Probabilistic Tracklet Scoring and Inpainting for Multiple Object Tracking

Fatemeh Saleh[1,2], Sadegh Aliakbarian[1,2], Hamid Rezatofighi[3], Mathieu Salzmann[4,5], Stephen Gould[1,2]
[1]Australian National University, [2]ACRV, [3]Monash University, [4]CVLab, EPFL, [5]ClearSpace
fatemehsadat.saleh@anu.edu.au

Abstract

Despite the recent advances in multiple object tracking (MOT), achieved by joint detection and tracking, dealing with long occlusions remains a challenge. This is due to the fact that such techniques tend to ignore the long-term motion information. In this paper, we introduce a probabilistic autoregressive motion model to score tracklet proposals by directly measuring their likelihood. This is achieved by training our model to learn the underlying distribution of natural tracklets. As such, our model allows us not only to assign new detections to existing tracklets, but also to inpaint a tracklet when an object has been lost for a long time, e.g., due to occlusion, by sampling tracklets so as to fill the gap caused by misdetections. Our experiments demonstrate the superiority of our approach at tracking objects in challenging sequences; it outperforms the state of the art in most standard MOT metrics on multiple MOT benchmark datasets, including MOT16, MOT17, and MOT20.

1. Introduction

Tracking multiple objects in a video is key to the success of many computer vision applications, such as sport analysis, autonomous driving, robot navigation, and visual surveillance. With the recent progress in object detection, tracking-by-detection [2] has become the de facto approach to multiple object tracking; it consists of first detecting the objects in the individual frames and then associating these detections with trajectories, known as tracklets. While these two steps were originally performed sequentially, recent advances have benefited from treating detection and tracking jointly [3, 53, 66]. These approaches cast MOT as a *local tracking* problem, utilizing either an object detector's regression head [3] or an additional offset head [53, 66] to perform temporal re-alignment of the object bounding boxes in consecutive frames. In other words, these approaches treat tracking as the problem of propagating detection identities across consecutive frames. While this strategy constitutes the state of the art on many benchmark datasets in terms of MOT metrics that highlight the quality of the detections, *e.g.*, MOTA, it fails to maintain identities throughout occlusions, and thus tends to produce many identity switches. In this paper, we address this issue by developing a stochastic motion model that helps the tracker to maintain identities, even in the presence of long-term occlusions. In other words, we show that, while largely ignored in the recent MOT literature, motion remains a critical cue for tracking, even with the great progress achieved by detectors. This is evidenced by our experimental results on multiple MOT benchmark datasets, in which our approach outperforms the state of the art by a large margin.

Motion has, of course, been considered in the past, mostly in the tracking-by-detection literature, via either model-based filtering techniques [5, 25, 56] or more sophisticated data-driven ones based on RNNs [14, 17, 41, 44, 47, 55, 59, 60]. However, all of these approaches treat human motion as a deterministic or a uni-modal process. Here, we argue that human motion is a stochastic multi-modal process, and should thus be modeled stochastically. Note that a similar concept has also been explored in the context of trajectory forecasting, where the problem is to often given perfect (ground-truth) trajectories, predict fixed-length continuations of those trajectories as a single path [1, 23], or a distribution over different paths [19, 24, 29, 39, 48, 49]. However, to the best of our knowledge, these techniques have not been incorporated in the context of MOT, where we deal with noisy observations (detections), frequent occlusions, and assignment uncertainties.

Therefore, we introduce a stochastic autoregressive motion model that explicitly learns the multi-modal distribution of natural trajectories. This allows us to estimate the likelihood of a tracklet given a sequence of bounding box locations and the tracklets of the surrounding agents. We then use this model to compute the likelihood of a tracklet after assigning it a new detection. Moreover, learning the multi-modal distribution of tracklets allows us to inpaint a tracklet in the presence of misdetections caused by occlusion by sampling from the learned distribution. This is also what the visual cortex of the human brain does when reasoning about dynamically occluded objects [15, 52].

To summarize, our contributions are as follows: **(1)** We introduce a stochastic autoregressive model to score a tracklet by the likelihood that it represents natural motion. **(2)** Since our model learns the multi-modal distribution of natural human motion, it can generate multiple plausible continuations of the tracklets and inpaint tracklets containing missed detections. **(3)** Our stochastic motion model can better preserve identities over longer time horizons than recent MOT approaches, especially when there are occlusions.

We conduct comprehensive ablation studies, demonstrating the effectiveness of the different components of our approach. Our method outperforms the state of the art in multiple MOT benchmark datasets, particularly improving the metrics related to long-term identity preservation, such as IDF1, ID Switch (IDs), and Mostly Tracked Tracklets (MT). This is further confirmed by our experiments on the challenging new MOT20 [13] dataset, targeting highly crowded scenarios. We refer to our model as **ArTIST**, for **A**uto**r**egressive **T**racklet **I**npainting and **S**coring for **T**racking. The code of ArTIST is publicly available[1].

2. Related Work

Tracking-by-Detection. Tracking-by-detection [2] has proven to be effective to address the MOT problem. In this context, tracking systems can be roughly grouped into online ones [3, 10, 11, 12, 31, 35, 38, 41, 47, 57, 64, 68], where the tracklets are grown at each time step, and batch-based (or offline) ones [9, 27, 28, 37, 54, 63], where the tracklets are computed after processing the entire sequence, usually in a multiple hypothesis tracking (MHT) framework [6, 27]. In this paper, we develop an online tracking system and thus, in this section, focus on this class of methods.

Closest to our approach are the ones that design or utilize a motion model for state prediction. In [5, 20, 56], this was achieved with a Kalman Filter [25] aiming to approximate the inter-frame displacements of each object with a linear constant velocity model, assuming independence across the objects and from the camera motion. As a linear motion model often poorly reflects reality, more sophisticated data-driven motion models have been proposed to permit more complex state prediction [14, 17, 41, 44, 47, 55, 59, 60]. In particular, the use of recurrent networks was introduced in [41] to mimic the behavior of a Bayesian filter for motion modeling and associations. Following [41], several recurrent approaches have been developed for MOT. In [17, 33, 44, 47], multiple streams of RNNs have been utilized to incorporate different forms of information, such as appearance and motion, to compute a score for the assignment, usually done by solving an assignment problem via the Munkres (a.k.a. Hungarian) algorithm [42].

In all the aforementioned approaches, human motion has been treated either deterministically [44, 47] or probabilistically in a uni-modal manner [17, 55]. The shortcoming of such techniques is that, while these are reasonable design choices when the state estimation uncertainty is low, they become poorly suited for tracking throughout long occlusions, where uncertainty increases significantly. This is particularly due to the stochastic nature of human motion, a property that has been overlooked by existing approaches.

