

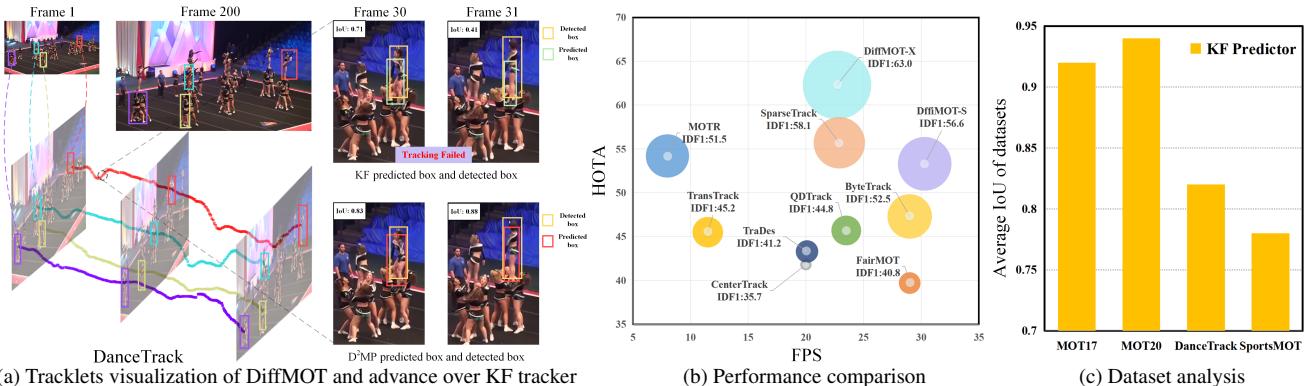
# DiffMOT: A Real-time Diffusion-based Multiple Object Tracker with Non-linear Prediction

Weiyi Lv<sup>1\*</sup> Yuhang Huang<sup>2\*</sup> Ning Zhang<sup>3</sup> Ruei-Sung Lin<sup>3</sup> Mei Han<sup>3</sup> Dan Zeng<sup>1†</sup>

<sup>1</sup>Shanghai University <sup>2</sup>National University of Defense Technology <sup>3</sup>PAII Inc.

<sup>1</sup>{kroery,dzeng}@shu.edu.cn, <sup>2</sup>huangai@nudt.edu.cn, <sup>3</sup>{ning.zhang,rueisung,meihan}@gmail.com

<https://diffmot.github.io/>



(a) Tracklets visualization of DiffMOT and advance over KF tracker

(b) Performance comparison

(c) Dataset analysis

Figure 1. (a) illustrates the trajectories of DiffMOT on sampled sequences of DanceTrack. Each object's center position along the 200 frames is plotted in the 3D coordinates. The objects in DanceTrack exhibit non-linear motion trajectories. Trackers with the KF predictor will fail in tracking in frame 30 for the inaccurate prediction, while our DiffMOT with D<sup>2</sup>MP can track successfully. (b) shows the HOTA-IDF1-FPS comparisons of different trackers. Our DiffMOT with the YOLOX-X detector achieves 62.3% HOTA, 63.0% IDF1 on the DanceTrack test set with 22.7 FPS. (c) shows the motion prediction of the linear Kalman Filter on different datasets. The average IoU of the predicted and ground truth bounding boxes are used as the metric to demonstrate the linear (high IoU) and non-linear (low IoU) characteristics of each dataset.

## Abstract

In Multiple Object Tracking, objects often exhibit non-linear motion of acceleration and deceleration, with irregular direction changes. Tacking-by-detection (TBD) trackers with Kalman Filter motion prediction work well in pedestrian-dominant scenarios but fall short in complex situations when multiple objects perform non-linear and diverse motion simultaneously. To tackle the complex non-linear motion, we propose a real-time diffusion-based MOT approach named DiffMOT. Specifically, for the motion predictor component, we propose a novel Decoupled Diffusion-based Motion Predictor (D<sup>2</sup>MP). It models the entire distribution of various motion presented by the data as a whole. It also predicts an individual object's motion conditioning on an individual's historical motion information. Furthermore, it optimizes the diffusion process with much fewer sampling steps. As a MOT tracker, the DiffMOT is real-time at 22.7FPS, and also outperforms the state-of-the-art on DanceTrack[30] and SportsMOT[6] datasets with 62.3% and 76.2% in HOTA metrics, respectively. To the best of

our knowledge, DiffMOT is the first to introduce a diffusion probabilistic model into the MOT to tackle non-linear motion prediction.

## 1. Introduction

Multiple object tracking (MOT) is one of the fundamental computer vision tasks that aims at continuously tracking objects in video sequences. A successful MOT benefits downstream research such as action detection and recognition, pose tracking, and video understanding. It also attracts much attention in various applications, including pedestrian-dominant smart-city, autonomous driving, and sports analysis.

Tracking-by-detection (TBD) paradigm has been a popular implementation in the MOT [3, 4, 10, 31, 39]. The TBD requires a robust detector and accurate motion predictor, which predicts an individual object's motion. These

\* equal contribution.

†corresponding author.

aforementioned approaches mainly focus on the pedestrian-dominant MOT17 [22] dataset whose objects possess linear motion patterns in terms of a constant-speed and monodirection. In this way, the Kalman Filter (KF) is the natural choice because of its linear prediction and fast speed.

Other scenarios have objects moving *non-linear motion* with less order and more variant. They are also less synchronous, which is different from the pedestrian-dominant scenes where people move more or less around the same speed. There are usually objects with (ac/de)celebration and irregular movement direction. Some typical *non-linear motions* examples are dancers on the stage jumping all-directional [30], or athletes in the field doing different movements [6]. In these cases, motion predictors based on the constant-velocity assumption may not be accurate anymore. As depicted in Fig. 1 (c), linear KF fails to follow the detection ground truth on non-linear datasets, compared to its linear counterpart of the MOT17/20.

