School of Computation,
Information and Technology

Technical University of Munich

Doctoral Thesis

# Deep Learning for Human Motion: Advancing Trajectory Prediction and Multi-Object Tracking

**Patrick Dendorfer**

School of Computation,
Information and Technology

Technical University of Munich

# Deep Learning for Human Motion: Advancing Trajectory Prediction and Multi-Object Tracking

**Patrick Dendorfer**

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

| | | |
|---|---|---|
| **Vorsitz:** | | Prof. Dr. Angela Dai |
| **Prüfer*innen der Dissertation:** | 1. | Prof. Dr. Laura Leal-Taixé |
| | 2. | Prof. Dr. Alexandre Alahi |

Die Dissertation wurde am 09.11.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 06.03.2023 angenommen.

# PREFACE

Researchers have been driven to study the dynamic of objects for centuries, ranging from planets and galaxies to the microscopic motion of electrons and protons. Understanding how objects move and why the world behaves as it does, was my motivation to study Physics after finishing high school.

Learning about Newtonian mechanics in my first theoretical physics course was a fascinating experience. I learned that one could precisely determine the object's trajectory of its past, present, and future by only following the *principle of least action*. The principle states that physical objects move along extremal trajectories to minimize energy, distance, or travel time. From there, equations that explain natural phenomena magically fall out of the equations; Fermat's principle describes light scattering in different media or the Brachistochrone curve that explains the shortest path between two points in a gravitational force field.

After my university graduation, I heard for the first time about machine learning and neural networks. Immediately, I was thrilled by their capabilities. "How is it possible that simple, functional layers result in a structure capable of executing tasks that usually require human intelligence." On this note, have you realized that the cover image of this thesis was generated by an AI? Indeed, the image was automatically created by OpenAI's DALL·E 2 [17] given a short text description of the visualized scenario.

It became quickly apparent that I wanted to dedicate myself to this topic to understand the magic behind neural networks and pursue a doctoral degree in the field. Hence, I applied to the Dynamic Vision and Learning (DVL) group and was fortunate to be accepted as a doctoral candidate.

My background in Physics made it easier for me to start with a project on pedestrian trajectory prediction project because it combined my interest in motion with state-of-the-art deep learning methods. Later, I became more involved in multi-object tracking, where motion plays a critical role in tracking objects across different time frames. Multi-object tracking and trajectory prediction are often treated as two separate fields of research. However, in our understanding, both disciplines could benefit each other. So, our goal was to unite both fields and demonstrate the beneficial synergies.

After more than four years, I am excited to present our research and advances in the field. Despite being the single author of this dissertation, I use the plural pronoun *we* throughout the text because research is a team sport, and my journey would not have been possible without the help of many people supporting me.

First and foremost, special thanks to my supervisor Laura. You gave me the chance and freedom to explore the world of neural networks and to let me work on my ideas. Thank you for all your support, your pieces of advice, your patience on last-minute paper submissions, our party times, and many more.

I also want to thank Aljoša for spending much time discussing ideas and writing papers with me. Without your help, our papers would not even be half as sound and structured.

Our DVL group rapidly grew in numbers. We shared good times at the retreat, journal clubs, and inspiring discussions on research ideas. Thanks to all of my colleagues: Andreas, Aysim, Franzi, Guillem, Ili, Jenny, Mark, Matthias, Maxim, Orcun, Qunjie, Robert, Sergio, Thomas, and Tim.

I want to thank Sabine for taking care of any administrative issues, and Quirin for his continuous support. Special kudos to two of my students at TUM, Sven, and Vladimir. You both showed outstanding dedication and motivation, which was awarded our published papers.

During my fourth year, I worked in the perception team of Argo AI in Munich for five months. It was an exciting experience to see the integration of academic research on self-driving cars and the complexity of hundreds of engineers working together towards one goal. I want to thank my supervisors Andreas, Alex, Andrew, Frie, and the entire perception team, who all were very welcoming and supportive.

Lastly, I want to thank my family, best friends, and partner, for all their support in whichever goal I wanted to pursue. Theresa, you supported me during stressful deadline periods, comforted me with rejections, and enthusiastically celebrated successes with me.

The journey of this PhD has been both challenging and exciting. I had the privilege to learn many new things and the time to study thoroughly the problems I am deeply interested in. The personal experiences and learnings are invaluable, and I am proudly finishing my PhD with many lessons learned. Now, my journey at TUM is ending, but I am excited about my future steps, which I hope will be as inspiring and promising as my PhD.

Patrick Dendorfer
Munich, 09.11.2022

# ABSTRACT

Visual perception and understanding of human motion are fundamental problems in computer vision and essential abilities for modern computer systems. Understanding a specific scenario involves perceiving, interpreting, and forecasting object motion. Self-driving cars and social robots must be capable of tracking pedestrians, predicting their future paths, and anticipating their actions. Therefore, systems need a model for human motion and interaction. In the computer vision literature, these tasks relate to the two separate fields of multi-object tracking and pedestrian trajectory prediction.

This thesis explores these fields and contributes new methodologies, metrics, and datasets to them. The focus is on deep learning methods, which have dominated research in these fields in recent years. While there have been several success stories, numerous challenges remain.

One of the most critical properties of pedestrian trajectory prediction methods is to provide multimodal forecasts and to reflect the uncertainty of the predictions. Most methods commonly use a vanilla GAN architecture that has limitations in generating multimodal distributions. In this study, we investigate the properties of multimodal trajectory prediction and propose two novel architectures for generating realistic and interpretable distributions of pedestrian trajectories. Our first solution, Goal GAN, generates a discrete distribution of realistic goal positions. The sampled goal positions help generate trajectories reaching these goals. The second method, MG-GAN, consists of multiple generators, each specializing in predicting trajectories of a specific mode of the target distribution.

Our contribution to multi-object tracking includes *MOTChallenge*, a platform for multi-object tracking datasets and model evaluation. With help of the collected information, we present an elaborative study on the development and trends of state-of-the-art pedestrian tracking methods and identify ongoing challenges for the task. Furthermore, we present a novel tracking metric, HOTA, that improves the balance between detection and association errors, and MOTCOM, a measure that describes the complexity of tracking sequences.

As a final contribution, we incorporate our state-of-the-art pedestrian trajectory prediction methods into multi-object tracking to overcome the challenge of bridging long-term occlusions in tracking. Our approach estimates a bird's-eye view transformation by fusing semantic and depth image information to represent tracklets in metric space. Once a track is lost, we predict future multimodal trajectories and try to re-match these inactive tracks with new detections in the association step. With this approach, we can significantly decrease the number of false associations even after an occlusion of multiple seconds and achieve state-of-the-tracking performance.

Ultimately, we advocate that both fields of multi-object tracking and trajectory prediction work closely together in the future and develop jointly as they benefit each other.

# CONTENTS

I Introduction 1

1 Motivation . . . 1

1.1 Real-World Applications . . . 2

2 Thesis Outline . . . 2

II Background and Contributions 4

1 Trajectory Forecasting . . . 5

1.1 Background . . . 5

1.1.1 Problem Formulation . . . 5

1.1.2 Trajectory Prediction Datasets . . . 6

1.1.3 Trajectory Prediction Methods . . . 6

1.1.4 Generative Trajectory Prediction Methods . . . 8

1.1.5 Evaluation Metrics . . . 13

1.2 Multimodal Pedestrian Trajectory Prediction . . . 15

2 Multi-Object Tracking . . . 18

2.1 Background . . . 18

2.1.1 Problem Formulation . . . 18

2.1.2 Multi-Object Tracking Datasets . . . 19

2.1.3 Evaluation Metrics . . . 20

2.1.4 Multi-Object Tracking Methods . . . 21

2.2 Benchmarking, Evaluating, and Analysing Multi-Object Trackers . 25

3 Camera Projection of Moving Objects . . . 28

3.1 Background . . . 28

3.1.1 Camera Projection . . . 28

3.1.2 Motion Projection . . . 29

3.1.3 Homography Transformation . . . 30

3.2 Solving Long-Term Occlusions with Trajectory Prediction . . . 31

III Summary of Selected Publications 33

1 Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation . . . 34

2 MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction . . . . . . . . . . . . . . . . . . . 36
3 MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking 38
4 Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

**IV Concluding Remarks 41**
1 Integrated Real-World Trajectory Prediction . . . . . . . . . . . . . . . . . . 41
2 Tracking-by-Forecasting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3 Alternative Data Sources for Training and Testing Models . . . . . . . . . . 42
4 “Epur si muove." – “And yet it does move." . . . . . . . . . . . . . . . . . 43

**Bibliography 44**

**Publications 51**
Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation 52
MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction . . . . . . . . . . . . . . . . . . . . . . . 70
MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking . . 81
Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119

# I Introduction

“Epur si muove." – “And yet it does move."
– Galileo Galilei

## 1 Motivation

Humans can track moving objects and anticipate their future positions. We experience the world with our senses and process visual information with our brains. Our ability to see develops at a very early age. It enables us to walk, cycle, or drive a car in our daily lives. However, building a computer system to do alike is not a trivial task.

Artificial computer systems that aim to perform these tasks require a visual understanding. With the advent of modern computers, researchers established the field of computer vision [61] to build computer programs that perceive and understand the world around us. Automatic processing and interpretation of visual information is the central aspect of computer vision where the computer is taught *to see*.

Along this line, the tasks of tracking and forecasting have a long and rich history in the field of computer vision. Continuous improvement of input sensors, computational power, and algorithms favored a steady development over time. However, only a decade ago, the revolutionary breakthrough of deep learning algorithms, in combination with a large amount of available data, led to a significant boost in performance and drew much attention to these tasks. These key technologies may enable new autonomous platforms, such as self-driving cars, to revolutionize the future of mobility.

Most real-world objects and problems are dynamic. So we need to track and forecast human motion from moving frames. While other tasks in computer vision, *i.e.,* object detection or segmentation, can be formulated on a static frame-by-frame setup, multi-object tracking, and pedestrian trajectory prediction rely on a dynamic video input stream. In this dissertation, we focus on understanding human motion in video sequences. To this end, we discuss the task of pedestrian trajectory prediction and video object tracking.

Pedestrian trajectory prediction is the task of predicting one or multiple future trajectories of objects with known observed history. Sophisticated prediction methods model interactions between multiple agents or the scene to generate realistic future paths. The task of multi-object tracking involves constructing trajectories of one or multiple moving objects in a video stream across different time frames.

## 1.1 Real-World Applications

Multiple real-world applications rely on tracking and predicting the positions of pedestrians. While humans usually have good intuition and the ability to anticipate pedestrians' motion and future positions, any artificial system interacting with humans needs those capabilities. These systems record raw sensor data such as RGB-video streams or lidar point clouds and process them with state-of-the-art computer vision methods. In Figure 1, we show relevant applications that require multi-object tracking and trajectory prediction methods.

**Autonomous Vehicles**
Tracking and forecasting pedestrians and other moving objects are essential components of an autonomous driving stack. The ability to anticipate the behavior of other agents is vital to route safely to the desired goal while preventing collisions, emergency breaks, or other unsafe actions. Safe navigation requires a real-time and robust prediction method. Since vehicles operate in dynamic and complex environments, incorporating scene structures such as roads, lanes, and traffic signs can enhance the accuracy of predicting the motion of traffic participants such as pedestrians and cyclists.

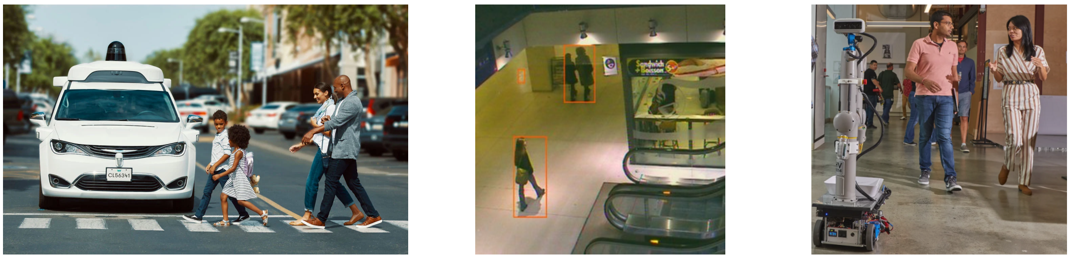
Not only fully self-driving vehicles benefit from those methods. Sophisticated trajectory prediction can support humans in safety-critical scenarios. Automatic driving assistants can inform the driver about pedestrians moving in the blind spot of the car or spontaneously crossing the road. As pedestrians represent 23% of annual 1.35 million road traffic deaths [60], most tragic events happen in crowded scenarios where the driver does not oversee all pedestrians near the vehicle. The number of accidents could be drastically reduced by introducing semi-automated driving systems.

**Visual Surveillance**
As security becomes more and more relevant, multi-object tracking systems help to track and follow targets across a network of stationary or moving cameras. Consistent tracks allow them to perform action recognition that may increase the chances of identifying criminal or suspicious behavior. The surveillance setup can vary from a single CCTV for a small convenience store to a multi-camera system in public areas. Besides security applications, the analysis of the recorded trajectories can lead to better and safer urban planning of public spaces.

**Human-Robot Interaction**
Human-robot interaction is an important topic, as we start to deploy robots in our homes and industry environments. Forecasting and anticipating the movements of human trajectories is a prerequisite for human-robot interaction to avoid collisions and dangerous situations.

# 2 Thesis Outline

In the section above, we motivate the significance of building computer-based vision systems that can complete the task of multi-object tracking and trajectory forecasting. Therefore, we present advancements in trajectory prediction and tracking. We develop different concepts and methods for motion understanding. Our contribution in this



**(a)** Autonomous Vehicles [1]

**(b)** Visual Surveillance [2]

**(c)** Human-Robot Interaction [3]

**Figure 1:** Illustration of different real-world applications that require multi-object tracking and trajectory prediction.

dissertation is threefold:

1. **Multimodal Pedestrian Trajectory Prediction (Section 1.2)**

   We present two novel methods for stochastic pedestrian trajectory prediction. These methods provide interpretable neural network architectures capable of learning a proper multimodal trajectory distribution.

2. **Benchmarking, Evaluating, and Analysing Object Trackers (Section 2.2)**

   We present our public evaluation benchmark for testing multi-object tracker - *MOTChallenge*, a symmetric evaluation metric HOTA, a dataset complexity metric MOTCOM, and an extensive analysis paper on the progress of multi-object trackers.

3. **Solving Long-Term Occlusions with Trajectory Prediction (Section 3.2)**

   We leverage state-of-the-art multimodal trajectory prediction models to improve the re-identification of tracking methods after long-term occlusions. Here, we establish a novel *tracking-by-forecasting* paradigm for single-camera multi-object tracking.

To provide sufficient background information and describe the contributions of our publications, we structure this dissertation into five chapters.

In Chapter I, we provide motivation for the relevance of the research topics presented in this thesis, and we delineate the scope of this dissertation.
In Chapter II, we provide comprehensive background information and an overview of the relevant research field of this dissertation. In detail, we discuss pedestrian trajectory prediction (Section 1.1), multi-object tracking (Section 2.1), and the effect of camera projection on recorded motion (Section 3.1). In every section, we state our contributions after the previous introductions to the topic and problem.
In Chapter III, we provide a more detailed summary of our cumulative content as a compilation of four research publications.
In Chapter IV we conclude this thesis and discuss future research directions.
The last chapter *Publications* includes the cumulative publications of this dissertation.

[1] https://s3-prod.autonews.com/s3fs-public/Waymo_SFO-MAIN_i_0_0.jpg

[2] https://upload.wikimedia.org/wikipedia/commons/thumb/2/2b/AphelionApplication21.jpg/300px-AphelionApplication21.jpg

[3] https://miro.medium.com/v2/resize:fit:1200/1*6W-eVi8rD7Hy2qN23iJQ_Q.jpeg

# II Background and Contributions

This section introduces the reader to the relevant topics addressed in this PhD dissertation and outlines the contributions to the respective fields. We provide comprehensive background information on relevant research fields that supports the reader's understanding of the relevance of our contributions presented in this thesis.

First, we introduce the task of pedestrian trajectory prediction and present state-of-the-art methods (Section 1.1). Second, we discuss the multi-object tracking task (Section 2.1). Ultimately, we provide background on camera projection and on how real-world motion of objects appears in video sequences (Section 3.1). In each section, we include a brief discourse on the particular problems we have identified and our proposed solution.

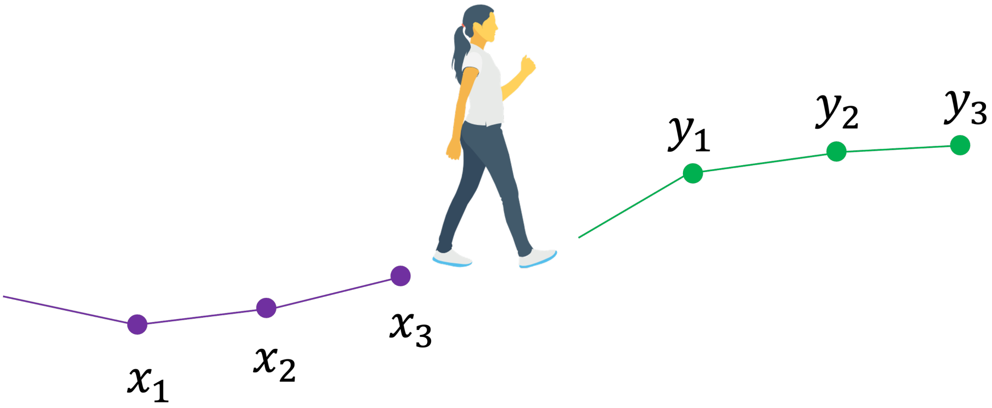


**Figure 2:** Illustration of Pedestrian Trajectory Forecasting.

# 1 Trajectory Forecasting

## 1.1 Background

Due to its relevance for multiple problems, the task of trajectory forecasting has already existed for some decades. Since its start, several approaches for multi-agent forecasting have been proposed. The approaches range from classical physics-based models to generative deep learning methods, as described by Rudenko et al. [71]. In the following, we will outline the problem formulation of trajectory forecasting and present existing forecasting methodologies.

The ability to predict human motion in different scenarios is precious for many applications. These applications range from autonomous vehicles to social robots and city planning. The scenarios which require trajectory forecasting can include very crowded scenes or restrictive scene topology that dominantly influences the movement of the pedestrians. Trajectory prediction is challenging because of the complexity of human behavior that arises from various internal and external stimuli. Human motion is affected by individual preferences, intentions, and complex interactions, most of which are not directly observable and difficult to model. Therefore, pedestrians can suddenly change their direction or speed. However, such behavior is difficult to anticipate from the previous observations and not trivial to predict.

### 1.1.1 Problem Formulation

The task of trajectory forecasting is defined as predicting the future trajectory $\hat{Y}$ of an agent *e.g.,* pedestrian, bicyclist, or vehicle for a given past observed trajectory $X$ as shown in Figure 2. A *trajectory* is defined as a sequence of the agent's position; changes in the position reflect the velocity. Here, the 2D input trajectory of a person $i$ is defined as $X_i = \left(x_i^t, y_i^t\right)$ for $t = 1, \dots, t_{obs}$ and the future trajectory $Y_i^t = \left(x_i^t, y_i^t\right)$ for $t = t_{obs+1}, \dots, t_{pred}$ For a scenario with multiple agents, the problem involves predicting the motion of all agents $\hat{\mathbf{Y}} = \{\hat{Y}_1, \dots, \hat{Y}_N\} \in \mathcal{Y}$ for a set of $N$ agent from past observations $\mathbf{X} = \{X_1, \dots, X_N\} \in \mathcal{X}$.

