and efficiency across a spectrum of complex and real-world scenarios. This comparison facilitates comprehensive and reliable performance evaluations, providing insights into algorithmic capabilities within diverse and authentic environments.

B. Evaluation Metrics

The experimental section will employ the following evaluation metrics to assess Hyper DeepSORT and other mainstream MOT models:

1) False Positives (FP): FP represents the number of detections of non-existent targets, namely the number of false alarms. A lower count of FP signifies higher precision in detection and tracking.

2) False Negatives (FN): FN represents the number of undetected genuine targets, indicating the number of missed detections. A lower count of FN indicates higher completeness in detection and tracking.

3) ID Switches (IDSW): IDSW represents the number of times target IDs change during the tracking process. A lower value of IDSW indicates more excellent stability in the tracking algorithm.

4) ID F1-Score (IDF1): IDF1 represents the degree of matching between tracked and ground truth trajectories, considering Precision and Recall. The calculation process is as Equation (16).

$$IDF1 = \frac{2 \cdot IDTP}{2 \cdot IDTP + IDFP + IDFN} \tag{16}$$

$IDTP$ is the number of correctly matched IDs, $IDFP$ is the number of incorrectly matched IDs, and $IDFN$ is the number of missed IDs. A higher IDF1 value indicates better tracking performance.

5) Multiple Object Tracking Accuracy (MOTA): MOTA represents a metric comprehensively considering FN, FP, and IDSW. The calculation process is as Equation (17).

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{17}$$

FN_t, FP_t, and $IDSW_t$ respectively represent the number of missed detections, false positives, and ID switches at time t, and GT_t denotes the number of ground truth targets at time t. A higher MOTA value indicates better tracking performance.

6) mean Average Precision (mAP): mAP is the mean of Average Precision (AP) across different object categories. AP measures the combined performance of precision and recall of a detector on a specific category. As the name suggests, mAP is calculated by averaging the APs of all categories. The calculation process is as Equation (18).

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \tag{18}$$

N is the number of categories, and AP_i is the average precision for the $i-th$ category.

7) Frames Per Second (FPS): FPS measures the real-time performance of a tracking algorithm, indicating the number of frames processed per second. A higher FPS value signifies greater efficiency in the algorithm and better meeting the requirements of real-time applications.

8) FLOPs (Floating Point Operations Per Second): FLOPs is a critical metric used to measure the performance of computational systems. It denotes the number of floating-point operations a system can execute in one second. FLOPs is essential when assessing high-performance computing systems' performance, efficiency, and deep learning models. Typically, FLOPs is combined with other accuracy metrics, where, under the premise of maintaining the same level of accuracy, a model with lower FLOPs is considered more efficient. Furthermore, the FLOPs value of a model can vary depending on the size of the input image.

Among these metrics, the values of IDF1 and mAP range from 0 to 1, while MOTA ranges from -∞ to 1. A negative MOTA value indicates inferior algorithm performance. Given that no negative MOTA values were observed in this experiment, the MOTA values fall within the 0 to 1 range. This experiment displays the IDF1, mAP, and MOTA values as percentages to ensure data presentation consistency. The values for the remaining evaluation metrics range from 0 to +∞. Higher values of mAP, MOTA, IDF1, and FPS indicate better model performance, whereas lower values of FP, FN, and IDSW suggest better model performance. Under the premise of maintaining the same level of model accuracy, a lower FLOPs value indicates higher computational efficiency of the model.

C. Baselines and Equipment

The experimental part uses Hyper DeepSORT and the following excellent MOT models for comparison.

1) Observation-Centric SORT (OC-SORT)[24]: OC-SORT represents a motion-model-centric multiobject tracking system designed to enhance tracking robustness in congested scenarios and instances of non-linear object motion. It addresses and rectifies limitations inherent in the Kalman filter and SORT, offering flexibility for integrating diverse detectors and matching modules, including appearance similarity. Emphasizing simplicity, online capability, and real-time operation, OC-SORT caters to dynamic tracking requirements.

2) BoT-SORT[25]: A novel and robust state-of-the-art tracker is introduced, capable of synergizing motion and appearance information. It presents a refined Kalman filter state vector to achieve higher accuracy in object tracking.

3) CenterTrack[26]: This model is a dedicated deep-learning framework tailored for MOT. The seamless integration of object detection and tracking significantly advances the real-time performance of object tracking. The core concept involves representing objects by estimating their centre points, departing from the conventional practice of using the corners of the bounding box. This center-point representation enhances the model's adaptability to object motion and deformation variations. CenterTrack also capitalizes on motion information and feature extraction to enhance tracking precision, adopting a real-time design to meet the demands of applications that require instantaneous response.

In addition to experimenting with the backbone network, this paper also conducts experimental validation on the ReID section. The following are mainstream ReID networks used in this paper for comparison:

1) RestNet-101[27]: RestNet is a deep neural network architecture that introduces the concept of residual learning, addressing the vanishing gradient problem in deep networks through skip connections. The core idea is for network layers to learn the residual between input and target mappings, making the network easier to train and capable of achieving greater depth. ResNet has demonstrated excellent performance in tasks such as image classification and has been a widely used architecture in deep learning. This paper selects RestNet-101, a variant with 101 layers, for experimental validation.

