ConsistencyTrack: A Robust Multi-Object Tracker with a Generation Strategy of Consistency Model

Lifan Jiang[a,b], Zhihui Wang[a,*], Siqi Yin[a], Guangxiao Ma[a], Peng Zhang[a], Boxi Wu[b]

[a] *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China*
[b] *School of Software Technology, Zhejiang University, Ningbo, China*

Abstract

Multi-object tracking (MOT) is a critical technology in computer vision, designed to detect multiple targets in video sequences and assign each target a unique ID per frame. Existed MOT methods excel at accurately tracking multiple objects in real-time across various scenarios. However, these methods still face challenges such as poor noise resistance and frequent ID switches. In this research, we propose a novel ConsistencyTrack, joint detection and tracking(JDT) framework that formulates detection and association as a denoising diffusion process on perturbed bounding boxes. This progressive denoising strategy significantly improves the model's noise resistance. During the training phase, paired object boxes within two adjacent frames are diffused from ground-truth boxes to a random distribution, and then the model learns to detect and track by reversing this process. In inference, the model refines randomly generated boxes into detection and tracking results through minimal denoising steps. ConsistencyTrack also introduces an innovative target association strategy to address target occlusion. Experiments on the MOT17 and DanceTrack datasets demonstrate that ConsistencyTrack outperforms other compared methods, especially better than DiffusionTrack in inference speed and other performance metrics. Our code is available at https://github.com/Tankowa/ConsistencyTrack.

Keywords:
Multi-object Tracking, Consistency Model, Joint Detection and Tracking, Denoising Diffusion Process, Inference Speed

*Corresponding author
Email addresses: lifanjiang@sdust.edu.cn,lifanjiang@zju.edu.cn (Lifan Jiang), zhihuiwangjl@gmail.com (Zhihui Wang), siqiyin@sdust.edu.cn (Siqi Yin), mgx@sdust.edu.cn (Guangxiao Ma), pengzhang_skd@sdust.edu.cn (Peng Zhang), boxiwu@zju.edu.cn (Boxi Wu)

1. Introduction

Multi-object tracking(MOT) is a critical task in computer vision [1, 2], enabling real-time localization and tracking of specific targets' positions, sizes, and motion states within video sequences. Typical targets include various categories such as pedestrians, vehicles, or animals [3, 4]. MOT algorithms take video sequences as input and output the targets' information, such as bounding boxes or trajectories.

Methodologies are primarily categorized into three paradigms: Tracking by detection (TBD) [5, 6, 7], joint learning of detection and embedding (JDE) [8, 9], and joint detection and tracking (JDT) [10, 11]. The TBD paradigm begins with object detection followed by data association, with typical association strategies such as SORT [12] and DeepSORT [7]. SORT uses Kalman filtering and the Hungarian algorithm for tracking, while DeepSORT enhances performance with deep learning-based appearance features. However, a notable drawback of TBD is its reliance on the initial detection accuracy. If object detection fails, the subsequent tracking will likely be compromised, especially in complex scenes with occlusions or overlapping objects. Advancing technology introduced the JDE paradigm, integrating feature extraction into the detector to eliminate the need for separate re-identification modules. However, a limitation of JDE is that it sometimes compromises detection quality due to the joint optimization of detection and feature extraction, which may lead to reduced performance in both areas under complex scenarios. JDT paradigm emerges from advancements in technology, seamlessly combining the detection and tracking stages to enhance efficiency and minimize computational overlap. However, the integration of detection and tracking in JDT can lead to challenges in distinguishing between closely spaced objects and maintaining consistent track identities in dynamic environments, where objects interact frequently and the scene changes rapidly. Therefore, the tracking paradigms still should be innovated to improve their tracking abilities.

In recent years, diffusion models [13, 14], also known as score-based generative models, have demonstrated significant effectiveness in various domains, such as object detection [15], image segmentation [16], and image generation [17]. A defining characteristic of diffusion models is their iterative sampling mechanism, systematically reducing noise from initial random vectors, thereby greatly enhancing the model's robustness during the training phase. Building on the principles of diffusion models, DiffusionTrack [11] has surpassed existing detectors in noise resistance, even exceeding the paradigm established by Transformer models [18]. However, its gradual denoising process, opposite to its iterative noise addition, imposes limitations on

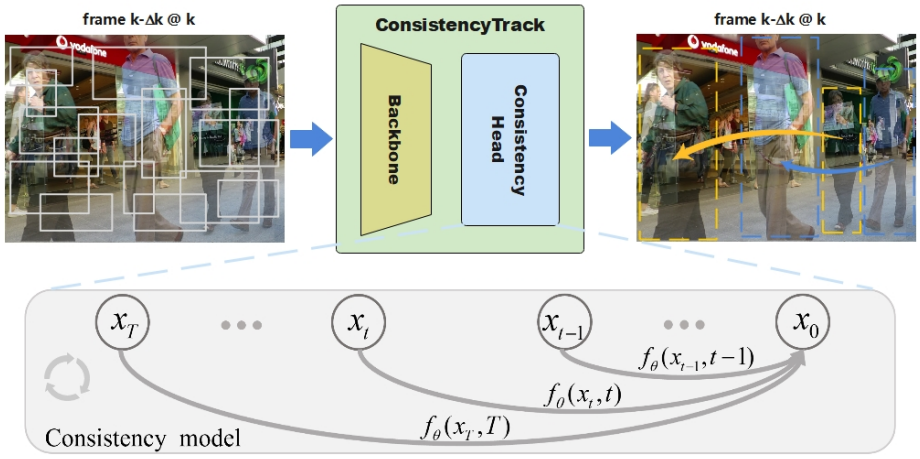

Figure 1: The denoising strategies of Consistency Model in the duty of MOT. ConsistencyTrack formulates object association as a denoising diffusion process from paired noise boxes to paired object boxes within two adjacent frames $(k - \Delta k, k)$. Here, $f_\theta(\cdot, \cdot)$ represents a one-step denoising process.

flexibility and computational efficiency. To adapt this model for real-world applications, its iterative processes still need further optimized. To address this challenge, we propose an innovative approach with a generation strategy of Consistency Model, denoted as ConsistencyTrack. Different with the foundational concepts of DiffusionTrack, the denoising strategy employed by Consistency Model is illustrated in Fig. 1. Notably, the self-consistency of Consistency Model allows denoising to be completed in a single step, significantly enhancing execution efficiency. Therefore, while maintaining detection accuracy, the number of denoising iterations can be significantly reduced.

As illustrated in Fig. 2, our approach leverages the ordinary differential equation (ODE) framework for probability flow (PF), akin to the continuous-time model utilized in DiffusionTrack. These models effectively guide sample paths, facilitating a seamless transition from the initial data distribution to a manageable noise distribution. ConsistencyTrack distinguishes itself by mapping any given point from an arbitrary time step back to the origin of its trajectory. This is made possible by the model's self-consistency feature, ensuring that points along the same trajectory correspond to the same starting point. This innovative method enables the generation of data samples by transforming random noise vectors through a single network evaluation, starting from the starting points of ODE trajectories. The proposed ConsistencyTrack achieves efficient iterative sampling, distinct from the extremely low execution efficiency of DiffusionTrack, thereby improving the cost-effectiveness

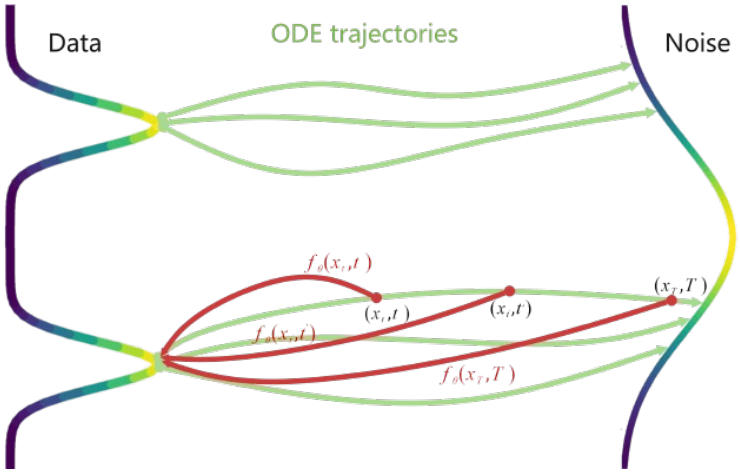

Figure 2: Consistency Model undergoes training process to establish a mapping that brings points along any trajectory of the PF ODE back to the origin of that trajectory [19]. The same as Fig. 1, $f_\theta(\cdot, \cdot)$ represents a one-step denoising process.

of the sampling process.

The proposed ConsistencyTrack innovatively integrates Gaussian noise into the center coordinates and dimensions extracted by the backbone network of bounding boxes, thereby generating corresponding noisy boxes. Subsequently, these generated noisy boxes are fed into a decoder for denoising prediction, primarily aligning them with ground truth (GT) boxes. It is noteworthy that to adapt ConsistencyTrack effectively to the JDT paradigm of MOT, two images at a fixed interval are simultaneously input into the network of ConsistencyTrack. This captures correlation information between instances of the same object across consecutive frames, thereby enhancing the model's ability for single-stage inference tracking.