Hence, the forecasting can be seen as a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ mapping

$$\hat{\mathbf{Y}} = f(\mathbf{X}). \tag{1}$$

Trajectory forecasting is dominantly a sequence-to-sequence modeling task that first includes encoding the long-term temporal relations of the observed trajectory into a latent representation. Interactions are modeled on top of the latent code before multimodal trajectories are decoded and generated from the final latent representation. The observation $X$ represents the past of the agent. It can contain the trajectory coordinates and additional scene information depending on the recording platform and available sensors. The predictions of future trajectories usually range over a time window of a couple of seconds.

### 1.1.2 Trajectory Prediction Datasets

Large datasets are indispensable for training and testing deep learning algorithms. In pedestrian trajectory prediction, datasets usually provide pedestrian trajectories represented as 2D points $(x, y)$ in metric space. A standard setup for data collection includes a video camera filming a scene from a static top-down view. Objects are detected and tracked in each frame and labeled with unique IDs. These detections are then projected into the ground plane and transformed by the scene homography.

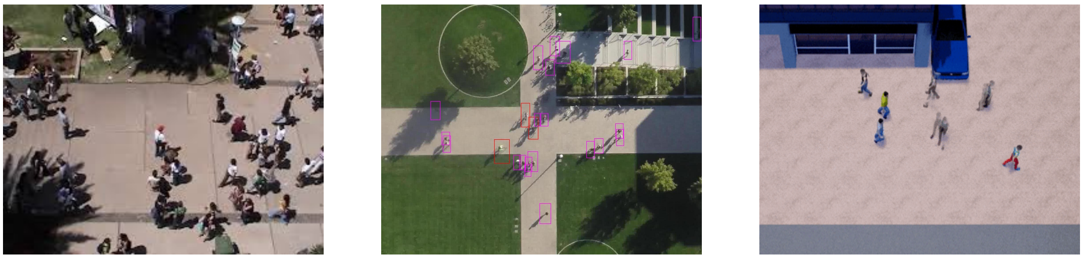
Researchers commonly work with the ETH [66] and UCY [48] datasets. These real-world datasets contain multi-human interaction scenarios captured at $2.5$ Hz $(t = 0.4s)$. There exist five datasets of four different scenes with $1536$ unique pedestrians. Both, ETH and UCY, are used jointly for training and testing. With this, it is standard to use the leave-one-out cross-validation approach. The model is trained on four sets and tested on the remaining one. This process is repeated five times to include all data combinations.

Another large-scale dataset is the Stanford Drone Dataset [70] (SDD). The dataset consists of annotated videos with more than $20,000$ targets, such as pedestrians, bikers, skateboarders, cars, buses, and golf carts. These objects navigate in eight unique scenes on the Stanford University campus. The dataset contains several spatially multimodal scenes with path crossings or roundabouts.

In contrast to the datasets mentioned above, the Garden of Forking Path [49] provides multiple future trajectories for a given set of observations. To obtain multiple possible future trajectories, the creators of the dataset rebuild some of the ETH and UCY scenarios inside a simulator. Human annotators then control selected agents to reach predefined spatial goals. Each agent has, on average, 5.9 future trajectories in 127 scenarios leading to 750 trajectories.

### 1.1.3 Trajectory Prediction Methods

Many approaches for multi-agent trajectory forecasting exist in the literature. The large trends range from physics-based models [38] to learned deterministic regressors [1] and generative probabilistic models [36, 45]. The following section will explore the different model categories, focusing on deep learning methods.



**(a)** ETH / UCY Dataset [66, 48] **(b)** Stanford Drone Dataset [70] **(c)** Garden of Forking Paths [49]

**Figure 3:** Different datasets for pedestrian trajectory prediction.

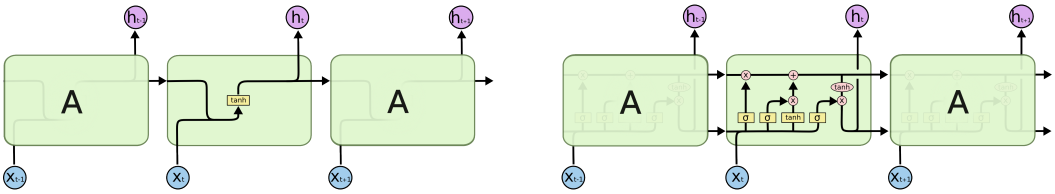
Traditional physics-based [38] models consider an explicit hand-crafted dynamic model that follows Newton's law of motion. The forces driving the dynamic include basic rules and consider the pedestrians' physical and social or psychological factors with interpretable parameters that specify the interactions' coupling. These methods try to model the microscopic behavior with physical dynamic systems.

While these methods are generally practical to simulate macroscopic phenomena in specific scenarios *e.g.,* occupation of escape routes in case of an emergency, the resulting dynamic heavily relies on the fixed parameters and the theoretical modeling framework. Hence, obtaining accurate trajectory predictions requires an ideal parameter fit which is very tricky. Once a set of parameters is fitted to a particular scene, the model shows low generalizability to other scenes. This limitation led to a shift toward data-driven methods.

Before researchers started to use deep learning networks, they applied statistical models like linear Kalman filters to propagate the current state with a dynamic model to predict the next steps. However, these simple models only consider pedestrians individually, assume linear motion, and cannot account for interactions. Consequently, these models are too inaccurate for many use cases that deal with complex dynamics.

With the rise of deep learning methods, neural networks have vastly outperformed physics-based models and have been stated as the foundation for all modern forecasting methods. The success of deep learning methods was enabled by a large amount of available data and the exponential improvement of high-end computing. Deep learning reduces the amount of feature engineering because data-driven models take minimal assumptions about the structure of the agent model. The networks learn to extract the relevant information from the data where the training objective guides the optimization. Hence, complex tasks can be solved by training neural networks on historically observed data. Deep learning architectures are nowadays the standard choice for pedestrian trajectory prediction models.

In order to deal with the temporal nature of the problem, recurrent neural networks are commonly used for the task. Sequence-to-sequence models overcome the problem of feed-forward networks. These networks have a fixed size of input and output vectors. In contrast, the recurrent architecture allows the model to process sequences with different time lengths. In the next paragraph, we will introduce the reader to recurrent neural network architectures.



**(a)** Vanilla Recurrent Neural Network (RNN) **(b)** Long Term-Short Memory (LSTM) Network

**Figure 4:** Architectures of different Recurrent Neural Networks[4].

**Recurrent Neural Networks**
A recurrent neural network (RNN) is an extension to a feed-forward layer to process sequential data. RNNs use the output of the previous timestep in addition to the data point as input for the current timestep. Thus, the architecture allows the network to store information in a memory state. An illustration of a vanilla RNN layer is shown in Figure 4a. While vanilla RNNs are effective at handling short-term information, they struggle to store information over longer time gaps due to the problem of *vanishing gradients*, as described by [39].

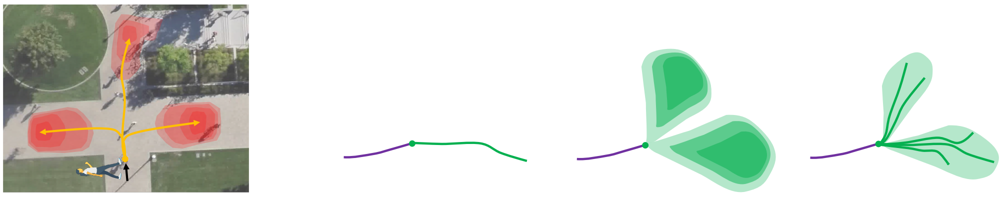
Long Short-Term Memory (LSTM) [40] networks address the problem of vanishing gradients and provide a solution to store long-term information. LSTMs successfully apply to numerous tasks such as speech recognition [14], machine translation [15], and video classification [59]. The LSTM has a gate structure to update the hidden and cell state of the network, as shown in Figure 4b. This architecture solves the vanishing gradient problem because the network's cell state (memory) propagates across different timesteps without being modified at each time step (depending on the activations of the gates).

An encoder-recurrent-decoder (ERD) architecture is the standard for sequence-to-sequence models. The RNN encoder first maps the entire sequence into a high-dimensional latent space. The latent encoding initializes the decoder then to generate the prediction. The ERD was first used for predicting the motion of human body poses [33] and is nowadays the most frequent architecture choice for prediction architectures.

### 1.1.4 Generative Trajectory Prediction Methods

An essential feature of a trajectory prediction model is the incorporation of uncertainty. Consider a pedestrian approaching a crossroad as shown in Figure 5a. The pedestrian may turn left or right or even walk straight. Therefore, we find that the distribution of future trajectories has multiple spatial modes. Multimodality means that the manifold of the whole distribution is the union of disconnected smaller manifolds, that may correspond to different directions. Trajectories of different modes are called multimodal, while trajectories within the same mode refer to the diversity of trajectories. This fact will become relevant as we discuss generative prediction methods later on. Consequently, we cannot yet clearly determine the route the pedestrian might take given the observation $X$. Ideally, we want the prediction method to express the prediction's uncertainty.

[4] https://colah.github.io/posts/2015-08-Understanding-LSTMs



**(a)** Spatial Multimodality **(b)** Single deterministic prediction **(c)** Distribution of future trajectories **(d)** Sample predictions of a GAN

**Figure 5:** Illustration of deterministic and stochastic trajectory prediction.

This fact leads to a paradigm shift from predicting the single best trajectory (Figure 5b) to producing a distribution of future trajectories (Figure 5c). Assuming there exists a *real* conditional probability distribution $p(Y|X)$, we want to estimate this distribution as well as we can. Knowing about the distribution of realistic future trajectories is more instructive than a single prediction for downstream tasks in autonomous platforms, *e.g.,* motion planning, and decision-making.

Ideally, we want to formulate an analytic form for the distribution $\hat{p}(Y|X)$. However, we can only construct a closed analytic form for parametric distributions *e.g.,* Gaussian mixture models. The downside of parametric distributions is that they are constrained to an explicit functional form and cannot learn more complex behavior. In the following, we will present how we can build a stochastic one-to-many trajectory forecasting method that is not constrained to a particular parametric distribution.

Deep generative models can learn to generate complex dynamics and interactions, replacing parametric models for trajectory prediction. There exists a large zoo of different generative models in the field. The most famous approaches are generative adversarial networks (GAN) [35], conditional variational autoencoders (CVAE) [44], normalizing flow networks [76] or diffusion models [77]. The standard generative model for trajectory prediction is an encoder-decoder model with a stochastic latent variable modeled by either a conditional variational autoencoder [74] or generative adversarial network [36].

The CVAE explicitly models multimodality with a bivariate Gaussian distribution in the latent space and maximizes the log-likelihood between the latent variables and the samples. In contrast, GANs implicitly learn to transform a known distribution into the target distribution. We can directly produce samples by drawing noise vectors from the known distribution and transforming them with the model into trajectories.

While GANs allow us to sample different trajectory forecasts of the estimated distribution, they do not provide a closed-form density function (Figure 5d). Suppose we need to estimate the probability density of the samples. In that case, we can use a kernel density estimate [62], a statistical tool fitting a probability density function to a set of generated samples.

In this dissertation, we will primarily focus on generative adversarial networks and discuss this class of models in more detail in the following section.

### Generative Adversarial Networks

A generative adversarial network [35] is a generative model capable of learning a target distribution and allows us to sample trajectories from the learned distribution. The main

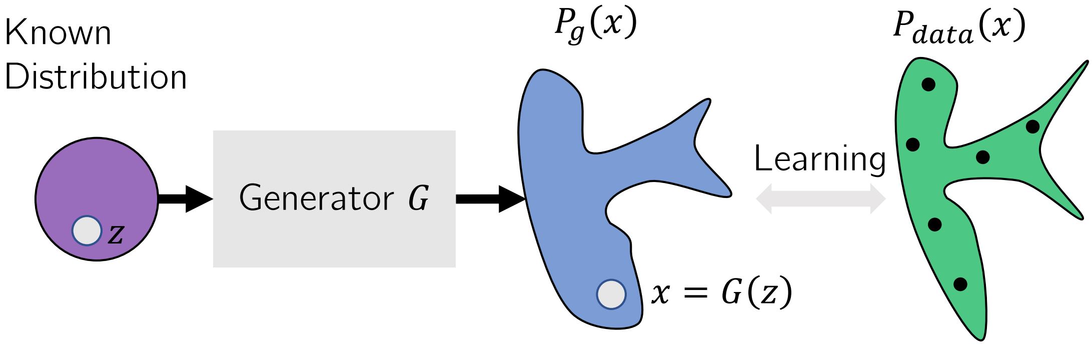


**Figure 6:** A generative adversarial network $G$ transforms a sample $z$ from a known distribution into a sample $x$ of the distribution $p_g$. During the network optimization, the model learns to estimate the data distribution $p_{\text{data}}$ with $p_g$.

idea behind GANs is the inverse transform method. In essence, the inverse transform method is a way to generate samples from a target distribution $p_{data}$ by transforming a pseudo-randomly generated variable $z$ from a known distribution $p(z)$ by a well-defined transformation function $G(\cdot)$. As shown in Figure 6, the purpose of the generator is to deform the input distribution to match the target distribution and explain the observed data samples. During the training of the model, the optimization aims to push the output distribution $p_g$ as close as possible to the data distribution $p_{data}$ such that $p_g = p_{data}$.

A GAN implicitly models the target distribution and transforms samples of a known distribution into those of the desired target distribution. Thus, the model has more capacity to learn complex behavior than parametric models because it is not constrained to a specific class of functions.

The GAN architecture, first proposed in 2014 by Goodfellow et al. [35], consists of a generator (G) and discriminator (D) (Figure 7). Inspired by game theory, the optimization principle follows a min-max objective in which $G$ and $D$ compete against each other while collectively becoming stronger. The G network attempts to fool the discriminator network, whereas the D learns to distinguish between real and fake data. In the original GAN paper, the min-max game training object reads:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{(z)}}[\log(1 - D(G(z)))]. \tag{2}$$

The generator learns to produce a distribution $p_g$ to match the real data distribution $p_{data}$. The global solution of the optimization in Equation (2) defines an equilibrium and reformulates to the Jensen-Shannon divergence. The optimal solution is in the global minimum of the objective where $p_g = p_{data}$. In the equilibrium, the discriminator D cannot distinguish between real and fake samples, and the best strategy is always to output a fixed probability $D(x) = \frac{1}{2}$.

Despite the success of GAN-based methods, training these models is challenging because of two main challenges. First, the unbalanced min-max game between the discriminator and generator can lead to divergence and instabilities of the training loss or settling in undesired local minima. Second, the vanilla GAN training results in a mode collapse, where the generator maps any random value $z$ to the same output and does not produce diverse samples.

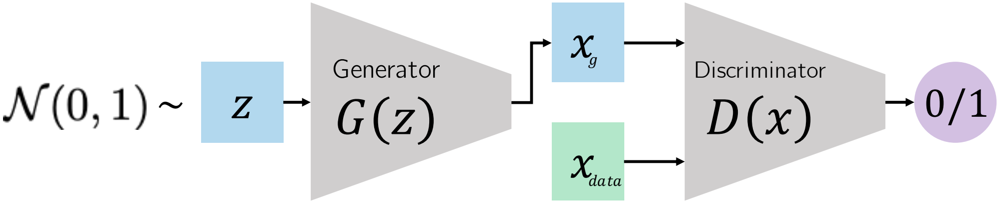


**Figure 7:** Illustration of a generative adversarial network [35]. The model consists of a generator $G$ and discriminator $D$. The generator transforms a standard Gaussian variable $z \sim \mathcal{N}(0,1)$ into a sample $x_g$. Then, the discriminator learns to distinguish between real $x_{data}$ and generated $x_g$ samples.

As the initial GAN paper discusses some of these limitations of the original formulation, multiple model variations have proposed remedies to the shortcomings mentioned above. In order to improve the learning of the generator, researchers introduce alternative training procedures [57], objectives [4, 55], or additional players to the min-max game [12, 90, 89].

### Generative Adversarial Networks for Trajectory Prediction

Social-GAN [36] is the first model that proposes a GAN architecture for trajectory prediction. The generator produces trajectory predictions, and the discriminator distinguishes between real and fake trajectories considering the dynamics and social behavior of the samples. Additionally, this model uses a *best-of-N* (BoN) $L_2$ training loss [8] to counteract mode collapse and to encourage the model to produce samples covering all modes. However, this training loss results in many unrealistic out-of-distribution samples, which we discuss in the contribution part of this thesis (Section 1.2). In Figure 8, we show trajectories generated by the generator for different noise samples for the same observation. The network learns to encode the different directions of the output distribution into the random latent space.

Social-Ways [2] leverages InfoGAN [12], an information-theoretic extension to the vanilla GAN, that introduces an additional latent code to improve multimodality. An information-based cost function replaces the standard BoN loss to encourage the model to be genuinely injective.

Along this line, Social-BiGAT [45] incorporates the idea of bicycle GAN training [90] to learn a bijective transformation between the latent space and the trajectory samples by applying cycle consistency [89].

In trajectory forecasting, researchers use GANs to learn a multimodal distribution of human motion. However, this raises a very general question (formalized by Khayatkhoet et al. [81]): *how can they [generators] fit disconnected manifolds when they are trained to transform continuously a unimodal latent distribution?*

So far, the trajectory forecasting community has bypassed this question, but it seriously suspects the use of generative adversarial models in the field. Moreover, the generator optimized during the GAN training learns a continuous transformation for samples from a unimodal distribution (usually uniform or Gaussian distribution) into the target distribution. This homeomorphic function can, therefore, not map the random samples into disconnected manifolds. Consequently, the learned distribution generated by the standard GAN models covers all modes by approximating the disconnected modes with

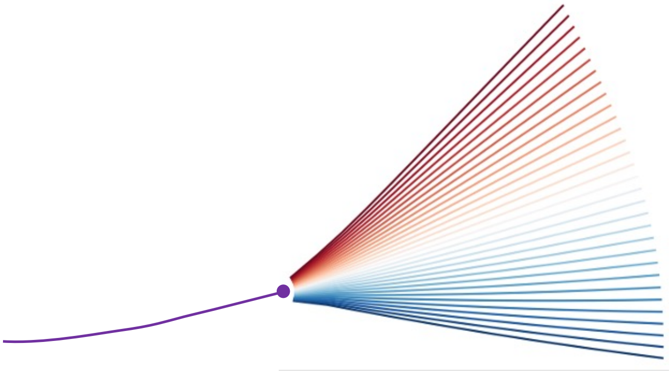


**Figure 8:** Generated trajectories of a GAN for different input noise vectors given a single past trajectory.

a single connected unimodal distribution. However, this produces out-of-distribution samples in regions with practically no support in the target distribution. Thus, these samples cannot be considered to be realistic. Strictly speaking, the previously proposed methods have theoretical limitations for modeling trajectory distributions. This thesis addresses these issues and proposes two solutions to generate multimodal predictions.

### Modeling Interactions

Humans do not walk around in empty spaces. Therefore, other agents or the environment affect their navigation and routing. Thus, trajectories are influenced by the interaction with other pedestrians or the layout and objects in the scene. Consequently, accounting for these types is essential to predicting pedestrians' future paths.

As modeling interactions is an integral part of trajectory forecasting, we will introduce the reader to relevant concepts in the following parts.

### Human-Human Interaction

In crowded scenes, humans navigate to their final position while avoiding collisions with other pedestrians. Their resulting trajectories can significantly differ from the shortest path between the start and end positions. To predict realistic trajectories in the presence of multiple agents, we must account for social interactions. Instead of treating each trajectory separately, models must predict trajectories jointly. The idea of social interactions was first formulated in the social force model [38] which interprets each pedestrian as a repelling particle that aims to reach a goal while avoiding collisions.