2) Efficient-B4[28]: EfficientNet is an efficient convolutional neural network architecture that maintains model accuracy while reducing computational complexity, parameter count, and memory usage. Efficient-B4 represents the fourth variant of EfficientNet, optimizing the network's depth, width, and resolution dimensions for balanced performance in a relatively smaller model size. Consequently, Efficient-B4 exhibits higher computational efficiency and is suitable for scenarios with limited computational resources. In this paper, Efficient-B4 is chosen as one of the comparative models.

3) Visual Geometry Group Network (VGGNet)[29]: VGGNet is a convolutional neural network architecture proposed by the Visual Geometry Group at the University of Oxford, achieving significant results in the 2014 ImageNet Large Scale Visual Recognition Challenge. One of the main features of VGGNet is its simple and regular structure, with fundamental building blocks consisting of multiple

3x3 convolutional layers followed by a max-pooling layer, resulting in an intense network. In this paper, VGGNet is selected as one of the comparative models for the ReID section and is used in experimental validation.

Those models mentioned above were implemented using the Python-based PyTorch deep learning framework. Model training and validation were conducted on a Linux server equipped with an Intel(R) Xeon(R) Platinum 8352V @ 3.50GHz processor and an NVIDIA RTX 4090 (24GB) GPU. As mentioned earlier, the input image size was set to default. All datasets were divided into training, validation, and testing sets in a 6:2:2 ratio. The experimental results presented in this chapter represent the average of five independent experiments. The best results are displayed in bold, while the second-best results are underscored. If multiple models achieve the same best or second-best results, they are marked in order of model sequence, with only the first being labelled.

V. Result and Analysis

A. Performance Validation and Analysis of MTRNet as a ReID Module in MOT Tasks

The experimental section first verifies the performance of MTRNet as a ReID module in Multiple Object Tracking (MOT) tasks. Using the original DeepSORT as the backbone network, the effectiveness of different ReID modules was evaluated through comparative experiments, with the results detailed in Table I. The results indicate MTRNet achieved the best or second-best scores across almost all evaluation metrics in various datasets. Among all the baseline models, ResNet-101 performed the best and is referred to as the optimal baseline model for subsequent comparisons.

Specifically, in the MOT-16 dataset, MTRNet achieved the best results in six metrics: mAP, MOTA, IDF1, IDSW, FP, and FN, improving by 2.71%, 5.37%, 5.47%, 16.34%, 4.97%, and 14.21%, respectively, compared to the optimal baseline model. It achieved the second-best results in FPS and FLOPs, trailing the best results by 12.62% and 15.79%, respectively. In the ETH dataset, MTRNet also achieved the best results in the same six metrics, with improvements of 2.72%, 5.37%, 5.38%, 16.33%, 4.67%, and 14.37%, respectively. It achieved the second-best results in FPS and FLOPs, trailing the best results by 14.47% and 16.30%, respectively. In the KITTI Tracking dataset, MTRNet achieved the best results in the same six metrics, with improvements of 8.12%, 7.17%, 8.10%, 12.50%, 5.98%, and 15.21%, respectively. It achieved the second-best results in FPS and FLOPs, trailing the best results by 14.69% and 9.20%, respectively. In the UA-DETRAC dataset, MTRNet achieved the best results in IDSW, FP, and FN, improving by 13.76%, 1.93%, and 11.94%, respectively. It achieved the second-best results in mAP, MOTA, IDF1, FPS, and FLOPs, trailing the best results by 3.65%, 3.43%, 3.39%, 9.83%, and 16.84%, respectively. For a more intuitive presentation of the results, the optimal baseline model ResNet-101 is set as the benchmark (100), and performing MTRNet in various evaluation metrics across different datasets is depicted proportionally, as shown in Fig. 4.

From the results in Table I and Fig. 4, it can be observed that MTRNet significantly improved the accuracy of DeepSORT in multiobject tracking when used as a ReID module. MTRNet achieved the best mAP scores in three out of four datasets. In ReID tasks, mAP refers explicitly to the average precision of the model in re-identifying targets, reflecting the performance of the ReID model in matching query images with images in the database. A higher mAP indicates that MTRNet can stably extract features and accurately reflect the distinctiveness of targets, effectively distinguishing different targets even in cases of similar appearances. MTRNet showed consistent performance across different queries and environmental conditions, reliably re-identifying targets.