Some efforts [5, 33, 36] attempt to tackle such *non-linear motion* prediction using neural networks but not yet successful. This is due to either rigid network structures that are not flexible to obtain good adaptation, or heavy computation that is not suitable for the application. For example, the vanilla neural network approaches (such as MLP and LSTM) [5] can hardly get a satisfactory performance, while the optical flow-based and transformer-based methods [33, 36] have much lower FPS. Naturally, our motivation is to devise an MOT tracker to achieve accurate non-linear motion prediction and real-time speed at the same time.

In this paper, we propose a novel multiple object tracker named DiffMOT with strong non-linear motion prediction and real-time speed. DiffMOT is based on the diffusion probabilistic model, which formulates bounding box position prediction as a denoising process conditioning on the previous bounding box motion trajectories. It offers two advantages over the traditional motion models with single-round learning and prediction. Specifically, at the learning stage, the multiple-step diffusion process has a thorough coverage of the input data with motion representations. During the prediction stage, an individual object’s history motion is used as a condition to guide the denoising process for a better result.

Moreover, to improve the low efficiency of diffusion probabilistic models [12], a Decoupled Diffusion-based Motion Predictor (D<sup>2</sup>MP) optimizes the diffusion framework with both high efficiency and performance inspired by literature work [14]. Compared to the standard thousand-step sampling of a typical diffusion model, a one-step sampling process is devised with decoupled diffusion theory, reducing the inference time significantly. As a result, DiffMOT achieves the best performances on two non-linear datasets (DanceTrack [30] and SportsMOT [6]) with real-

time at 22.7 FPS on an RTX 3090 machine.

As we can see in Fig. 1 (b), our DiffMOT with a YOLOX-X detector combines both fast speed and superior performance, while methods with other motion predictors either suffer from low FPS (MOTR [35] and TransTrack [29]) or low HOTA (FairMOT [38], ByteTrack [39], and QDTrack [23]). Additionally, DiffMOT with a YOLOX-S detector can achieve much faster speed.

In summary, our contribution is three-fold:

- We propose a novel multiple object tracker named DiffMOT with strong non-linear motion prediction and real-time speed. To the best of our knowledge, our work is the first to introduce a diffusion model into the MOT to tackle non-linear motion prediction.
- We introduce a decoupled diffusion-based motion predictor D<sup>2</sup>MP to model the motion distribution of objects with non-linear movements. Compared to previous motion models, D<sup>2</sup>MP excels in fitting the non-linear motion and fast inference speed.
- DiffMOT outperforms SOTA methods on major public datasets in non-linear motions. DiffMOT achieves 62.3% and 76.2% HOTA metrics, on DanceTrack and SportsMOT, respectively.

## 2. Related Work

**Motion Model in MOT.** The motion model in MOT is employed to predict the future position of objects in the previous frame, categorized into linear and non-linear motion models. The most classic linear motion model is the KF which is used in many literature, including the SORT [3], DeepSORT [31], FairMOT [38], ByteTrack [39], OC-SORT [4] and so on. KF assumes that objects’ motion velocity and direction remain constant within a small time interval.

For the non-linear motion models, certain literature proposes various non-linear approaches. For example, Optical flow-based motion models[36] calculate pixel displacements between adjacent frames to obtain motion information, Long Short-Term Memory-based motion models[5] capture sequence motion in latent space, and Transformer-based motion models[33] capture long-range dependencies to model motion. However, none of the aforementioned can simultaneously achieve accurate non-linear motion prediction and real-time speed. In this paper, the proposed D<sup>2</sup>MP optimizes the typical diffusion framework and combines the superiority of performance and speed.

**Diffusion Probabilistic Models.** Benefiting from the powerful fitting capabilities, DPMs have drawn extensive attention due to remarkable performances in image generation [7, 13, 16, 24, 40]. However, conventional DPMs suffer from a standard thousand-step sampling during inference, and recent efforts [8, 9, 14, 18, 27, 28] have focused on DPMs with few-step sampling. DDM[14] splits the typical

diffusion process into two sub-processes to realize the few-step sampling. Inspired by DDM, we propose a specific decoupled diffusion-based motion predictor to perform motion prediction. DiffusionTrack [20] is a concurrent work that utilizes diffusion models to construct the relationships between the paired boxes. In contrast, we focus on non-linear motion modeling and aim to learn the entire motion distribution.

### 3. Method

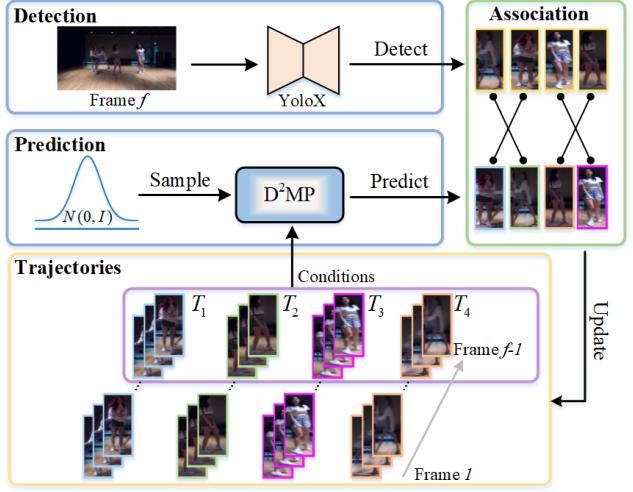
#### 3.1. Framework of DiffMOT

In this section, we introduce DiffMOT, a real-time diffusion-based MOT tracker, to track realistic objects with non-linear motion patterns. As shown in Fig. 2, DiffMOT follows the tracking-by-detection framework that associates the detection of the current frame with the trajectories of the previous frame. The overall framework includes three parts: detection, motion prediction, and association. Given a set of video sequences, DiffMOT first uses a detector to detect the objects’ bounding boxes in the current frame. Next, the future position of the target object in the previous frame is predicted via motion prediction. The motion prediction is where our proposed D<sup>2</sup>MP is devised. In particular, D<sup>2</sup>MP is a diffusion-based motion predictor, which utilizes previous  $n$  frames as conditions and generates the future motion of the objects from the previous frame. Details of D<sup>2</sup>MP are described in Sec. 3.2. As a result, the motion prediction process will output the predicted bounding boxes of objects in the previous frame. Third and last, the association process matches the detected and predicted bounding boxes, thus updating the trajectories.