Nowadays, deep learning methods have replaced hand-crafted models, and modeling interactions between agents is usually done by applying social pooling to the latent representation of the agents. The latent vector representing all features of the observation of the scenario is commonly obtained by encoding the past trajectory with a recurrent neural network. The basic idea behind these approaches is to share the latent information across the agents in the scene.

In the following, we present some of the common social pooling layers presented in Table 1. The first neural network architecture that introduced the concept of social pooling was Social LSTM [1]. The model gathers the hidden states of neighboring pedestrians from a defined grid surrounding the agent of interest. While social LSTM relies on a deterministic

**Table 1:** Social pooling mechanism proposed in different models for pedestrian trajectory prediction.

| Model | Social Pooling |
|---|---|
| Social LSTM [1] | Grid Pooling |
| Social GAN [36] | Feature max-pooling |
| Social Ways [2] | Social Soft Attention |
| Social-BiGAT [45] | Graph Attention Network (GAT) |

ERD, Social-GAN [36] uses a generative adversarial network as a predictor and replaces the social grid pooling with a global feature-wise max-pooling to compute the social interactions. The social module in Social Ways [2] advances the max-pooling with a soft attention [83] network. In contrast to max-pooling, the attention layer allows gradient backpropagation through social pooling during training. To account for even higher-order interactions and feedback responses in the interaction, Social-BiGAT [45] leverages a graph attention network as the social pooling layer for modeling interaction.

**Human-Scene Interaction**

Human trajectories are not only influenced by social interactions but also by the environment and topology of the scene. Pedestrians adjust their motion to walk around spatial obstacles such as trees, benches, and parked cars. Their trajectory is also determined by social norms and physical constraints.

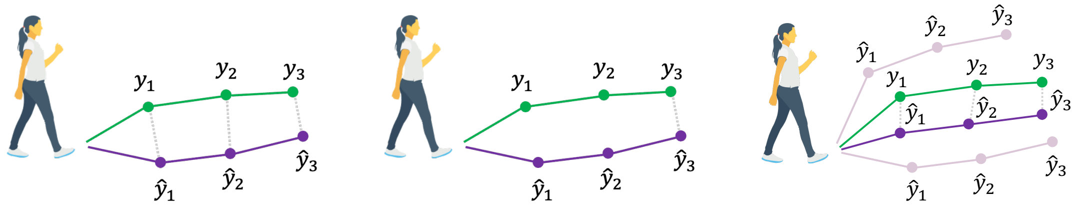
A common approach is to use the scene's geometry and semantics to reason about the long-term goal of the agent. Semantic aware models usually extract visual features from a top-down scene image with a convolutional neural network (CNN) to extract visual features. These visual features are combined with the trajectory encodings of the pedestrian. Hence, models can learn the relation between the outline of the scene and its effect on the resulting trajectory.

Various approaches exist to incorporate visual features into the prediction model. SoPhie [73] fuses the visual features in an attention module to assign soft attention weights to different spatial regions of interest in the scene image. These weights are then passed to the decoder to generate the forecasting trajectories. Instead of soft attention, Social-BiGAT [45] replaces the attention layer with a graph neural network.

In summary, forecasting models incorporate social and scene interactions using different layer architectures. While agent-agent interactions rely on the exchange of individual features across the set of pedestrians in a scene, scene-aware models use visual features from a CNN to combine them with the position and motion of an agent for trajectory prediction.

### 1.1.5 Evaluation Metrics

Evaluation metrics are crucial to measuring the performance and progress of any machine learning method. There are several popular metrics for evaluating pedestrian trajectory



**(a)** Average displacement error **(b)** Final displacement error **(c)** Best-of-N

**Figure 9:** Illustration of ADE, FDE, and BoN between the ground-truth trajectory $\mathbf{y}$ (**green**) and prediction $\hat{\mathbf{y}}$ (**purple**). While the FDE only evaluates the final position of the prediction, the ADE averages the distance between all positions of the prediction and the ground-truth trajectory. The BoN only considers the sample out of $N$ predictions with the lowest ADE or FDE.

prediction models. Metrics compare the ground-truth sample with the predictions for a given observation. In the literature, many researchers report the average displacement error (ADE) and final displacement error (FDE) as illustrated in Figures 9a and 9b.

- **Average Displacement Error (ADE):** Average $L2$ distance between the ground-truth trajectory and the prediction.
- **Final Displacement Error (FDE):** The $L2$ distance between the final point of the ground-truth trajectory and the prediction.

These metrics are unimodal and only consider one forecast per input sample. Instead of predicting a single trajectory, stochastic methods (*i.e.,* GANs or VAEs) can output multiple predictions to account for the multimodality of human trajectories as they learn a distribution of possible future trajectories. However, these methods do not provide the distribution in a closed form and can only be evaluated by their empirical distribution. To evaluate stochastic methods, we sample $N$ output trajectory forecasts $\{\hat{y}_1, \dots, \hat{y}_N\}$. We evaluate these trajectories under the *best-of-N* (BoN) protocol. The BoN where $N = 3$ is illustrated in Figure 9c. Therefore, most benchmarks allow for multiple predictions per input sample and report the following metrics:

- **BoN ADE:** Computes ADE of $N$ samples and reports the minimum ADE loss.
- **BoN FDE:** Computes FDE of $N$ samples and reports the minimum FDE loss.

In most evaluation protocols, researchers report $N = 20$. However, we argue that we must evaluate the model with fewer samples because most real-world applications cannot usefully process $N = 20$ trajectory predictions. Fixing the evaluation incentive is relevant because it significantly affects the focus of the research effort. Consequently, many methods have focused on producing highly diverse trajectories, reducing the BoN error. Accordingly, [75] shows that one can construct a simple linear model that produces $N$ trajectories evenly fanning a cone in the direction of motion, yielding very competitive results on the test benchmarks. This method caricatures current state-of-the-art methods with an oversimplified model seriously questioning the merit of BoN metrics.

Besides $L2$-based methods, researchers occasionally evaluate the ground-truth trajectory's negative log-likelihood (NLL) *wrt.* the predicted sample distribution. For each timestep, a kernel density estimate [62] (KDE) of the sampled trajectories is computed to obtain a probability density function of the predictions. Then, the estimated KDE is used to compute the mean NLL of the ground-truth distribution.

As mentioned above, the downside of BoN metrics is that they primarily focus on the recall of the ground-truth trajectory, *i.e.,* creating highly diverse trajectories. The large spatial variance of the predicted samples increases the chance of generating one trajectory close to the ground-truth trajectory. That means only the trajectory with the lowest error counts for this metric. As a result, very different prediction outputs can lead to the same BoN performance as long as they all share one trajectory being the closest to the ground-truth trajectory and having the same distance error. All other trajectories do not affect the loss value.

While rewarding a high recall of the ground-truth trajectory, the overall performance of all predictions is not evaluated. Primarily for real-world applications like autonomous driving, this behavior is inappropriate because the system does not have knowledge of the future and cannot select the trajectory closest to the ground truth. Therefore, it considers all trajectories as possible future paths and needs to adjust its actions accordingly. Unrealistic trajectories, however, confuse this process and may lead to disturbance or dangerous situations. To overcome these shortcomings, we will propose an alternative new recall-precision metric that measures the *realism* of all samples and not only the best trajectory.

## 1.2 Multimodal Pedestrian Trajectory Prediction

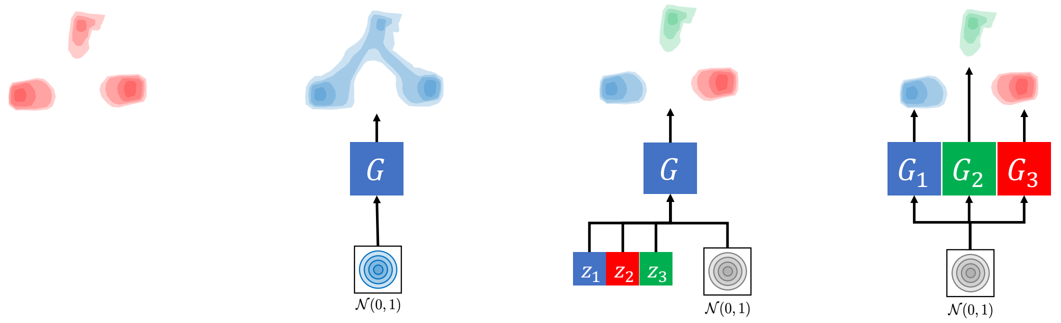
Modeling stochastic trajectory forecasting methods is critical for multiple downstream tasks where trajectory prediction is needed for planning and decision-making. With this, modeling prediction uncertainty is an essential feature. While first generative methods transform a known, connected distribution into the desired target distribution, they do not consider the true multimodal nature of pedestrian trajectories.

**Table 2:** Our publications on *multimodal trajectory prediction.*

**Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation**
**Patrick Dendorfer**, Aljoša Ošep, and Laura Leal-Taixé
*Asian Conference on Computer Vision (ACCV).* 2020.

**MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction**
**Patrick Dendorfer***, Sven Elflein*, and Laura Leal-Taixé
*International Conference on Computer Vision (ICCV).* 2021.
(* denotes equal contribution)

In the previous part, we introduced generative models such as GANs as the solution at hand when generating a distribution of future trajectories and modeling the uncertainty of trajectory forecasts. State-of-the-art GAN-based trajectory prediction models [36, 45, 73] significantly improve the test error performance over deterministic models under



**(a)** Multimodal Target Distribution

**(b)** Vanilla GAN

**(c)** GAN with discrete variables

**(d)** Multi-Generator GAN

**Figure 10:** Illustration of the multimodal nature of trajectory prediction and the ability of different model architectures to learn a multimodal target distribution.

the *best-of-many* evaluation protocol. However, are these models eventually capable of learning the underlying multimodal nature of the distribution of pedestrian trajectories?

A closer look at the generated samples and distribution reveals that these models also tend to predict undesired out-of-distribution (OOD) samples. This phenomenon is not broadly discussed in recent papers because many state-of-the-art methods still evaluate their methods with the BoN protocol that primarily focuses on recall and does not punish unrealistic predictions. Based on real-world downstream tasks, precision is as important as recall because applications can only consider a limited number of predicted paths and cannot select the best predictions a priori.

There are several explanations why this undesired behavior occurs: Certainly, the BoN loss encourages the model to increase the diversity of the output for the sake of realism *wrt.* the data distribution. Another relevant reason is that single-generator models are theoretically incapable of learning a transformation that produces a multimodal distribution.

We already presented the concept of multimodality in the previous section. Limitations of single generators arise as they are modeled as neural networks transforming a known distribution. By construction, neural networks are continuous functions because they need to be differentiable for training their weights. The random noise vectors usually come from a multivariate unimodal Gaussian distribution on a connected topology. The generators transform the connected manifold of random variables with a continuous transformation. The resulting distribution preserves the topology of that manifold and hence is also connected. Therefore, a connected manifold cannot be transformed by continuous functions into a disconnected one. Consequently, the learned distribution smears over all modes and still has non-zero probability mass at regions that are outside of the target distribution (Figure 10b). Practically, there are only two ways to build a sophisticated model capable of generating a multimodal distribution, as shown in Figure 10a.

Following [81], we can change the topology of the latent space by either (i) introducing discrete latent variables (Figure 10c) or (ii) constructing a discontinuous transformation consisting of multiple smaller generator networks (Figure 10d).

### Contribution

We explore the two possibilities to achieve a multimodal distribution. We discretize the generator's input (Figure 10c) or apply a discontinuous multi-generator model (Figure 10d) to build a multimodal trajectory forecasting model.

As the first concept, we present Goal-GAN [21]. In this model, we decompose the trajectory prediction into a two-stage process: (i) we estimate a goal position from an estimated goal probability distribution and (ii) route the trajectory toward that goal. The model estimates an interpretable probability distribution of the goal positions of the trajectory. The categorical distribution allows a discrete sampling of goal positions introducing a discontinuous process into the prediction models. The goal estimates influence the decoding of the trajectory and fix the last point of the predictions.

Secondly, we propose MG-GAN [19], a multi-generator model for trajectory prediction. In contrast to Goal-GAN, we use the idea of a discontinuous function to ensure a multimodal target distribution. The key here is that we have a set of different generators with their individual model parameters. Each of the generators produces a trajectory distribution and represents one of the modes of the multimodal distribution. Furthermore, we introduce the path mode network that outputs a discrete probability distribution over the different generators conditioned on the observed trajectory and the scene image. The model uses these probabilities to sample randomly or select the generators. Therefore, we can efficiently cover all relevant modes of the distribution with as few samples as necessary. Hence, there is a considerable advantage compared to vanilla GAN methods, where the data distribution is only learned implicitly and can hardly be controlled to cover specific modes.

The problem of out-of-distribution samples is not adequately addressed in existing works because the standard metrics used in the field do not assess the quality of all samples except the one closest to the ground truth. To address this issue, we additionally propose a precision-recall metric for trajectory prediction, which is standard in studying GANs in other research fields. While we can produce a distribution of generated trajectories, we only have one observation of the ground-truth trajectory. To overcome this shortcoming, we either synthetically simulate multiple possible future trajectories for the same observation or try to cluster real ground-truth trajectories based on dynamic similarities of the observation.

### Conclusion

This part presents the relevance and challenges of predicting multimodal trajectories and our contribution to solving this task. Once more, we want to emphasize the importance of multimodal and high-precision predictions for many real-world applications. We improve stochastic trajectory prediction methods by studying the multimodal nature of trajectory distributions. Based on that knowledge, we choose accurate model architectures which reflect these properties. Therefore, we derive our two proposed solutions from the first principles and hope that our work encourages the community to think about multimodality more carefully and consider other metrics, which include the idea of precision for their methods.

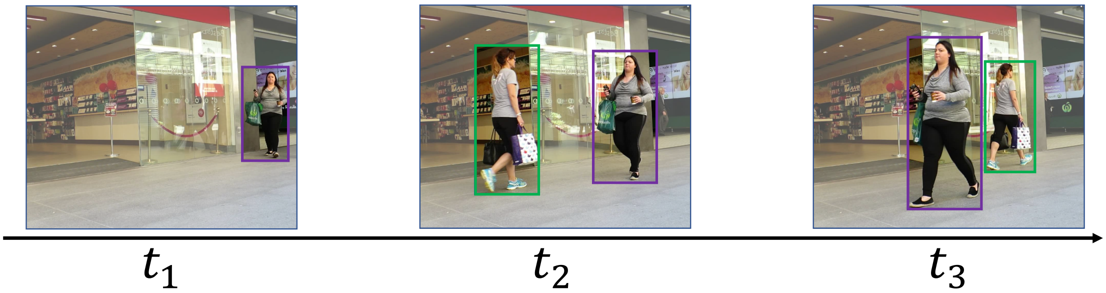


**Figure 11:** Visualization of a pedestrian tracking scenario. The multi-object tracking task involves detecting and localizing objects of interest and associating them across different timesteps.

# 2 Multi-Object Tracking

## 2.1 Background

Multi-object tracking (MOT) is essential in studying human motion and defines one of the core research fields of computer vision. In this section, we will introduce the reader to the task of multi-object tracking. Along this line, we will present different tracking paradigms and evaluation protocols, and outline open challenges and problems.

**Overview**
Tracking moving objects over space and time is fundamental for reasoning and acting in a dynamic and visual world. Whether driving on a highway, walking in the streets or playing team sports, one often maintains attention on multiple moving objects simultaneously. To explore this ability with a computer system, researchers have established the task of multi-object tracking as a core field of computer vision which already dates back to the late 80s [67].

Today, multi-object tracking provides the technology for several applications in various fields considering a wast spectrum of objects. Objects of interest include pedestrians, vehicles, sports players, animals, organisms, or other moving objects like footballs or airplanes. Moreover, following [53], more than 70% of current MOT research is towards pedestrian tracking, whereas vehicle tracking has become increasingly important.

### 2.1.1 Problem Formulation

The task of multi-object tracking (MOT) involves detecting and localizing all objects from a known set of classes as bounding boxes in each frame of a video sequence and assigning unique IDs to each entity. These targets may enter and leave the scene at any time. During the tracking period, the unique objects are assigned individual IDs that must persist even after long-time occlusions and under appearance changes. The problem of multi-object tracking can be split into object detection and detection association. A tracking example is shown in Figure 11.

The concept of MOT also applies to segmentation (MOTS), for which each object is not represented by a bounding box but by an instance segmentation mask. The target objects can be from completely different classes. Datasets exist that target pedestrians

(*MOT16* [58], *MOT17* [58], and *MOT20* [26]), biological cells (CTMC [3]), human heads, and zebrafish (3DZeF [63]).

**Tracking Modes**
Given a video stream, there are two different ways of tracking objects, namely *offline* and *online tracking.* In online tracking, the tracker obtains the video frames sequentially and outputs a prediction for each frame without information about the future. Online tracking is relevant for any autonomous platform that navigates through a scene and interacts with objects in real-time. In contrast, offline tracking can be seen as the postprocessing of a sequence where the methods have access to the entire sequence information. The tracker can optimize its predictions by using the past and future of a track. While unsuitable for interacting systems, offline tracking can help analyze recorded scenarios, *e.g.,* creating motion profiles of pedestrians and groups in public areas.

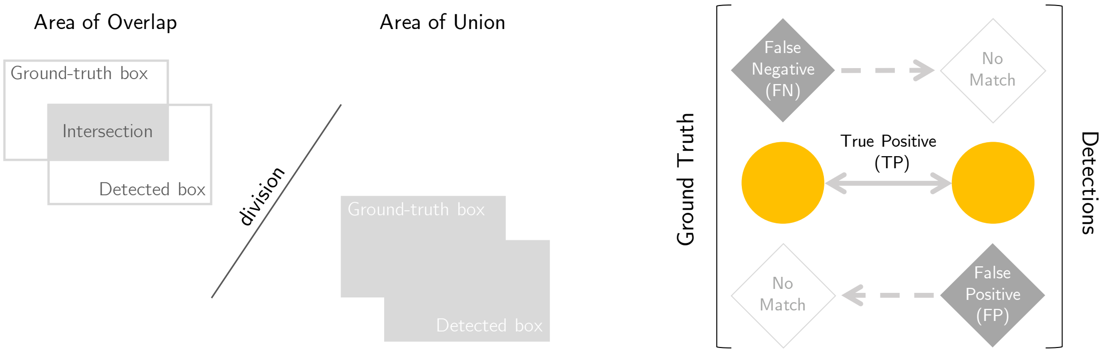
### 2.1.2 Multi-Object Tracking Datasets

Data is indispensable for training and testing deep learning methods. The tracking community has shown effort to create standardized datasets and benchmarks. These datasets provide raw video sequences and annotations files for each frame containing the location of the object bounding boxes and target IDs.

This thesis will present our benchmark for multi-object tracking, *MOTChallenge*, which provides tracking datasets for various objects. The most relevant leaderboards in pedestrian multi-object tracking are the *MOT15*, *MOT16*, *MOT17*, *MOT20*, and *MOTSynth* datasets. Additionally, there are other publicly available datasets for human tracking such as DanceTrack [79] and open world tracking [18].

Autonomous driving datasets such as nuscenes [10], KITTI [34], Waymo [78], Argoverse [11] and BDD100K [86] provide large-scale video data recorded from a driving vehicle in various traffic scenarios, day times, and weather conditions. While these autonomous driving datasets are very helpful in training trackers on visible objects and have more and longer video sequences than *MOTChallenge*, we find that less than 0.6% of tracks in BDD100K and 4% of tracks in KITTI contain occlusion gaps longer than 2s. In contrast, in *MOTChallenge*, 19.4% of tracks contain long (over 2s) occlusion gaps. The low number of annotated tracks containing long occlusions in the driving datasets is likely due to the semi-automated annotation process. Reappearing objects in these large-scale datasets are often assigned new identities. This limitation makes these datasets less insightful to work on solving long-term occlusions, which is still a challenging problem.