Joint Detection and Tracking. As an alternative to the two-stage tracking-by-detection, the recent trend in MOT has moved toward jointly performing detection and tracking. This is achieved by converting an object detector to predict the position of an object in the next frame, thus inherently utilizing it for tracking. To this end, Tracktor (and its variants, Tracktor++ and Tracktor++v2) [3] exploits the regression head of a Faster R-CNN [45] to perform temporal realignment of the object bounding boxes. CenterTrack [66] adapts the CenterNet object detector [67] to take two frames and a heatmap rendered from the tracked object centers as input, and computes detection and tracking offsets for the current frame. Chained-Tracker [53] uses two adjacent frames as input to regress a pair of bounding boxes for the same target in the two adjacent frames. Although these approaches yield impressive results, their effectiveness depends on the feasibility of detecting the objects. In fact, these approaches look at the tracking problem from a *local* perspective, and thus, even though they use techniques such as person ReID [3, 22], CMC [3, 16], or ReBirth [3, 66] to re-identify occluded objects, tend to struggle to preserve identities.

This paper. To address the shortcomings of the aforementioned approaches, we introduce a MOT framework with a focus on designing a non-linear *stochastic* motion model by learning the multi-modal distribution of the next plausible states of a pedestrian so as to reason about uncertainties in the scene when facing occlusions. It not only allows us to estimate the likelihood of a tracklet and directly use it for scoring a new detection, but also enables us to fill in the gaps in case of misdetection caused by occlusion by sampling from the learned distribution. As a result, we considerably improve identity preservation, as confirmed by our results on several MOT benchmark datasets.

3. Proposed Method

We address the problem of online tracking of multiple objects in a scene by designing a stochastic motion model. In this section, we first define our notation, and then provide an overview of our ArTIST algorithm, followed by the details of its different components.

3.1. Notation

As many other online tracking systems, we follow a tracking-by-detection paradigm [2]. Let us consider a video of T frames, where, for each frame, we are provided with a

[1] https://github.com/fatemeh-slh/ArTIST

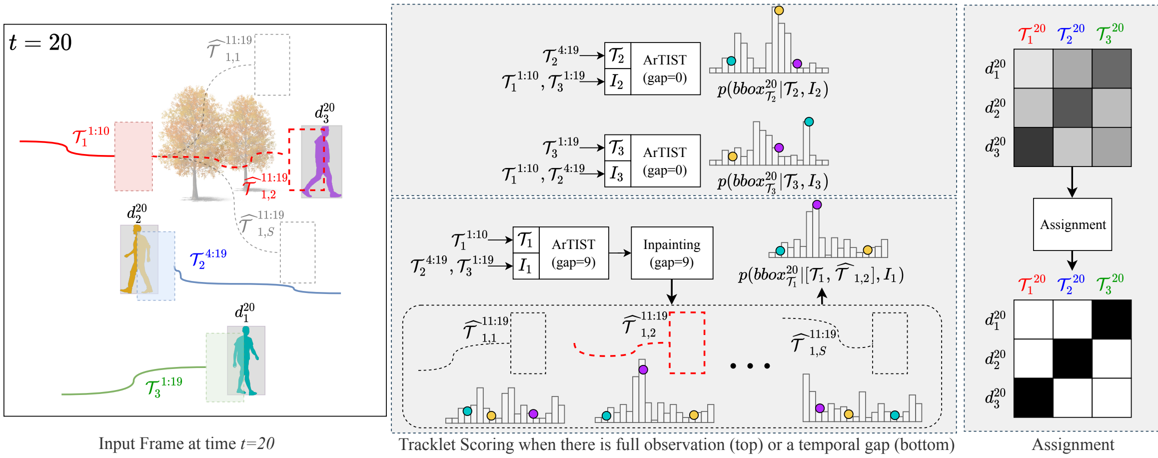

Figure 1. An overview of our framework. **Left**: At time $t = 20$, we are provided with detections illustrated by solid gray boxes and a set of alive tracklets $\mathcal{T}^{\cdot}_{\cdot}$, shown in different colors. The task is to assign the new detections to existing tracklets. **Middle**: An illustration of tracklet scoring (top) in the case of full observation and tracklet inpainting and scoring (bottom) in case of misdetection due to *e.g.*, occlusion. **Right**: Given the computed scores, we solve the Munkres algorithm to perform assignment before processing the next frame.

set of detections computed by, *e.g.*, Faster-RCNN [45]. This yields an overall detection set for the entire video denoted by $\mathcal{D}^{1:T} = \{D^1, D^2, ..., D^T\}$, where $D^t = \{d_1^t, d_2^t, ...\}$ is the set of all detections at time t, with $d_i^t \in \mathbb{R}^4$, *i.e.*, the 2D coordinates (x, y) of the top-left bounding box corner, its width w and height h. We tentatively initialize a first set of tracklets $\mathbb{T}$ with the detections D^1 in the first frame. From the second time-step to the end of the video, the goal is to expand the tracklets by assigning the new detections to their corresponding tracklets. Throughout the video, new tracklets may be created, and incorporated into the set of tracklets $\mathbb{T}$, and existing tracklets may be terminated and removed from $\mathbb{T}$. We write $\mathbb{T} = \{\mathcal{T}_1^{s_1:e_1}, \mathcal{T}_2^{s_2:e_2}, ..., \mathcal{T}_m^{s_m:e_m}\}$, where $\mathcal{T}_j^{s_j:e_j}$ is a tracklet representing the j^{th} identity that has been alive from time s_j to time e_j, and is defined as $\mathcal{T}_j^{s_j:e_j} = \{d_{\Pi_j}^{s_j}, d_{\Pi_j}^{s_j+1}, ..., d_{\Pi_j}^{e_j}\}$, where $d_{\Pi_j}^t$ is the detection (or an inpainted box) at time t that has been assigned to tracklet $\mathcal{T}_j^{s_j:e_j}$. For each tracklet $\mathcal{T}_j^{s_j:e_j}$, we define a learnable interaction representation $I_j^{s_j:e_j}$ which captures the latent representations of all other tracklets whose lifespan overlaps with the temporal range $[s_j, e_j]$. We also define z_j^t that captures the hidden representation of $\mathcal{T}_j^{s_j:t}$. Both I_j and z_j are described in detail below.