Additionally, MTRNet achieved the best IDF1 scores in three out of four datasets. A high IDF1 score indicates that the MTRNet module can consistently maintain the identity

TABLE I
MTRNet and Other ReID Baseline Models Comparison

Datasets	ReID Networks	Metrics							
		mAP	MOTA	IDF1	IDSW	FP	FN	FPS	FLOPs
MOT-16	ResNet-101	36.86	74.50	75.62	202	8054	21644	23.20	**9.60**
	Efficient-B4	35.68	73.80	74.87	202	7939	22680	18.70	19.50
	VGGNet	29.56	65.50	62.43	240	8865	25648	**32.50**	15.30
	MTRNet	**37.86**	**78.50**	**79.76**	**169**	**7654**	**18569**	28.40	11.40
ETH	ResNet-101	36.39	73.56	74.61	205	8143	21925	21.10	**13.50**
	Efficient-B4	36.23	74.87	73.95	205	8026	23066	20.60	27.30
	VGGNet	29.18	64.67	61.61	243	8963	25981	**31.10**	21.40
	MTRNet	**37.38**	**77.51**	**78.62**	**171**	**7738**	**18773**	26.60	15.70
KITTI Tracking	ResNet-101	42.02	87.17	89.21	180	7563	20367	21.10	**16.30**
	Efficient-B4	43.33	87.66	83.89	172	**7020**	20208	20.00	34.40
	VGGNet	31.33	68.78	63.65	229	8502	24545	**28.60**	26.00
	MTRNet	**45.43**	**93.42**	**96.44**	**158**	7111	**17269**	24.40	17.80
UA-DETRAC	ResNet-101	51.24	103.56	105.08	190	7587	20389	26.70	**26.90**
	Efficient-B4	**54.23**	**112.18**	**113.85**	189	7547	21274	28.40	54.60
	VGGNet	41.98	93.01	88.61	224	8262	23904	**32.40**	42.80
	MTRNet	52.25	108.33	109.99	**163**	**7401**	**17956**	29.50	31.70

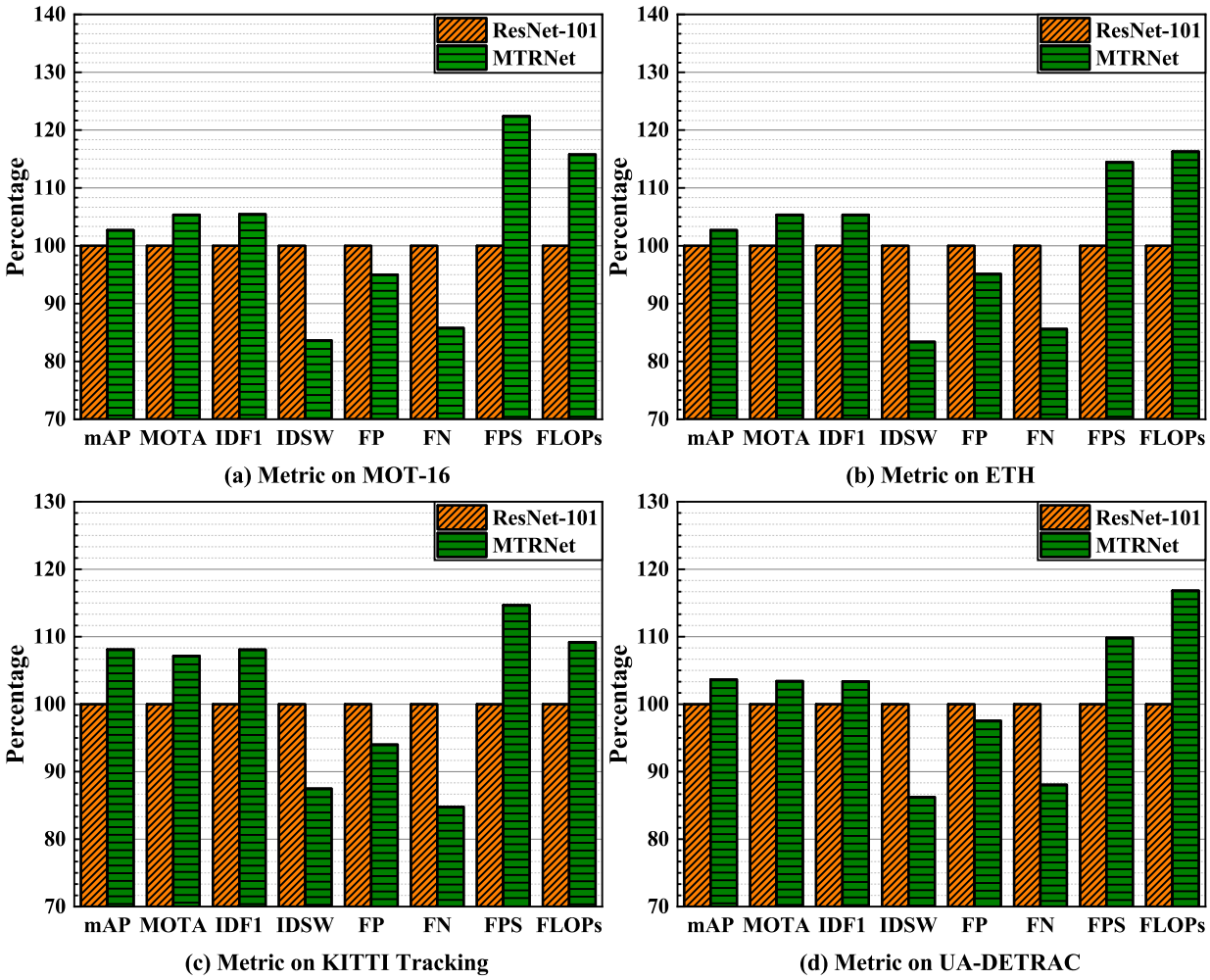

Fig. 4. Proportional Evaluation of Hyper DeepSORT Based on the Optimal Baseline Model