Therefore, precise and temporal consistent annotation is necessary for model training and testing, but highly accurate data is very costly and annotations are time-consuming. The successive improvement in computer graphics allows us to generate synthetic data mimicking real data to train tracking methods. Training the trackers with a large amount of synthetic data actually achieves very competitive results compared to real data training [32]. The benefit of synthetic data is that we can automatically generate the raw data together with the annotation. At this end, we include *MOTSynth* containing 1.3M frames and 33M person instances into *MOTChallenge* as we believe that synthetic data is a promising approach to comply with the high data demand of state-of-the-art methods.



**(a)** Intersection over Union (IoU)

**(b)** Illustration of detection association and matching

**Figure 12:** Concepts of measuring track and bounding box associations.

### 2.1.3 Evaluation Metrics

Tracking evaluation metrics define a similarity between the tracker output and the corresponding ground-truth track. As measuring the quality of the tracking output is not well-defined and ambiguous, many different ways of scoring the similarity between predictions and targets exist. The choice of the metric is significant as these metrics assess different behavior and errors. The agreement in the community on particular metrics to focus on also heavily dictates the research direction of future methods because it becomes increasingly important to rank top on public benchmarks. All tracking metrics measure detection and association performance but emphasize the importance of these performances differently.

In the following, we will briefly introduce the reader to Hungarian matching [46], CLEAR MOT [6], and IDF1 [69]. For most metrics, the first step usually involves matching detected bounding boxes with ground-truth boxes. The matching is usually done with a Hungarian algorithm that solves the linear assignment between the predicted objects and the ground-truth targets. The algorithm computes a mapping that minimizes the global matching score $\mathcal{S}$ of the bounding boxes. The similarity $\mathcal{S}$ is commonly the intersection-over-union (IoU) between the two bounding boxes where we only consider matches $\mathcal{S} \leq \alpha$ for the matching. The IoU of two bounding boxes $A$ and $B$ is defined as

$$\mathrm{IoU}\,(A, B) = \frac{A \cap B}{A \cup B} \tag{3}$$

and displayed in Figure 12a. A standard procedure in MOT evaluation is bijective matching (one-to-one) between the ground-truth object and the detection. In Figure 12b we illustrate different matching associations:

- True Positives (TP) are correct matches between ground-truth and detection pairs.
- False Negatives (FN) are ground-truth objects that are not matched (missed).
- False Positives (FP) are detections that are not matched to ground-truth objects.

TPs are correct matches, while FPs and FNs are two types of incorrect predictions. In 2006, the CLEAR (Classification of Events, Activities, and Relationships) Workshop collected existing metrics and unified them into the CLEAR MOT [6] framework. The central metric in CLEAR MOT is MOTA (Multi-Object Tracking Accuracy) which considers the sum of detection errors, *i.e.,* FNs and FPs, as well as identity switches (IDSW) divided by the total number of ground-truth objects (gtDet).

$$\text{MOTA} = 1 - \frac{|\text{FN}| + |\text{FP}| + |\text{IDSW}|}{|\text{gtDet}|} \tag{4}$$

The metric has a maximum value of 1 but is unbounded for negative values. Identity Switches (IDSW) define the association errors in MOTA. An IDSW occurs when a tracker wrongfully associates a new detection with a different identity to an existing ground-truth track.

In contrast to MOTA which matches on a detection level, IDF1 [69] calculates a matching between a set of ground-truth (gtTrajs) and predicted tracks (prTrajs). Identity true positives (IDTP) are matching bounding boxes, *i.e.,* TP, of the overlapping part of the trajectories. Identity false negatives (IDFN) are non-overlapping sections of matched trajectories, and identity false positives (IDFP) are unmatched detections of the remaining part of the prTraj. Hence, we can define the following scoring functions:

$$\text{ID-Recall} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFN}|} \tag{5}$$

$$\text{ID-Precision} = \frac{|\text{IDTP}|}{|\text{IDTP}| + |\text{IDFP}|} \tag{6}$$

$$\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5\,|\text{IDFN}| + 0.5\,|\text{IDFP}|} \tag{7}$$

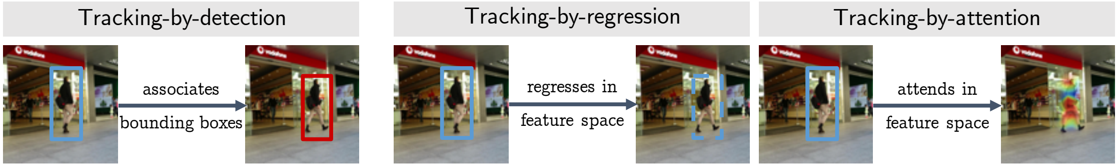
CLEAR MOT and IDF1 metrics have served the MOT community over the past years. However, there exist multiple drawbacks to these metrics that restrict tracking research. The main shortcomings of MOTA include an imbalance between detection and association performance, missing association precision (ID transfers), and unbounded values. IDF1 has a strong bias towards association but ignores the quality of associations outside of matched sections. In Section 2.2, we resume the discussion about evaluation metrics and present a new metric compensating for some of these shortcomings.

### 2.1.4 Multi-Object Tracking Methods

Before discussing different MOT methods, we want to motivate the object cues, which enable a tracking system to preserve the same unique object identity across different frames. If we try to track pedestrians as shown in Figure 11, we can either use the person's appearance or motion to associate the detections across different time steps.

**Tracking Cues**

At first, we discuss the *Appearance Model.* The appearance model consists of two components: visual representation and statistical measuring. Visual representation describes an object's visual characteristics based on features extracted from the video sequence. Here, the representation can consist of a single cue or a combination of multiple cues. The space of features used for multi-object tracking is enormous, but standard



**(a)** Tracking-by-detection **(b)** Tracking-by-regression **(c)** Tracking-by-attention

**Figure 13:** Illustration of different multi-object tracking paradigms.

features include visual encodings from convolutional neural networks [13, 47], optical flow [72], motion cues [7], or depth estimates [42]. Based on the set of different features, statistical measuring provides a function or measure $C(\cdot)$ that compares and rates the similarity $S_{ij}$ of two features $f_i$ and $f_j$.

$$S_{ij} = C\left(f_i, f_j\right) \tag{8}$$

In addition, the frame-by-frame association can be supported by motion and location cues of existing trajectory tracks and detections. The simplest motion model assumes constant velocity for the object. The velocity is computed from the position displacements in previous frames. Objects propagating with constant velocity move as

$$\left(x, y\right)_t = \left(x, y\right)_{t-1} + (u, v)_{t-1} \cdot \Delta t \tag{9}$$

where $(x, y)$ and $(u, v)$ are the 2D image pixel positions and displacements, respectively. To include uncertainty of the localization and motion of objects, researchers use stochastic processes *e.g.,* Kalman Filter [7]. The Kalman Filter provides an optimal estimator of a linear system with Gaussian error. Most motion models operate in pixel space. While these models are helpful for short-term predictions, long-term predictions in pixel space are not very accurate due to the camera projection effects discussed in Section 3.1.

### Dominant Tracking Paradigms

The field of multi-object tracking has rapidly evolved in the past, and modern state-of-the-art methods can be classified into three different types of models, which are demonstrated in Figure 13, namely tracking-by-detection, tracking-by-regression, and tracking-by-attention. These paradigms differ in how they leverage object information and cues for tracking.

In the last years, the *tracking-by-detection* [85, 84, 87, 82] paradigm has made considerable progress in the field of multi-object tracking. This approach consists of two independent steps, including (i) localizing and detecting objects in each frame and (ii) associating these detections across frames. The first step is performed by state-of-the-art object detectors [68, 50, 16]. In the association step, the model forms trajectories for each identity utilizing motion, location, and appearance cues. The association can either be solved frame-by-frame or track-wise optimized over the entire sequence in an offline application. Trackers assume that object displacements are small for short-term preservation, given two nearby frames. This assumption allows them to utilize spatial proximity for matching by exploiting simple motion models.

The *tracking-by-regression* [5, 88] paradigm assumes that objects' displacements between two temporarily close frames are small. Under this assumption, these methods leverage

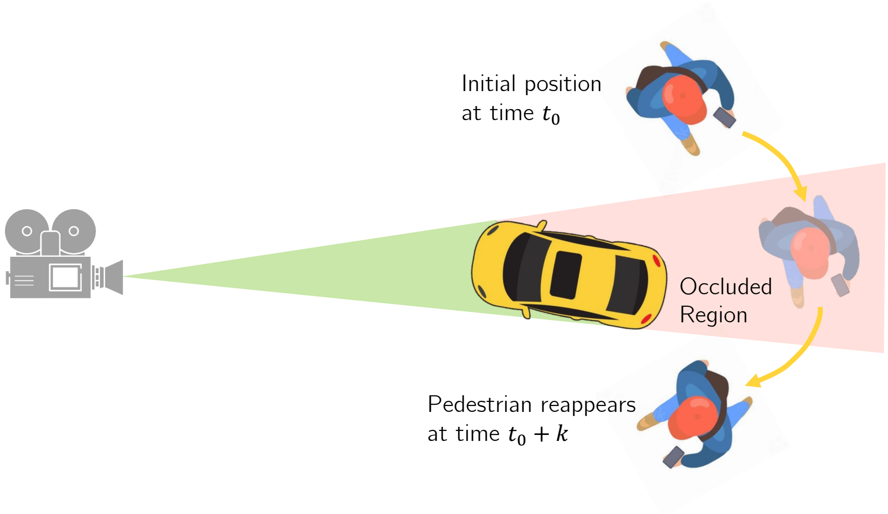

Initial position at time $t_0$
Occluded Region
Pedestrian reappears at time $t_0 + k$

**Figure 14:** Illustration of an occlusion scenario in a tracking sequence. The target pedestrian is occluded to the camera by a static occluder *i.e.,* car for $k$ timesteps. Therefore, the object is not detected by the object detector. The longer the occlusion time, the further the object may have traveled, and the harder the re-identification of the track after the occlusion.

the bounding box regression head of their object detectors to regress the bounding box positions of a tracked object in the next frame. These methods only require the first detection to initialize the track. All other bounding boxes of a particular track are then directly regressed from the model. While this setup can already solve most simple cases, advanced models are equipped with an additional re-identification Siamese network [31] and simple motion models, significantly improving performance and resolving short-term occlusions.

More recently, methods advanced to use transformer networks [83] for tracking and established the *tracking-by-attention* paradigm [56]. This end-to-end MOT approach is based on encoder-decoder transformer architecture and omits additional graph optimization, matching, or motion modeling with identity-preserving track queries. These approaches reach outstanding performance but require large amounts of training data.

### Challenges in Multi-Object Tracking

Nowadays, object detectors have become very robust, and tracking visible objects can be seen as a *solved* problem [5]. However, there are remaining challenges in multi-object tracking. One of the main challenges states long-term occlusions where objects are not visible for $> 2s$. An occlusion scenario is illustrated in Figure 14. These occlusions occur in crowded scenes with many objects (object-object occlusion) or when the target is hidden behind objects like a tree or building (object-scene occlusion). With increasing occlusion time, the positional uncertainty increases, and the appearance and size of the lost objects change drastically, making the target re-identification harder.

Methods use appearance-based re-identification networks [5, 9, 43, 47, 72] to achieve better performance in long-term association scenarios. These methods encode appearance cues or simple motion models in pixel space to re-identify persons after occlusions. Building appearance-based models require a large amount of training data for the model to learn a rich representation of the objects. Also, appearance-based re-identification becomes computationally expensive as the number of possible matching combinations drastically increases over time. Thus, an unconstrained search space for re-identification in crowded scenes after long occlusions becomes intractable. Conversely, motion models can help to decrease the re-identification search space. However, simple motion models in pixel space are not accurate enough because the localization error caused by the effect of non-linear camera projection (see Section 3.1) is significant for long-term occlusions.

To overcome these challenges, we analyze these scenarios and propose a tracking-by-forecasting paradigm to solve long-term occlusion as one of our contributions in Section 3.2.

**Table 3:** Our publications on *benchmarking, evaluating, and analyzing multi-object trackers.* Publications highlighted in gray do not count toward the list of publications of this cumulative thesis but are mentioned in Section 2.2.

**CVPR19 Tracking and Detection Challenge: How crowded can it get?**
**Patrick Dendorfer**, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé
*arXiv*. 2019.

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

**MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking**
Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé
*International Journal of Computer Vision (IJCV)*. 2020.

**HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking**
Jonathon Luiten, Aljoša Ošep, **Patrick Dendorfer**, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe
*International Journal of Computer Vision (IJCV)*. 2020.

**MOTCOM: The Multi-Object Tracking Dataset Complexity Metric**
Malte Pedersen, Joakim Bruslund Haurum, **Patrick Dendorfer**, and Thomas B. Moeslund
*European Conference on Computer Vision (ECCV)*. 2022.

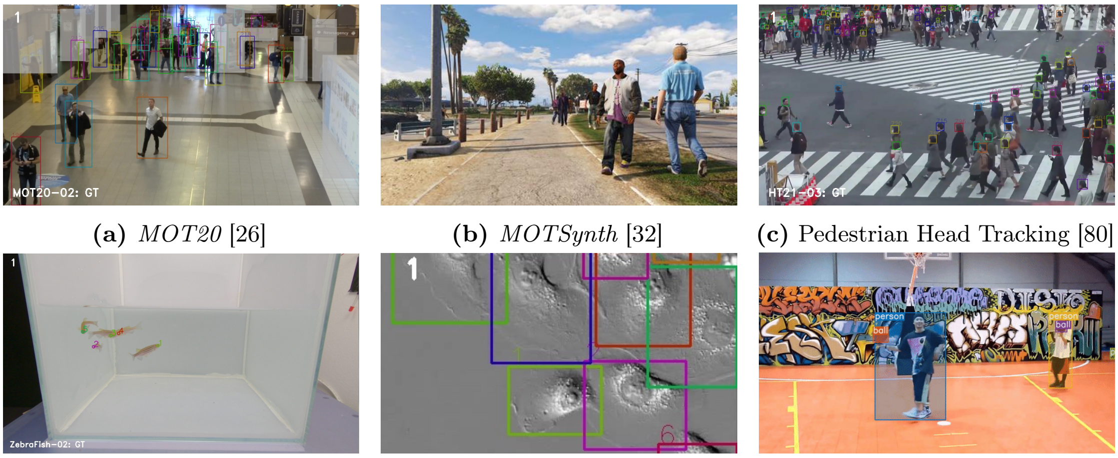
## 2.2 Benchmarking, Evaluating, and Analysing Multi-Object Trackers

Advancing deep learning multi-object trackers requires not only new model architectures but also a performant and efficient data and evaluation infrastructure. Thus, the progress goes hand-in-hand with the quality of available datasets and the ability to measure performance accurately.

We contribute to the field of multi-object tracking by providing a platform for diverse datasets, a centralized evaluation benchmark, novel tracking evaluation metrics, and a measure to characterize the complexity of data sequences. The number of newly published tracking methods has skyrocketed in the last few years. However, the number of new datasets, metrics, and evaluation benchmarks only increases incrementally.

### Contribution

The need for platform and infrastructure contributions is undisputed, but only a few researchers tackle these challenges. In the following, we present our contributions to the aforementioned challenges.



**(a)** *MOT20* [26] **(b)** *MOTSynth* [32] **(c)** Pedestrian Head Tracking [80]

**(d)** Zebra Fish (3D-ZeF20) [63] **(e)** Cell Tracking [3] **(f)** TAO [18]

**Figure 15:** Visualization of different tracking challenges hosted on *MOTChallenge*.

### MOTChallenge

The *MOTChallenge* benchmark was released in 2014, providing publically available datasets and a centralized infrastructure to evaluate and benchmark multi-object trackers. Since then[5], more than 3500 users have registered who evaluated 9000 trackers. At this date, *MOTChallenge* provides 13 open tracking benchmarks with varying target objects and tasks as shown in (Figure 15). These challenges include tracking pedestrians, human heads, zebrafish, cells, or other moving objects.

In particular, we recorded, annotated, and realized two challenges in 2019 [25] and 2020 [26], providing challenging sequences under crowded conditions. *MOT20* has scenarios with more than 150 pedestrians at the same time aiming to evaluate methods on their ability to deal with a large number of objects.

[5] Date: 25.10.2022

In addition to the training datasets, users can download the raw test videos and produce predictions for these sequences. We provide a centralized evaluation server where tracking outputs can be uploaded and evaluate their tracking performance on unknown test data. The results are displayed on publically accessible leaderboards, and we automatically generate video sequences visualizing the tracking results.

By providing the *MOTChallenge* benchmark to the community, we collect much data from different tracking methods that allow us to analyze the current model trends, performances, and failure cases of state-of-the-art tracking methods. Furthermore, we have detailed access to historical data on how methods' performances improved over time for particular scenarios and sequences. Therefore, we present an extensive analysis of 205 trackers on the *MOT15*, *MOT16*, and *MOT17* datasets in [22].

**HOTA**

Expressive metrics that measure performance on a particular task are essential for building deep learning methods. The research community agrees that precision, recall, and average precision clearly define the model performance for detection tasks. However, evaluating the quality of multi-object tracking is ambiguous and not trivial. For this reason, there exists more than one metric.

Generally, one can see multi-object tracking consisting of two subtasks: detection and association. Different applications focus on either one or the other tasks and prioritize different aspects of tracking. Hence, multiple metrics exist that emphasize different aspects of the task.

The most prominent evaluation metric was the Multi-Object Tracking Accuracy (MOTA) [6] and Identification F1 (IDF1) [69]. However, MOTA primarily focuses on detection and completely ignores identity transfers. On the other hand, IDF1 has a strong bias toward association. The problem with these two metrics is that they either focus on measuring detection or association. As a means of creating a metric that better balances the subtasks of detection and association, we propose the Higher Order Tracking Accuracy (HOTA) [52]. We extensively discuss the shortcomings of the metrics mentioned above and demonstrate the advantages of HOTA. We also present a user experiment showing that HOTA agrees more closely with human assessment of the quality of a tracking output than MOTA. Since the publication of this paper in 2020, we see that the community accepts HOTA as a new metric for multi-object tracking and reports it frequently in new publications. The metric is now also implemented in evaluation platforms such as *MOTChallenge* [19] and *KITTI* [34].

**MOTCOM**

In contrast to evaluation metrics for trackers, no comprehensive metrics exist for describing the complexity of multi-object tracking sequences. Until now, sequences are described by the number of tracks and object density. However, these numbers are not very informative when it comes to describing the complexity of the tracking problem a priori. The absence of complexity metrics decreases the explainability and comparison of tracking results across different datasets. To overcome this limitation, we present a novel MOT dataset complexity metric [65] (MOTCOM). In this metric, we investigate measurable properties of tracking sequences that explain the complexity of the tracking sequence. To compare the complexity of different sequences, we use the average HOTA scores as a proxy for the ground-truth sequence complexity. We identify three critical properties of tracking sequences that strongly correlate with complexity: object occlusions, erratic motion, and visual similarity between different objects. We develop a specific sub-metric to measure each of the aspects mentioned above. Finally, we combine these metrics to compute the final MOTCOM score. An experimental evaluation of MOTCOM on the comprehensive MOT17, MOT20, and MOTSynth datasets shows a significant correlation between our proposed MOTCOM metrics and the proxy complexity of the sequences.

We hope our metric can provide further insights into understanding and interpreting tracking performance results, curating and collecting new datasets, and developing specialized tracking methods.