Furthermore, we conducted rigorous evaluations of the proposed ConsistencyTrack's performance on the MOT17 and DanceTrack datasets. The experiments demonstrate that ConsistencyTrack exhibits robust noise resistance and fast inference speeds. Our work contributes to the field in the following aspects:

- ConsistencyTrack conceptualizes the process of object tracking as a generative denoising process and introduces a novel denoising paradigm. In contrast to the established paradigm in DiffusionTrack, which employs a very small number of iterations for noise addition and removal, our method represents a substantial advancement in enhancing the efficiency of the MOT task.

- In crafting the loss function for the proposed ConsistencyTrack, we aggregate the individual loss values at time steps $(t-1, t)$ subsequent to the model's

predictions to compute the total loss. This methodology guarantees that the mapping of any pair of adjacent points along the temporal dimension to the axis origin maintains the highest degree of consistency. This attribute mirrors the inherent self-consistency principle central to Consistency Model.

- We designed a novel target association strategy, distinct from DiffusionTrack within the JDT paradigm. This association strategy emphasizes the process of matching low-confidence detection boxes with tracking trajectories, significantly enhancing the ability to recognize occlusion issues and markedly improving performance metrics.

The structure of the paper is as follows: Section 2 presents a concise review of the development of one-stage JDT methods and the application of traditional diffusion models in tracking tasks, and discusses the foundational principles of Consistency Model. Section 3 delineates the specific methodologies for noise addition and removal within Consistency Model, elucidates the model's architecture, and provides essential details regarding the training and sampling methodologies. Section 4 details the empirical findings from evaluating ConsistencyTrack and conducts a comparative analysis against other leading models in the field. Finally, Section 5 encapsulates the salient features of the newly proposed ConsistencyTrack and contemplates avenues for future research.

2. Related works

2.1. One-stage JDT methods

In recent years, there have been several explorations into the one-stage paradigm, which combines object detection and data association into a single pipeline. Query-based methods, a burgeoning trend, utilize DETR [20] extensions for MOT by representing each object as a query regressed across various frames. Techniques such as TrackFormer [18] perform simultaneous object detection and association using concatenated object and track queries. TransTrack [21] employs cyclical feature passing to aggregate embeddings, while MeMOT [22] encodes historical observations to preserve extensive spatiotemporal memory.

Offset-based methods, in contrast, bypass inter-frame association and instead focus on regressing past object locations to new positions. Examples include Tracktor++ [22] for temporal realignment of bounding boxes, CenterTrack [23] for object localization and offset prediction, and PermaTrack [24], which fuses historical memory to reason target location and occlusion. TransCenter [25] further advances this

category by adopting dense representations with image-specific detection queries and tracking.

Trajectory-based methods extract spatial-temporal information from historical tracklets to associate objects. GTR [26] groups detections from consecutive frames into trajectories using trajectory queries, and TubeTK [27] extends bounding boxes to video-based bounding tubes for prediction. Both efficiently handle occlusion issues by utilizing long-term tracklet information.

2.2. DiffusionTrack

Diffusion models [13, 14, 15, 16] originate from randomly distributed samples and progressively reconstruct the desired data through a denoising process. As powerful tools, these models have achieved significant success across a range of fields, including computer vision, natural language processing, and audio signal processing. In the task of MOT, diffusion models have been adopted into a tracking task known as DiffusionTrack [11]. DiffusionTrack designs a novel tracker that performs tracking implicitly by predicting and associating the same object across two adjacent frames within the video sequence. This represents a groundbreaking application of Diffusion Model to the field of object detection. Building upon the foundations of DiffusionTrack, this work seeks to optimize the balance between detection accuracy and computational speed. We aim to enhance detection efficiency through a single-step processing approach, while preserving the essential benefits derived from the process of iterative sampling.

2.3. Consistency Model

Diffusion Model operates on an iterative generation process, which often results in limited execution efficiency, thereby restricting its applicability in real-time scenarios. To mitigate this limitation, OpenAI introduced Consistency Model, a novel class of generative models that can swiftly produce high-quality samples without the necessity for adversarial training. Consistency Model enables rapid one-step generation while also providing the flexibility for multi-step sampling to balance computational efficiency with the quality of generated samples. Additionally, it offers zero-shot data manipulation capabilities, including tasks such as image restoration, colorization, and super-resolution, without the need for task-specific training. This work formally acknowledges these capabilities and, for the first time, integrates Consistency Model in the field of MOT.

Table 1: Nomenclature with related notations.

Notation	Definition
t_r	A random time step in the range $[0, T]$
t	Current time step
T	Number of total time steps
Δt	Time step interval for sampling
k	The k-th frame in the video
Δk	Time interval of the selected frames
K	Number of total frames in the video
u_{roi}^k	RoI-features at frame k
q_{pro}^k	Self-attention output query
q_k	The object query
c_x^i/c_y^i	x/y-axis coordinate of the i-th box's center point
w^i/h^i	Width / Height of the i-th box
B_i^k	(c_x^i, c_y^i, w^i, h^i) of the i-th box at frame k
α_t/σ_t	Parameter in Denoiser at the t-th time step
θ	Model parameter
$F_\theta(\cdot,\cdot)$	A designed free-form deep neural network
$Split$	Split function description
$f_{\mathrm{BMM}}(\cdot,\cdot)$	Batch Matrix Multiplication function
$Linear(\cdot)$	Fully-connected layer
$\mathcal{N}(\cdot,\cdot)$	Normal distribution
$f_\theta(\cdot,\cdot)$	Final answer for Consistency Model
$p_\theta(\cdot,\cdot)$	Prediction function parameterized by θ
$c_{skip/out/in}(\cdot)$	Calculation factor for f_θ
$\lambda(\cdot)$	A positive weighting function
$\mathcal{L}$	Total loss function in training phase
$\mathcal{L}_{cls/L1/GIoU3d}$	Focal / L1 / GIoU3d loss item
$GIoU_{3d}(\cdot,\cdot)$	Three-dimensional Generalized Intersection over Union
$\lambda_{cls/L1/GIoU3d}$	Weight for Focal / L1 / GIoU3d loss item
$\sigma_{max/min}$	Maximum / Minimum threshold of noise parameter
σ_{data}	Noise parameter between σ_{min} and σ_{max}
ϵ	Randomly generated Gaussian noise
B_t	Random noise at the t-th time step in sampling
N	Batch size of concurrently processed samples
R	Number of regions analyzed within each sample
d	Dimension of feature
$r(\cdot)$	Generate random noise with given dimensions
$E(\cdot)$	Image feature extraction with backbone network

Table 1 – continued from previous page

Notation	Definition
$P_c(\cdot,\cdot)$	Prediction of Consistency Model in each time step
$nms(\cdot,\cdot)$	Non-max suppression (NMS) operation
N_{th}	Threshold of NMS operation
B_{th}	Threshold of Box-renewal operation
$dcm(\cdot,\cdot)$	Decoder of ConsistencyTrack with head network
$conc(\cdot,\cdot)$	Concatenate function
$score(\cdot)$	Estimation function of association score for each target
$frame(\cdot)$	Number counter of adjacent untracked frames for each target
n_{ss}	Number of sampling steps
N_{train}	Number of total proposed boxes in training phase
n_{rp}	Number of times the prior box repeats
n_p	Number of total proposed boxes in inference
n_r	Number of current proposed boxes
x_s	Padded box information at time axis origin
x_t	Noised box information at the t-th time step
x_b	Predicted box information in each time step
x_0	Predicted box information at time axis origin
$x_{box/cls/score}$	Predicted box coordinate / category / association scores
AP	Average Precision

2.4. Nomenclature

For the sake of clarity in the ensuing discussion, we provide a summary of the symbols and their corresponding descriptions as utilized in this study. This is encapsulated in Table 1, which meticulously outlines the nomenclature employed. The symbols encompass a variety of elements including training samples, components of the loss function, strategies for training, and metrics for evaluation, among others.

3. The proposed tracking method

In this section, we introduced the proposed tracking method with generation strategy of Consistency Model, denoted as ConsistencyTrack. This tracker is designed to perform the tracking duty implicitly by predicting and associating the same object across two adjacent frames within the video sequence. We first reviewed the pipeline of MOT, Diffusion Model, and Consistency Model. Finally, detailed discussions were provided on the training and inference procedures of the model.