of targets throughout the tracking process. This is crucial for multiobject tracking tasks, as frequent identity switches severely affect tracking continuity and accuracy. A high IDF1 score means the model reduces FP and FN and correctly maintains target identities. This corresponds to a low IDSW, indicating that MTRNet rarely makes identity switch errors during tracking. Even when targets temporarily disappear or are occluded, the module can accurately re-identify the target without mistaking it for another. That MTRNet achieved the best scores in FP, FN, and IDSW metrics in multiple datasets in Table 1 corroborates this analysis. This improvement is mainly attributed to the bidirectional attention computation within the MTRNet SE layer and the utilisation of larger convolution kernels.

Lower FP, FN, and IDSW naturally lead to higher MOTA scores, indicating that MTRNet can accurately distinguish and identify different targets, maintaining performance even with changes in target appearance, partial occlusion, or environmental variations. It can continuously track the same target, avoiding frequent trajectory switches and erroneous re-identifications, significantly improving the overall performance and stability of the tracking system. Notably, although the bidirectional attention computation in the SE layer increases computational overhead, leading to a slight decrease in FPS, it also brings higher tracking accuracy and sustained target identity. This trade-off is acceptable in practical applications, as continuous and accurate tracking is crucial for the effectiveness of multiobject tracking systems. While surpassing the optimal baseline model in accuracy, MTRNet's FLOPs metric in different datasets were, on average, only 14.78% higher, indicating that MTRNet achieved performance improvements with a slight increase in computational complexity, indirectly enhancing the model's generalisation capability.

B. Performance Validation and Analysis of Hyper DeepSORT in MOT Tasks

After validating the excellent performance of MTRNet as a ReID network, comparative experiments were conducted between the proposed HyperSORT and other baseline models on the four previously mentioned datasets. Detailed information can be found in Table II. The data in the table shows that HyperSORT almost achieved the most optimal and second-best scores across all evaluation metrics in all datasets, positioning it as the optimal model. CenterTrack performed the best and is considered the optimal baseline model among the baseline models.

Specifically, in the MOT-16 dataset, HyperSORT achieved the best results in six metrics: mAP, MOTA, IDF1, IDSW, FP, and FN, improving by 12.75%, 21.62%, 7.20%, 29.03%, 8.37%, and 12.25%, respectively, compared to the optimal baseline model. It was 9.29% lower in FPS and 6.93% higher in FLOPs. In the ETH dataset, HyperSORT also achieved the best results in the same six metrics, improving

TABLE II
HYPER DEEPSORT AND OTHER MOT BASELINE MODELS COMPARISON

Datasets	BackBone	ReID	Metrics							
			mAP	MOTA	IDF1	IDSW	FP	FN	FPS	FLOPs
MOT-16	DeepSORT	ResNet-101	26.58	69.89	50.14	205	8214	22688	24.00	24.80
	DeepSORT	MRTNet	30.59	74.52	52.45	171	8054	21644	22.30	26.40
	OC-SORT	\	28.98	70.58	55.79	266	8655	24688	24.80	**23.70**
	BoT-SORT	\	27.89	69.54	60.51	279	8898	25498	25.70	24.10
	CenterTrack	\	33.58	64.56	74.54	217	8359	21264	**26.90**	32.40
	Hyper DeepSORT	MRTNet	**37.86**	**78.52**	**79.91**	**154**	**7659**	**18659**	24.40	30.30
ETH	DeepSORT	ResNet-101	25.86	68.00	48.74	207	8288	23006	23.30	35.80
	DeepSORT	MRTNet	29.86	72.36	50.98	174	8175	21969	21.70	37.00
	OC-SORT	\	28.26	68.82	54.40	270	8785	25058	24.20	**30.40**
	BoT-SORT	\	27.19	67.80	59.00	283	9031	25880	25.10	30.60
	CenterTrack	\	34.01	63.46	78.03	221	8509	21604	**26.20**	47.90
	Hyper DeepSORT	MRTNet	**37.06**	**76.79**	**79.07**	**156**	**7766**	**18902**	24.90	42.20
KITTI Tracking	DeepSORT	ResNet-101	29.53	77.66	55.66	202	8097	22477	22.70	59.40
	DeepSORT	MRTNet	33.95	83.36	58.12	168	7938	21354	21.00	71.90
	OC-SORT	\	32.24	78.79	61.90	264	8574	24432	23.67	**60.70**
	BoT-SORT	\	31.05	77.43	67.37	277	8824	25285	24.20	63.70
	CenterTrack	\	38.74	72.28	89.34	216	8356	21237	**25.60**	82.70
	Hyper DeepSORT	MRTNet	**42.51**	**88.00**	**90.62**	**152**	**7549**	**18505**	24.30	74.00
UA-DETRAC	DeepSORT	ResNet-101	27.40	72.13	51.44	199	7984	22121	24.60	69.90
	DeepSORT	MRTNet	31.45	76.61	53.92	186	7837	21060	22.90	81.80
	OC-SORT	\	29.65	72.63	57.52	259	8387	23972	25.50	**67.30**
	BoT-SORT	\	28.67	71.49	62.20	271	8658	24810	**27.60**	68.50
	CenterTrack	\	36.81	**81.98**	81.05	211	8150	20754	26.20	78.40
	Hyper DeepSORT	MRTNet	**37.27**	80.56	**82.39**	**172**	**7422**	**18137**	25.10	72.90