3.2. ArTIST Overview

ArTIST relies on two main steps for every video frame: scoring how well a detection fits in an existing tracklet (as in Fig. 1-middle) and assigning the detections to the tracklets (as in Fig. 1-right), thus updating them.

Specifically, given an input frame at time t, *e.g.*, $t = 20$ in Fig. 1-left, a set of tracklets up to time $t-1$, *e.g.*, $\mathcal{T}_1^{1:10}$, $\mathcal{T}_2^{4:19}$, and $\mathcal{T}_3^{1:19}$, and a set of detections at time t, *e.g.*, d_1^{20}, d_2^{20}, and d_3^{20}, shown as solid gray boxes, we perform scoring for the tracklets were last assigned a detection at time $t-1$, *i.e.*, the non-occluded tracklets. This is denoted by gap $= 0$ in Fig. 1-middle. We refer to these tracklets as *alive*, and to others as *tentatively alive*. For each alive tracklet, for instance $\mathcal{T}_2^{4:19}$, ArTIST computes the probability distribution of the next plausible bounding box ($bbox^{20}_{\mathcal{T}_2}$) that can be assigned to $\mathcal{T}_2^{4:19}$, given information about both this tracklet and the other tracklets that interact with it, *i.e.*, $\mathcal{T}_2$ and I_2, respectively. We then evaluate all the detections $d_i^t \in D^t$ at time t under the estimated distribution.

For any tentatively alive tracklets (*e.g.*, $\mathcal{T}_1^{1:10}$), whose last assignment occurred prior to $t-1$, resulting in a non-zero gap, we first perform tracklet inpainting to fill the gap up to $t-1$, so that it can be considered as a fully-observed tracklet. As ArTIST estimates a multi-modal distribution of natural motion, we generate S plausible tracklets to fill in this gap, denoted by $\{\widehat{\mathcal{T}}_{1,1}^{11:19}, ..., \widehat{\mathcal{T}}_{1,S}^{11:19}\}$ in the bottom part of Fig. 1-middle. We then select the best inpainted tracklet (the second one in this example) among the S candidates to complete $\mathcal{T}_1^{1:19}$ as $[\mathcal{T}_1^{1:10}, \widehat{\mathcal{T}}_{1,2}^{11:19}]$. We can now treat this tracklet as having zero gap and thus compute the distribution over next plausible bounding box assignments.

Finally, as illustrated in Fig. 1-right, we construct a cost matrix from the likelihoods of each detection under the estimated distributions for all tracklets, and obtain an optimal assignment using the Munkres algorithm [42]. We then update all the tracklets with the assigned detections, and repeat the entire process for the next time-step. In the following sections, we provide more details about our ArTIST architecture and the different steps of this algorithm, with a complete specification given in the supplementary material.

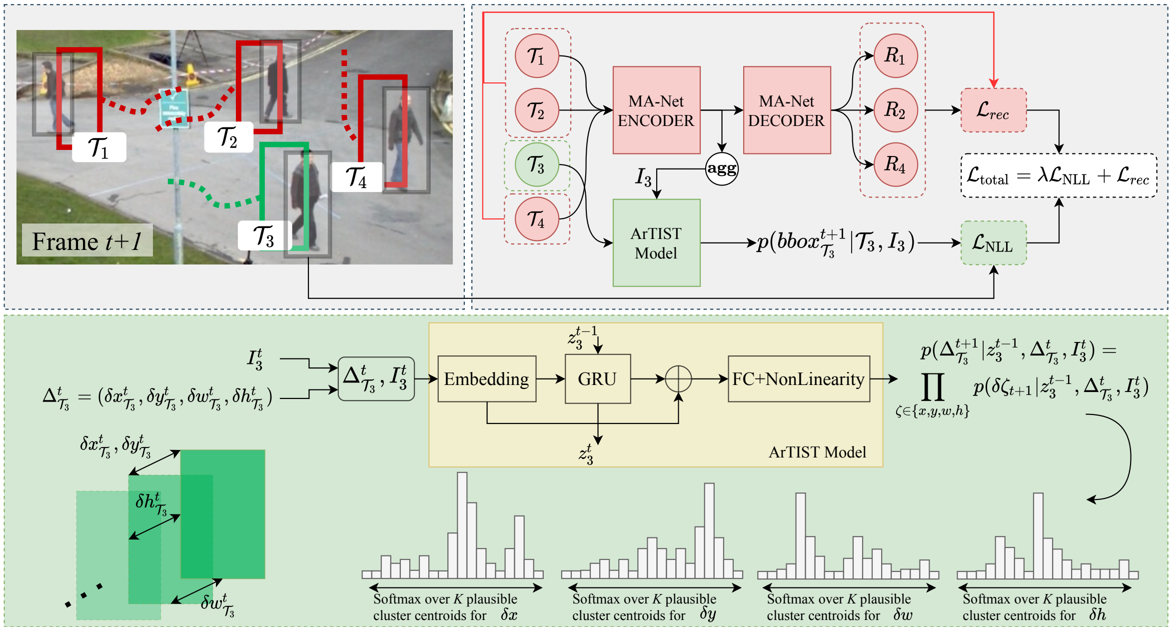

Figure 2. **Top**: An overview of our approach during training. Given a set of GT bounding boxes (gray boxes) at time $t+1$, we show the training procedure of our model that aims to maximize the likelihood of tracklet $\mathcal{T}_3$ when assigned with the correct detection by incorporating additional information from all other tracklets, I_3. MA-Net autoencoder is trained jointly with the ArTIST model to provide an expressive representation of I_3 by learning to reconstruct other tracklets (shown in red). **Bottom**: An overview of the recurrent residual architecture of ArTIST model for processing a tracklet at time t to compute the probability distribution of the next plausible bounding box. Such distribution is used to either evaluate the likelihood of assigning a new detection to the tracklet or for inpainting a tracklet.

3.3. ArTIST Architecture

ArTIST is a stochastic autoregressive motion model that aims to explicitly learn the distribution of natural tracklets. As an estimator, ArTIST is capable of determining the likelihood of each tracklet. As a generative model, ArTIST is capable of generating multiple plausible continuations of a tracklet by multinomial sampling from the estimated multi-modal distribution at each time-step.