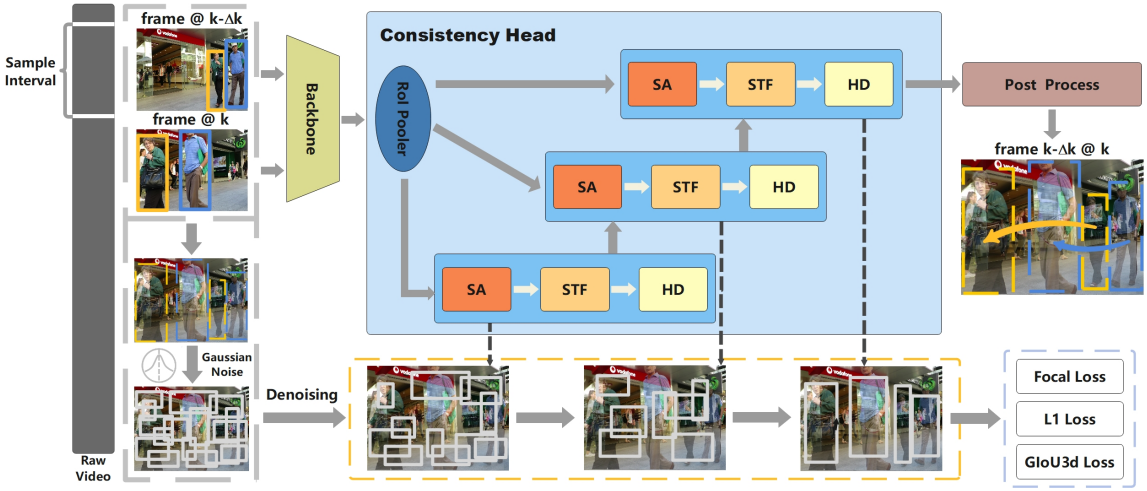

Figure 3: Training procedure of the proposed ConsistencyTrack. Features are extracted through the backbone network which extracts them from adjacent frames $(k - \Delta k, k)$ in a video sequence. Then, random Gaussian noise is added to the GT boxes according to the noise addition strategy of Consistency Model. These noisy boxes, with corresponding features, are processed by the RoI pooler and then input into the ConsistencyHead for iterative noise removal using three basic modules, ultimately yielding the final detection results. Each basic module contains a self-attention mechanism (SA), a Spatial-temporal fusion module (STF), and a correlation score head (HD). After the post process, the objects between adjacent frames $(k-\Delta k, k)$ are one-to-one associated with their matching scores.

3.1. Preliminaries

Multi-object tracking. The training samples of MOT are a set of input-target pairs (X_k, B_k, C_k) per k-th frame, where X_k is the input image, B_k and C_k are a set of bounding boxes and ID information for objects in the video on the k-th frame respectively. More specifically, we formulate the i-th box in the set B_k as $B_i^k = (c_i^x, c_i^y, w_i, h_i)$, where (c_i^x, c_i^y) is the center coordinates of the bounding box and (w_i, h_i) are width and height of that bounding box, i is the identity number respectively. Specially, $B_i^k = \emptyset$ when the i-th object is missing in X_k.

Diffusion Model. Diffusion models [13, 14, 16] emulate the image creation process through a sequence of stochastic diffusion steps. The core of diffusion models involves commencing with random noise and progressively refining it until it closely matches a sample from the target distribution. In the forward diffusion process, starting with a data point drawn from the real distribution, $x_0 \sim q(x)$, Gaussian noise is incrementally introduced over T steps with the following iterative process:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right), \tag{1}$$

where β_t schedules the noise for the current timestep $t \in (1, T]$. In the reverse diffusion process, the random noise $x_T \sim \mathcal{N}(0, I)$ is denoised into the target distribution by modeling $q(x_{t-1}|x_t)$. At each reverse step t, the conditional probability distribution is represented approximately by a network $\epsilon_\theta(x_t, t)$ using the timestep t and the previous output x_t as input:

$$x_{t-1} \sim p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right), \beta_t I\right), \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Through iterative operations, the noise in the current state is gradually reduced, eventually bringing it close to a real data point when approaching the original timestep of sample x_0.

Consistency Model. Within the framework of Consistency Model which utilizes deep neural networks, two cost-effective methodologies are investigated for enforcing boundary conditions. Let $F_\theta(x, t)$ represent a free-form deep neural network with the input x. The first method directly parameterizes Consistency Model as:

$$f_\theta(x, t) = \begin{cases} x, & \text{if } t = \tau, \\ F_\theta(x, t), & \text{if } t \in (\tau, T), \end{cases} \tag{3}$$

where τ is an integer in the range $[0, T-1]$. The second method parameterizes Consistency Model by incorporating skip connections and is formalized as follows:

$$f_\theta(x, t) = c_{skip}(t)x + c_{out}(t)F_\theta(x, t), \tag{4}$$

where $c_{skip}(t)$ and $c_{out}(t)$ are differentiable functions [19], satisfying $c_{skip}(\tau) = 1$ and $c_{out}(\tau) = 0$. By employing this construction, Consistency Model becomes differentiable at $t = \tau$, provided that $F_\theta(x, t)$, $c_{skip}(t)$, and $c_{out}(t)$ are all differentiable. This differentiability is crucial for the training process of continuous-time Consistency Models.

3.2. Architecture

The overall framework of our ConsistencyTrack is visualized in Fig. 3, which consists of two major components: a feature extraction backbone and a data association denoising head (diffusion head). The feature extraction backbone processes two adjacent input images $(X_{k-\Delta k}, X_k)$ to extract deep feature representations. The data association denoising head uses these features as conditions, rather than the raw images, to progressively refine paired association box predictions from paired noise boxes. In our setup, data samples consist of paired bounding boxes $z_0 = (B_{k-\Delta k}, B_k)$, where $z_0 \in \mathbb{R}^{N\times 8}$. A neural network $f_\theta(z_s, s, X_{k-\Delta k}, X_k)$ for $s = \{0, \cdots, K\}$ is trained

to predict z_0 from paired noise boxes z_s, conditioned on the corresponding two adjacent images $(X_{k-\Delta k}, X_k)$. The corresponding association confidence score S are produced accordingly. If $X_{k-\Delta k} = X_k$, the task of MOT degenerates into an object detection problem. This consistent property allows ConsistencyTrack to simultaneously solve both the tasks of object detection and tracking. It is noteworthy that Δk is set to 1 only during the tracking of matching process, seen as Fig. 5.

Backbone. We adopt the YOLOX backbone [28], which utilizes Feature Pyramid Networks (FPN) [29] to extract high-level features from two adjacent frames. These features are then fed into the diffusion head for the denoising process of conditioned data association.

Diffusion head. The diffusion head takes a set of proposal boxes as input to crop RoI features from the backbone's feature map. These RoI features are processed through different blocks to perform box regression, classification, and the prediction of association confidence score. To address the object tracking problem, each block of the diffusion head incorporates a Spatial-Temporal Fusion module (STF) and an association score head.

Spatial-Temporal Fusion Module. STF module is proposed to enable temporal information exchange between paired boxes across two consecutive frames, facilitating complete data association. Given the RoI features $\{u_{roi}^{k-\Delta k}, u_{roi}^{k}\} \in \mathbb{R}^{N\times R\times d}$ for two consecutive timesteps, where N is the batch size, R is the number of regions, d is the feature dimension, and the self-attention output queries $\{q_{pro}^{k-\Delta k}, q_{pro}^{k}\} \in \mathbb{R}^{N\times d}$. Then, we proceed with the following transformations:

1. **Linear Transformation and Splitting:** Each query q_{pro}^{i} is first transformed using a linear projection, and the result is then split into two separate equidimensional tensors:
$$P_1^i, P_2^i = \mathit{Split}(\mathit{Linear}(q_{\text{pro}}^i)). \tag{5}$$
2. **Batch Matrix Multiplications:** RoI features from the two timesteps are concatenated and then subjected to two consecutive batch matrix multiplications (BMM) with the split parts:
$$\mathit{feat} = f_{\text{BMM}}\left(f_{\text{BMM}}\left(\mathit{conc}\left(u_{\text{roi}}^i, u_{\text{roi}}^j\right), P_1^i\right), P_2^i\right). \tag{6}$$
3. **Final Linear Transformation:** The resulting feature tensor from the BMM operations is further processed using another linear transformation to produce the object queries for the current block:
$$q^i = \mathit{Linear}(\mathit{feat}), \quad q^i \in \mathbb{R}^{N\times d}. \tag{7}$$

4. **Index Relationships:** The indices (i, j) are taken from the adjacent time pairs $[(k - \Delta k, k), (k, k - \Delta k)]$, indicating the operation considers transitions between consecutive timesteps, in both forward and backward directions.

Association Score Head. In addition to the box head and class head, we introduce an additional association score head. This head utilizes the fused features of paired boxes, obtained from the spatial-temporal fusion module, and feeds them into a linear layer. The output of this head provides the confidence score for data association. It determines whether the paired box outputs belong to the same object during the subsequent post-processing of NMS.

3.3. Model Training

During the training phase, the algorithm randomly selects a pair of frames from the video sequence as input to the model. First, GT boxes in the images are supplemented to a total number of N_{train}. Then, based on the noise addition strategy of Consistency Model, random noise is added to the original GT boxes in both frames. All these noised boxes are then fed into the model for the denoising process. Finally, model extracts the association relationships between the instance boxes in the two adjacent frames, calculates the loss, and performs the backpropagation operation. The detailed process of the training phase is described in Algorithm 1.