**Detection.** We adopt the commonly used YoloX [11] as our detector. For a video sequence, the detector detects the bounding boxes of objects frame by frame.

**Motion prediction.** First, we retrieve the previous  $n$  frames of information from the trajectories to serve as the condition for D<sup>2</sup>MP. Subsequently, we employ D<sup>2</sup>MP to sample from the normal distribution, obtaining the motion of each object. Finally, diffusion-based D<sup>2</sup>MP generates the motion and finalizes the predicted bounding boxes of the current frame.

**Association.** The association process is similar to the ByteTrack [39]. First, the predictions are matched with high-scoring bounding boxes from the detection using the Hungarian algorithm [15]. The matching cost is defined by the re-id feature distance and Intersection-over-Union(IoU). We also incorporate the dynamic appearance and adaptive weighting techniques as introduced in [21]. Second, the unmatched predictions are matched with low-scoring boxes from detections using the Hungarian algorithm, employing the IoU as the cost function. At last, we use the matched results to update the trajectories.



height and width of the box. We define the object motion at frame  $f$  as the difference between current frame and previous frame:

$$\mathbf{M}_f = \mathbf{B}_f - \mathbf{B}_{f-1} = (\Delta x_f, \Delta y_f, \Delta w_f, \Delta h_f). \quad (1)$$

In this paper, we define  $\mathbf{M}_f$  as the clean motion data in the diffusion process.

Compared to the typical diffusion model [12] that only has a data-to-noise mapping, we follow the decoupled diffusion process [14] that contains two sub-processes: data to zero and zero to noise. To formulate the forward process, we introduce an additional subscript  $t$ , i.e. we use  $\mathbf{M}_{f,0} = \mathbf{M}_f$  and  $\mathbf{M}_{f,t}$  to denote the clean and noisy motion data. Considering a continuous time axis  $t \in [0, 1]$ , we simultaneously conduct the data-to-zero and zero-to-noise processes over time  $t$  to map the clean motion data  $\mathbf{M}_{f,0}$  to be a pure noise. Specifically, the data-to-zero process utilizes an analytic attenuation function to decrease the clean motion data to be zero data over time. In particular, we adopt the constant function as the analytic attenuation function and the data-to-zero process can be represented by:

$$\mathcal{D}_{f,t} = \mathbf{M}_{f,0} + t\mathbf{c}, \quad (2)$$

where  $\mathbf{c}$  is the constant function and  $\mathbf{c} = -\mathbf{M}_{f,0}$  that can be obtained by solving  $\mathbf{M}_{f,0} + \int_0^1 \mathbf{c} dt = \mathbf{0}$  with reference to [14]. In this way, the clean motion data is attenuated gradually over time  $t$  and to be zero when  $t = 1$ , i.e.  $\mathcal{D}_{f,1} = \mathbf{0}$ . At the same time, the zero-to-noise process adds the normal noise to the zero data gradually, increasing it to be the pure normal noise when  $t = 1$ , which is written as:

$$\mathcal{W}_{f,t} = \mathbf{0} + \sqrt{t}\mathbf{z}, \quad (3)$$

where  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Combining the two sub-processes, we can obtain the noisy motion data  $\mathbf{M}_{f,t}$  through our forward process:

$$\begin{aligned} \mathbf{M}_{f,t} &= \mathcal{D}_{f,t} + \mathcal{W}_{f,t} \\ &= \mathbf{M}_{f,0} + t\mathbf{c} + \sqrt{t}\mathbf{z}. \end{aligned} \quad (4)$$

### 3.2.2 Reversed process

The reversed process utilizes the conditional probability  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$  to recover the object motion from the pure noise,  $\mathbf{M}_{f,1}$ . The reconstructed motion is represented by  $\hat{\mathbf{M}}_{f,0}$ . Following [14], the reversed conditional probability  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$  can be written as:

$$\begin{aligned} q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0}) &= \mathcal{N}(\mathbf{M}_{f,t-\Delta t}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu} &= \mathbf{M}_{f,t} - \Delta t \mathbf{c} - \frac{\Delta t}{\sqrt{t}} \mathbf{z} \\ \boldsymbol{\Sigma} &= \frac{\Delta t(t - \Delta t)}{t} \mathbf{I}, \end{aligned} \quad (5)$$

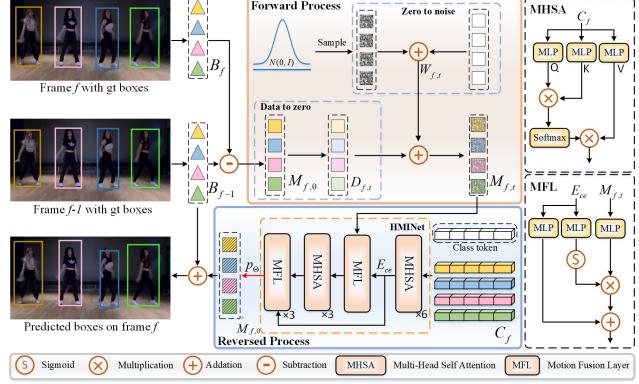


Figure 3. The overall architecture of D<sup>2</sup>MP. D<sup>2</sup>MP consists of the forward process and the reversed process. In the forward process, data to zero and zero to noise processes are enclosed within the blue dashed box. In the reversed process, HMINet is enclosed within the orange dashed box.  $p_\Theta$  refers to the operation introduced in Eq. 6 to reconstruct  $\hat{\mathbf{M}}_{f,0}$ .