by 8.98%, 21.00%, 1.34%, 29.17%, 8.73%, and 12.51%, respectively, compared to the optimal baseline model. It was 4.96% lower in FPS and 13.50% higher in FLOPs. In the KITTI Tracking dataset, HyperSORT achieved the best results in the same six metrics, improving by 9.74%, 21.75%, 1.43%, 29.46%, 9.66%, and 12.87%, respectively, compared to the optimal baseline model. It achieved the second-best result in FPS, 5.34% lower than the optimal baseline model, and was 11.76% higher in FLOPs. In the UA-DETRAC dataset, HyperSORT achieved the best results in mAP, IDF1, IDSW, FP, and FN, improving by 12.50%, 1.64%, 18.62%, 8.93%, and 12.61%, respectively, compared to the optimal baseline model. It achieved the second-best result in MOTA, 1.73% lower than the optimal score, was 4.20% lower in FPS, and 7.02% higher in FLOPs. Like the ReID comparison, the optimal baseline model CenterTrack was set as the benchmark (100). HyperSORT's performance in various evaluation metrics across different datasets was depicted proportionally, as shown in Fig. 5.

From the results in Table II and Fig. 5, HyperSORT achieved the best or second-best scores in multiple evaluation metrics across the four datasets, demonstrating its effectiveness in addressing challenges in MOT tasks. HyperSORT achieved the best mAP scores in all four datasets. mAP is a crucial metric for evaluating object detection performance. A higher mAP indicates that the model performs excellently in object detection, accurately detecting multiple targets in images or videos and providing relatively precise bounding boxes. mAP considers the confidence scores of each detection box, meaning HyperSORT can confidently identify targets after detection, reducing FP and FN, as reflected in Table II, where HyperSORT had the lowest FP and FN scores in all four datasets.

Accurate object detection is foundational for subsequent tracking effectiveness in multi-object tracking tasks. HyperSORT achieved the best IDF1 scores in all four datasets, indicating it maintains target identity consistency throughout the tracking process, reducing IDSW. This is due to the proposed corrected IoU confidence calculation strategy and the improved Hyper Kalman Filter. The former allows HyperSORT to adjust according to target dynamics, significantly reducing erroneous switches, while the latter reduces noise during the tracking process, enhancing tracking performance. In contrast, DeepSORT uses original confidence measurements and a Kalman filter, resulting in higher FP scores than HyperSORT, indicating many misidentified targets in its tracking results. The comparison of IDSW scores also reflects this phenomenon, with HyperSORT achieving the best IDSW scores in all four datasets.

IDF1 primarily evaluates the comprehensive ability of target detection and identity matching. A higher IDF1 value means that HyperSORT can detect and accurately associate targets across different frames, ensuring the completeness of the tracking chain. This is crucial for practical MOT tasks, especially in complex and dynamic environments. A model that maintains a high IDF1 value is more reliable and practical. The combined effects of lower FP, FN, and IDSW naturally enhance the MOTA metric. HyperSORT achieved