Algorithm 1 Training loss of ConsistencyTrack

Input: Images $(X_{k-\Delta k}, X_k)$ with GT boxes at two adjacent frames $(k - \Delta k, k)$
Output: Loss $\mathcal{L}_{t_r,t_{r-1}}$ per iteration

for each iteration **do**
  Sample $(X_{batch1}, X_{batch2}) \in (X_{k-\Delta k}, X_k)$
  Extract features $E(X_{batch1}, X_{batch2})$
  Pad (X_{batch1}, X_{batch2}) with GT boxes and features as x_s
  Generate a random timestep $t_r \in [0, T]$
  /* Calculate noise parameters */
  Calculate $(\sigma_{t_{r-1}}, \sigma_{t_r})$ by Eqn. (8)
  Add noise to x_s by Eqn. (9) as x_{t_r}
  Predict $x_{t_{r-1}}$ with $(x_{t_r}, \sigma_{t_r}, \sigma_{t_{r-1}}, x_s)$
  $d_{t_{r-1}} \leftarrow dcm(x_{t_{r-1}}, \sigma_{t_{r-1}})$
  $d_{t_r} \leftarrow dcm(x_{t_r}, \sigma_{t_r})$
  $\mathcal{L}_{t_r,t_r-1} \leftarrow \mathcal{L}(d_{t_{r-1}}, G) + \mathcal{L}(d_{t_r}, G)$
  return Loss $\mathcal{L}_{t_r,t_{r-1}}$
end for

GT boxes padding. In open-source benchmarks for MOT, as cited in [30, 31], there is typically a variance in the number of annotated instances across images. To address this inconsistency, we implement a padding strategy by introducing auxiliary boxes around the GT boxes. This ensures that the total number of boxes reaches a predetermined amount, N_{train}, during the training phase. These padded instances are denoted as x_s, representing the original padded samples. For the i-th GT box, denoted as b_i, Gaussian noise is applied to its four parameters (c_x^i, c_y^i, w^i, h^i) at a randomly selected timestep t.

Box corruption. The range of the noised box at the t-th timestep is constrained. Initially, the scale factor of the noise is determined as follows:

$$\sigma_t = \left(\sigma_{max}^{1/\rho} + \frac{t}{T-1} \cdot \left(\sigma_{min}^{1/\rho} - \sigma_{max}^{1/\rho}\right)\right)^{\rho}. \tag{8}$$

Subsequently, noise is introduced to the original padded sample x_s:

$$x_t = x_s + \epsilon \cdot \sigma_t, \tag{9}$$

where ϵ denotes randomly generated Gaussian noise. Finally, the range of the noised box is restricted by:

$$x_t \leftarrow \frac{c_{in}(t)}{2} \cdot x_t, \tag{10}$$

where $c_{in}(\cdot)$ represents the scale factor of the noised box and is defined as:

$$c_{in}(t) = \frac{1}{\sqrt{{\sigma_t}^2 + \sigma_{data}^2}}. \tag{11}$$

This formulation ensures that the noise scale factor is properly adjusted across the time steps, and that the noised box remains within the specified value ranges.

Loss Function. The loss function used to evaluate the predicted bounding boxes follows the framework established by DiffusionTrack, which incorporates both $\mathcal{L}_{\text{L1}}$ loss and $\mathcal{L}_{\text{GIoU3d}}$ loss item. The former represents the standard L1 loss item, while the latter represents the Generalized Intersection over Union (GIoU) loss. Notably, we extend the definition of GIoU to make it compatible with the paired boxes design. 3D GIoU and 3D IoU are the volume-extended versions of the original area-based ones. Additionally, focal loss $\mathcal{L}_{cls}$ is used to evaluate the classification of each predicted bounding box. To balance the relative impact of each loss component, a positive real-valued weight $\lambda_{\text{cls/L1/GIoU3d}} \in \mathbb{R}^+$ is assigned to each loss item. Therefore, the total loss function is formulated at the t-th timestep as follows:

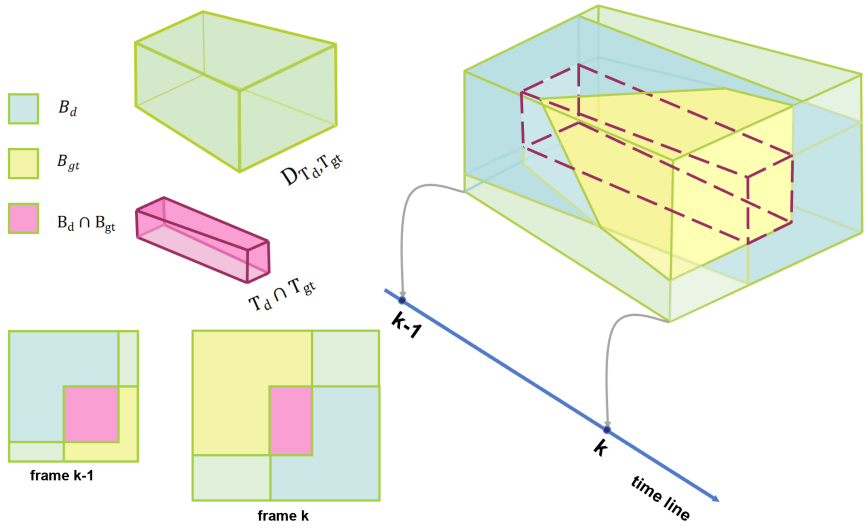

Figure 4: Visualization of the computation methodology for 3D GIoU. The volumetric intersection and the minimal bounding volume between target representations across consecutive frames are characterized as square frustums.

$$\mathcal{L}_t = \lambda_{cls} \cdot \mathcal{L}_{cls_t} + \lambda_{L1} \cdot \mathcal{L}_{L1_t} + \lambda_{GIoU3d} \cdot \mathcal{L}_{GIoU3d_t}, \tag{12}$$

with

$$\mathcal{L}_{GIoU3d} = 1 - GIoU_{3d}(T_d, T_{gt}), \tag{13}$$

where T_d and T_{gt} are square frustums consisting of estimated detection boxes and ground-truth bounding boxes for the same target in two adjacent frames respectively.

As shown in Fig. 4, 3D GIoU of paired predicted boxes is defined as:

$$\begin{aligned} GIOU_{3D}(T_d, T_{gt}) = &- \frac{\left|\sum_{i=k-1}^{k} \left(Area(D_{B_d^i, B_{gt}^i}) - Area(B_d^i \cup B_{gt}^i) \right)\right|}{\left|\sum_{i=k-1}^{k} Area(D_{B_d^i, B_{gt}^i})\right|} \\ &+ IOU_{3D}(T_d, T_{gt}), \end{aligned} \tag{14}$$

where $D_{B_d^i, B_{gt}^i}$ represents the smallest convex hull that includes the estimated detection box B_d and the ground-truth bounding box B_{gt} at frame i. T_d and T_{gt} are similar to Eqn.(13). The intersection $T_d \cap T_{gt}$ also forms a square frustum, encompassing the overlaps $B_d^{k-1} \cap B_{gt}^{k-1}$ and $B_d^k \cap B_{gt}^k$.

Leveraging the self-consistency property of Consistency Model, the perturbed bounding boxes associated with sample x_s at consecutive timesteps $t-1$ and t

undergo a joint denoising process. The corresponding loss values are accumulated to derive the final comprehensive loss:

$$\begin{aligned}\mathcal{L} =&\lambda_{cls} \cdot (\mathcal{L}_{cls_{t-1}} + \mathcal{L}_{cls_t}) + \lambda_{L1} \cdot (\mathcal{L}_{L1_{t-1}} + \mathcal{L}_{L1_t}) \\ &+ \lambda_{GIoU3d} \cdot (\mathcal{L}_{GIoU3d_{t-1}} + \mathcal{L}_{GIoU3d_t}).\end{aligned} \tag{15}$$

3.4. Inference

The inference mechanism employed in ConsistencyTrack resembles that of DiffusionTrack, utilizing a denoising sampling strategy that starts from initial bounding boxes, similar to the processed noisy samples during the training phase, and progresses to the final object detections. In the absence of GT annotations, these initial bounding boxes are randomly generated following a Gaussian distribution. The model iteratively refines these predictions through multiple sampling steps. Ultimately, the final detections include refined bounding boxes and category classifications. After completing all iterative sampling steps, the predictions undergo enhancement through a post-processing module, resulting in the final outcomes. The detailed procedure is outlined in Algorithm 2. Regarding the detailed demonstration of the inference phase, refer to Fig. 5. Intuitive comparison of ConsistencyTrack and DiffusionTrack during the inference phase, as illustrated in the following Fig. 9.

Algorithm 2 Inference of ConsistencyTrack

Input: Images $(X_{k-\Delta k}, X_k)$ at the frame pair $(k-\Delta k, k)$, total timestep T, the number of sampling steps n_{ss}

Output: Final predictions $nms(x_{box}, x_{cls}, x_{score})$

/* Initialization */

$\Delta t = T/n_{ss}$

Generate random noise B_0 with the dimensions of presupposed boxes' amount

Extract features $E(X_{k-\Delta k}, X_k)$

/* Iterative operation */

for $t = 0$ **to** $T-1$ **step** Δt **do**

    Calculate σ_t by Eqn. (8)

    $x_0, x_b, x_{cls}, x_{box}, x_{score} \leftarrow P_c(E(X_{k-\Delta k}, X_k), B_t)$

    Perform Box-renewal operation for x_b and x_0

    $\nabla_\sigma x \leftarrow (x_b - x_0)/\sigma_t$

    $B_t \leftarrow x_b + \nabla_\sigma x(\sigma_{t+\Delta t} - \sigma_t)$

    /*Supplement new proposals */

    $B_t \leftarrow \text{conc}(B_t, r([1, n_p - n_r, 4]) \cdot \sigma_{t+\Delta t})$

end for

12: **return** $nms(x_{box}, x_{cls}, x_{score})$

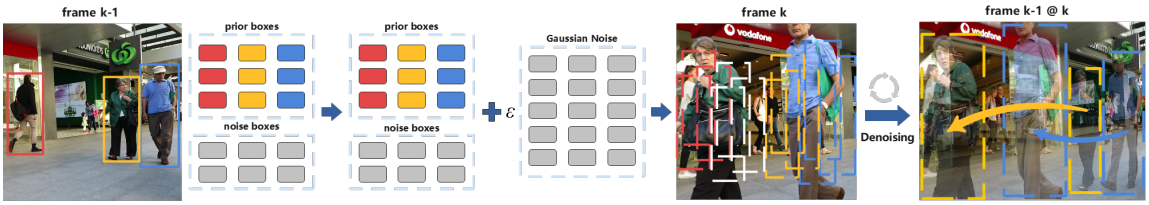

Figure 5: Illustration of the inference process with Consistency Model. First, padding repeated prior boxes with Gaussian boxes until the predefined number N_{test} is reached. Then, adding Gaussian noise to the input boxes according to $x_t = x_s + \epsilon \cdot \sigma_t$ under the control of ϵ. Finally, obtaining tracking results through a denoising process with one sampling step of Consistency Model.