The probability of a tracklet $\mathcal{T}^{s_j:t}_j$, where t is the current time-step, in an autoregressive framework is defined as

$$p(\mathcal{T}^{s_j:t}_j|I^{s_j:t}_j) = p(d^{s_j}_{\Pi_j}|I^{s_j}_j)\prod_{k=s_j+1}^{t} p(d^k_{\Pi_j} \mid d^{<k}_{\Pi_j}, I^{<k}_j)\,, \quad (1)$$

where $d^k_{\Pi_j}$ is the detection assigned to $\mathcal{T}_j$ at time k and I^k_j is the representation of the interactions computed from other tracklets co-occurring with $\mathcal{T}_j$ at time k. Since each detection is represented by continuous bounding box coordinates, one could attempt to regress its position in the next frame given previous positions. However, regression does not explicitly provide a distribution over natural tracklets. Furthermore, regression can only generate a single deterministic continuation of a tracklet, which does not reflect the stochastic nature of, *e.g.*, human motion, for which multiple continuations may be equally likely.

To remedy this, inspired by PixelRNN [43], we propose to discretize the bounding box position space. This allows us to model $p(\mathcal{T}^{s_j:e_j}_j)$ as a discrete distribution, with every conditional distribution in Eq. 1 modeled as a multinomial (categorical) distribution with a *softmax* layer. However, unlike PixelRNN-like generative models that discretize the space by data-independent quantization, *e.g.*, through binning, we define a data-dependent set of discrete values by clustering the motion velocities, *i.e.*, δx, δy, δw, and δh, between consecutive frames, normalized by the width and height of the corresponding frames. This makes our output space shift and scale invariant. In practice, we use non-parametric k-means clustering [36] to obtain K clusters, and treat each cluster centroid as a discrete motion class.

Our ArTIST architecture is depicted by Fig. 2, whose top portion illustrates the high-level overview of ArTIST during training. In general, during training the model takes as input all alive tracklets $\mathbb{T}$, and jointly learns the distribution of each tracklet, shown in green in Fig. 2-top, together with a representation of the interactions, shown in red. Since we aim to estimate a probability distribution over the bounding box position in the next time-step, we train our model with a negative log-likelihood loss function. Additionally, to learn an expressive representation of the interactions, we use a moving agent autoencoder network (MA-Net) that is trained to reconstruct all the interacting tracklets, as discussed in more detail below. Thus, to train our model, we minimize

$$\mathcal{L}_{\text{total}} = \lambda\mathcal{L}_{\text{NLL}} + \mathcal{L}_{rec}\,, \quad (2)$$

where $\mathcal{L}_{rec}$ is the mean squared error loss, $\mathcal{L}_{\text{NLL}}$ is the negative log-likelihood of the GT boxes given the estimated tracklet distribution, and λ is an annealing function. We start from $\lambda = 0$, forcing the model to learn better interaction representations first, and gradually increase it to $\lambda = 1$, following a logistic curve, to account for both terms equally.

As shown in Fig. 2-bottom, ArTIST itself relies on a recurrent residual architecture to represent motion velocities. At each time-step t, it takes as input a motion velocity represented by $\Delta^t_{\mathcal{T}_j} = (\delta x^t_{\mathcal{T}_j}, \delta y^t_{\mathcal{T}_j}, \delta w^t_{\mathcal{T}_j}, \delta h^t_{\mathcal{T}_j})$ and an interaction representation I^t_j, discussed below. Given these inputs and the hidden state computed in the last time-step z^{t-1}_j, it predicts a distribution over the motion velocity for time $t+1$, *i.e.*, $p(\Delta^{t+1}_{\mathcal{T}_j} \mid z^{t-1}_j, \Delta^t_{\mathcal{T}_j}, I^t_j)$. This approximates the definition in Eq. 1, since z^{t-1}_j carries information about all previous time-steps.

Moving Agent Interactions. Most of the existing MOT frameworks [5, 17, 41, 56] treat each agent as independent from other agents in the scene. A few approaches [32, 37, 47] have nonetheless shown the benefits of modeling the interactions between agents. We believe that an effective modeling of interactions will lead to better tracking quality as the motion of each pedestrian may be affected by the behaviour of the other agents in the scene. In this paper, we do so using the Moving Agent Network, MA-Net, illustrated in Fig. 2. MA-Net is a recurrent autoencoder neural network that learns to reconstruct the tracklets of all moving agents potentially interacting with the tracklet of interest, *e.g.*, $\mathcal{T}_j$. During training, the encoder compresses the tracklets into a latent representation (*i.e.*, the hidden state of the last time-step), and the decoder reconstructs all tracklets given their compressed latent representations. To learn distribution of $\mathcal{T}_j$, ArTIST then needs a representation of the interacting agents that depends neither on their number nor on their order. We achieve this via max-aggregation of the latent representations of all interacting agents, $\mathbb{T} \setminus \{\mathcal{T}_j\}$. Specifically, we take the hidden-state of the last recurrent cell in the MA-Net encoder for the N_{I_j} interacting agents, leading to a matrix in $\mathbb{R}^{\text{N}_{I_j} \times L}$, where L is the hidden state dimension. We then perform max-pooling over the first dimension of this matrix, giving us $I_j \in \mathbb{R}^L$. Note that, during tracking (*i.e.*, at test time), we remove the MA-Net decoder and only rely on the representation computed by its encoder.

3.4. Tracklet Scoring

Given the trained ArTIST model, we can score how likely a detection at time t is to be the continuation of a tracklet $\mathcal{T}_j$. To this end, given $\mathcal{T}_j$'s velocity sequence and I_j, the model estimates a probability distribution over the location of the bounding box at time t. We then take the likelihood of the observed detection given the estimated distribution as a score for the tracklet-detection pair. Specifically, we compute the Δ, *i.e.*, the potential velocity of changes in x, y, w, and h made by any detection with respect to the previous observation (or inpainted bounding box if the previous time-step was inpainted). We then take the probability estimated for the centroid closest to this Δ as likelihood. In practice, we assume independence of the bounding box parameters, *i.e.*, $\delta x^t_{\mathcal{T}_j}$, $\delta y^t_{\mathcal{T}_j}$, $\delta w^t_{\mathcal{T}_j}$, and $\delta h^t_{\mathcal{T}_j}$. Therefore, we have four sets of clusters and thus four probability distributions estimated at each time-step, as shown in Fig. 2-bottom. We then compute the likelihood of a bounding box as the product of the probabilities of the components, as

$$p(\Delta^{t+1}_{\mathcal{T}_j} \mid z^{t-1}_j, \Delta^t_{\mathcal{T}_j}, I^t_j) = \prod_{\xi \in \{x,y,w,h\}} p(\delta \xi^{t+1}_{\mathcal{T}_j} \mid z^{t-1}_j, \Delta^t_{\mathcal{T}_j}, I^t_j). \tag{3}$$

In practice, we do this in log space, summing over the log of the probabilities.