where  $\mathcal{N}$  represents the normal distribution,  $\mu$  and  $\Sigma$  are the mean and variance of  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$ , and  $\mathbf{I}$  is the identity matrix. Inconveniently,  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$  cannot be accessed directly since  $\mu$  contains the unknown terms  $\mathbf{c}$  and  $\mathbf{z}$ . Hence, we need to use the parameterized  $p_\Theta(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t})$  to approximate  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$ .

$$\begin{aligned} p_\Theta(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}) &= \mathcal{N}(\mathbf{M}_{f,t-\Delta t}; \boldsymbol{\mu}_\Theta, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu}_\Theta &= \mathbf{M}_{f,t} - \Delta t \mathbf{c}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f) - \frac{\Delta t}{\sqrt{t}} \mathbf{z}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f) \\ \boldsymbol{\Sigma} &= \frac{\Delta t(t - \Delta t)}{t} \mathbf{I}. \end{aligned} \quad (6)$$

Here,  $\mathbf{c}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f)$  and  $\mathbf{z}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f)$  are parameterized via a neural network  $\Theta$ . Additionally,  $\mathbf{C}_f$  is the conditioned historical information whose details will be described in Sec. 3.2.3. For the simplicity, we replace  $\mathbf{c}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f)$  and  $\mathbf{z}_\Theta(\mathbf{M}_{f,t}, t, \mathbf{C}_f)$  with  $\mathbf{c}_\Theta$  and  $\mathbf{z}_\Theta$  in the following paper.

The original decoupled diffusion model proposes a two-branch architecture to parameterize  $\mathbf{c}$  and  $\mathbf{z}$  simultaneously, however, it consumes additional computational cost, hurting its efficiency. To speed up the process, we propose a specific reversed process that only needs to parameterize  $\mathbf{c}$  and maintain the performance. From Eq. 4, we can represent  $\mathbf{z}$  by:

$$\begin{aligned} \mathbf{z} &= \frac{1}{\sqrt{t}} (\mathbf{M}_{f,t} - \mathbf{M}_{f,0} - t\mathbf{c}) \\ &= \frac{1}{\sqrt{t}} (\mathbf{M}_{f,t} - (t-1)\mathbf{c}). \end{aligned} \quad (7)$$

In the similar way, we can represent  $\mathbf{z}_\Theta$  by  $\mathbf{M}_{f,t}$  and  $\mathbf{c}_\Theta$ :

$$\mathbf{z}_\Theta = \frac{1}{\sqrt{t}}(\mathbf{M}_{f,t} - (t-1)\mathbf{c}_\Theta). \quad (8)$$

Substituting Eq. 8 into Eq. 6, we have:

$$\boldsymbol{\mu}_\Theta = \frac{t - \Delta t}{t}\mathbf{M}_{f,t} - \frac{\Delta t}{t}\mathbf{c}_\Theta \quad (9)$$

Therefore, we only need  $\mathbf{c}_\Theta$  to solve the reversed process. In general, solving the reversed process of typical diffusion models [12] needs thousands of steps via numerical integration. Differently, benefiting from the analyticity of the decoupled diffusion process [14], the proposed reversed process enables one-step sampling, which further increases the inference speed to achieve real-time performance.

### 3.2.3 Historical Memory Information Network (HMINet)

In this section, we introduce a specific historical memory information network to parameterize  $\mathbf{c}_\Theta$  in Eq. 9. HMINet first utilizes the multi-head self-attention layers to extract the condition embedding from the conditioned input  $\mathbf{C}_f$ . Afterward, we use the condition embedding as guidance and integrate it into the noisy motion feature. Finally, we use an MLP layer to obtain the final prediction  $\mathbf{c}_\Theta$ .

**Extracting condition embedding.** As the literature shows [26, 37], the conditioned guidance is important in providing clues for generating the final result. Figure 3 illustrates the details. We leverage the motion information of previous  $n$  frames as conditioned input  $\mathbf{C}_f$ . The motion information of frame  $f$  is defined as the combination of the object’s bounding box and movement:

$$I_f = (x_f, y_f, w_f, h_f, \Delta x_f, \Delta y_f, \Delta w_f, \Delta h_f). \quad (10)$$

Thus, the conditioned input is represented by:  $\mathbf{C}_f = [I_{f-1}; I_{f-2}, \dots, I_{f-n}]$ ,  $\mathbf{C}_f \in \mathbb{R}^{n \times 8}$ . We aim to capture the long-range dependencies between different frames in  $\mathbf{C}_f$ , so we use the multi-head self-attention (MHSA) as the base layer. In practice, we feed  $\mathbf{C}_f$  to  $6 \times$  MHSA layers to extract condition embedding. We concatenate a learnable *class token*  $\mathbf{E} \in \mathbb{R}^{1 \times 512}$  and  $\mathbf{C}_f$ , and then incorporate historical motion information contained in  $\mathbf{C}_f$  into the *class token* using multi-head self-attention layers. In this way, we use the updated *class token* as condition embedding, denoted by  $\mathbf{E}_{ce} \in \mathbb{R}^{1 \times 512}$ .