3.5. Target association strategy

Our target association process adopts the JDT paradigm, and the entire object matching process no longer involves additional feature matching. Initially, paired detection boxes are filtered based on the association scores obtained by the detector, retaining only those detection boxes with higher association scores. This is a specific embodiment of the JDT paradigm. Subsequently, the detection boxes are classified into high and low confidence for separate tracking matching. To address potential occlusions, a simple Kalman filter is implemented to reassociate lost objects. The pseudo-code of ConsistencyTrack is listed in Algorithm 3.

Algorithm 3 ConsistencyTrack

Input: Video sequence V, a single track m, consistency track CT, association score threshold τ_{conf}, detection score threshold τ_{det}, track score threshold τ_{track}, number of boxes for association N_a, the upper threshold n_{lost} of adjacent frame amounts before lost, the high / low confidence detection boxes detected from the first half of the input information $D_{lpre/spre}$ and the second half of the input information $D_{lcur/scur}$, the intermediate parameter for returned collection $T_{act\_remain/lost\_remain/rm\_remain}$, trajectories with lost tags in this tracking match T_{lost_remain}, trajectories with remove tags in this tracking match T_{rm_remain}, trajectories waiting to be activated and already activated trajectories being updated during this tracking match T_{act_remain}.

Output: Tracked targets' status $T_{activated/unactivated/lost/remove}$ per frame

1: **for** frame (u_{k-1}, u_k) in V **do**
2: $\quad D_k \leftarrow CT(u_{k-1}, u_k)$

$D_{pre}, D_{cur}, D_{new} \leftarrow \emptyset$
for (idx, d_{k-1}, d_k) in D_k **do**
  if $score(d_k) > \tau_{conf}$ **then**
    if $idx < N_a$ **then**
      $D_{pre} \leftarrow D_{pre} \cup \{d_{k-1}\}$
      $D_{cur} \leftarrow D_{cur} \cup \{d_k\}$
    else
      $D_{new} \leftarrow D_{new} \cup \{d_{k-1}, d_k\}$
    end if
  end if
end for
/* Partitioning detection boxes */
for (idx, d_{k-1}, d_k) in D_{pre}, D_{cur} **do**
  if $score(d_{k-1}) < \tau_{track}$ & $score(d_{k-1}) > 0.1$ **then**
    $D_{spre} \leftarrow d_{k-1}$
  end if
  if $score(d_{k-1}) > \tau_{track}$ **then**
    $D_{lpre} \leftarrow d_{k-1}$
  end if
  if $score(d_k) < \tau_{track}$ & $score(d_k) > 0.1$ **then**
    $D_{scur} \leftarrow d_k$
  end if
  if $score(d_{k-1}) > \tau_{track}$ **then**
    $D_{lcur} \leftarrow d_k$
  end if
end for
for (idx, d_{k-1}, d_k) in D_{new} **do**
  if $score(d_{k-1}) \in (0.1, \tau_{track})$ & $score(d_k) \in (0.1, \tau_{track})$ **then**
    $D_{snew} \leftarrow \{d_{k-1}, d_k\}$
  end if
  if $score(d_{k-1}) > \tau_{track}$ & $score(d_k) > \tau_{track}$ **then**
    $D_{lnew} \leftarrow \{d_{k-1}, d_k\}$
  end if
end for
/* Partitioning tracking trajectory */
$T_{lost} \leftarrow \{m \in T_{activated} \mid m \text{ is lost}\}$
$T_{non\_lost} \leftarrow \{m \in T_{activated} \mid m \text{ is not lost}\}$
/* First target association*/

Associate T_{non_lost} and D_{lpre} using IoU Similarity
$U_{track} \leftarrow \{m \in T_{non\_lost} \mid m \text{ not matched}\}$
$U_{detection} \leftarrow \{d \in D_{pre} \mid d \text{ not matched}\}$
$T_{act} \leftarrow \{m \in T_{non\_lost} \mid m \text{ matched}\}$
Associate T_{act} and D_{lnew} using IoU Similarity
$U_{lnew} \leftarrow \{d \in D_{lnew} \mid d \text{ not matched with } T_{act}\}$
$T_{act\_remain} \leftarrow \{m \in T_{act}\}$
/* Second target association*/
Associate T_{lost} and U_{lnew} using IoU Similarity
$T_{act\_remain} \leftarrow \{m \in T \mid m \text{ matched with } U_{lnew}\}$
$T_{unactivated} \leftarrow \{d \in U_{lnew} \mid d \text{ unmatched with } T_{lost}\}$
$U_{track\_now} \leftarrow \{m \in T \mid m \text{ unmatched with } U_{lnew}\}$
$T_{rm\_remain} \leftarrow \{m \in T \mid m \text{ unmatched with } U_{lnew} \text{ and } frame(m) > n_{lost}\}$
/* Third target association*/
Associate U_{track} and D_{sper} using IoU Similarity
$U_{track} \leftarrow \{m \in U_{track} \mid m \text{ not matched}\}$
$U_{detection} \leftarrow \{d \in D_{sper} \mid d \text{ not matched}\}$
$T_{act} \leftarrow \{m \in U_{track} \mid m \text{ matched}\}$
Associate T_{act} and D_{snew} using IoU Similarity
$D_{scur} \leftarrow \{D_{snew} \mid d \text{ matched with } T_{act}\}$
$T_{act\_remain} \leftarrow \{m \in T_{act}\}$
$T_{lost\_remain} \leftarrow \{m \in T \mid m \text{ not matched with } U_{track}\}$
/* Fourth target association*/
Associate $T_{unactivated}$ and $U_{detection}$ using IoU Similarity
$T_{act\_remain} \leftarrow \{m \in T \mid m \text{ matched with } U_{detection} \text{ and } m \text{ is activated}\}$
$T_{refind} \leftarrow \{m \in T \mid m \text{ matched with } U_{detection} \text{ and } m \text{ is not activated}\}$
$T_{unactivated} \leftarrow \{d \in U_{detection} \mid d \text{ unmatched with } T_{unactivated}\}$
/* Update tracking status*/
$T_{rm\_remain} \leftarrow \{m \in T_{lost} \mid frame(m) > n_{lost}\}$
$T_{activated} \leftarrow T_{activated} \cup T_{refind} \cup T_{act\_remain}$
$T_{lost} \leftarrow ((T_{lost} \setminus T_{refind}) \cup T_{lost\_remain}) \setminus T_{rm\_remain}$
$T_{remove} \leftarrow T_{rm\_remain}$
return $T_{activated}, T_{unactivated}, T_{lost}, T_{remove}$
end for

4. Experiments

In this section, the performance of the proposed ConsistencyTrack is evaluated on two popular datasets: MOT17 and DanceTrack [32, 33, 34, 35]. Firstly, the noise

robustness of ConsistencyTrack is tested through experiments. Then, the proposed ConsistencyTrack framework is compared with a series of established MOT models on several evaluation indicators. Finally, ablation studies are conducted to compare the optimal parameters of the ConsistencyTrack model, highlighting the significant efficiency advantages of our proposed model over DiffusionTrack.

MOT17 Dataset. The MOT17 dataset [32, 33] is a widely used benchmark for MOT tasks, consisting of 14 challenging video sequences from various indoor and outdoor environments. It provides detailed annotations for object bounding boxes and identities, allowing for the evaluation of tracking accuracy using metrics like MOTA and MOTP. The dataset is known for its diversity and complexity, including scenarios with occlusions, varying lighting conditions, and dense crowds.

DanceTrack Dataset. The DanceTrack dataset [34, 35] is designed to evaluate tracking algorithms in dynamic and complex scenarios, specifically focusing on dance performances. It includes sequences with fast, non-linear movements and frequent occlusions. The dataset provides detailed annotations for dancers, including bounding boxes and identities, and uses standard tracking metrics like MOTA and MOTP to assess performance. DanceTrack is particularly challenging due to the high-speed and intricate interactions among dancers.