3.5. Tracklet Inpainting

In the real world, detection failures for a few frames are quite common due to, *e.g.*, occlusion. Such failures complicate the association of upcoming detections with the tracklets, and thus may lead to erroneous tracklet terminations. Our approach overcomes this by inpainting the tracklets for which no detections are available. Let us consider the scenario where a tracklet was not assigned any detection in the past few frames. We now seek to check whether a new detection at the current time-step belongs to it. To compute a likelihood for the new observation, we need to have access to the full bounding box sequence up to the previous time-step. To this end, we use our model to inpaint the missing observations, as illustrated in the bottom of the Fig. 1-middle, by *multinomial* sampling from the learned tracklet distribution. Sampling can in fact be done autoregressively to create a diverse set of full sequence of observations and inpainted boxes, which, in turn, allows us to score a new detection. To account for the fact that motion is stochastic by nature, especially for humans, we sample S candidates for the whole subsequence to inpaint from the estimated distribution and get multiple plausible inpainted tracklets. Since ArTIST relies solely on geometric information, on its own, it cannot estimate which of the S inpainted options are valid. To select one of these candidates, we use a tracklet rejection scheme (TRS), as follows: if there is a candidate to be selected, we compute the intersection over union (IoU) of the last generated bounding box with all the detections in the scene. The model then selects the candidate with highest IoU, if it surpasses a threshold. However, in some cases, the last generated bounding box of one of the candidates may overlap with a false detection or a detection for another object, *i.e.*, belonging to a different tracklet. To account for these ambiguities, we continue predicting boxes for all candidates for another 1–2 frames and compute the

IoUs for these frames as well. ArTIST then selects the candidate with the maximum sum of IoUs. This allows us to ignore candidates matching a false detection or a detection for another object moving in a different direction. However, this may not be enough to disambiguate all cases, *e.g.*, the detections belonging to other tracklets that are close-by and moving in the same direction. We treat these cases in our assignment strategy discussed below.

3.6. Assignment

To assign the detections to the tracklets at each time-step, we use the linear assignment found by the Munkres algorithm [42]. This method relies on a cost matrix C, storing the cost of assigning each detection to each tracklet. In our case, the costs are negative log-likelihoods computed by ArTIST. Let us denote by $C^t_{ij} = \log p(\langle d^t_i, \mathcal{T}^t_j \rangle)$ the negative log-likelihood of assigning detection i to tracklet j at time t. The Munkres algorithm then returns the indices of associated tracklet-detection pairs by solving $A^\star = \arg\min_{A^t} \sum_{i,j} C^t_{ij} A^t_{ij}$, where $A^t \in [0,1]^{N\times M}$ is the assignment probability matrix, with N the number of detections and M the number of tracklets. This matrix satisfies the constraints $\sum_j A^t_{ij} = 1,\ \forall i$ and $\sum_i A^t_{ij} = 1,\ \forall j$.

In practice, to account for the fact that we are less confident about the tracklets that we inpainted, we run the Munkres algorithm twice. First, using only the tracklets whose scores at the previous time-step were obtained using actual detections; second, using the remaining tracklets obtained by inpainting and the unassigned detections.

4. Experiments

In this section, we evaluate different aspects of ArTIST and compare it with existing methods. In our experiments, bold and underlined numbers indicate the best and second best results, respectively. We provide the implementation details of our approach in the supplementary material.

Datasets. We use MOTChallenge benchmarks[2]. MOTChallenge consists of several challenging pedestrian tracking sequences with moving and stationary cameras capturing the scene from various viewpoints and at different frame rates. We report our results on the three benchmarks of this challenge, MOT16 [40], MOT17 [40], and the recently introduced MOT20 [13]. MOT17 contains 7 training-testing sequence pairs with similar statistics. Three sets of public detections, namely DPM [18], Faster R-CNN [45] and SDP [61], are provided with the benchmark. The sequences of MOT16 are similar to those of MOT17, with detections computed only via DPM. MOT20 contains videos of very crowded scenes, in which there are many long occlusions occurring frequently. This dataset consists of 8 different sequences from 3 different scenes that are captured in both indoor and outdoor locations, during day and night. This dataset contains over 2M bounding boxes and 3,833 tracks, 10 times more than MOT16. For the ablation studies, we follow the standard practice of [66] and thus split each training sequence into two halves, and use the first half for training and the rest for validation. Note that our main results are reported on the test sets of each benchmark dataset. In all of our experiments, unless otherwise stated, we follow the standard practice of refining the public detections, which is allowed by the benchmarks and commonly done by the challenge participants [3, 9, 26, 58, 66].

Evaluation Metrics. To evaluate MOT approaches, we use the standard metrics [4, 46] of MOT Accuracy (MOTA), Identity F1 Score (IDF1), number of identity switches (IDs), mostly tracked (MT), mostly lost (ML), false positives (FP), and false negatives (FN). The details of these metrics are provided in the supplementary material.

4.1. Comparison with the State of the Art

We compare ArTIST with existing approaches that, as ours, use the public detections provided by the benchmarks. In this section, we evaluate ArTIST in two settings, ArTIST-T, which utilizes the bounding box regression of [3], and ArTIST-C, which uses the bounding box regression of [66], both acting on the public detections provided by the MOT benchmark datasets. For the sake of completeness, we consider both online and offline approaches. However, only online approaches are directly comparable to ArTIST.

As clearly shown in our results in Tables 1, 2, and 3, thanks to its stochastic, multi-modal motion model, ArTIST is capable of maintaining the identities of the tracklets for longer time periods, evidenced by superior IDF1 scores. Doing so allows ArTIST to keep more tracklets for more than 80% of their actual lifespan, resulting in very high MT and very low IDs, outperforming all other competing methods. Another important aspect of ArTIST is its capability to inpaint the gaps due to detection failures. Filling such gaps not only has a great impact on identity preservation, but also significantly reduces the FN, a metric that is often ignored by existing trackers. As such, it directly affects the MOTA metric[3], as there exist considerably more FN than FP and IDs, according to which our approach again outperforms existing methods by a considerable margin.

As clearly evidenced by the performance of our approach on the challenging MOT20 [13] benchmark dataset, ArTIST is also a better tracker in highly crowded scenarios with frequent occlusions. In this benchmark, the mean crowd density is 10 times higher than in MOT16 and MOT17, reaching 246 pedestrians per frame. ArTIST's significant improvement in almost all MOT metrics demonstrates the benefit of using a better motion model, perform-

[2] https://motchallenge.net/

[3] MOTA is defined as $1 - \sum_t (FN_t + FP_t + IDw_t) / \sum_t GT_t$.