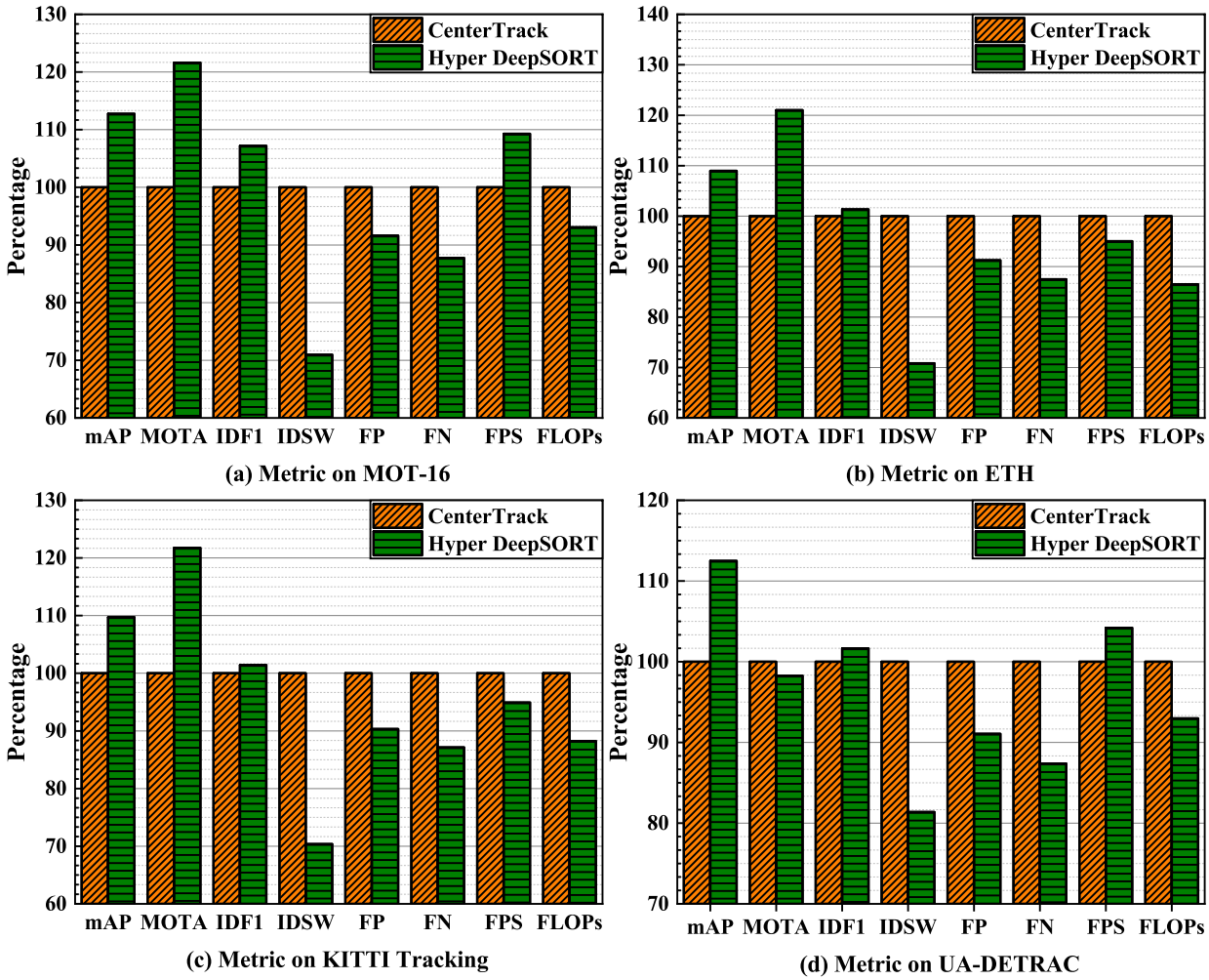

Fig. 5. Proportional Evaluation of Hyper DeepSORT Based on the Optimal Baseline Model

three best and one second-best MOTA scores across the four datasets, indicating its outstanding performance in tracking tasks.

Although HyperSORT lags CenterTrack in FPS, this is because CenterTrack significantly reduces computational complexity by using centre positioning, achieving the best FPS scores. However, this method increases IDSW values, leading to reduced tracking performance. The slight decrease in FPS with HyperSORT increases computational overhead and enhances its ability to track continuous and accurate targets. This balance is crucial in practical applications as it ensures the overall performance and stability of the multi-object tracking system. HyperSORT's FLOPs scores were better than CenterTrack's in all datasets because CenterTrack's CenterNet framework generates centre points and bounding box regressions requiring extensive convolutional computations. HyperSORT introduces an attention mechanism and optimizes the Kalman Filter, achieving lower FLOPs scores while surpassing CenterTrack in accuracy. This makes it more resource- and energy-efficient, essential for deployment on resource-constrained devices such as mobile and embedded systems. HyperSORT can reduce computational resource usage and costs in large-scale deployment scenarios, improving system scalability and service quality.

C. Ablation Experiments

To validate the various modules and their functionalities proposed in this paper, a series of ablation experiments were conducted on the MOT-16 dataset to assess the performance of Hyper DeepSORT and its variants. Specifically, three variants based on Hyper DeepSORT were designed:

1) HS-HKF: This variant replaces the Hyper Kalman Filter in Hyper DeepSORT with the original Kalman Filter, aiming to evaluate the performance of the Hyper Kalman Filter.

2) HS-HNMS: This variant replaces the Hyper Non-Maximum Suppression in Hyper DeepSORT with traditional Non-Maximum Suppression, aiming to assess the performance of Hyper Non-Maximum Suppression.

3) HS-MTRNet: This variant replaces the MultiTrack ReID Net in Hyper DeepSORT with the currently mainstream ReID module, ResNet-101, to validate the performance of MultiTrack ReID Net.