**Incorporating  $\mathbf{E}_{ce}$  into noisy motion feature.** After obtaining the condition embedding  $\mathbf{E}_{ce}$ , we aim to incorporate its information into the noisy motion data. In particular, we construct a motion fusion layer (MFL) that fuses  $\mathbf{E}_{ce}$  and  $\mathbf{M}_{f,t}$  into a unified motion feature  $\bar{\mathbf{M}}_{f,t}$ . Specifically, MFL utilizes an MLP layer to encode  $\mathbf{M}_{f,t}$ , letting the dimension of  $\mathbf{M}_{f,t}$  be same as  $\mathbf{E}_{ce}$ . At the same time, the

Method	HOTA↑	IDF1↑	AssA↑	MOTA↑	DetA↑
FairMOT[38]	39.7	40.8	23.8	82.2	66.7
CenterTrack[41]	41.8	35.7	22.6	86.8	78.1
TraDes[32]	43.3	41.2	25.4	86.2	74.5
TransTrack[29]	45.5	45.2	27.5	88.4	75.9
QDTrack[23]	45.7	44.8	29.2	83.0	72.1
DiffusionTrack[20]	52.4	47.5	33.5	89.5	82.2
MOTR[35]	54.2	51.5	40.2	79.7	73.5
DeepSORT[31]	45.6	47.9	29.7	87.8	71.0
ByteTrack[39]	47.3	52.5	31.4	89.5	71.6
SORT[3]	47.9	50.8	31.2	91.8	72.0
MotionTrack[33]	52.9	53.8	34.7	91.3	80.9
OC-SORT[4]	55.1	54.2	38.0	89.4	80.3
StrongSORT[10]	55.6	55.2	38.6	91.1	80.7
SparseTrack[17]	55.7	58.1	39.3	91.3	79.2
C-BIoU[34]	60.6	61.6	45.4	91.6	81.3
Deep OC-SORT[21]	61.3	61.5	45.8	92.3	82.2
DiffMOT	<b>62.3</b>	<b>63.0</b>	<b>47.2</b>	<b>92.8</b>	<b>82.5</b>

Table 1. Comparison with SOTA MOT trackers on the DanceTrack test sets without using any extra training data. Trackers in the blue block use the same YOLOX detector. ↑ means the higher the better and ↓ means the lower the better. **Bold** numbers indicate the best result.

other two MLP layers are applied to  $\mathbf{E}_{ce}$ , generating two variables that are used as the scale and shift coefficients. The above process can be described by:

$$\bar{\mathbf{M}}_{f,t} = \text{Sigmoid}(\text{MLP}(\mathbf{E}_{ce})) \cdot \text{MLP}(\mathbf{M}_{f,t}) + \text{MLP}(\mathbf{E}_{ce}). \quad (11)$$

Afterward, we concatenate  $\mathbf{E}_{ce}$  and  $\bar{\mathbf{M}}_{f,t}$  together and feed it into the stacked MHSA and MFL layers to conduct the further feature fusion. Finally, the fused feature is fed into an MLP layer to get the final prediction  $\mathbf{c}_\Theta$ . After obtaining  $\mathbf{c}_\Theta$ , we can use Eq. 6 to reconstruct  $\hat{\mathbf{M}}_{f,0}$ .

### 3.2.4 Training loss

We use the ground truth  $\mathbf{c}$  to supervise the network to optimize the  $\Theta$ . We adopt the smooth L1 loss and the final loss function is written as:

$$L = \begin{cases} 0.5(\mathbf{c}_\Theta - \mathbf{c})^2 & |\mathbf{c}_\Theta - \mathbf{c}| < 1 \\ |\mathbf{c}_\Theta - \mathbf{c}| - 0.5 & \text{otherwise.} \end{cases} \quad (12)$$

## 4. Experiments

### 4.1. Datasets and Evaluation Metrics

**Datasets.** We conducted the main experiments on DanceTrack [30] and SportsMOT [6] datasets in which the objects possess non-linear motion patterns. DanceTrack is a dataset comprised of dance videos, consisting of 40 training sequences, 25 validation sequences, and 35 testing sequences. DanceTrack exhibits a highly similar appearance

Method	HOTA↑	IDF1↑	AssA↑	MOTA↑	DetA↑
FairMOT[38]	49.3	53.5	34.7	86.4	70.2
GTR[42]	54.5	55.8	45.9	67.9	64.8
QDTrack[23]	60.4	62.3	47.2	90.1	77.5
CenterTrack[41]	62.7	60.0	48.0	90.8	82.1
TransTrack[29]	68.9	71.5	57.5	92.6	82.7
ByteTrack[39]	62.8	69.8	51.2	94.1	77.1
BoT-SORT[1]	68.7	70.0	55.9	94.5	84.4
OC-SORT[4]	71.9	72.2	59.8	94.5	86.4
DiffMOT	72.1	72.8	60.5	94.5	86.0
*ByteTrack[39]	64.1	71.4	52.3	95.9	78.5
*MixSort-Byte[6]	65.7	74.1	54.8	96.2	78.8
*OC-SORT[4]	73.7	74.0	61.5	96.5	88.5
*MixSort-OC[6]	74.1	74.4	62.0	96.5	88.5
*DiffMOT	<b>76.2</b>	<b>76.1</b>	<b>65.1</b>	<b>97.1</b>	<b>89.3</b>

Table 2. Comparison with SOTA MOT trackers on the SportsMOT test sets. Following the MixSORT [6] convention, the methods with \* indicate that their detectors are trained on the SportsMOT train and validation sets.

with complex non-linear motion patterns such as irregular direction changes. Therefore, DanceTrack’s evaluation requires a significant demand on the trackers’ capacity to robustly handle non-linear motions. SportsMOT introduces video sequences from three different sporting events: soccer, basketball, and volleyball. The dataset comprises a total of 45 training sequences, 45 validation sequences, and 150 testing sequences. It displays extensive acceleration and deceleration motions, thereby demanding robustness in handling non-linear motions from trackers. MOT17 [22] is a conventional and commonly used pedestrian-dominant dataset in MOT. The motion patterns of objects in MOT17 are approximated linearly.

**Metrics.** We utilize Higher Order Metric [19] (HOTA, AssA, DetA), IDF1 [25], and CLEAR metrics [2] (MOTA) as our evaluation metrics. Among various metrics, HOTA is the primary metric that explicitly balances the effects of performing accurate detection and association. IDF1 and AssA are used for association performance evaluation. DetA and MOTA primarily evaluate detection performance. Furthermore, we employ the frames per second (FPS) metric to assess the speed of the algorithm.