4.1. Implementation details

We adopted the pre-trained YOLOX detector from DiffusionTrack [11], and trained ConsistencyTrack on the training sets of MOT17 and DanceTrack separately. For MOT17, the training schedule contains 40 training epochs of detection on the combined datasets (includes MOT17, CrowdHuman, Cityperson, and ETHZ), and 40 training epochs solely on MOT17 for tracking. For DanceTrack, no additional training data were used, and the model was trained by 60 epochs. During the detection and tracking training phases, we also employed data augmentation techniques such as Mosaic [36] and Mixup [37]. Each training sample (frame pair) was directly sampled within each video with a frame interval of $\Delta k = 5$. The input image size was resized to 1440×800. The AdamW optimizer [38] was used with an initial learning rate of 1e-4, which decreased according to a cosine function with a final reduction factor of 0.1. We used a warm-up learning rate of 2.5e-5 with a warm-up factor of 0.2 for the first epoch. The model was trained on a single NVIDIA GeForce RTX A100 GPU with FP32 precision and a constant seed for all experiments. The mini-batch size was set to 3, with each GPU hosting two batches with $N_{train} = 500$. Our approach was implemented in Python 3.8 with PyTorch 1.10.

4.2. Main Properties

The core characteristic of ConsistencyTrack is its self-consistency, which ensures that the mapping effect from any point along the time axis back to the origin remains relatively stable. This stability indicates that once the model is adequately trained, it allows for flexible adjustment of the number of sampling steps n_{ss} during inference. By appropriately increasing the sampling time steps, ConsistencyTrack can enhance its detection accuracy. Consequently, in different situations, the model can adjust the number of sampling steps according to specific requirements, thereby balancing accuracy and algorithmic efficiency. Additionally, the noise augmentation during the training process significantly improves the model's robustness against noise.

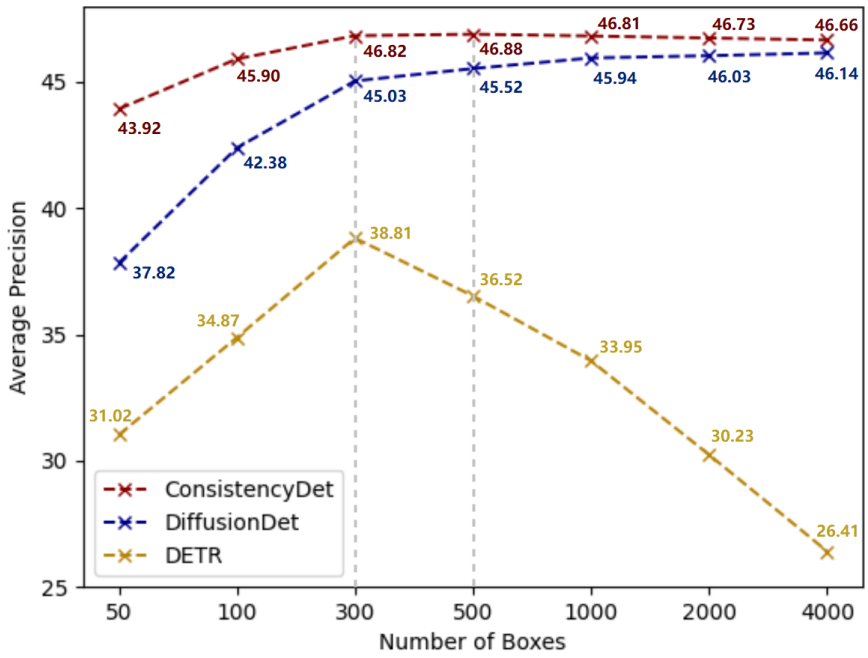

Figure 6: Performance comparisons of ConsistencyDet and DETR on COCO val dataset with increasing number of noisy boxes [39].

Robustness to detection perturbation. To rigorously assess the robustness of ConsistencyTrack to noise during the detection phase, we independently trained and tested its detection component, ConsistencyDet [39], on the MS-COCO dataset. This evaluation was benchmarked against leading detectors such as DiffusionDet and DETR, whose performance metrics are derived from [15]. As depicted in Fig. 6, ConsistencyDet exhibits a marked enhancement in performance correlating with the incremental inclusion of bounding boxes, achieving stability at $n_p > 300$ and peaking

at $n_p = 500$. In contrast, DETR reaches its maximum AP at $n_p = 300$, thereafter experiencing a precipitous decline, notably decreasing to 26.4% at $n_p = 4000$, a 12.4% drop from its highest AP of 38.8%. Although DiffusionDet also improves as more boxes are considered, its performance consistently trails behind that of ConsistencyDet. Consequently, ConsistencyDet not only proves more robust against noise but also showcases superior transferability and generalizability across diverse scenarios involving variable object counts.

Dynamic boxes. Once the model is trained, it can be utilized by varying the number of boxes and the sampling time steps during inference. Consequently, a single ConsistencyTrack can be deployed across multiple scenarios, achieving the desired speed-accuracy trade-off without the need for retraining the network. In Fig. 7, we evaluate ConsistencyTrack with 1000, 2000, and 4000 proposal boxes by increasing n_{ss} from 1 to 6. The results indicate that highest MOTA in ConsistencyTrack can be achieved by increasing the number of random boxes. Moreover, the highest MOTA and IDF1 is achieved when $n_{ss} = 2$, this aligns with the one-step mapping characteristic of Consistency Model. Note that when $n_{ss} > 2$, the MOTA metric remains in a state of oscillating fluctuations, but its peak is lower than the case of $n_{ss} = 2$.

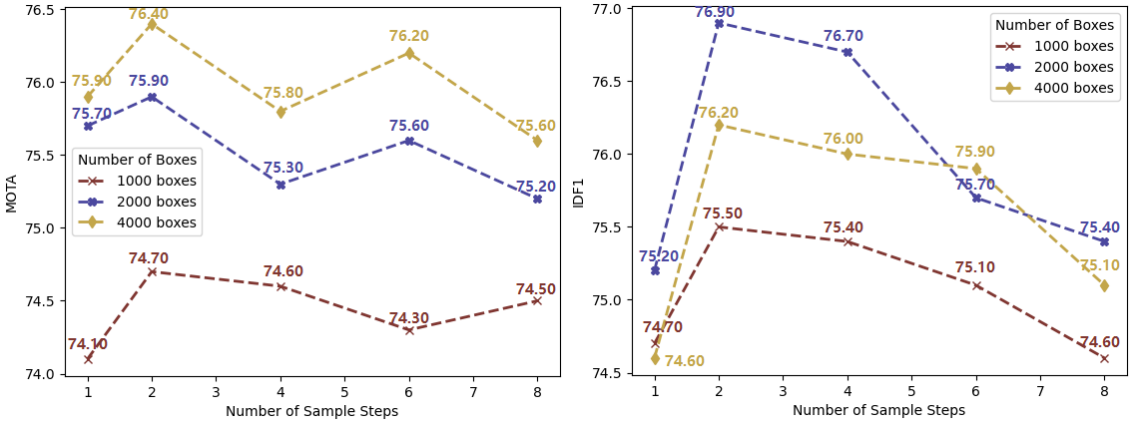

Figure 7: The performance of ConsistencyTrack is valuated on the MOT17 val-half set with different numbers of proposal boxes and different numbers of sampling time steps.

4.3. Simulation Analysis

The performance of ConsistencyTrack is evaluated against other tracking methods [22, 23, 27, 40, 41] on the MOT17 and DanceTrack datasets. The proposed ConsistencyTrack's screenshots of sampled tracking results on the MOT17 and DanceTrack

datasets are shown in Fig. 8. This subsection provides an analysis of the simulation results.

MOT17 dataset. The test performances on the MOT17 dataset are shown in Table 2. The proposed ConsistencyTrack outperforms in several key metrics, achieving the best scores in MOTA (69.9%), IDF1 (65.7%), HOTA (54.4%), and DetA (58.2%). Compared to UTM, which uses the same public detections, the proposed ConsistencyTrack improves MOTA and IDF1 by 4.4% and 0.6%, respectively. Additionally, ConsistencyTrack outperforms PCL in several metrics, such as IDF1, with an improvement of 4.5%, demonstrating better tracking consistency and accuracy. ConsistencyTrack also excels in Mostly Tracked targets (MT) and Minimal Localization error (ML), with scores of 907 and 428, respectively, indicating strong capabilities in target location and its error control. Compared to other methods like CJTracker and TrajE, the proposed method shows significant advantages across multiple metrics, demonstrating that ConsistencyTrack offers superior overall accuracy and reliability.

Table 2: Performance comparison on MOT17 test dataset on several metrics

Methods	MOT17										
	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN↓	AssA↑	DetA↑	IDs↓	Frag↑
Tracktor++v2[22]	56.3	55.1	/	498	831	**8866**	235449	/	/	1987	/
TubeTK*[27]	63.0	58.6	48.0	735	468	27060	177483	45.1	51.4	4137	5727
CTTrack17[23]	67.8	64.7	52.2	816	579	18498	160332	51.0	53.8	3039	6102
CJTracker[40]	58.7	58.2	48.4	621	909	32448	197790	48.0	49.1	2877	5031
TrajE[42]	67.4	61.2	49.7	820	587	18652	161347	46.6	53.5	4019	6613
Sp_Con [43]	61.5	63.3	50.5	622	754	14056	200655	52.0	49.2	2478	5079
PCL[44]	58.8	61.2	49.0	612	837	12072	218912	51.1	47.2	**1219**	**2197**
UTM[45]	63.5	65.1	52.5	881	635	33683	170352	**53.2**	52.2	1686	2562
ConsistencyTrack	**69.9**	**65.7**	**54.4**	**907**	**428**	24186	**142145**	51.2	**58.2**	3774	5854

[1] Results of MOTA/IDF1/HOTA/AssA/DetA are percentage data (%).
[2] Bold font indicates the best performance while underlined font indicates the second best.