Besides these three variants, the experiments also included the original DeepSORT network and the proposed Hyper DeepSORT as controls. The specific experimental settings were consistent with the previous comparative experiments. The best results are highlighted in bold, while the second-best results are underlined. The detailed experimental results are presented in Table III. As in the previous two experimental sections, this paper sets Hyper DeepSORT as the baseline (100) and plots the proportional evaluation metrics of the three variants of Hyper DeepSORT and the original

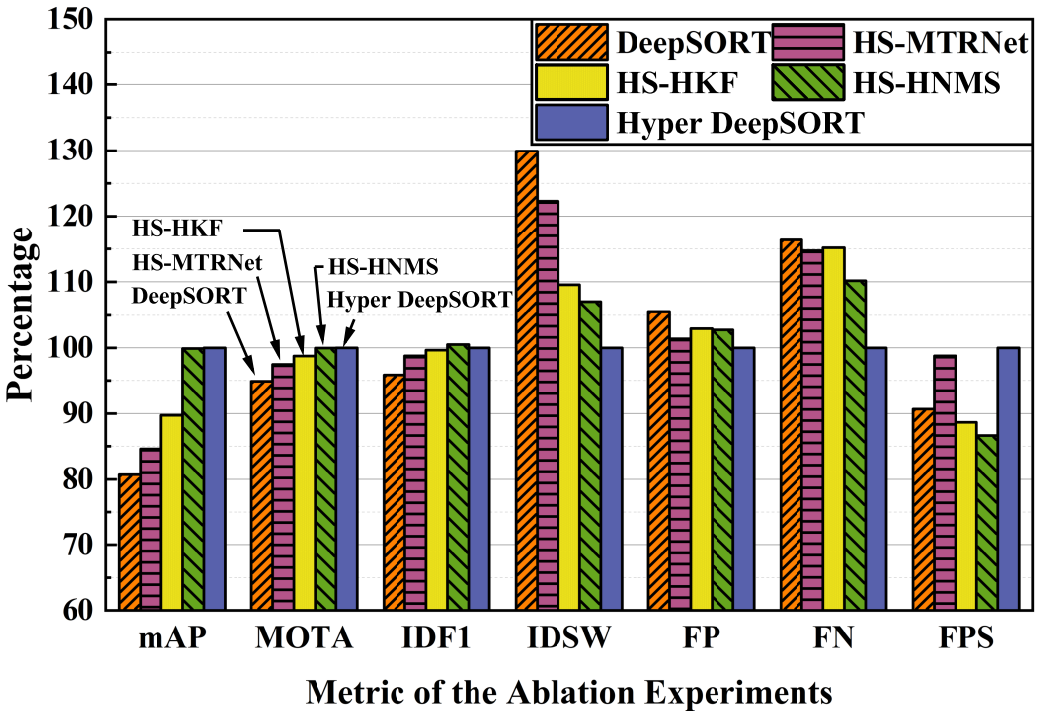

Fig. 6. Proportional Evaluation of Ablation Experiment Based on Hyper DeepSORT

DeepSORT across various datasets. The specific results are shown in Fig. 6.

From the data in Table III, it can be observed that, except for the FPS metric, Hyper DeepSORT achieved the best results across all metrics. HS-MTRNet exhibited the worst performance among the Hyper DeepSORT variants. The performance of HS-HKF showed that the FP metric was the highest among all Hyper DeepSORT variants, exceeding Hyper DeepSORT by 15.17%. Noise often leads to incorrect identity switches, resulting in a high IDSW metric, the second-highest among the Hyper DeepSORT variants, 9.55% higher than Hyper DeepSORT. This is because the traditional Kalman filter algorithm uses a constant noise scale matrix that does not adapt to changes in detection confidence. In contrast, the proposed Hyper Kalman Filter measures noise covariance to represent the noise scale in the current frame, effectively filtering out background noise. During the experiments, HS-HKF was more prone to losing targets and misidentifications under varying lighting conditions or complex backgrounds.

The performance of HS-HNMS showed that the FPS metric was the lowest among the Hyper DeepSORT variants, with an 8.11% difference compared to Hyper DeepSORT, and the FP and FN metrics were the second-highest among the variants. The traditional Non-Maximum Suppression algorithm iteratively selects the highest-scoring bounding box. It calculates the intersection over union (IoU) with all other bounding boxes, deleting those with an IoU above a certain threshold, which requires substantial computational resources, reflected in its lower FPS metric. In contrast, Hyper NMS reduces the confidence scores of boxes below the threshold, decreasing the number of bounding boxes that need to be deleted entirely and recalculated. This approach improves processing speed and mitigates the increase in FP and FN caused by the aggressive deletion of all boxes below a fixed threshold.

TABLE III
ABLATION EXPERIMENTS OF HYPER DEEPSORT

Method	Metrics						
	mAP	MOTA	IDF1	IDSW	FP	FN	FPS
DeepSORT	30.59	74.52	75.25	204	8054	21644	22.40
HS-MTRNet	32.06	76.55	77.55	192	7739	21339	24.40
HS-HKF	34.01	77.57	78.23	172	7859	21420	21.90
HS-HNMS	36.86	78.52	78.91	168	7844	20492	21.40
Hyper DeepSORT	**37.90**	**78.54**	**79.50**	**157**	**7633**	**18599**	**24.70**

The performance of HS-MTRNet was the poorest, primarily because ResNet-101, used as the ReID module, performed worse than the proposed MTRNet, as showed in earlier experiments. The attention calculation mechanism in both x and y directions in MultiTrack ReID Net enhances the model's tracking ability, resulting in significantly higher IDSW values for HS-MTRNet, the highest among all Hyper DeepSORT variants, 22.29% higher than Hyper DeepSORT. MTRNet achieves consistent target ReID across different times, locations, and even cameras, significantly improving the accuracy and robustness of Hyper DeepSORT.