**Conclusion**

Our presented projects provide the necessary infrastructure, insight, and tools for developing new tracking methods. Therefore, we hope that our work benefits future research and helps to improve current state-of-the-art methods. By studying new evaluation metrics for multi-object trackers and understanding the complexity of tracking sequences, we can identify the challenges of current trackers and gain insights into scenarios that pose difficulties for the models. This analysis can guide future research directions and new model concepts for the tracking task.

# 3 Camera Projection of Moving Objects

## 3.1 Background

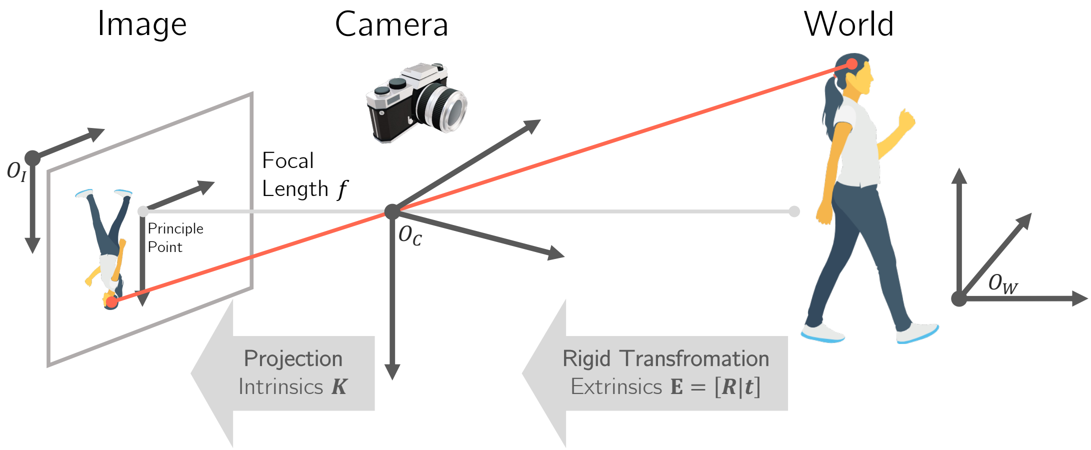


**Figure 16:** Illustration of a pinhole camera model projecting a real-world point to an image position. The projection **P** is defined by a rigid transformation **E** that transforms the coordinates into the camera reference system and the intrinsic matrix **K** projecting a point from the camera reference system onto the image plane.

In the previous section, we introduced the task of trajectory prediction. For this task, objects are represented as 2D world coordinates on a plane in metric space. While lidar sensors provide a 3D point cloud of objects, a single camera setup only captures objects in an RGB-video sequence. Hence, we discuss how a 3D world object appears in the two-dimensional image plane as illustrated in Figure 16.

### 3.1.1 Camera Projection

A monocular camera maps 3-dimensional real-world points to the 2-dimensional camera plane. The pinhole camera model is a simple but effective model that explains this transformation between the real world and the recorded image. Hence, researchers in computer vision often consult this model. For the pinhole estimation, the image plane is positioned at a distance $f$ from the camera's center $O_c$. A 3D-point $(X, Y, Z)^T$ projects to the image positions $(u, v)^T$ as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{f \cdot X}{Z} \\ \frac{f \cdot Y}{Z} \end{pmatrix}. \tag{10}$$

The image projection from $\mathbb{R}^3 \longrightarrow \mathbb{R}^2$ leads to a loss of depth information. All world coordinates along optical rays are projected to the same pixel position. Typically, the image plane has its origin $O_I$ not in the principal point (center of the image), which is why there is an offset $p_x$ and $p_y$, respectively, for both axes added. The focal length,

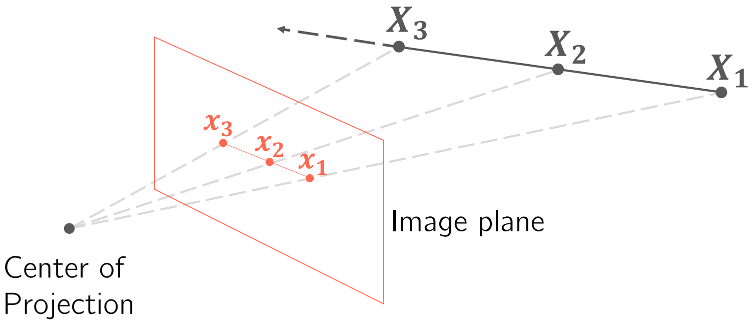


**Figure 17:** Illustration of projected motion into the image plane. The two segments $\mathbf{X}_1 - \mathbf{X}_2$ and $\mathbf{X}_3 - \mathbf{X}_2$ have the same euclidean length. As the object is moving towards the camera, the segments $\mathbf{x}_1 - \mathbf{x}_2$ appear to be shorter than $\mathbf{x}_3 - \mathbf{x}_2$ in the image.

together with the offset parameters, defines the intrinsic matrix,

$$\mathbf{K} = \begin{pmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{pmatrix}. \tag{11}$$

The intrinsic matrix provides a projective mapping from camera coordinates into the image plane.

The camera reference coordinates system is usually different from the world coordinates system. Therefore, we must transform 3D-world points into the camera coordinate reference frame. The relationship between these two coordinate systems is defined by a $3 \times 1$ translation vector $\mathbf{t}$ and a $3 \times 3$ rotation matrix $\mathbf{R}$. Combined, they define a rigid transformation given by the extrinsic matrix

$$E = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array}\right). \tag{12}$$

A monocular camera technically projects 3D world points to 2D points in the image. Given a 3D world coordinate $\mathbf{X}_W$, the camera projection matrix $\mathbf{P}$ transforms the homogeneous coordinates into the image points $\mathbf{x}$ following

$$\mathbf{x} = \mathbf{P}\mathbf{X}_w. \tag{13}$$

The projection matrix $\mathbf{P}$ is the product of the intrinsic matrix $\mathbf{K}$ and extrinsic matrix $\begin{pmatrix} \mathbf{R} & | & \mathbf{t} \end{pmatrix}$.

### 3.1.2 Motion Projection

Objects move around in the real world. In a video sequence, we can only observe the projection of that motion. Depending on the orientation of the recording camera, the

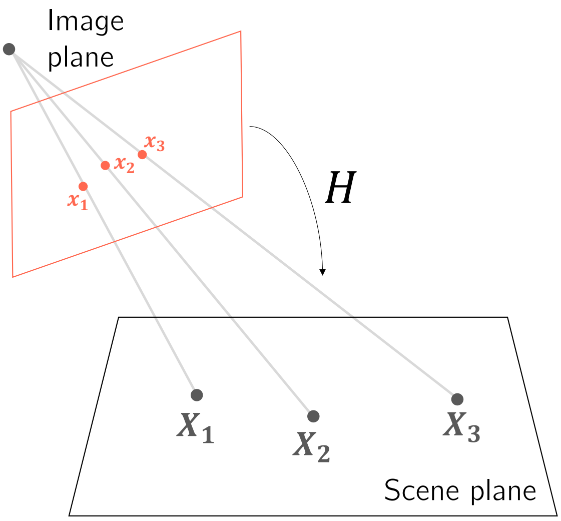


**Figure 18:** Illustration of the homography matrix transforming positions $\mathbf{x}_i$ on the image plane to the points $\mathbf{X}_i$ on the scene plane.

motion of objects moving with the same speed but differently directed velocities appear to be different depending on the projection matrix. An object moving orthogonally to the image plane appears static in the image. Therefore, the projection depends on the object's position relative to the camera position.

The projection matrix $\mathbf{P}$ influences the measurement of motion in the image plane. As illustrated in Figure 17, the segments have the same euclidean distance in metric space; however, the corresponding projection of $\mathbf{X}_1 - \mathbf{X}_2$ into the image space is shorter than $\mathbf{X}_2 - \mathbf{X}_3$. Here, the distance to the camera plays a role as $\mathbf{X}_3$ is closer to the camera than $\mathbf{X}_1$. The relative distance between the camera pose and the scene plane affects, as well as the position of objects on that plane, the projected points. Consequently, even simple linear motion in 3D appears to be non-linear in the projected image space.

### 3.1.3 Homography Transformation

We can restore the proper 3D world coordinates by reverting Equation (10). However, this requires the knowledge of depth measurements, which are usually not recorded in an RGB-camera setup. If we assume that objects are located and moving on a scene plane, then there exists a transformation between the image and scene planes given by the homography matrix $H$ [37]. The homography matrix is a $3x3$ matrix with 8 degrees of freedom that transforms points from the image plane to the scene plane, as shown in Figure 18. The homography matrix $H$ can be estimated using pairwise correspondence between the two planes.

For this section, we want to emphasize the importance of understanding the effect of the camera projection on the recorded motion. It is crucial to understand that the camera projection distorts real motion and a description of trajectories in the image space is insufficient. This insight is essential to building motion models for moving objects in camera sequences. To do so, we must account for the projection effect to describe real

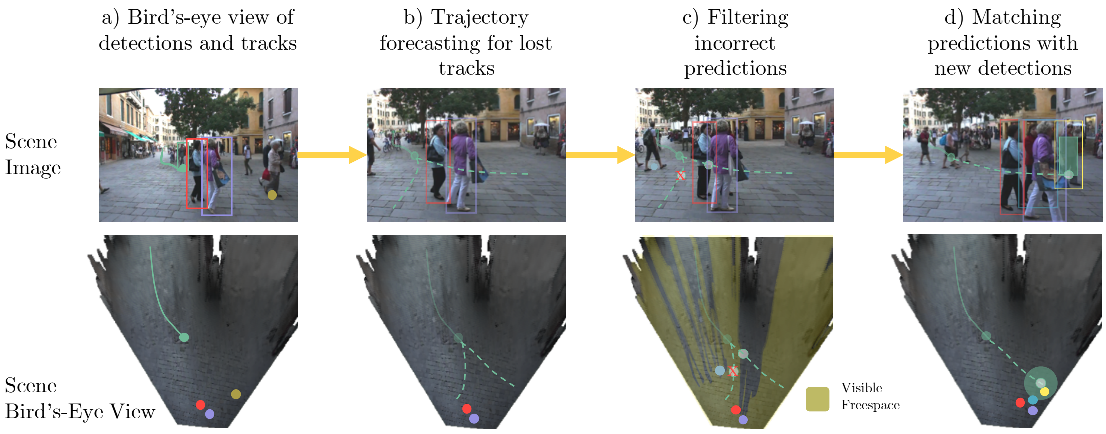

a) Bird's-eye view of detections and tracks
b) Trajectory forecasting for lost tracks
c) Filtering incorrect predictions
d) Matching predictions with new detections
Scene Image
Scene Bird's-Eye View
Visible Freespace

**Figure 19:** Visualization of QuoVadis [29]. We bridge long-term occlusions by (a) localizing object tracks in bird's-eye view via the estimated homography and (b) forecasting future trajectories for *lost* tracks. We (d) continually aim to match these *inactive* track predictions with new object detections and remove incorrect predictions under a visibility constraint (c) of the tracking task.

motion.

As discussed in Section 2.1, many tracking methods rely on simple constant and linear motion models operating in pixel space. As pixel space methods help to bridge short time gaps for slowly moving objects, they are too inaccurate to predict large displacements because they neglect the non-linearity of the trajectories in image space.

## 3.2 Solving Long-Term Occlusions with Trajectory Prediction

Developments in multi-object tracking have been very successful in tracking visible or shortly occluded objects. Despite the advancement in solving short-term occlusion, recovering track identities after long-term occlusions remain challenging. Nevertheless, identity preservation is an important feature in multi-object tracking and essential for video editing, surveillance, or autonomous navigation tasks.

**Table 4:** Our publication on *solving long-term occlusions with trajectory prediction.*

**Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?**
**Patrick Dendorfer**, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé
*Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

Most state-of-the-art methods rely on visual similarity and linear motion to re-identify lost tracks after occlusions. However, these methods do not account for the projective camera effect explained in Section 3.1. While sufficient for short-term predictions, simple linear motion models are not sophisticated enough to predict complex multimodal trajectory forecasts that are needed for long-term occlusions. Long-term occlusions are not the primary concern of current research; these occlusions are generally rare and contribute only little to the overall tracking performance on standard tracking datasets. Our performance analysis shows that state-of-the-art MOT trackers rarely solve occlusion gaps $t > 2s$. This is a relevant problem that one must address.

**Contribution**

We aim to bridge long-term occlusions by leveraging trajectory forecasts of the objects. At this end, we include state-of-the-art pedestrian trajectory forecasting in the tracking pipeline. To establish our tracking-by-forecasting paradigm, we present Quo Vadis [29]. Our method consists of multiple steps, as shown in Figure 19.

In the first step, we additionally represent the objects in the image as two-dimensional points on the scene plane in a bird's-eye view representation. This transformation into metric space resolves the non-linearity of the camera projection (discussed in Section 3.1). We utilize semantic segmentation and depth estimates to compute the transformation between the image positions and the world plane. The transformation into a bird's-eye view allows us to use sophisticated forecasting models. In contrast to models with additional depth or lidar sensors, we focus on monocular tracking, where we only work with 2D images without additional information. Therefore, we need to estimate the homography matrix.

When an object is lost, we store that track in the memory and predict a set of future trajectories in the bird's-eye view representation. Whenever a new object appears, the detection is matched to the existing prediction using the spatial position in the bird's-eye view and appearance features from the image space. In other words, the trajectory prediction spatially decreases the area of the search space for the re-identification of the object.

Our paper aims to improve the performance of multi-object tracking by applying trajectory forecasting. On the other hand, we analyze the benefit of different trends in the trajectory forecasting literature by using different prediction methods and sampling approaches.

**Conclusion**

We have shown how pedestrian trajectory prediction can benefit multi-object tracking to solve long-term occlusions. The significant improvement comes from representing trajectories in a bird's-eye view that enables us to use state-of-the-art multimodal trajectory prediction.

For tracking objects in the 2D image plane, it is advantageous to consult a 3D understanding of the motion and scenario since the 2D image is a projection of the underlying 3D reality. Contrarily, running motion models in pixel space completely ignores the underlying physicality of the objects and their motion. Future work should continue working on 3D motion and object representation supporting 2D tracking with a monocular camera setup.

# III Summary of Selected Publications

This publication-based dissertation comprises the following four publications:

## MULTIMODAL PEDESTRIAN TRAJECTORY PREDICTION

**Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation**
**Patrick Dendorfer**, Aljoša Ošep, and Laura Leal-Taixé
*Asian Conference on Computer Vision (ACCV).* 2020.

**MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction**
**Patrick Dendorfer***, Sven Elflein*, and Laura Leal-Taixé
*International Conference on Computer Vision (ICCV).* 2021.
(* denotes equal contribution)

## BENCHMARKING, EVALUATING, AND ANALYSING MULTI-OBJECT TRACKERS

**MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking**
**Patrick Dendorfer**, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé
*International Journal of Computer Vision (IJCV).* 2020.

## SOLVING LONG-TERM OCCLUSIONS WITH TRAJECTORY PREDICTION

**Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?**
**Patrick Dendorfer**, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé
*Conference on Neural Information Processing Systems (NeurIPS).* 2022.

# 1 Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation

### Citation

**Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation**
**Patrick Dendorfer**, Aljoša Ošep, and Laura Leal-Taixé
*Asian Conference on Computer Vision (ACCV).* 2020.

### Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

### Summary

It appears to be intuitive that humans first determine a goal or an intention before they route toward a goal in a scenario. However, this natural behavior does not reflect in standard trajectory forecasting models, where the decoder usually generates the prediction autoregressively without having a pre-determined goal.

Arguably, existing standard trajectory prediction models theoretically encode the final location of the prediction inside the encoder-decoder bottleneck variable, but this is neither enforced to the model during training nor during inference. The decoder uses this latent information to create the expected trajectory. However, these predictions can suffer from divergence where the hidden information modifies during the decoding process, and the trajectory ends at an unfavorable end position. In general, these network architectures do not allow deeper insight into the prediction processes and do not provide much interpretation of the decisions.

In order to tackle the problem mentioned above, we closely follow the natural behavior of human navigation and model the task of trajectory prediction as an intuitive two-stage process: we first estimate a categorical probability distribution of future goals based on the scene allowing us to sample different positions. Then, we condition a routing module on the selected goal and generate the trajectory leading to this goal.

The goal module leverages information on the past trajectory and the surrounding scene to determine the probabilities for future positions. The routing module is conditioned on the preselected goal and generates feasible paths toward that goal. The module exists of a recurrent neural network that uses local visual information to react to physical constraints in the near surrounding. To this end, the model is capable of learning a multimodal distribution of final goal positions. Since the goal samples are from a discrete distribution where some values will have zero probability, the model input can be seen as disconnected and is, therefore, capable of solving the problem of generating out-of-distribution samples. Furthermore, the estimated goal probability distribution is interpretable and provides a solution for multimodal trajectory prediction. The two-stage prediction process stabilizes the prediction process and prevents divergence of the trajectories.

## 2 MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction

### Citation

**MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction**
**Patrick Dendorfer***, Sven Elflein*, and Laura Leal-Taixé
*International Conference on Computer Vision (ICCV).* 2021.
(* denotes equal contribution)

### Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

### Summary

Pedestrian trajectory prediction is challenging due to its uncertain and multimodal nature. While generative adversarial networks can learn a distribution over future trajectories, they tend to predict out-of-distribution samples when the distribution of future trajectories is a mixture of multiple, possibly disconnected modes. These out-of-distribution samples are particularly harmful to real-world applications when the system needs to act on all forecasts and can only consider a small number of predictions. Therefore, it is necessary to develop forecasting methods that cover all significant modes with as few samples as possible.

Standard generative methods for pedestrian trajectory prediction are based on a single-generator architecture. These models transform a sample from a known distribution into a target trajectory. While trying to transform the unimodal distribution into a multimodal distribution, they oversee the theoretical limitations of single-generator GANs. The problem arises as the generator continuously transforms the probability density of a distribution with connected support into the target distribution. As this transformation cannot change the topology of the underlying unimodal input distribution, the output itself is connected and will have non-zero probability mass at undesired out-of-distribution regions.

This paper proposes a multi-generator model for pedestrian trajectory prediction to overcome the limitations of single-generator GANs. Each generator specializes in learning a distribution over trajectories of one mode of the distribution, while the path mode network estimates a categorical distribution over these generators. As a consequence, the overall model is a discontinuous function because each generator has its own set of model parameters.

The path mode network outputs scores dependent on the dynamics and scene input that allow us to effectively sample the generators, Therefore, the model controls the different modes of the learned distribution. It also provides an interpretable probability distribution of each mode. This is especially useful when we only sample a small number of trajectories because we can select the most suited generators for a specific scenario. Sampling trajectories from specialized generators that cover only one mode reduces the number of out-of-distribution samples compared to single-generator methods.

To train our network, we demonstrate an alternating training schema similar to the expectation-maximization algorithm. Finally, we introduce recall and precision metrics for pedestrian trajectory prediction to measure the quality of the entire generated trajectory distribution. These metrics extend the current $L_2$ based recall metrics and additionally focus on measuring out-of-distribution samples.

# 3 MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking

### Citation

**MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking**
**Patrick Dendorfer**, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé
*International Journal of Computer Vision (IJCV)*. 2020.

### Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

### Summary

Standardized benchmarks have been essential in the development of deep learning computer vision algorithms. Public leaderboards provide a fair and objective measure of the performance of multi-object tracking methods. This paper discusses the need for standardized benchmarks and the history of *MOTChallenge*.