**Implementation Details.** In the training stage, we set the smallest time step of the diffusion process to 0.001. We select previous  $n = 5$  frames of historical motion information as the condition. For the optimization, we adopt Adam optimizer training for 800 epochs. The learning rate is set to  $10^{-4}$ . The batch size is set to 2048. All experiments are trained on 4 GeForce RTX 3090 GPUs. In experiments, we use the YOLOX-X as the default detector as recent works[1, 4] for a fair comparison unless there is

Method	HOTA↑	IDF1↑	AssA↑	MOTA↑	DetA↑
DiffusionTrack[20]	60.8	73.8	58.8	77.9	63.2
MotionTrack[33]	61.6	75.1	60.2	78.6	-
ByteTrack[39]	63.1	77.3	62.0	80.3	64.5
OC-SORT[4]	63.2	77.5	63.4	78.0	63.2
MixSort-OC[6]	63.4	77.8	63.2	78.9	63.8
MixSort-Byte[6]	64.0	78.7	64.2	79.3	64.1
C-BIoU[34]	64.1	79.7	63.7	79.7	64.8
StrongSORT[10]	64.4	79.5	64.4	79.6	64.6
Deep OC-SORT[21]	64.9	<b>80.6</b>	<b>65.9</b>	79.4	64.1
SparseTrack[17]	<b>65.1</b>	80.1	65.1	<b>81.0</b>	<b>65.3</b>
DiffMOT	64.5	79.3	64.6	79.8	64.7

Table 3. Comparison with SOTA MOT trackers on the MOT17 test sets under the “private detector” protocol. All methods use the same YOLOX detector.

a specific announcement. For the denoising process, we employ D<sup>2</sup>MP to conduct one-step sampling from a standard normal distribution to obtain the objects’ motion. In the association stage, similar to ByteTrack, we perform separate matching for high-scoring and low-scoring boxes with the high threshold  $\tau_{high} = 0.6$  and the low threshold  $\tau_{low} = 0.4$ . Besides, we adopt the same dynamic appearance and adaptive weighting techniques as [1].

## 4.2. Benchmark Evaluation

We conduct the experiments on the test set of DanceTrack and SportsMOT with non-linear motion patterns to demonstrate the effectiveness of our model. We put all methods that use the same YOLOX detector results in the blue block. We also compare the tracking performances on the test set of MOT17 under the “private detection” protocol to demonstrate that our tracker can achieve satisfactory performance even in pedestrian-dominant scenarios.

**DanceTrack.** We report DiffMOT’s performance on DanceTrack in Tab. 1. It can be seen that DiffMOT consistently achieves the best results across all metrics with the 62.3% HOTA, 63.0% IDF1, 47.2% AssA, 92.8% MOTA, and 82.5% DetA. Compared with the previous SOTA tracker Deep OC-SORT, DiffMOT outperforms it in HOTA by 1.0%. Moreover, we would like to emphasize the improvement of the association-related metrics, such as IDF1 and AssA. DiffMOT outperforms the Deep OC-SORT in IDF1 and AssA by 1.5% and 1.4%, respectively. The results demonstrate the robustness of the proposed DiffMOT in dealing with rich non-linear motion. Note that DiffusionTrack is a concurrent work similar to ours that introduces the diffusion model in MOT. They employ the DDPM[12] to model the distribution of the relationship between paired boxes without considering the non-linear motion of objects. Differently, we model the entire motion distribution and generate future motion from a normal distribution during

Detector	HOTA $\uparrow$	IDF1 $\uparrow$	MOTA $\uparrow$	FPS $\uparrow$
YOLOX-S	53.3	56.6	88.4	<b>30.3</b>
YOLOX-M	57.2	58.6	91.2	25.4
YOLOX-L	61.5	61.7	92.0	24.2
YOLOX-X	<b>62.3</b>	<b>63.0</b>	<b>92.8</b>	22.7

Table 4. Comparison of different detectors in DiffMOT on the DanceTrack test sets. The best results are shown in bold.

the prediction process. In comparison to DiffusionTrack, we make full use of the diffusion model’s strong fitting capabilities and we can see in the table that DiffMOT outperforms it in HOTA by 8.1%.

Additionally, we conduct experiments on the test sets of Dancetrack using different detectors as shown in Tab. 4. We observe that more refinements cost brings more performance gain and results in less FPS. We observed that smaller detectors can achieve higher FPS. DiffMOT can achieve the 30.3 FPS with the YOLOX-S detector. This indicates that DiffMOT can flexibly choose different detectors for various real-world application scenarios.

**SportsMOT.** To further demonstrate the performance of DiffMOT in other non-linear scenarios, we conduct experiments on the SportsMOT benchmark which is characterized by a large amount of (ac/de)celeration object motion. Following [6], we conduct experiments under two different detector setups. As shown in Tab. 2, methods with \* indicate their detectors are trained on both train and validation sets, while others are trained only on the train set. DiffMOT achieves SOTA results in both setups. In the train-only setup, DiffMOT obtains 72.1% HOTA, 72.8% IDF1, 60.5% AssA, 94.5% MOTA, and 86.0% DetA, which surpasses previous methods comprehensively. In the other setup, DiffMOT also achieves the best performance across all metrics and outperforms the previous SOTA tracker MixSort-OC by 2.1% in HOTA, 1.7% in IDF1, 3.1% in AssA, 0.6% in MOTA, and 0.8% in DetA with the 76.2% HOTA, 76.1% IDF1, 65.1% AssA, 97.1% MOTA, and 89.3% DetA. The results further indicate that DiffMOT exhibits strong robustness in scenarios with (ac/de)celeration object motion.

**MOT17.** We also conduct the experiment on the conventional and commonly used pedestrian-dominant MOT17 dataset. On MOT17, it can be seen from the Tab. 3 that DiffMOT achieves 64.5% HOTA, 79.3% IDF1, 64.6% AssA, 79.8% MOTA, and 64.7% DetA. Although the proposed DiffMOT is designed specifically for non-linear motion scenes, it can still achieve comparable performances to other SOTA methods under pedestrian-dominant scenarios.