Table 1. Results on MOT16 benchmark dataset on test set. "RB" indicates whether the baselines use bounding box refinement.

Method	RB	Mode	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
JPDA [20]	✗	Off	26.2	-	365	4.1	67.5	3,689	130,549
BiLSTM [28]	✗	Off	42.1	47.8	753	14.9	44.4	11,637	93,172
MHT-DAM [27]	✗	Off	45.8	46.1	590	16.2	43.2	6,412	91,758
LMP [54]	✗	Off	48.8	51.3	481	18.2	40.1	6,654	86,245
MPNTrack [9]	✓	Off	58.6	61.7	354	27.3	34.0	4,949	70,252
EAMTT [50]	✗	On	38.8	42.4	965	7.9	49.1	8,114	102,452
AMIR [47]	✗	On	47.2	46.3	774	14.0	41.6	2,681	92,856
DMAN [68]	✗	On	46.1	54.8	532	17.4	42.7	7,909	89,874
MOTDT [35]	✗	On	47.6	50.9	792	15.2	38.3	9,253	85,431
STRN [57]	✗	On	48.5	53.9	747	17.0	34.9	9,038	84,178
Tracktor++ [3]	✓	On	54.4	52.5	682	19.0	36.9	3,280	79,149
Tracktor++v2 [3]	✓	On	56.2	54.9	617	20.7	35.8	**2,394**	76,844
DeepMOT-T [58]	✓	On	54.8	53.4	645	19.1	37.0	2,955	78,765
UMA [62]	✓	On	50.5	52.8	685	17.8	33.7	7,587	81,924
ArTIST-T	✓	On	56.6	57.8	**519**	22.4	37.5	3,532	75,031
ArTIST-C	✓	On	**63.0**	**61.9**	635	**29.1**	**33.2**	7,420	**59,376**

Table 2. Results on MOT17 benchmark dataset on test set. The second column (RB) indicates whether the baselines use bounding box refinement. Note that CenterTrack* [66] utilizes re-birthing.

Method	RB	Mode	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
IOU17 [7]	✗	Off	45.5	39.4	5,988	15.7	40.5	19,993	281,643
EEB&LM [37]	✗	Off	44.2	57.2	1,529	16.1	44.3	29,473	283,611
BiLSTM [28]	✗	Off	47.5	51.9	2,069	18.2	41.7	25,981	268,042
MHT-DAM [27]	✗	Off	50.7	47.2	2,314	20.8	36.9	22,875	252,889
eHAF [51]	✗	Off	51.8	54.7	1,843	23.4	37.9	33,212	236,772
MPNTrack [9]	✓	Off	58.8	61.7	1,185	28.8	33.5	17,413	213,594
MOTDT [35]	✗	On	50.9	52.7	2,474	17.5	35.7	24,069	250,768
DMAN [68]	✗	On	48.2	55.7	2,194	19.3	38.3	26,218	263,608
EAMTT [50]	✗	On	42.6	41.8	4,488	12.7	42.7	30,711	288,474
GMPHD [30]	✗	On	39.6	36.6	5,811	8.8	43.3	50,903	284,228
SORT17 [5]	✗	On	43.1	39.8	4,852	12.5	42.3	28,398	287,582
STRN [57]	✗	On	50.9	56.5	2,593	20.1	37.0	27,532	246,924
Tracktor++ [3]	✓	On	53.5	52.3	2,072	19.5	36.6	12,201	248,047
Tracktor++v2 [3]	✓	On	56.5	55.1	3,763	21.1	35.3	**8,866**	235,449
DeepMOT-T [58]	✓	On	53.7	53.8	1,947	19.4	36.6	11,731	247,447
FAMNet [11]	✓	On	52.0	48.7	3,072	19.1	33.4	14,138	253,616
UMA [62]	✓	On	53.1	54.4	2,251	21.5	31.8	22,893	239,534
CenterTrack* [66]	✓	On	61.5	59.6	2,583	26.4	31.9	14,076	200,672
CenterTrack [66]	✓	On	61.4	53.3	5,326	27.9	**31.4**	15,520	196,886
ArTIST-T	✓	On	56.7	57.5	**1,756**	22.7	37.2	12,353	230,437
ArTIST-C	✓	On	**62.3**	**59.7**	2,062	**29.1**	34.0	19,611	**191,207**

Table 3. Results on MOT20 benchmark dataset on test set. Note that the methods denoted by * are the ones reported on CVPR2019 Challenge in which the videos are similar to MOT20 with very minor corrections in the ground-truth. The second column (RB) indicates whether the baselines use bounding box refinement.

Method	RB	Mode	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
IOU19 [7]*	✗	Off	35.8	25.7	15,676	10.0	31.0	24,427	319,696
V-IOU [8]*	✗	Off	46.7	46.0	2,589	22.9	24.4	33,776	261,964
MPNTrack [9]	✓	Off	57.6	59.1	1,210	38.2	22.5	16,953	201,384
SORT20 [5]	✗	On	42.7	45.1	4,470	16.7	26.2	27,521	264,694
TAMA [65]*	✗	On	47.6	48.7	2,437	27.2	**23.6**	38,194	252,934
Tracktor++ [3]*	✓	On	51.3	47.6	2,584	24.9	26.0	16,263	253,680
ArTIST-T	✓	On	**53.6**	**51.0**	**1,531**	**31.6**	28.1	**7,765**	**230,576**

ing stochastic sampling for tracklet inpainting, and employing a probabilistic scoring function.

4.2. Ablation Study

In this section, we evaluate different components of ArTIST using ArTIST-C on the MOT17 validation set with the public Faster-RCNN [45] detections.

Table 4. Evaluating the effect of utilizing interactions.

Setting	MOTA ↑	IDF1↑	IDs↓	MT↑	ML↓	FP↓	FN↓
Isolation	58.6	62.0	293	30.4	28.0	**1,466**	20,556
Interaction	**59.8**	**66.5**	**231**	**33.3**	**27.7**	1,675	**19,769**

Table 5. Evaluating the effect of tracklet inpainting.