*MOTChallenge* is a benchmark for single-camera multi-object tracking; it was launched in 2014 to curate existing datasets, collect new sequences for training, and create a framework for the standardized evaluation of trackers. The benchmark focuses on multiple people tracking since pedestrians are the most studied object class in the tracking community, with applications ranging from robot navigation to self-driving cars. As part of the benchmark, we provide a detailed overview of the different datasets and challenges. We present the first release *MOT15*, along with numerous state-of-the-art results that were submitted in the last years, *MOT16*, which contains new challenging videos, and *MOT17*, which extends *MOT16* sequences with more precise labels and evaluates tracking performance on three different object detectors.

With the help of our collected data, we provide an extensive analysis of state-of-the-art trackers. This investigation includes comprehensive error analysis and a discussion of different trends in multi-object tracking. This will help newcomers to understand the MOT community's related work and research trends. Hopefully, it will shed light on potential future research directions.

# 4 Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?

### Citation

**Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?**
**Patrick Dendorfer**, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé
*Conference on Neural Information Processing Systems (NeurIPS).* 2022.

### Author Contributions

The author of this dissertation significantly contributed to

- developing the main concepts
- implementing the algorithm
- evaluating the numerical experiments
- writing the paper

### Summary

Recent developments in monocular multi-object tracking have been very successful in tracking visible objects and bridging short occlusion gaps, mainly relying on data-driven appearance models. While performance in short-term tracking has advanced significantly, bridging longer occlusion has remained a challenge. We find that state-of-the-art object trackers bridge less than 10% of occlusions longer than two seconds. However, the persistence of identities is essential for self-driving vehicles, safety cameras, and video editing. Therefore, solving long-term occlusions is a necessary yet unsolved problem.

We suggest that the missing key is reasoning about the future trajectory of the occluded object to rematch it again after the occlusion. Intuitively, the longer the occlusion gap, the larger the search space for possible associations. In this paper, we demonstrate that a small set of trajectory predictions for moving agents will significantly reduce this search space and thus improve *long-term tracking* robustness. To do so, we show that it is crucial to transform the objects' tracks into a bird's-eye view to compensate for the camera projection effect on the recorded motion. The transformation normalizes the trajectories across scenes and image positions into a common metric space. This representation allows us to run state-of-the-art trajectory prediction models and leverage their benefits over simple linear prediction models in pixel space. Moreover, we generate a small yet multimodal set of forecasts while accounting for their localization uncertainty. Hence, objects have representations in the image and bird's-eye view, which enable matching in both spaces.

The disciplines of pedestrian trajectory prediction and multi-object tracking have co-existed with little interaction or exchange. Consequently, trajectory prediction has evolved as an isolated task working on idealized 2D trajectories without dealing with localization uncertainty usually present in real data. The research mainly focuses on the recall of the predictions and decreasing the best-of-N scores on the test data. Multiple trends such as social and physical interactions or multimodality have evolved in the field. We aim to evaluate these trends in prediction models to assess if they actually benefit or even harm the application in a downstream task like tracking. We identify that the best-of-N distance metrics currently used in the field lead to models generating highly diverse trajectories. However, this behavior harms the tracking performance because it does not decrease the search space significantly and increases the number of identity transfers. Moreover, we do not find that a social module adds value to the prediction model experimentally. A small number ($K = 3 - 5$) of multimodal trajectories covering the distribution's main modes, however, increases successful rematches of lost tracks after occlusions.

This paper presents an approach that allows us to leverage state-of-the-art trajectory forecasting methods to resolve long-term occlusions. This way, we can advance state-of-the-art trackers on the *MOTChallenge* dataset and significantly improve their long-term tracking performance.

# IV Concluding Remarks

## 1 Integrated Real-World Trajectory Prediction

Most work published in the field of trajectory prediction still operates in an idealized space without considering localization, measurement, and association uncertainty. As the trajectories in the standard datasets are laboriously preprocessed and already transformed into 2D metrics space, the actual task boils down to building a sequence-to-sequence model while minimizing a target loss.

Although these simplifications have helped to develop the first deep learning prototypes and allowed for the experimentation of different interaction modules, we argue that they nowadays lead to a distracted focus in the field. Research primarily focuses on building very complex models in these idealized and unrealistic settings. Most of the time, the project's goal is to boost the performance in some heavily used datasets. However, this motivation loses sight of the big picture and the applicability of their methods. We advocate that we need to look outside the prediction box and account for uncertainties in incoming data (upstream) and outgoing predictions (downstream).

Given the perception system of an autonomous platform, it detects, classifies, and tracks objects of interest from the raw sensor input. If one of these sequential processes produces errors, the following tasks must consider the estimates' uncertainties. Most trajectory forecasting models are designed for a specific task in a well-defined environment and confidently take upstream input. Nevertheless, when integrating trajectory forecasts into a system pipeline, the question of probability and certainty of particular predictions becomes relevant. There, errors in localization and measurement propagated by upstream tasks can be arguably more significant than the incremental improvement in the performance of even bigger state-of-the-art methods.

A similar argument holds for downstream tasks. Most trajectory prediction methods do not propagate the estimated uncertainty but instead take the prediction's mean or maximum. Consequently, the perceptual uncertainties are not propagated through the process, and predictions are overconfident. Missing confidence intervals are dangerous because risk may be over or underrated.

To this end, we argue that the task of trajectory prediction should not be considered isolated, and the uncertainty of an upstream task must be included. In the past, researchers have developed their methods on public datasets that already provided extracted bird's-eye view trajectories. However, this is an artificial and not realistic setup. So far, we only find [41] recently addressing this issue and we surely see a need for further investigation into the integration of the uncertainty of the predictions.

Moving towards greater application of trajectory prediction, we also emphasize the need for new evaluation metrics and training objects. As discussed in our publication [19], the current best-of-N $L_2$ metrics encourage highly diverse trajectory predictions, which show strong performance on these measures but also ignore out-of-distribution samples. However, these samples can become very harmful as the system does not know which prediction is finally correct, and resulting action to unrealistic predictions can cause dangerous situations. This thesis proposes a precision-recall metric as the first step towards a more comprehensive and accurate evaluation of the performance of prediction algorithms. Nevertheless, future research can focus on improving and developing novel formulations of this metric.

## 2 Tracking-by-Forecasting

Artificial neural networks have made enormous progress in multi-object tracking in recent years. It is possible to track objects during short-term occlusions based on visual appearance features and simple motion cues. For long-term occlusions, however, sophisticated trajectory prediction models in bird's-eye view space are necessary.

In this dissertation, we present in Section 3.2 our method Quo Vadis and establish the concept of *tracking-by-forecasting* for which we combine state-of-the-art trajectory prediction with tracking.

While our work encourages a proof-of-concept on how to integrate real-world trajectory prediction into different tracking methods effectively, future work has to improve transforming tracklets into a bird's-eye view, dealing with track uncertainties, and combining inactive tracks with forecasts. To this end, a line of work is trying to learn an automatic transformation between an image and its bird-eye view representation.

Tracking and forecasting should not be handled separately and instead be trained in an end-to-end and combined fashion. Especially track association matching requires more contextual methods beyond simple Hungarian matching and hardcoded feature thresholding.

The current integration of forecasting only uses the extracted positions of objects but does not make use of the entire semantic information of the scene. Identified structures like buildings, entrances, or bus stops can be regions where pedestrians enter or leave the scene, even in the middle of the image. Scene information can also help resolve occlusions, as these often happen during interactions with static objects like vegetation or buildings. Intuitively, humans semantically understand a scene layout and anticipate where the object most likely re-appears. However, tracking and forecasting methods so far lack this ability. Semantic understanding can play a significant role to improve tracking scenarios when dynamic observations are missing.

## 3 Alternative Data Sources for Training and Testing Models

To solve more complex tasks we increase the model capacity and size. The bigger we build a model, the more important the availability of large data sources for training this model becomes. The model's training requires a large amount of annotated data when trained in a supervised fashion. While novel research primarily studies more performant models, the urge for new datasets remains. Besides the quantity of data, the quality and diversity of scenarios are essential to guarantee the model's generalization to any real-world scenario, as most datasets are still not very diverse.

Unfortunately, manual annotation of video sequences is very costly, so researchers must explore alternatives. On the other hand, a vast amount of raw video data is available that could be used for unsupervised methods. One can imagine a symbiosis between trajectory prediction and multi-object tracking. Movement observation (tracking) and trajectory prediction are fundamentally intertwined tasks. If we track an object across a video sequence, we simultaneously obtain a trajectory of that object. The trajectory can then be used for training the forecasting method. In reverse, we can better track objects if we can predict their future positions. As a starting point for the unsupervised approach, we can use pre-trained optical flow and detection models to extract the initial positions of the object. Until now, [54] is the only work addressing this opportunity for pedestrian trajectory prediction.

The lack of expensively annotated real-world data can also be substituted with synthetically created data rendered by game engines such as GTA, Unity, or UnrealEngine. The impressive development in computer graphics allows us to generate visually almost indistinguishable sequences for the human eye compared to real recordings. These sequences are especially effective for tracking methods that primarily work on appearance features. However, simulating realistic motion, trajectories, and interactions of humans is more challenging and still an ongoing problem. Future research needs to focus on how to generate valuable trajectory data helping training of trajectory forecasting methods and tracking-by-forecasting models.

## 4 “Epur si muove." – “And yet it does move."

This dissertation presents research in pedestrian trajectory prediction and multi-object tracking to develop and combine new prediction and tracking methods to better understand human motion for real-world applications.

Given the solo effort and progress in trajectory prediction and tracking, future research must address problems jointly to achieve robust and applicable solutions. In that fashion, we need to aim for the development of end-to-end deep learning models incorporating tracking and prediction.

In the last few years, we have worked on the problems and solutions presented in this dissertation. We spent many hours understanding, studying, and thinking about the current challenges of these tasks. Despite the huge number of open problems, we hope that we have advanced the field, at least a very small step. Computer systems are on a promising path towards understanding and anticipating human motion, and academic research is increasingly being applied to real-world applications. As there are many open questions, we are excited to see what future research can achieve.

# Bibliography

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. **Social LSTM: Human Trajectory Prediction in Crowded Spaces**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. **Social Ways: Learning multi-modal distributions of pedestrian trajectories with GANs**. In: *Conference on Computer Vision and Pattern Recognition (Workshops)*. 2019.

[3] Samreen Anjum and Danna Gurari. **CTMC: Cell Tracking with Mitosis Detection dataset challenge**. In: *Conference on Computer Vision and Pattern Recognition (Workshops)*. 2020.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. **Wasserstein Generative Adversarial Networks**. In: *International Conference on Machine Learning (ICML)*. 2017.

[5] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. **Tracking without bells and whistles**. In: *International Conference on Computer Vision (ICCV)*. 2019.

[6] Keni Bernardin and Rainer Stiefelhagen. **Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics**. In: *Journal on Image and Video Processing*. 2008.

[7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. **Simple online and realtime tracking**. In: *International Conference on Image Processing (ICIP)*. 2016.

[8] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. **Accurate and diverse sampling of sequences based on a "Best of Many" sample objective**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[9] Guillem Braso and Laura Leal-Taixé. **Learning a Neural Solver for Multiple Object Tracking**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[10] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. **nuScenes: A multimodal dataset for autonomous driving**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[11] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. **Argoverse: 3d tracking and forecasting with rich maps**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[12] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. **InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets**. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2016.

[13] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. **Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[14] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. **End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results**. In: *Conference on Neural Information Processing Systems (Workshops)*.

[15] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. **A Recurrent Latent Variable Model for Sequential Data**. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2015.

[16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. **R-FCN: Object Detection via Region-based Fully Convolutional Networks**. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2016.

[17] *DALL·E 2*. https://openai.com/dall-e-2/. Accessed: 2022-10-25.

[18] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. **TAO: A large-scale benchmark for tracking any object**. In: *European Conference on Computer Vision (ECCV)*. 2020.

[19] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. **MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction**. In: *International Conference on Computer Vision (ICCV)*. 2021.

[21] Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé. **Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation**. In: *Asian Conference on Computer Vision (ACCV)*. 2020.

[22] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. **MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking**. In: *International Journal of Computer Vision (IJCV)*. 2020.

[25] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. **CVPR19 Tracking and Detection Challenge: How crowded can it get?** In: *arXiv*. 2019.

[26] 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**. In: *arXiv*. 2020.

[29] Patrick Dendorfer, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé. **Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?** In: *Conference on Neural Information Processing Systems (NeurIPS).* 2022.

[31] Xingping Dong and Jianbing Shen. **Triplet Loss in Siamese Network for Object Tracking**. In: *European Conference on Computer Vision (ECCV).* 2018.

[32] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. **MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?** In: *International Conference on Computer Vision (ICCV).* 2021.

[33] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. **Recurrent Network Models for Human Dynamics**. In: *International Conference on Computer Vision (ICCV).* 2015.

[34] Andreas Geiger, Philip Lenz, and Raquel Urtasun. **Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite**. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2012.

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. **Generative Adversarial Nets**. In: *Conference on Neural Information Processing Systems (NeurIPS).* 2014.

[36] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. **Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks**. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2018.

[37] Richard Hartley and Andrew Zisserman. **Multiple View Geometry in Computer Vision**. 2nd ed. Cambridge University Press, 2003. ISBN: 0521540518.

[38] Dirk Helbing and Péter Molnár. **Social force model for pedestrian dynamics**. In: *Physical Review E.* 1995.

[39] Sepp Hochreiter. **The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions**. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.* 1998.

[40] Sepp Hochreiter and Jürgen Schmidhuber. **Long Short-Term Memory**. In: *Neural Computation.* 1997.

[41] Boris Ivanovic, Yifeng Lin, Shubham Shrivastava, Punarjay Chakravarty, and Marco Pavone. **Propagating State Uncertainty Through Trajectory Forecasting**. In: *International Conference on Robotics and Automation (ICRA).* 2022.

[42] Tarasha Khurana, Achal Dave, and Deva Ramanan. **Detecting Invisible People**. In: *International Conference on Computer Vision (ICCV).* 2021.

[43] Chanho Kim, Li Fuxin, Mazen Alotaibi, and James M. Rehg. **Discriminative Appearance Modeling with Multi-track Pooling for Real-time Multi-object Tracking**. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2021.

[44] Diederik P. Kingma and Max Welling. **Auto-Encoding Variational Bayes**. In: *International Conference on Learning Representations (ICLR)*. 2014.

[45] 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: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[46] H. W. Kuhn. **The Hungarian method for the assignment problem**. In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

[47] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. **Learning by Tracking: Siamese CNN for Robust Target Association**. In: *Conference on Computer Vision and Pattern Recognition (Workshops)*. 2016.

[48] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. **Crowds by Example**. In: *Comput. Graph. Forum*. 2007.

[49] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. **The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[50] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. **Feature Pyramid Networks for Object Detection**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[52] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. **HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking**. In: *International Journal of Computer Vision (IJCV)*. 2020.

[53] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. **Multiple object tracking: A literature review**. In: *Artificial Intelligence*. Vol. 293. 2021.

[54] Yuexin Ma, Xinge ZHU, Xinjing Cheng, Ruigang Yang, Jiming Liu, and Dinesh Manocha. **AutoTrajectory: Label-free Trajectory Extraction and Prediction from Videos using Dynamic Points**. In: *European Conference on Computer Vision (ECCV)*. 2020.

[55] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. **Least Squares Generative Adversarial Networks**. In: *International Conference on Computer Vision (ICCV)*. 2016.

[56] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. **TrackFormer: Multi-Object Tracking with Transformers**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[57] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. **Unrolled Generative Adversarial Networks**. In: *International Conference on Learning Representations (ICLR)*. 2017.

[58] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. **MOT16: A Benchmark for Multi-Object Tracking**. In: *arXiv*. 2016.

[59] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. **Beyond short snippets: Deep networks for video classification**. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2015.

[60] World Health Organization. **Global status report on road safety 2018**. World Health Organization, 2018.

[61] Seymour Papert and Marvin Minsky. *Summer Vision Project.* Tech. rep. 16. Massachusetts Institute of Technology, 1966.

[62] Emanuel Parzen. **On Estimation of a Probability Density Function and Mode**. In: *The Annals of Mathematical Statistics.*

[63] Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B. Moeslund. **3D-ZeF: A 3D Zebrafish Tracking Benchmark Dataset**. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2020.

[65] Malte Pedersen, Joakim Bruslund Haurum, Patrick Dendorfer, and Thomas B. Moeslund. **MOTCOM: The Multi-Object Tracking Dataset Complexity Metric**. In: *European Conference on Computer Vision (ECCV).* 2022.

[66] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. **Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings**. In: *European Conference on Computer Vision (ECCV).* 2010.

[67] Zenon W. Pylyshyn and Raymond Storm. **Tracking multiple independent targets: evidence for a parallel tracking mechanism.** In: *Spatial Vision.* 1988.

[68] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. In: *Conference on Neural Information Processing Systems (NeurIPS).* 2015.

[69] 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 (ECCV).* 2016.

[70] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. **Learning social etiquette: Human trajectory understanding in crowded scenes**. In: *European Conference on Computer Vision (ECCV).* 2016.

[71] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. **Human motion trajectory prediction: a survey**. In: *The International Journal of Robotics Research.* 2020.

[72] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. **Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies**. In: *International Conference on Computer Vision (ICCV).* 2017.

[73] 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: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2019.

[74] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. **Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data**. In: *European Conference on Computer Vision (ECCV)*. 2020.

[75] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. **What the Constant Velocity Model Can Teach Us About Pedestrian Motion Prediction**. In: *Robotics and Automation Letters (RA-L)*. 2020.

[76] Christoph Schöller and Alois Knoll. **FloMo: Tractable Motion Prediction with Normalizing Flows**. In: *International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[77] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. **Deep Unsupervised Learning using Nonequilibrium Thermodynamics**. In: *International Conference on Machine Learning (ICML)*.

[78] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. **Scalability in perception for autonomous driving: Waymo open dataset**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[79] 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: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[80] Ramana Sundararaman, Cedric De Almeida Braga, Eric Marchand, and Julien Pettre. **Tracking Pedestrian Heads in Dense Crowd**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[81] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. **Learning disconnected manifolds: a no GANs land**. In: *Proceedings of Machine Learning and Systems (MLSys)*. 2020.

[82] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. **Learning to track with object permanence**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is All you Need**. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

[84] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. **Towards Real-Time Multi-Object Tracking**. In: *European Conference on Computer Vision (ECCV)*. 2020.

[85] Yihong Xu, Aljoša Ošep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. **How To Train Your Deep Multi-Object Tracker**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[86] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. **BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning**. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[87] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. **FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking**. In: *International Journal of Computer Vision (IJCV)*. 2021.

[88] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. **Tracking objects as points**. In: *European Conference on Computer Vision (ECCV)*. 2020.

[89] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. **Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks**. In: *International Conference on Computer Vision (ICCV)*. 2017.

[90] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. **Toward Multimodal Image-to-Image Translation**. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

# Publications

# Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation

**Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation**
**Patrick Dendorfer**, Aljoša Ošep, and Laura Leal-Taixé
*Asian Conference on Computer Vision (ACCV)*. 2020.

# Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation

Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé

Technical University Munich
{patrick.dendorfer,aljosa.osep,leal.taixe}@tum.de

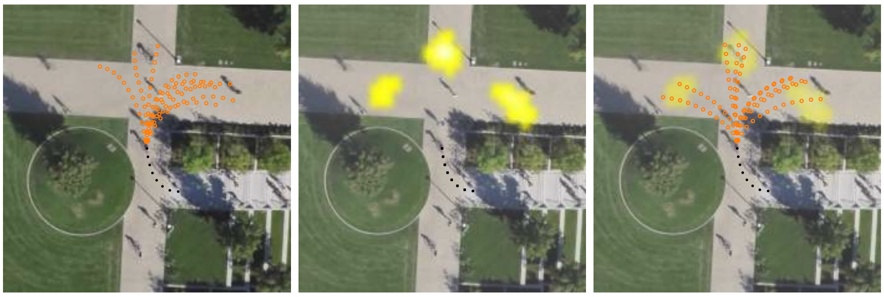
**Abstract.** In this paper, we present Goal-GAN, an interpretable and end-to-end trainable model for human trajectory prediction. Inspired by human navigation, we model the task of trajectory prediction as an intuitive two-stage process: (i) goal estimation, which predicts the most likely target positions of the agent, followed by a (ii) routing module which estimates a set of plausible trajectories that route towards the estimated goal. We leverage information about the past trajectory and visual context of the scene to estimate a multi-modal probability distribution over the possible goal positions, which is used to sample a potential goal during the inference. The routing is governed by a recurrent neural network that reacts to physical constraints in the nearby surroundings and generates feasible paths that route towards the sampled goal. Our extensive experimental evaluation shows that our method establishes a new state-of-the-art on several benchmarks while being able to generate a realistic and diverse set of trajectories that conform to physical constraints.

## 1 Introduction

Modeling human motion is indispensable for autonomous systems that operate in public spaces, such as self-driving cars or social robots. Safe navigation through crowded scenes and collision prevention requires awareness not only of the present position but also of the future path of all moving objects. Human trajectory prediction is particularly challenging since pedestrian trajectories depend primarily on their intention – and the destination of a pedestrian is inherently unknown to the external observer. Consider the example of a pedestrian reaching a crossroad such as the one depicted in Figure 1. Based solely on past observations, we cannot infer the future path of the pedestrian: turning right, left, or going straight, are all equally likely outcomes.

For this reason, a powerful prediction model should be able to capture the *multimodality* of this task, *i.e.*, forecast trajectories that cover the distinctive modes present in the scene. Furthermore, it should produce a *diverse* set of the paths within each mode, reflecting inherent uncertainty in walking style, velocity, and different strategies for obstacle avoidance.

To capture the stochastic nature of trajectory prediction, state-of-the-art methods leverage generative the power of variational autoencoders (VAEs) [1,2,3]



(a) Vanilla GAN (b) Goal Probabilities (c) Goal-GAN

Fig. 1: Visual comparison between predictions of our proposed Goal-GAN and a vanilla GAN. In contrast to the baseline, our proposed model covers all three modes and predicts diverse and feasible trajectories by explicitly estimating realistic goals.

and/or generative adversarial networks (GANs) [4,5,6] to predict a set of trajectories for every observation.

While generative methods are widely used to generate diverse outputs, they are unable to explicitly capture the inherent multimodality of pedestrian trajectories. Often, these methods generate highly diverse trajectories but tend to neglect the physical structure of the environment. The resulting trajectories are not necessarily feasible, and often do not fully cover multiple possible directions that a pedestrian can take (Figure 1a). A more natural way of capturing all feasible directions is to first determine an intermediate goal sampled from a distribution of plausible positions, as shown in Figure 1b. In the second step, the model generates the trajectories reaching the sampled positions (Figure 1c). While social interactions among agents [7,4,5,6] and local scene interaction have been extensively studied, there are almost no methods tackling the challenge of explicitly learning the inherent multimodal distribution of pedestrian trajectories.

In this paper, we aim to bridge this gap and explicitly focus on the underexplored problem of generating diverse multimodal trajectories that conform to the physical constraints. Influenced by recent studies on human navigation [8] we propose an end-to-end trainable method that separates the task of trajectory prediction into two stages. First, we estimate a posterior over possible goals, taking into account the dynamics of the pedestrian and the visual scene context, followed by the prediction of trajectories that route towards these estimated goals. Therefore, trajectories generated by our model take both local scene information and past motion of the agent explicitly into account. While the estimated distribution of possible goal positions reflects the multimodality in the scene, the routing module reacts to local obstacles and generates diverse and feasible paths. We ensure diversity and realism of the output trajectories by training our network in a generative adversarial setup.

In summary, our main **contribution** is three-fold: (i) we propose Goal-GAN, a two-stage end-to-end trainable trajectory prediction method inspired by human navigation, which separates the prediction task into goal position estimation and routing. (ii) To this end, we design a novel architecture that explicitly estimates an interpretable probability distribution of future goal positions and allows us to sample from it. Using the Gumbel Softmax trick [9] enables us to train the network through the stochastic process. (iii) We establish a new state-of-the-art on several public benchmarks and qualitatively demonstrate that our method predicts realistic end-goal positions together with plausible trajectories that route towards them. The code for Goal-GAN[1] is publicly available.

## 2 Related Work

Several methods focus on modelling human-human [4,7], human-space interactions [10,2,11], or both [5]. Recent methods leverage generative models to learn a one-to-many mapping, that is used to sample multimodal future trajectories.
**Trajectory Prediction.** Helbing and Molar introduced the Social Force Model (SFM) [12], a physics-based model, capable of taking agent-agent and agent-space interactions into account. This approach was successfully applied in the domain of multi-object tracking [13,14,15,16]. Since then, data-driven models [17,18,7,19,4] have vastly outperformed physics-based models. Encoder-decoder based methods [2,7] leverage recurrent neural networks (RNNs) [20] to model the temporal evolution of the trajectories with long-short term memory (LSTM) units [21]. These deterministic models cannot capture the stochastic nature of the task, as they were trained to minimize the $L_2$ distance between the prediction and the ground truth trajectories. This often results in implausible, average-path trajectories.

Recent methods [22,11] focus on human-space interactions using bird-view images [5] and occupancy grids [10,23] to predict trajectories that respect the structural constraints of the scene. Our method similarly leverages bird-eye views. However, we use visual information to explicitly estimate feasible and interpretable goal positions, that can, in turn, be used to explicitly sample end-goals that ease the task of future trajectory estimation.
**Generative Models for Trajectory Prediction.** Recent works [4,5,6] leverage generative models to sample diverse trajectories rather than just predicting a single deterministic output. The majority of methods either use variational autoencoders (VAEs) [24,3,2,25,26,27,11] or generative adversarial networks (GANs) [28,4,5,6,29]. Social GAN (S-GAN) [4] uses a discriminator to learn the distribution of socially plausible paths. Sadeghian *et al.* [5] extend the model to human-environment interactions by introducing a soft-attention [30] mechanism. GANs have shown promising results for the task of trajectory prediction, but tend to suffer from mode collapse. To encourage the generator to

[1] `https://github.com/dendorferpatrick/GoalGAN`

produce more diverse predictions, [1] uses a best-of-many sampling approach during training while [6] enforces the network to make use of the latent noise vector in combination with BicycleGAN [31] based training. While producing trajectories with high variance, many trajectories are not realistic, and a clear division between different feasible destinations (reflecting inherent multi-modality of the inherent task) is not clear. To account for that, we take inspiration from prior work conditioning the trajectory prediction on specific target destinations.

**Goal-conditioned forecasting.** In contrast to the aforementioned generative models that directly learn a one-to-many mapping, several methods propose two-stage prediction approaches. Similarly to ours, these methods predict first the final (goal) position, followed by trajectory generation that is conditioned on this position. Early work of [32] models the distribution over possible destinations using a particle filter [33] while other approaches [34] propose a Bayesian framework that estimates both, the destination point together with the trajectory. However, these purely probabilistic approaches are highly unstable during training. The Conditional Generative Neural System (CGNS) [35] uses variational divergence minimization with soft-attention [30] and [36] presents a conditional flow VAE that uses a conditional flow-based prior to effectively structured sequence prediction. These models condition their trajectory generator on initially estimated latent codes but do not explicitly predict a goal distribution nor sample an explicit goal position. Most recently, [37] proposes P2TIRL that uses a maximum entropy inverse reinforcement learning policy to infer goal and trajectory plan over a discrete grid. P2TRL assigns rewards to future goals that are learned by the training policy which is slow and computationally expensive. In contrast, we directly learn the multimodal distribution over possible goals using a binary cross-entropy loss between the (discrete) probability distribution estimate and the ground truth goal position. This makes our work the first method (to the best of our knowledge) that directly predicts an explicit (and discrete) probability distribution for multimodal goals and is efficiently end-to-end trainable.

## 3 Problem Definition

We tackle the task of predicting the future positions of pedestrians, parametrized via $x$ and $y$ coordinates in the 2D ground plane. As input, we are given their past trajectory and visual information of the scene, captured from a bird-view.

We observe the trajectories $X_i = \{(x_i^t, y_i^t) \in \mathbb{R}^2 | t = 1, \ldots, t_{obs} \}$ of $N$ currently visible pedestrians and a top-down image $I$ of the scene, observed at the timestep $t_{obs}$. Our goal is to predict the future positions $Y_i = \{(x_i^t, y_i^t) \in \mathbb{R}^2 | t = t_{obs} + 1, \ldots, t_{pred}\}$.

In the dataset, we are only given one future path for $t_{obs}$ – in particular, the one that was observed in practice. We note that multiple distinctive trajectories could be realist for this observed input trajectory. Our goal is, given the input past trajectory $X_i$, to generate $k \in \{1, \ldots, K\}$ multiple future samples $\hat{Y}_i^k$ for all pedestrians $i \in \{1, \ldots, N\}$. These should cover all feasible modes and be compliant with the physical constraints of the scene.

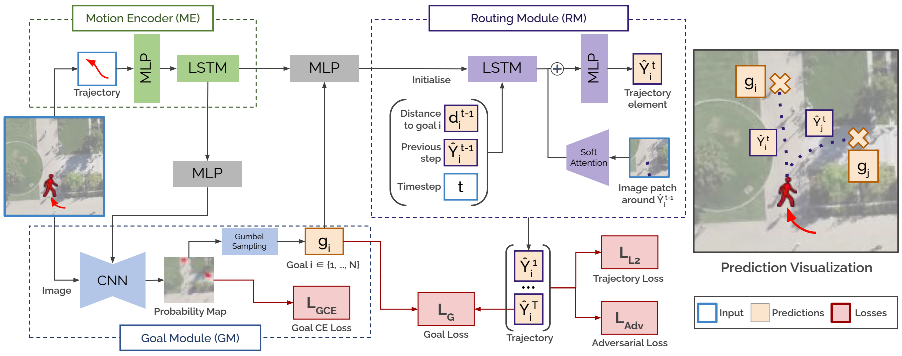


Fig. 2: **Overview of model architecture:** Our model consists of three components: 1) Motion Encoder, 2) Goal Module, and the 3) Routing Module. The Goal Module combines the dynamic features of the Motion Encoder and the scene information to predict a final goal $g$. The Routing Module generates the future trajectory while taking into account the dynamic features and the estimated goal. During inference, we generate multiple trajectories $i$ by sampling goals from the estimated goal probability map.

## 4 Goal-GAN

When pedestrians walk through public spaces, they aim to reach a predetermined goal [8], which depends on their intentions and the scene context. Once the goal is set, humans route to their final destination while reacting to obstacles or other pedestrians along their way. This observation motivates us to propose a novel two-stage architecture for trajectory prediction that first estimates the end-goal position and then generates a trajectory towards the estimated goal. Our proposed Goal-GAN consists of three key components, as shown in Figure 2.

- *Motion Encoder (ME):* extracts the pedestrians' dynamic features recursively with a long short-term memory (LSTM) unit capturing the speed and direction of motion of the past trajectory.
- *Goal Module (GM):* combines visual scene information and dynamic pedestrian features to predict the goal position for a given pedestrian. This module estimates the probability distribution over possible goal (target) positions, which is in turn used to sample goal positions.
- *Routing Module (RM):* generates the trajectory to the goal position sampled from the GM. While the goal position of the prediction is determined by the

GM, the RM generates feasible paths to the predetermined goal and reacts to obstacles along the way by using visual attention.

Figure 2 shows an overview of our model. In the following sections, we motivate and describe the different components in detail.

### 4.1 Motion Encoder (ME)

The past trajectory of a pedestrian is encoded into the Motion Encoder (ME), which serves as a dynamic feature extractor to capture the speed and direction of the pedestrian, similarly to [7,4]. Each trajectory's relative displacement vectors $(\Delta x_i^t, \Delta y_i^t)$ are embedded into a higher dimensional vector $e^t$ with a multi-layer perceptron (MLP). The output is then fed into an LSTM, which is used to encode the trajectories. The hidden state of the LSTM, $h_{ME}$, is used by the other modules to predict the goal and to decode the trajectory for each pedestrian.

### 4.2 Goal Module (GM)

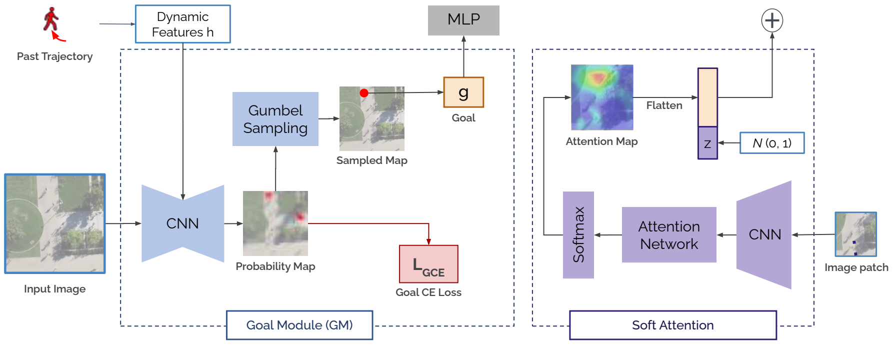


Fig. 3: **Goal Module (GM) and Soft Attention (SA).** The Goal Module samples a goal coordinate $g$ while the soft attention assigns attention scores to spatial positions.

In our work, we propose a novel Goal Module (GM). The Goal Module combines visual and dynamic features of the pedestrian to estimate a distribution of possible end goals. As can be seen in Figure 1, the scene dictates the distinctive modes for possible trajectories. Here, the pedestrian can go left, right, or straight. The Goal Module is responsible for capturing all the possible modes and predicting a final goal position, *i.e.*, choosing one of the three options.

**Architecture.** In order to estimate the goal distribution, the network assesses the visual scene and the dynamics of the pedestrian. The visual scene is represented as an RGB image (or a semantic map) of size $H \times W$, captured from a bird-eye view. This image is input to the goal module.

The scene image is passed through an encoder-decoder CNN network with skip connections similar to [38]. Before the decoder, the scene image features in the bottleneck layer are concatenated with the motion features $h_{ME}$ from the Motion Encoder. Intuitively, the CNN decoder should analyze both the past trajectory and the scene to estimate the future target positions – goals. The module outputs a probability distribution that reflects the multimodal directions for a given input trajectory and scene.

**Training through sampling.** The CNN decoder outputs a score map $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ for which each $\alpha_i$ reflects the probability of a particular cell being the end-goal location of the agent.

The discrete probability distribution $\alpha$ is used to sample an end-goal by using the Gumbel-Softmax-Trick [9]. This allows us to sample a discrete distribution over possible goal locations while being able to backpropagate the loss through the stochastic process. The resulting two-dimensional goal position $g$ is sampled randomly from the 2D grid representing the scene,

**Goal Sampling vs. Soft Attention.** A major novelty of our work is the Goal Module that replaces soft attention [30] to process the scene contextual information [5,11]. Both approaches are illustrated in Figure 3. A soft attention module assigns attention scores to spatially relevant positions based on the visual CNN features. In [5], the attention values are combined with random noise and fed to the trajectory decoder to generate scene-aware multimodal trajectories. However, this often leads to unsatisfying results when the network simply ignores the spatial attention scores or has difficulties combining the attention values with the noise to capture all modes in the scene.

We argue that the attention module is useful when predicting the route towards a goal (as we show in Section 4.3), as it encourages the feasibility of the predicted trajectories. However, the model that solely relies on soft visual attention mechanism tends to generate trajectories that do not capture the multimodal nature of the task, as illustrated in Figure 1. Furthermore, in Section 5, we experimentally confirm that stochasticity of the task is reflected better when sampling from the learned probability distribution, produced by our Goal Module, compared to merely relying on noise injection.

We can directly train the module for the goal position estimation using the Gumbel Softmax trick [9], in combination with the standard cross-entropy loss, which is directly applied to the estimated goal distribution based on the observed (final) position of the ground truth trajectories. We emphasize that we do not use nor need any other data than what is provided in the standard training set.

During the inference, we simply sample the goal from the learned probability distribution and pass it to the decoder. This significantly eases the task for the decoder, as the Goal Module already assesses the visual surroundings and only passes a low dimensional input into the routing module.

### 4.3 Routing Module (RM)

The Routing Module (see Figure 2) is the third component of our method. It combines the dynamic features and the global goal estimate to generate the final trajectory prediction. The RM consists of an LSTM network, a visual soft attention network (ATT), and an additional MLP layer that combines the attention map with the output of the LSTM iteratively at each timestep.

First, we forward the goal estimate embedding $e_g$ and the object dynamics embedding $h_{ME}$ (given by the motion encoder, ME) to an MLP to initialise the hidden state $h^0_{RM}$ of the RM.

Then, we recursively estimate predictions for the future time steps. To this end, the LSTM in the RM obtains three inputs: the previous step prediction $\hat{Y}^{t-1}$, the remaining distance to the estimated goal $d_{t-1} = g - \hat{Y}^{t-1}$ and the current scalar timestep value $t$.

To assess the traversability of the local surroundings, we apply soft attention [30] on the image patch centered around the current position of the pedestrian. As shown in the Figure 3, we combine the output of the LSTM with the attention map $F^t$ to predict the next step $\hat{Y}^t$. The visual attention mechanism allows the RM to react to obstacles or other nearby structures. Finally, we use both the dynamic and visual features to predict the final prediction $\hat{Y}^t$.

### 4.4 Generative Adversarial Training

In our work, we use a Generative Adversarial Network (GAN) to train our trajectory generator to output realistic and physically feasible trajectories. The GAN consists of a Generator and Discriminator network competing in a two-player min-max game. While the generator aims at producing feasible trajectories, the discriminator learns to differentiate between real and fake samples, i.e., feasible and unfeasible trajectories. Adversarial training is necessary because, in contrast to prediction accuracy, it is not possible to formulate a differential loss in a closed mathematical form that captures the concept of feasibility and realism of the generated trajectories.

The discriminator network consists of an LSTM network that encodes the observed trajectory $X$. This encoding is used to initialize the second LSTM that processes the predicted trajectory $Y$ together with visual features (obtained from the CNN network, that encodes the image patch centered around the current position) at each time step. Finally, the last hidden state of the LSTM$_{pred}$ is used for the final output of the discriminator.

### 4.5 Losses

For training our Goal-GAN we use multiple losses addressing the different modules of our model. To encourage the generator to predict trajectories, that are closely resembling the ground truth trajectories, we use a best-of-many [1] distance loss $\mathcal{L}_{L2} = \min_k \|Y - \hat{Y}^{(k)}\|_2$ between our predictions $\hat{Y}$ and the ground truth $Y$. As an adversarial loss, we employ the *lsgan* [39] loss:

$$\mathcal{L}_{Adv} = \frac{1}{2}\mathbb{E}\left[\left(D\left(X, Y\right) - 1\right)^2\right] + \frac{1}{2}\mathbb{E}\left[D(X, \hat{Y})^2\right], \tag{1}$$

due to the fact, the original formulation [28] using a classifier with sigmoid cross-entropy function potentially leads to the vanishing gradient problem.