Hyper DeepSORT and all its variants outperformed the traditional DeepSORT network, validating not only Hyper DeepSORT's superior performance but also the effectiveness of the three mechanisms. The Hyper Kalman Filter, MultiTrack ReID Net, and Hyper NMS can each serve as independent modules to enhance the performance of other MOT models.

VI. CONCLUSION

This paper initiates an exploration into the common challenges in MOT tasks and subsequently proposes a novel MOT model named Hyper DeepSORT, building upon the

foundation of DeepSORT. The model introduces the Hyper Kalman Filter, replacing the original Kalman Filter and effectively enhancing the model's capability to eliminate noise. Incorporating Hyper NMS reduces the model's false negative rate, decreases computational overhead, and consequently boosts the processing speed reflected in the FPS metric. MTRNet improves upon traditional ReID models, notably reducing the false positive rate. Experimental results show the model excels compared to other MOT models on the MOT-16 dataset, positioning itself as a cutting-edge model in the MOT domain. Furthermore, the various submodules of the Hyper DeepSORT model can seamlessly integrate into other models, providing new avenues for breakthroughs in related research. This paper enhances the robustness and performance of target tracking systems and establishes a foundation for future research and applications.

REFERENCES

[1] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," *Artificial Intelligence*, vol. 293, pp. 103 448–103 497, 2021.

[2] H. W. Sorenson, "Kalman Filtering Techniques," in *Advances in Control Systems*. Elsevier, 1966, vol. 3, pp. 219–292.

[3] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[4] S. Lee and E. Kim, "Multiple Object Tracking via Feature Pyramid Siamese Networks," *IEEE Access*, vol. 7, pp. 8181–8194, 2018.

[5] X. Liu, Y. Dai, L. Liu, and Z. Hu, "Research on Online Multi-Target Tracking Algorithm Combining Full Convolutional Twinnetworks and ReID Networks," in *2023 IEEE 13th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2023, pp. 817–822.

[6] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online Multi-object Tracking Using CNN-Based Single Object Tracker with Spatial-Temporal Attention Mechanism," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4836–4845.

[7] L. Zhang and L. Van Der Maaten, "Preserving Structure in Model-Free Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756–769, 2013.

[8] E. Bochinski, V. Eiselein, and T. Sikora, "High-Speed Tracking-by-Detection without Using Image Information," in *2017 14th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*. Springer, 2012, pp. 702–715.

[10] Z. Sun, J. Chen, L. Chao, W. Ruan, and M. Mukherjee, "A Survey of Multiple Pedestrian Tracking Based on Tracking-by-Detection Framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1819–1833, 2020.

[11] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with A Deep Association Netric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[12] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir, and H. Hashim, "People Tracking System Using DeepSORT," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, 2020, pp. 137–141.

[13] D. Simon, "Kalman Filtering," *Embedded Systems Programming*, vol. 14, no. 6, pp. 72–79, 2001.

[14] X. He, Y. Wang, and S. Yang, "Gaussian Mixture CBMeMBer Filter for Multi-Target Tracking with Non-Gaussian Noise," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 3, pp. 568–575, 2022.

[15] Y. Chen, D. Zhao, and H. Li, "Deep Kalman Filter with Optical Flow for Multiple Object Tracking," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 3036–3041.

[16] G. Guo and S. Zhao, "3D Multi-Object Tracking with Adaptive Cubature Kalman Filter for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 512–519, 2022.

[17] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep Learning for Person Re-Identification: A Survey and Outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2872–2893, 2021.

[18] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The Re-Identification Challenge," *Person Re-Identification*, pp. 1–20, 2014.

[19] L. Yang, Q. Song, Z. Fan, C. Liu, and M. Hu, "Rethinking the Activation Function in Lightweight Network," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 1355–1371, 2023.

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[21] Y. Wang, Y. Li, and Q. Han, "Vehicle-Mounted Infrared Pedestrian Tracking Based on Scale Adaptive Kernel Correlation Filter," *IAENG International Journal of Computer Science*, vol. 49, no. 2, pp. 349–356, 2022.

[22] L. T. Duong, P. T. Nguyen, C. Di Sipio, and D. Di Ruscio, "Automated Fruit Recognition using EfficientNet and MixNet," *Computers and Electronics in Agriculture*, vol. 171, p. 105326, 2020.

[23] R. Zhang, "Making Convolutional Networks Shift-invariant Again," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7324–7334.

[24] 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*, 2023, pp. 9686–9696.

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

[26] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking Objects as Points," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning PMLR*, 2019, pp. 6105–6114.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR*, 2015, pp. 1439–1452.