DanceTrack dataset. In Table 3, we compared the ConsistencyTrack method with other traditional MOT methods on the DanceTrack validation set. Overall, our method demonstrated a balanced performance across various metrics and achieved a significant lead in the MOTA metric, reaching 88.1%, far surpassing the metrics of the method in second place. Due to the inherent unfairness in comparing One-Stage strategy object tracking algorithms with those of other strategies, we chose to compare our algorithm with CenterTrack [23] and FairMoT [46] on DanceTrack test set. The results are shown in Table 4. The experimental results indicate that, except for the DetA metric, which is slightly lower than CenterTrack, all other metrics are higher than both CenterTrack and FairMoT.

Table 3: Performance comparison on DanceTrack val set

Methods	DanceTrack				
	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
IoU[22]	44.7	**79.6**	25.3	<u>87.3</u>	36.8
DeepSORT[7]	**45.8**	70.9	**29.7**	87.1	**46.8**
MOTDT[47]	39.2	68.8	22.5	84.3	39.6
ConsistencyTrack	<u>45.5</u>	<u>77.7</u>	<u>26.9</u>	**88.1**	<u>43.4</u>

[1] Results are all percentage data (%).
[2] Bold font indicates the best performance while underlined font indicates the second best.

Table 4: Performance comparison on DanceTrack test set

Methods	DanceTrack				
One-Stage	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterTrack[23]	41.8	**78.1**	22.6	86.8	35.7
FairMOT[46]	39.7	66.7	23.8	82.2	40.8
ConsistencyTrack	**42.3**	76.4	**25.4**	**87.8**	**41.2**

[1] Results are all percentage data (%).
[2] Bold font indicates the best performance.

4.4. Ablation studies

Figure 8: Screenshots of sampled tracking results on the proposed ConsistencyTrack on MOT17 and DanceTrack datasets. The frame numbers corresponding to the images are marked in the upper left corner.

Comprehensive ablation studies were conducted to elucidate the characteristics of the proposed ConsistencyTrack on the MOT17 val-half set. These simulations utilized YOLOX as the primary backbone, with no further modifications or enhancements specified.

Proportion of prior information. In contrast to object detection, which operates without prior knowledge of object positions, MOT benefits from prior frame information. By incorporating this prior knowledge, we can adjust the proportion of such information in the construction of N_{test} proposal boxes by repeating the previous frame's boxes. In the experiments, we tested the impact of $n_{rp} \in \{1, 2, \cdots, 10\}$ on metrics such as MOTA, IDF1, and IDP, with the specific results for $n_{rp} = 6/8/10$ presented on the left of Table 5. The results indicate that setting the Repeat parameter to 8, which involves repeating the prior boxes eight times from the $(k-1)$-th frame, yields the best performance.

Box-renewal threshold. The right column of Table 5 describes the performances with varied threshold B_{th} on the metrics of MOTA, IDF1, and IDP. The

Table 5: Performance comparison with varied thresholds of Box-renewal and repeat times

n_{rp}	MOTA↑	IDF1↑	IDP↑
6	<u>75.5</u>	**76.6**	**83.7**
8	**75.8**	76.2	<u>82.9</u>
10	75.2	<u>76.4</u>	82.8

B_{th}	MOTA↑	IDF1↑	IDP↑
0.5	**75.8**	75.3	82.0
0.6	**75.8**	**76.2**	<u>82.9</u>
0.7	75.4	<u>76.1</u>	**83.1**

[1] Results are all percentage data (%).
[2] Bold font indicates the best performance while underlined font indicates the second best.
[3] Left table represents the proportion of prior information results and right table represents Box-renewal results.

case $B_{th} = 0$ signifies that no threshold is applied. Analysis of the MOT17 validation set indicates that a threshold of 0.6 obtains a slightly better performance, compared to other thresholds.

Accuracy vs. speed. In Table 6, the inference speeds of ConsistencyTrack and DiffusionTrack are compared on the MOT17 val-half set. Operational efficiency was measured using a single NVIDIA RTX 3090 GPU with a batch size of one. The evaluation of DiffusionTrack is operated with varied sampling time steps ($n_{ss} = 2/4/6$) and a dynamic box count ($n_p = 2000$), and corresponding frames per second (FPS) are recorded. ConsistencyTrack was tested with steps of ($n_{ss} = 1/2/4/6$) and a dynamic box count ($n_p = 2000$). Experimental results indicate that ConsistencyTrack not only achieves a sharp increase in operational efficiency (FPS) compared to DiffusionTrack under the same n_{ss}, but also maintains stable FPS as n_{ss} increases incrementally, unlike DiffusionTrack. This demonstrates ConsistencyTrack's ability to significantly enhance inference speed.

Table 6: Comparison of operational efficiency (FPS) between ConsistencyTrack and DiffusionTrack

n_{ss}	FPS↑	
	DiffusionTrack	ConsistencyTrack
1	/	**10.53**
2	2.50	10.51
4	1.25	10.39
6	0.84	10.27

[1] Bold font indicates the best performance.

Table 7: Performance comparison of stretching methods on advanced metrics

Stretching Method $f(x)$	MOTA↑	IDF1↑	IDP↑
$\frac{x-\mu}{\sigma}$	75.6	74.9	81.7
e^x	75.6	<u>75.7</u>	<u>82.7</u>
$\sqrt{x}$	<u>75.7</u>	75.5	82.4
$\tanh(x)$	<u>75.7</u>	75.4	82.2
$\log(x)$	**75.8**	**76.2**	**82.9**

[1] Results are all percentage data (%).
[2] Bold font indicates the best performance while underlined font indicates the second best.

Stretch association function. In order to classify detection objects into high and low confidence levels, it is necessary to stretch data that is confined to a small range into a larger range. In Table 7, we investigated the impact of different stretching methods on the tracking performance. Specifically, we compared the effects of normalization stretching, exponential function stretching, square root function stretching, and hyperbolic tangent function stretching, and contrasted them with our logarithmic stretching method with a base of 1.01. All experiments were conducted under identical conditions, using $n_p = 2000$, $n_{ss} = 6$, $n_{rp} = 8$, and $B_{th} = 0.6$. The results indicate that, compared to the other methods, our approach demonstrates a distinct advantage in terms of the MOTA metric and also shows improved performance in IDF1 and IDP metrics.

Table 8: Performance comparison between ConsistencyTrack and DiffusionTrack on basic metrics

Method	MOTA↑	IDF1↑	IDP↑	MT↑	ML↓	FN↓	IDs↓
DiffusionTrack	74.4	74.5	82.7	46.6	10.6	21.3	0.8
ConsistencyTrack	**75.7**	**76.5**	**83.3**	**52.8**	**18.6**	**19.4**	**0.6**

[1] Results are all percentage data (%).
[2] Bold font indicates the best performance.

New tracking and matching strategy. We validated the effectiveness of the tracking and matching strategy designed in this work on the MOT17 val-half set, compared to DiffusionTrack. We set unified parameters, such as $n_p = 2000$, $n_{ss} = 6$, and $n_{rp} = 8$. The experiments showed improvements of 1.3% in the MOTA, 2% in IDF1, and 0.6% in IDP. The visualization results are presented in Fig. 10, which intuitively demonstrates the superior performance of the proposed ConsistencyTrack.

In this section, the outstanding performance and prominent features of ConsistencyTrack are demonstrated on the MOT17 and DanceTrack datasets. Notably,

this model significantly surpasses DiffusionTrack in terms of execution efficiency and maintains stability as the sampling timesteps increase, marking one of its most critical innovations. However, due to the model's reliance on very few denoising steps, a decline in accuracy is inevitable. This is primarily manifested in the frequent loss of tracked targets and delayed identification of new targets, with typical failed cases presented in the Fig. 11. These deficiencies still need to be enhanced with more perfect theoretical support or the trade-off between tracking effect and efficiency.

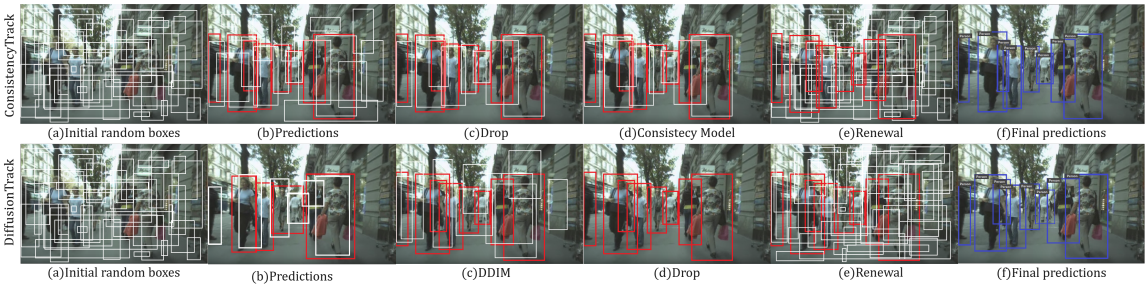

Figure 9: The comparison of the visual reasoning process with one typical sampling step between ConsistencyTrack and DiffusionTrack. The initial noised boxes or verified boxes with low confidence are marked in white, while the boxes with high confidence are marked in red and the final predictions are marked in blue.