Setting	MOTA ↑	IDF1↑	IDs↓	MT↑	ML↓	FP↓	FN↓
No Inpainting	56.2	60.7	292	24.8	30.7	**1,150**	22,168
Invisible Inpainting	56.6	64.4	**216**	25.4	30.1	1,173	22,004
Visible Inpainting	**59.8**	**66.5**	231	**33.3**	**27.7**	1,675	**19,769**

Effect of Utilizing Interactions. Most existing trackers treat individual tracklets independently of the other agents in the scene, ignoring the fact that the motion of each person is affected by that of the other pedestrians. This typically results in an increase of identity switches when different pedestrians are moving toward and passing each other, thus directly affecting the identity preservation capability of a tracker, which can be seen in the IDF1, IDs, and MT metrics. In Table 4, we evaluate the effect of our approach to accounting for interactions across agents, as described in Section 3.3, by comparing it to the "Isolation" setting, where no interactions are utilized. Note that exploiting interactions improves almost all metrics, except for FP. A better identity preservation capability leads to an inevitable slight increase in FP since there are more attempts toward inpainting continuations of tracklets in case of occlusions, which is discussed below.

Effect of Inpainting. As discussed in Section 3.5, filling in the gap to compensate detector's failure leads to better identity preservation in a tracking framework. We demonstrate this effect in Table 5, where we compare the no-inpainting case, with inpainting in visible or invisible mode. In the invisible mode, we do not consider the inpainted bounding boxes in the evaluations, whereas in the visible mode we do. As shown by our results, inpainting significantly improves the identity-sensitive metrics, such as IDF1, IDs, MT, and ML. This experiments also shows that incorporating the inpainted bounding boxes (*i.e.*, the visible mode) improves FN significantly which has a direct impact on MOTA. We observe that, while the inpainted tracklets resemble natural human motion, not all inpainted boxes correctly match the ground truth[4], leading to a slight increase in FP and IDs. However, since FN is typically two to three orders of magnitude higher than FP and IDs, we see an overall improvement in tracking. In Fig. 3, we provide a qualitative evaluation of the effect of inpainting, showing that our approach can better handle multiple occlusions.

Effect of Multinomial Sampling. As discussed in Section 3.5, ArTIST is capable of generating multiple plausible motion continuations by multinomial sampling from the

[4]This could also be due to the fact that GT provides an approximation of the position in case of occlusion.

Table 6. Evaluating the effect of multinomial sampling and TRS.

Setting	MOTA ↑	IDF1↑	IDs↓	MT↑	ML↓	FP↓	FN↓
Top-1	58.8	62.5	293	30.7	28.0	**1,444**	20,484
Multi.	59.4	64.9	257	32.7	28.0	1,733	19,870
Multi.+TRS	**59.8**	**66.5**	**231**	**33.3**	**27.7**	1,675	**19,769**

Table 7. Evaluating the effect of our motion modeling.

Setting	MOTA ↑	IDF1↑	IDs↓	MT↑	ML↓	FP↓	FN↓
No Motion	58.1	60.7	614	31.3	**25.6**	1,572	20,379
Kalman Filter	56.8	61.9	247	24.7	30.7	**1,185**	21,865
CenterTrack [66]	58.0	60.6	509	31.2	25.9	1,795	20,337
Ours	**59.8**	**66.5**	**231**	**33.3**	27.7	1,675	**19,769**

Table 8. Evaluating the effect of public bounding box refinement.

Setting	MOTA ↑	IDF1↑	IDs↓	MT↑	ML↓	FP↓	FN↓
Not Refined	48.1	57.0	255	27.1	30.7	1,411	26,327
ArTIST-C	59.8	66.5	231	33.3	27.7	1,675	19,769
ArTIST-C (Private)	68.5	71.1	229	42.2	21.5	2,343	14,409

learned distribution. In Table 6, we compare a model that ignores the stochasticity in human motion, and thus greedily generates a single continuation of a tracklet for inpainting (denoted by “Top-1”), with one that takes stochasticity into account (denoted by “Multi.”). Note that, with more inpainted options, the model achieves better performance. However, large numbers of samples may introduce ambiguities, causing a decrease in tracking performance. To handle this, we disambiguate such scenarios using our tracklet rejection strategy (TRS), as shown in the third row of Table 6. As shown in Table 1 of supp. material, for sequences captured by a static camera, and for tracklets with relatively long observations, Top-1 sampling performs reasonably well, almost on par with multinomial sampling. This is due to the fact that, with long observations, our approach captures the motion pattern and can reliably fill in the gaps. However, when it comes to moving cameras or newly born tracklets (with relatively short observations), multinomial sampling (with TRS) leads to more reliable tracking.

Effect of Stochastic Motion Modeling. The key component of our approach is our stochastic motion model that is capable of capturing the multi-modal nature of human motion. To evaluate its effectiveness, given the same set of detections, we compare it with no motion model (setting CenterTrack’s offsets [66] to zero), a linear and uni-modal probabilistic motion model (Kalman Filter [5]), and a nonlinear and deterministic motion model (existing SOTA CenterTrack [66]) in Table 7. As shown in this table and in Fig. 3, the effect of learning a multi-modal distribution in scoring and inpainting is directly proportional to the success of the model at handling occlusions and thus at preserving identities for a longer time, resulting in a considerable improvement in metrics such as IDF1, IDs, and MT.

Effect of Bounding Box Refinement. A number of recent tracking techniques [3, 37, 58, 66] refine the bounding boxes computed by the detectors. In particular, [3, 58] use Faster R-CNN [45] with ResNet-101 [21] and Feature Pyramid Networks [34] trained on the MOT17Det [40]

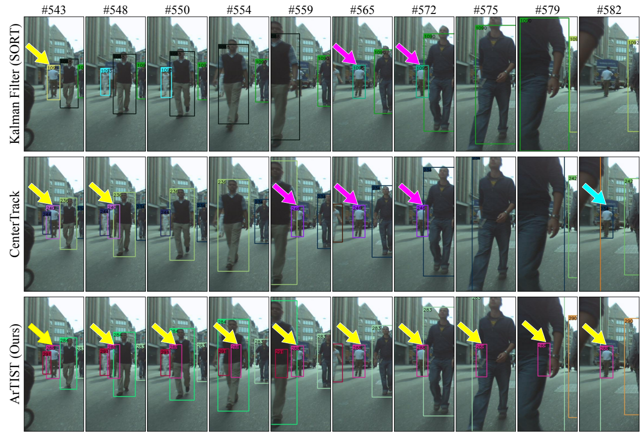

Figure 3. Qualitative result of handling occlusion in a moving camera scenario where colored arrows point to the bounding boxes of the pedestrian of interest and changing in the color of arrows shows a change in the identity of that pedestrian. Unlike Kalman Filter [5] and CenterTrack [66], our method preserves the identity after two occlusions and also inpaints the bounding boxes in occlusions. Note, all the methods are using exactly the same detections.

pedestrian detection dataset to refine the public detections provided with the MOTChallenge. Following [3], CenterTrack [66] also utilizes such refinement. Note that, as acknowledged by [3, 66], for the comparison with the methods that use the public detections to be fair, the new trajectories are still initialized from the public detection bounding boxes, and thus refinement is not used to detect a new bounding box. In this experiment, we evaluate the effectiveness of this refinement step. As shown by Table 8, refinement leads to better tracking quality compared to the “Not Refined” setting, where the public detections are directly used in our tracking framework. Moreover, we evaluate the effect of using more accurate detected bounding boxes provided by a different detector, CenterNet [67], which not surprisingly leads to even better tracking performance.