### 4.3. Ablation Studies

We conduct ablation studies on the validation set of DanceTrack. we would like to note that all ablation study focuses on the motion prediction step based on the D<sup>2</sup>MP. We use

Method	HOTA $\uparrow$	IDF1 $\uparrow$	AssA $\uparrow$	MOTA $\uparrow$	DetA $\uparrow$
IoU Only	44.7	36.8	25.3	87.3	<b>79.6</b>
Kalman Filter	46.8	52.1	31.3	87.5	70.2
LSTM	51.2	51.6	34.3	87.1	76.7
Transformer	54.6	54.6	38.1	89.2	78.6
D <sup>2</sup> MP(ours)	<b>55.7</b>	<b>55.2</b>	<b>39.5</b>	<b>89.3</b>	78.9

Table 5. Comparison of different motion models on the DanceTrack validation sets. The best results are shown in bold.

Method	sampling steps	HOTA $\uparrow$	IDF1 $\uparrow$	MOTA $\uparrow$	FPS $\uparrow$
D <sup>2</sup> MP-TB	1	44.5	48.0	89.2	20.9
D <sup>2</sup> MP-TB	10	52.3	50.5	89.2	13.1
D <sup>2</sup> MP-TB	20	54.6	53.6	<b>89.3</b>	7.5
D <sup>2</sup> MP-OB	1	<b>55.7</b>	<b>55.2</b>	<b>89.3</b>	<b>22.7</b>

Table 6. Comparison of D<sup>2</sup>MP with two or one branch on the DanceTrack validation sets. The best results are shown in bold.

the same YOLOX-X detector throughout all experiments. The ablation studies focus on investigating the impact of different motion models, different architectures of D<sup>2</sup>MP, different conditions, and the length of historical motion information on the proposed DiffMOT.

**Different motion models.** To compare our method with other motion models regarding its modeling capability in non-linear motion scenarios, we conduct the experiment in Tab. 5. The motion models we compared include the linear motion model (KF) and non-linear motion models (LSTM and Transformer-based motion models). We also present IoU only association, without any motion model for reference. From the table, it can be seen that our motion model achieves the best performance with 55.7% in HOTA, 55.2% in IDF1, 39.5% in AssA, 89.3% in MOTA, and 78.9% in DetA. This demonstrates that D<sup>2</sup>MP can ensure more robust non-linear motion predictions because our diffusion-based motion model directly learns the distribution of all objects motion across the entire dataset rather than individual object trajectories.

**Different architectures of D<sup>2</sup>MP.** We compared the experimental results of different D<sup>2</sup>MP architectures which are mentioned in Sec. 3.2.2 in Tab. 6. In the table, ”D<sup>2</sup>MP-TB” refers to the two-branch architecture, which optimizes both  $c_{\Theta}$  and  $z_{\Theta}$  simultaneously. During the reversed process, D<sup>2</sup>MP-TB uses the Eq. 6 to approximate  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$ . On the other hand, ”D<sup>2</sup>MP-OB” refers to the one-branch architecture that is used in our DiffMOT. The one-branch architecture only requires the optimization of  $c_{\Theta}$ , and during the reversed process, D<sup>2</sup>MP-OB uses the Eq. 8 to approximate  $q(\mathbf{M}_{f,t-\Delta t} | \mathbf{M}_{f,t}, \mathbf{M}_{f,0})$ . As shown in the table, D<sup>2</sup>MP-OB outperforms D<sup>2</sup>MP-TB

Condition	HOTA $\uparrow$	IDF1 $\uparrow$	AssA $\uparrow$	MOTA $\uparrow$	DetA $\uparrow$
$\mathbf{B}_{f-1}$	51.0	<b>49.1</b>	33.3	88.6	78.5
$\mathbf{M}_{f-1}$	50.4	46.8	32.2	89.0	79.3
$\mathbf{I}_{f-1}$	<b>51.7</b>	48.5	<b>33.8</b>	<b>89.1</b>	<b>79.4</b>

Table 7. Comparison of different conditions on the DanceTrack validation sets. The best results are shown in bold.

by 11.2% in HOTA when using one-step sampling. This is attributed to the increased complexity of learning in the two-branch network. To achieve comparable performance to the D<sup>2</sup>MP-OB, the D<sup>2</sup>MP-TB requires multiple sampling steps. D<sup>2</sup>MP-TB can achieve 54.6% HOTA, 53.6% IDF1, and 89.3% MOTA after 20 sampling steps. However, at this point, the speed is only 7.5 FPS, making it hard for practical applications.

**Different conditions.** To demonstrate the advantages of our designed condition used in HMINet, we conducted the ablation experiment as shown in Tab. 7. In the table,  $\mathbf{B}_{f-1}$  refers to using the object box from the previous frame as a condition which is defined as  $\mathbf{B}_{f-1} = (x_{f-1}, y_{f-1}, w_{f-1}, h_{f-1})$ .  $\mathbf{M}_{f-1}$  refers to the object motion which is defined as Eq. 1.  $\mathbf{I}_{f-1}$  refers to the motion information which is used in our DiffMOT. It is defined as Eq. 10. From the table, it can be seen that utilizing  $\mathbf{I}_{f-1}$  as the condition yields the best results. This is because  $\mathbf{M}_{f-1}$  contains only the positional information of the bounding boxes, using it as a condition can lead to motion predictions with larger deviations. On the other hand,  $\mathbf{I}_{f-1}$  contains only the information on the motion change, lacking the positional information, thereby introducing difficulty in generation.

**Length of historical motion information.** To determine the optimal length of historical motion information used in HMINet for controlling motion prediction, we conduct the ablation experiment in Tab. 8. As shown in the table, the best results were achieved when the length was set to  $n = 5$ . The results gradually improve as  $n < 5$  and deteriorate as  $n > 5$ . We posit that this phenomenon arises from the inadequacy of effectively directing motion prediction when the length of historical motion information is too short, and the interference introduced by excessive information when the length of historical motion information is too long.