5. Conclusions

In this work, we have introduced the generative principles of Consistency Model into an end-to-end MOT approach, implementing a JDT paradigm. Our noise-to-tracking pipeline possesses several attractive features, such as self-consistency and single-step denoising. The model's structure and unique self-consistency enable to achieve faster inference results with the same parameter settings. Extensive experiments demonstrate that ConsistencyTrack achieves excellent performance compared to previous methods. This work provides a novel insight into MOT from the perspective of Consistency Model and open up a new avenue in the field of MOT.

It is noteworthy that due to ConsistencyTrack's adoption of a single-step denoising method, its excessive noise addition and reduction amplitude has compromised its accuracy in MOT tasks. Future research will focus on enhancing the detection and tracking precision of ConsistencyTrack and exploring the way to integrate the core principles of Consistency Model into other advanced tracking models.

Figure 10: Performance comparison between ConsistencyTrack and DiffusionTrack on MOT17 val-half set. Fig. 10(a) illustrates the robust performance of ConsistencyTrack when handling occlusions. The thicker yellow boxes highlight the areas where cases of incorrect detection occur. Fig. 10(b) shows the cases that the exceptional ability of ConsistencyTrack in addressing the ID-switch problem after resolving occlusions. Bold boxes of the same color indicate the same ID.

Acknowledgments

This work is supported in part by the Qingdao Natural Science Foundation under Grant 24-4-4-zrjj-126-jch and Grant 24-4-4-zrjj-90-jch, in part by the National Natural Science Foundation of China under Grant 62202280, and in part by Shandong Provincial Natural Science Foundation under Grant ZR2021QF017.

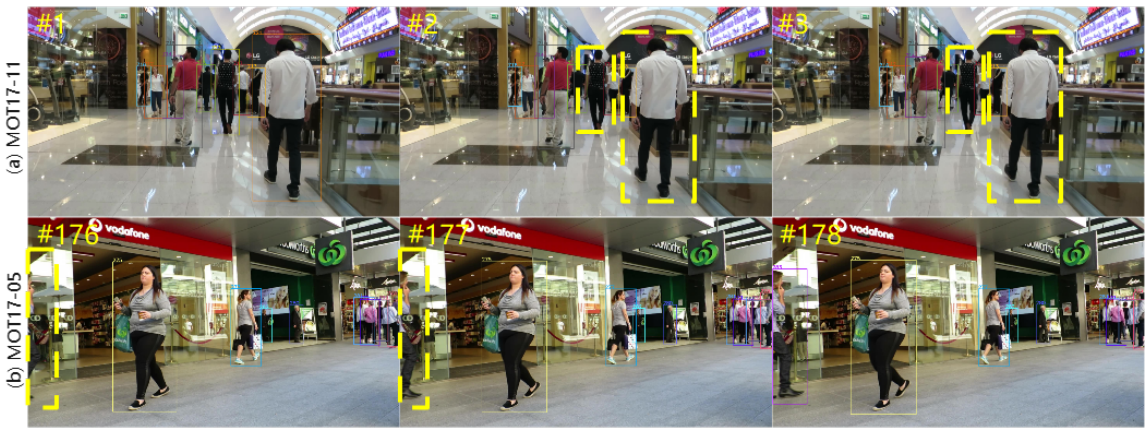

Figure 11: Tracking failures of the proposed ConsistencyTrack in MOT task. Few targets marked in yellow dashed bounding boxes are failed to be tracked continuously, with the situations of failed associations or missed detections with partial occlusions.

References

[1] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, T.-K. Kim, Multiple object tracking: A literature review, Artificial intelligence 293 (2021) 103448.

[2] P. Li, D. Wang, L. Wang, H. Lu, Deep visual tracking: Review and experimental comparison, Pattern Recognition 76 (2018) 323–338.

[3] H. F. Yang, J. Cai, C. Liu, R. Ke, Y. Wang, Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning, Transportation research part C: emerging technologies 148 (2023) 103982.

[4] W. Liu, Y. Lin, Q. Li, Y. She, Y. Yu, J. Pan, J. Gu, Prototype learning based generic multiple object tracking via point-to-box supervision, Pattern Recognition 154 (2024) 110588.

[5] Y. Zhang, Y. Liang, J. Leng, Z. Wang, Scgtracker: Spatio-temporal correlation and graph neural networks for multiple object tracking, Pattern Recognition 149 (2024) 110249.

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

[7] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing, 2017, pp. 3645–3649.

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

[9] C.-Y. Tsai, G.-Y. Shen, H. Nisar, Swin-jde: Joint detection and embedding multi-object tracking in crowded scenes based on swin-transformer, Engineering Applications of Artificial Intelligence 119 (2023) 105770.

[10] H. Kieritz, W. Hubner, M. Arens, Joint detection and online multi-object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops, 2018, pp. 1459–1467.

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

[12] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing, 2016, pp. 3464–3468.

[13] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, Advances in neural information processing systems 34 (2021) 8780–8794.

[14] F.-A. Croitoru, V. Hondru, R. T. Ionescu, M. Shah, Diffusion models in vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (9) (2023) 10850–10869.

[15] S. Chen, P. Sun, Y. Song, P. Luo, Diffusiondet: Diffusion model for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19830–19843.

[16] E. A. Brempong, S. Kornblith, T. Chen, N. Parmar, M. Minderer, M. Norouzi, Denoising pretraining for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4175–4186.

[17] Z. Yuan, C. Hao, R. Zhou, J. Chen, M. Yu, W. Zhang, H. Wang, X. Sun, Efficient and controllable remote sensing fake sample generation based on diffusion model, IEEE Transactions on Geoscience and Remote Sensing 61 (2023) 1–12.

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

[19] Y. Song, P. Dhariwal, M. Chen, I. Sutskever, Consistency models, arXiv preprint arXiv:2303.01469 (2023).

[20] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, arXiv preprint arXiv:2010.04159 (2020).

[21] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, P. Luo, Transtrack: Multiple object tracking with transformer, arXiv preprint arXiv:2012.15460 (2020).

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

[23] X. Zhou, V. Koltun, P. Krähenbühl, Tracking objects as points, in: European conference on computer vision, 2020, pp. 474–490.

[24] P. Tokmakov, J. Li, W. Burgard, A. Gaidon, Learning to track with object permanence, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10860–10869.

[25] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, X. Alameda-Pineda, Transcenter: Transformers with dense representations for multiple-object tracking, IEEE transactions on pattern analysis and machine intelligence 45 (6) (2022) 7820–7835.

[26] X. Zhou, T. Yin, V. Koltun, P. Krähenbühl, Global tracking transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8771–8780.

[27] B. Pang, Y. Li, Y. Zhang, M. Li, C. Lu, Tubetk: Adopting tubes to track multi-object in a one-step training model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6308–6318.

[28] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint arXiv:2107.08430 (2021).

[29] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[30] S. Hassan, G. Mujtaba, A. Rajput, N. Fatima, Multi-object tracking: a systematic literature review, Multimedia Tools and Applications 83 (14) (2024) 43439–43492.

[31] M. A. Islam, M. Kowal, S. Jia, K. G. Derpanis, N. D. Bruce, Position, padding and predictions: A deeper look at position information in cnns, International Journal of Computer Vision (2024) 1–22.

[32] C. Shan, C. Wei, B. Deng, J. Huang, X.-S. Hua, X. Cheng, K. Liang, Tracklets predicting based adaptive graph tracking, arXiv preprint arXiv:2010.09015 (2020).

[33] Y. Zhang, T. Wang, 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, 2023, pp. 22056–22065.

[34] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, B. Schiele, Arttrack: Articulated multi-person tracking in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6457–6465.

[35] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, 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, 2022, pp. 20993–21002.

[36] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[37] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[38] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[39] L. Jiang, Z. Wang, C. Wang, M. Li, J. Leng, X. Wu, Consistencydet: Robust object detector with denoising paradigm of consistency model, arXiv preprint arXiv:2404.07773 (2024).

[40] J. Cao, J. Zhang, B. Li, L. Gao, J. Zhang, Retinamot: rethinking anchor-free yolov5 for online multiple object tracking, Complex & Intelligent Systems 9 (5) (2023) 5115–5133.

[41] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, F. Yu, Quasi-dense similarity learning for multiple object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 164–173.

[42] A. Girbau, X. Giró-i Nieto, I. Rius, F. Marqués, Multiple object tracking with mixture density networks for trajectory estimation, arXiv preprint arXiv:2106.10950 (2021).

[43] G. Wang, Y. Wang, R. Gu, W. Hu, J.-N. Hwang, Split and connect: A universal tracklet booster for multi-object tracking, IEEE Transactions on Multimedia 25 (2022) 1256–1268.

[44] Z. Lu, B. Shuai, Y. Chen, Z. Xu, D. Modolo, Self-supervised multi-object tracking with path consistency, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[45] S. You, H. Yao, B.-K. Bao, C. Xu, Utm: A unified multiple object tracking model with identity-aware feature enhancement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21876–21886.

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

[47] L. Chen, H. Ai, Z. Zhuang, 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, 2018, pp. 1–6.