5. Conclusion

We have introduced an online MOT framework based on a stochastic autoregressive motion model. Specifically, we have employed this model to both score tracklets for detection assignment purposes and inpaint tracklets to account for missing detections. Our results on the MOT benchmark datasets have shown the benefits of relying on a probabilistic multi-modal representation of motion, especially when dealing with challenging crowded scenarios with frequent occlusions, as in MOT20. Notably, without using any complex components, such as person Re-ID, our framework yields state of the art performance.

Acknowledgment. This research was supported by the Australian Research Council Centre of Excellence for Robotic Vision and funded by the Australian Government.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 1

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1, 2

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 941–951, 2019. 1, 2, 6, 7, 8

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008. 6

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *International Conference on Image Processing*, pages 3464–3468. IEEE, 2016. 1, 2, 5, 7, 8

[6] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004. 2

[7] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017. 7

[8] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending iou based multi-object tracking by visual information. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018. 7

[9] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020. 2, 6, 7

[10] Peng Chu, Heng Fan, Chiu C Tan, and Haibin Ling. Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 161–170. IEEE, 2019. 2

[11] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6172–6181, 2019. 2, 7

[12] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845, 2017. 2

[13] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 2, 6

[14] Caglayan Dicle, Octavia I Camps, and Mario Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2311, 2013. 1, 2

[15] Gennady Erlikhman and Gideon P Caplovitz. Decoding information about dynamically occluded objects in visual cortex. *NeuroImage*, 146:778–788, 2017. 1

[16] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008. 2

[17] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision*, pages 466–475. IEEE, 2018. 1, 2, 5

[18] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 6

[19] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 1

[20] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055, 2015. 2, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 8

[22] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2

[23] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6272–6281, 2019. 1

[24] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2375–2384, 2019. 1

[25] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 1960. 1, 2

[26] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple unsupervised multi-object tracking. *arXiv preprint arXiv:2006.02609*, 2020. 6

[27] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceed-*

ings of the IEEE International Conference on Computer Vision, pages 4696–4704, 2015. 2, 7

[28] Chanho Kim, Fuxin Li, and James M Rehg. Multi-object tracking with neural gating using bilinear lstm. In *European Conference on Computer Vision*, pages 200–215, 2018. 2, 7

[29] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems*, pages 137–146, 2019. 1

[30] Tino Kutschbach, Erik Bochinski, Volker Eiselein, and Thomas Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–5. IEEE, 2017. 7

[31] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–40, 2016. 2

[32] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 120–127. IEEE, 2011. 5

[33] Yiming Liang and Yue Zhou. LSTM multiple object tracker combining multiple cues. In *International Conference on Image Processing*, pages 2351–2355. IEEE, 2018. 2

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 8

[35] Chen Long, Ai Haizhou, Zhuang Zijie, and Shang Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, volume 5, page 8, 2018. 2, 7

[36] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. 4

[37] Andrii Maksai and Pascal Fua. Eliminating exposure bias and metric mismatch in multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4639–4648, 2019. 2, 5, 7, 8

[38] Andrii Maksai, Xinchao Wang, Francois Fleuret, and Pascal Fua. Non-markovian globally consistent multi-object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2544–2554, 2017. 2

[39] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, August 2020. 1

[40] 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. 6, 8

[41] Anton Milan, S Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1, 2, 5

[42] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 2, 3, 6

[43] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 4

[44] Nan Ran, Longteng Kong, Yunhong Wang, and Qingjie Liu. A robust multi-athlete tracking algorithm by exploiting discriminant features and long-term dependencies. In *International Conference on Multimedia Modeling*, pages 411–423. Springer, 2019. 1, 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2, 3, 6, 7, 8

[46] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 6

[47] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017. 1, 2, 5, 7

[48] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. SoPhie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 1

[49] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *European Conference on Computer Vision*, 2020. 1

[50] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016. 7

[51] Hao Sheng, Yang Zhang, Jiahui Chen, Zhang Xiong, and Jun Zhang. Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3269–3280, 2018. 7

[52] Scott D Slotnick, William L Thompson, and Stephen M Kosslyn. Visual mental imagery induces retinotopically organized activation of early visual areas. *Cerebral cortex*, 15(10):1570–1583, 2005. 1

[53] Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection

and tracking. *European Conference on Computer Vision*, 2020. 1, 2

[54] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017. 2, 7

[55] Xingyu Wan, Jinjun Wang, and Sanping Zhou. An online and flexible multi-object tracking framework using long short-term memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1230–1238, 2018. 1, 2

[56] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *International Conference on Image Processing*, pages 3645–3649. IEEE, 2017. 1, 2, 5

[57] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3988–3998, 2019. 2, 7

[58] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6787–6796, 2020. 6, 7, 8

[59] Bo Yang and Ram Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1918–1925. IEEE, 2012. 1, 2

[60] Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *European Conference on Computer Vision*, pages 484–498. Springer, 2012. 1, 2

[61] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016. 6

[62] Junbo Yin, Wenguan Wang, Qinghao Meng, Ruigang Yang, and Jianbing Shen. A unified object motion and affinity model for online multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6768–6777, 2020. 7

[63] Kwangjin Yoon, Young-min Song, and Moongu Jeon. Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views. *IET Image Processing*, 12(7):1175–1184, 2018. 2

[64] Young-chul Yoon, Abhijeet Boragule, Young-min Song, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. In *IEEE International conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018. 2

[65] Young-Chul Yoon, Du Yong Kim, Young-min Song, Kwangjin Yoon, and Moongu Jeon. Online multiple pedestrians tracking using deep temporal appearance matching association. *Information Sciences*, 2020. 7

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *European Conference on Computer Vision*, 2020. 1, 2, 6, 7, 8

[67] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 8

[68] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *European Conference on Computer Vision*, pages 366–382, 2018. 2, 7