#### 4.4. Visualization

Fig. 4 illustrates the qualitative comparison between using KF or D<sup>2</sup>MP as the motion model on the test set of DanceTrack. The upper row represents the results predicted by KF, while the lower row represents the results predicted by D<sup>2</sup>MP in each case. It is obvious that when the objects exhibit non-linear motion such as rolling, jumping, and crouching in dance, KF is unable to accurately predict the trajectories' position, resulting in the generation of new

$n$	HOTA $\uparrow$	IDF1 $\uparrow$	AssA $\uparrow$	MOTA $\uparrow$	DetA $\uparrow$
1	51.7	48.5	33.8	89.1	<b>79.4</b>
2	52.5	49.9	34.9	89.2	79.3
3	53.0	52.0	35.9	89.2	78.6
5	<b>55.7</b>	<b>55.2</b>	<b>39.5</b>	<b>89.3</b>	78.9
7	52.5	49.7	34.9	89.2	79.3
10	51.1	48.7	33.3	89.1	78.8

Table 8. Evaluation of  $n$  on the DanceTrack validation sets. The best results are shown in bold.



Figure 4. Qualitative comparison between using KF or D<sup>2</sup>MP as the motion model on the DanceTrack test set. The upper row represents the results predicted by KF, while the lower row represents the results predicted by D<sup>2</sup>MP. The red arrow indicates the noteworthy objects. Boxes of the same color represent the same ID. Best viewed in color and zoom-in.

ID numbers. In contrast, D<sup>2</sup>MP exhibits greater robustness in handling these non-linear motions, accurately predicting trajectory positions, and maintaining the trajectories. For more visualization results, please refer to the supplementary materials.

## 5. Conclusion

In this paper, we propose a diffusion-based MOT, named DiffMOT. In contrast to previous trackers that focus on pedestrians, DiffMOT aims to track objects in non-linear motion. To deal with the more complex non-linear pattern, we carefully design the decoupled diffusion motion predictor. DiffMOT exceeds all previous trackers in two non-linear datasets (DanceTrack and SportsMOT) and gets comparable performances to SOTA methods on the general pedestrian-dominant dataset (MOT17). The results show that the DiffMOT has great potential in realistic applications.

**Acknowledgement.** This work was supported in part by the National Natural Science Foundation of China under Grant 62372284.

## References

- [1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 6
- [2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 6
- [3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016. 1, 2, 5
- [4] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *CVPR*, pages 9686–9696, 2023. 1, 2, 5, 6
- [5] Mohamed Chaabane, Peter Zhang, J Ross Beveridge, and Stephen O’Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021. 2
- [6] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. Sportsmot: A large multi-object tracking dataset in multiple sports scenes. In *ICCV*, pages 9921–9931, 2023. 1, 2, 5, 6, 7
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2
- [8] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *NeurIPS*, 35: 30150–30166, 2022. 2
- [9] Arnaud Doucet, Will Grathwohl, Alexander G Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. *NeurIPS*, 35:21482–21494, 2022. 2
- [10] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023. 1, 5, 6
- [11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 4, 5, 6
- [13] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 2
- [14] Yuhang Huang, Zheng Qin, Xinwang Liu, and Kai Xu. Decoupled diffusion models with explicit transition probability. *arXiv preprint arXiv:2306.13720*, 2023. 2, 3, 4, 5
- [15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3
- [16] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *CVPR*, pages 19434–19445, 2023. 2
- [17] Zelin Liu, Xinggang Wang, Cheng Wang, Wenyu Liu, and Xiang Bai. Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. *arXiv preprint arXiv:2306.05238*, 2023. 5, 6
- [18] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 35:5775–5787, 2022. 2
- [19] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2): 548–578, 2021. 6
- [20] Run Luo, Zikai Song, Lintao Ma, Jinlin Wei, Wei Yang, and Min Yang. Diffusiontrack: Diffusion model for multi-object tracking. *arXiv preprint arXiv:2308.09905*, 2023. 3, 5, 6
- [21] Gerard Maggiolini, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023. 3, 5, 6
- [22] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 2, 6
- [23] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173, 2021. 2, 5, 6
- [24] Seung Ho Park, Young Su Moon, and Nam Ik Cho. Flexible style image super-resolution using conditional objective. *IEEE Access*, 10:9774–9792, 2022. 2
- [25] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016. 6
- [26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 5
- [27] Mohamad Kazem Shirani Faradonbeh, Mohamad Sadegh Shirani Faradonbeh, and Mohsen Bayati. Thompson sampling efficiently learns to control diffusion processes. *NeurIPS*, 35:3871–3884, 2022. 2
- [28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 2
- [29] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 2, 5, 6
- [30] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, pages 20993–21002, 2022. 1, 2, 5
- [31] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 1, 2, 5
- [32] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, pages 12352–12361, 2021. 5

- [33] Changcheng Xiao, Qiong Cao, Yujie Zhong, Long Lan, Xiang Zhang, Huayue Cai, Zhigang Luo, and Dacheng Tao. Motiontrack: Learning motion predictor for multiple object tracking. *arXiv preprint arXiv:2306.02585*, 2023. [2](#), [5](#), [6](#)
- [34] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *WACV*, pages 4799–4808, 2023. [5](#), [6](#)
- [35] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, pages 659–675. Springer, 2022. [2](#), [5](#)
- [36] Jimuyang Zhang, Sanping Zhou, Xin Chang, Fangbin Wan, Jinjun Wang, Yang Wu, and Dong Huang. Multiple object tracking by flowing and fusing. *arXiv preprint arXiv:2001.11180*, 2020. [2](#)
- [37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. [5](#)
- [38] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. [2](#), [5](#), [6](#)
- [39] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21. Springer, 2022. [1](#), [2](#), [3](#), [5](#), [6](#)
- [40] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *ICLR*, 2021. [2](#)
- [41] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490. Springer, 2020. [5](#), [6](#)
- [42] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, pages 8771–8780, 2022. [6](#)