To encourage the network to take into account the estimated goal positions for the prediction, we propose a goal achievement losses $\mathcal{L}_G$ that measures the $L_2$ distance between the goal prediction $g$ and the actual output $\hat{Y}^{t_{pred}}$,

$$\mathcal{L}_G = \|g - \hat{Y}^{t_{pred}}\|_2. \tag{2}$$

In addition, we use a cross-entropy loss

$$\mathcal{L}_{GCE} = -\log\left(p_i\right), \tag{3}$$

where $p_i$ is the probability that is predicted from the Goal Module for the grid cell $i$ corresponding to the final ground-truth position. The overall loss is the combination of the partial losses weighted by $\lambda$:

$$\mathcal{L} = \lambda_{Adv}\,\mathcal{L}_{Adv} + \mathcal{L}_{L2} + \lambda_G\,\mathcal{L}_G + \lambda_{GCE}\,\mathcal{L}_{GCE}. \tag{4}$$

## 5 Experimental Evaluation

In this section, we evaluate our proposed Goal-GAN on three standard datasets used to assess the performance of pedestrian trajectory prediction models:
ETH [40], UCY [41] and Stanford Drone Dataset (SDD) [19]. To assess how well our prediction model can cover different possible modes (splitting future paths), we introduce a new, synthetically generated scene.

We compare our method with several state-of-the-art methods for pedestrian trajectory prediction and we qualitatively demonstrate that our method produces multi-modal, diverse, feasible, and interpretable results.

**Evaluation measures.** We follow the standard evaluation protocol and report the prediction accuracy using Average Displacement Error (ADE) and Final Displacement Error (FDE). Both measures are computed using the $L_2$ distance between the prediction and ground truth trajectories. The generative models are tested on these metrics with a $N-K$ variety loss [1,4,5]. As in the previous work [19,7], we observe 8 time steps (3.2 seconds) and predict the future 12 time steps (4.8 seconds) simultaneously for all pedestrians in the scene.

**Visual input and coordinates.** As in [5], we use a single static image to predict trajectories in a given scene. We transform all images into a top-down view using the homography transformation provided by the respective datasets. This allows us to perform all predictions in real-world coordinates.

### 5.1 Benchmark Results

In this section, we compare and discuss our method's performance against state-of-the-art on ETH [40], UCY [41] and SDD [19] datasets.

**Datasets.** ETH [40] and UCY datasets [41] contain 5 sequences (ETH:2, UCY: 3), recorded in 4 different scenarios. All pedestrian trajectories are converted

Table 1: Quantitative results for ETH [40] and UCY [41] of Goal-GAN and baseline models predicting 12 future timesteps. We report ADE and FDE in meters.

| | Baseline | | | | | | Ours |
|---|---|---|---|---|---|---|---|
| Dataset | S-LSTM<br>[7] | S-GAN<br>[4] | S-GAN-P<br>[4] | SoPhie<br>[5] | S-BiGAT<br>[6] | CGNS<br>[35] | Goal<br>GAN |
| K | 1 | 20 | 20 | 20 | 20 | 20 | 20 |
| **ETH** | 1.09/2.35 | 0.81/1.52 | 0.87/1.62 | 0.70/1.43 | 0.69/1.29 | 0.62/1.40 | **0.59**/**1.18** |
| **HOTEL** | 0.79/1.76 | 0.72/1.61 | 0.67/1.37 | 0.76/1.67 | 0.49/1.01 | 0.70/0.93 | **0.19**/**0.35** |
| **UNIV** | 0.67/1.40 | 0.60/1.26 | 0.76/1.52 | 0.54/1.24 | 0.55/1.32 | **0.48**/1.22 | 0.60/**1.19** |
| **ZARA1** | 0.47/1.00 | 0.34/0.69 | 0.35/0.68 | **0.30**/0.63 | **0.30**/0.62 | 0.32/**0.59** | 0.43/0.87 |
| **ZARA2** | 0.56/1.17 | 0.42/0.84 | 0.42/0.84 | 0.38/0.78 | 0.36/0.75 | 0.35/0.71 | **0.32**/**0.65** |
| **AVG** | 0.72/1.54 | 0.58/1.18 | 0.61/1.21 | 0.54/1.15 | 0.48/1.00 | 0.49/0.97 | **0.43**/**0.85** |

into real-world coordinates and interpolated to obtain positions every 0.4 seconds. For training and testing, we follow the standard leave-one-out approach, where we train on 4 datasets and test on the remaining one. The Stanford Drone Dataset (SDD) [19] consists of 20 unique video sequences captured at the Stanford University campus. The scenes have various landmarks such as roundabouts, crossroads, streets, and sidewalks, which influence the paths of pedestrians. In our experiments, we follow the train-test-split of [42] and focus on pedestrians.
**Baselines.** We compare our model to several published methods. S-LSTM [7] uses a LSTM encoder-decoder network with social pooling. S-GAN [4] leverages a GAN framework and S-GAN-P [4] uses max-pooling to model social interactions. SoPhie [5] extends the S-GAN model with a visual and social attention module, and Social-BiGAT [6] uses a BicycleGAN [43] based training. DESIRE [2] is an inverse optimal control based model, that utilizes generative modeling. CARNet [11] is a physically attentive model. The Conditional Generative Neural System (CGNS) [35] uses conditional latent space learning with variational divergence minimization to learn feasible regions to produce trajectories. CF-VAE [36] leverages a conditional normalizing flow-based VAE and P2TIRL [37] uses a grid-based policy learned with maximum entropy inverse reinforcement learning policy. As none of the aforementioned provide publicly available implementation, we outline the results reported in the respective publications.

**ETH and UCY.** We observe a clear trend – the generative models improve the performance of the deterministic approaches, as they are capable of sampling a diverse set of trajectories. Compared to other generative models, Goal-GAN achieves state-of-the-art performance with an overall decrease of the error of nearly 15% compared to *S-BiGAT* and *CGNS*. While *SoPhie* and *S-BiGAT* also use visual input, these models are unable to effectively leverage this information to discover the dominant modes for the trajectory prediction task, thus yielding a higher prediction error. It has to be pointed out that Goal-GAN decreases the average FDE by $0.12m$ compared to the current state-of-the-art method. We explain the drastic increase in performance with our new Goal Module as we can cover the distribution of all plausible modes and are therefore able to generate trajectories lying close to the ground truth.

**Stanford Drone Dataset.** We compare our model against other baseline methods on the SDD and report ADE and FDE in pixel space. As it can be seen in Table 2, Goal-GAN achieves state-of-the-art results on both metrics, ADE and FDE. Comparing Goal-GAN against the best non-goal-conditioned method, *SoPhie*, Goal-GAN decreases the error by 25%. This result shows clearly the merit of having a two-stage process of predicting a goal estimate over standard generator methods using only soft attention modules but does not explicitly condition their model on a future goal. Further, it can be understood that multimodal trajectory predictions play a major role in the scenes of the SDD. Also, Goal-GAN exceeds all other goal-conditioned methods and is on par with P2TIRL (which was not yet published during the preparation of this work).

Table 2: Quantitative results for the Stanford Drone Dataset (SDD) [19] of Goal-GAN and baseline models predicting 12 future timesteps. We report ADE and FDE in pixels.

| | Baseline | | | | | | | | Ours |
|---|---|---|---|---|---|---|---|---|---|
| | S-LSTM [7] | S-GAN [4] | CAR-NET [11] | DESIRE [2] | SoPhie [5] | CGNS [35] | CF-VAE [36] | P2TIRL [37] | Goal GAN |
| K | 1 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| **ADE** | 57.0 | 27.3 | 25.7 | 19.3 | 16.3 | 15.6 | 12.6 | 12.6 | **12.2** |
| **FDE** | 31.2 | 41.4 | 51.8 | 34.1 | 29.4 | 28.2 | 22.3 | **22.1** | **22.1** |

### 5.2 Assessing Multimodality of Predictions on Synthetic Dataset

In this section, we conduct an additional experiment using synthetically generated scenarios to study the multimodality of the predictions. We compare the performance of Goal-GAN against two vanilla GAN baselines, with and without visual soft attention [30]. The synthetic dataset allows us to explicitly control multimodality and feasibility of the (generated) ground truth trajectories, as the other datasets do not provide that information.

**Dataset.** We generate trajectories using the Social Force Model [12] in the *hyang 4* scene of the SDD dataset [19]. To ensure the feasibility of the generated trajectories, we use a two-class (manually labeled) semantic map, that distinguishes between feasible (walking paths) from unfeasible (grass) areas. We simulate 250 trajectories approaching and passing the two crossroads in the scene.

**Additional Evaluation Measures.** In addition to ADE and FDE, we follow [26,44] to measure the multimodality of the distribution of generated trajec-

Table 3: Quantitative results on our synthetic dataset. We show results obtained by a GAN baseline [4] and different versions of our Goal-GAN model, predicting 12 future time steps. We report ADE, FDE, F (feasibility) and MC (mode coverage) for $k = 10$ sampled trajectories for each scene. We also report the negative log-likelihood (NLL) of the ground truth trajectory computed with the KDE (Kernel Density Estimate), following [26].

| Model | Loss | ADE ↓ | FDE ↓ | F ↑ | MC ↑ | NLL ↓ |
|---|---|---|---|---|---|---|
| GAN w/o visual | $\mathcal{L}_{L2} + \mathcal{L}_{Adv}$ | 0.70 | 1.49 | 59.94 | 78.51 | 4.54 |
| GAN w visual | $\mathcal{L}_{L2} + \mathcal{L}_{Adv}$ | 0.68 | 1.27 | 66.51 | 85.12 | 4.47 |
| Goal-GAN | $\mathcal{L}_{GCE} + \mathcal{L}_{G}$ | 2.09 | 1.27 | 76.78 | 88.22 | **3.76** |
| Goal-GAN | $\mathcal{L}_{L2} + \mathcal{L}_{GCE} + \mathcal{L}_{G}$ | 0.62 | 1.20 | 85.05 | 89.27 | 3.90 |
| Goal-GAN w/o GST | $\mathcal{L}_{L2} + \mathcal{L}_{Adv} + \mathcal{L}_{GCE} + \mathcal{L}_{G}$ | 0.84 | 1.45 | 76.84 | 86.27 | 4.18 |
| Goal-GAN (full model) | $\mathcal{L}_{L2} + \mathcal{L}_{Adv} + \mathcal{L}_{GCE} + \mathcal{L}_{G}$ | **0.55** | **1.01** | **89.47** | **92.48** | 3.88 |

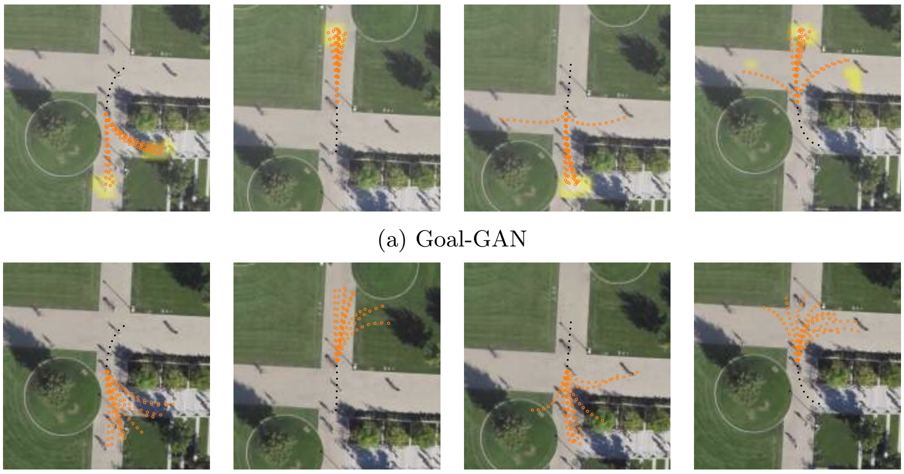
tories. Here we evaluate the negative log-likelihood (NLL) of the ground truth trajectories using a Kernel Density Estimate (KDE) from the sampled trajectories at each prediction timestep. In addition, we define a new mode coverage (MC) metric. For each scene, MC assesses if at least one of the $k$ generated trajectories $\hat{y}$ reaches the final position of the ground truth final up to a distance of $2m$:

$$\mathrm{MC} = \frac{1}{n}\sum_{i}^{n} S\left(\hat{\mathbf{y}}_i\right) \text{ with } S\left(\hat{\mathbf{y}}\right) = \begin{cases} 1 & \text{if } \exists k,\ \|\hat{y}^k - y\|_2 < 2m \\ 0 & \text{else.} \end{cases} \tag{5}$$

To evaluate the feasibility of the trajectories, we report the ratio of trajectories lying inside the feasible area $\mathcal{F}$, *i.e.*, predictions staying on the path:

$$\mathrm{F} = \frac{1}{n}\sum_{i,k}^{n} f\left(\hat{y}_i^k\right) \text{ with } f\left(\hat{y}\right) = \begin{cases} 1 & \text{if } \hat{y} \in \mathcal{F} \\ 0 & \text{else.} \end{cases} \tag{6}$$

**Results.** As can be seen in Table 3, the vanilla GAN baseline [4] that is not given access to the visual information, yields ADE/FDE of 0.70/1.49, respectively. Adding visual information yields a performance boost (0.68/1.27), however, it is still not able to generate multimodal and feasible paths. When we add our proposed goal module (Goal-GAN) and train it using our full loss, we observe a large boost of performance w.r.t. multimodality (7.36 increase in terms of MC) and feasibility (10.26 increase in terms of F). To ablate our model, we train our network using different loss components, incentivizing the network to train different modules of the network. A variant of our model, trained only with goal achievement loss $\mathcal{L}_G$ and adversarial loss $\mathcal{L}_{Adv}$ can already learn to produce multimodal trajectories (MC of 88.22), however, yields a high ADE error of 2.09. The addition of $L_2$ loss $\mathcal{L}_{L2}$ significantly increases the accuracy of the predictions (1.47 reduction in ADE), and at the same time, increases the quality and feasibility (8.26 increase in F), of the predictions. This confirms that our proposed goal module, which explicitly models the distribution over the future goals, is vital for accurate and realistic predictions. Furthermore, we note that the performance drastically drops if we train the full model without the Gumbel-Softmax Trick (GST) (see Section 4.2) which seems to be crucial for stable training, enabling the loss back-propagation through the stochastic sampling process in the Goal Module.



(a) Goal-GAN

(b) Vanilla GAN

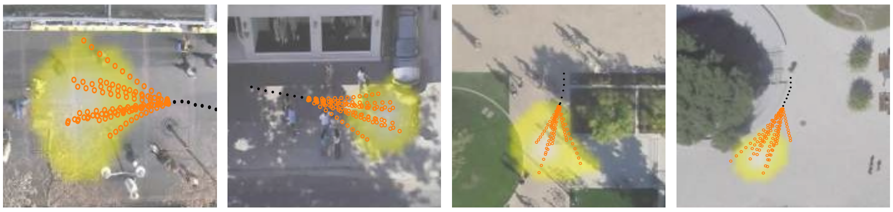
Fig. 4: Visualisation of multiple generated trajectories (orange) for past trajectory (black) on the synthetic dataset. We compare the output of our Goal-GAN against the performance of the vanilla GAN using visual attention for $t_{pred} = 12$. For Goal-GAN, the yellow heatmap corresponds to the goal probability map.

### 5.3 Qualitative Evaluation

In this section, we visually inspect trajectories, generated by our model, and assess the quality of the predictions.

**Synthetic Dataset:** In Figure 4 we visualize trajectories of the synthetic dataset for our proposed Goal-GAN (top) and the vanilla GAN baseline [4] (bottom). Next to the predicted trajectories (orange circles), we display the probability distribution (yellow heatmap) of goal positions, estimated by the Goal Module. As shown in Figure 4, Goal-GAN predicts a diverse set of trajectories routing to specific estimated modes. Here, we observe that Goal-GAN outputs an interpretable probability distribution that allows us to understand where the model “sees“ the dominant modes in the scene. Comparing the quality of the predictions, we can demonstrate that Goal-GAN produces distinct modes while the

GAN baseline tends to instead span its trajectory over a wider range leading to unfeasible paths.



(a) Hotel (b) Zara 2 (c) Hyang 4 (d) Coupa 1

Fig. 5: Visualisation of generated trajectories (orange circles) and estimated global goal probabilities (yellow heatmap). The figures show that the model interacts with the visual context of the scene and ensures feasibility predictions.

**Real Data:** Furthermore, we present qualitative results of the datasets ETH/UCY and SDD in Figure 5. The two figures show predictions on the *Hotel* (Figure 5a) and *Zara 2* (Figure 5b) sequences. Our model assigns high probability to a large area in the scene as in *Hotel* sequence, as several positions could be plausible goals. The broad distribution ensures that we generate diverse trajectories when there are no physical obstacles. Note that the generated trajectories do not only vary in direction but also in terms of speed. In *Zara 2*, the model recognizes the feasible area on the sidewalk and predicts no probability mass on the street or in the areas covered by the parked cars. In the scene *Hyang 4* SDD dataset, we observe that the model successfully identifies that the pedestrian is walking on the path, assigning a very low goal probability to the areas, overgrown by the tree. This scenario is also presented successfully with synthetic data which shows that we can compare the results of the synthetic dataset to the behavior of real data. The trajectories shown for *Coupa 1* demonstrate that the model generates solely paths onto concrete but avoids predictions leading towards the area of the tree.

## 6 Conclusion

In this work, we present Goal-GAN, a novel two-stage network for the task of pedestrian trajectory prediction. With the increasing interest in the interpretability of data-driven models, Goal-GAN allows us to comprehend the different stages iduring the prediction process. This is an alternative to the current generative models, which use a latent noise vector to encourage multimodality and diversity of the trajectory predictions. Our model achieves state-of-the-art results on the ETH, UCY, and SDD datasets while being able to generate multimodal, diverse, and feasible trajectories, as we experimentally demonstrate.

## References

1. Bhattacharyya, A., Schiele, B., Fritz, M.: Accurate and diverse sampling of sequences based on a “best of many” sample objective. In: Conference on Computer Vision and Pattern Recognition. (2018)
2. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Conference on Computer Vision and Pattern Recognition. (2017)
3. Felsen, P., Lucey, P., Ganguly, S.: Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In: European Conference on Computer Vision. (2018)
4. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In: Conference on Computer Vision and Pattern Recognition. (2018)
5. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In: Conference on Computer Vision and Pattern Recognition. (2019)
6. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., Savarese, S.: Social-BiGAT: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: Neural Information Processing Systems. (2019)
7. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Conference on Computer Vision and Pattern Recognition. (2016)
8. Bellmund, J.L.S., Gärdenfors, P., Moser, E.I., Doeller, C.F.: Navigating cognition: Spatial codes for human thinking. Science (2018)
9. Jang, E., Gu, S., Poole, B.: Categorical Reparameterization with Gumbel-Softmax. arXiv e-prints (2016) arXiv:1611.01144
10. Ridel, D.A., Deo, N., Wolf, D.F., Trivedi, M.M.: Scene compliant trajectory forecast with agent-centric spatio-temporal grids. IEEE Robotics Autom. Lett. (2020)
11. Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., Savarese, S.: CAR-Net: Clairvoyant attentive recurrent network. In: European Conference on Computer Vision. (2018)
12. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E (1995)
13. Scovanner, P., Tappen, M.: Learning pedestrian dynamics from the real world. In: International Conference on Computer Vision. (2009)
14. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You’ll never walk alone: Modeling social behavior for multi-target tracking. In: International Conference on Computer Vision. (2009)
15. Yamaguchi, K., Berg, A., Ortiz, L., Berg, T.: Who are you with and where are you going? In: Conference on Computer Vision and Pattern Recognition. (2011)
16. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: International Conference on Computer Vision Workshop. (2011)
17. Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: Conference on Computer Vision and Pattern Recognition. (2014)
18. Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K.: Online multi-target tracking using recurrent neural networks. In: International Conference on Computer Vision. (2017)

19. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European Conference on Computer Vision. (2016)
20. E. Rumelhart, D., E. Hinton, G., J. Williams, R.: Learning representations by back propagating errors. Nature (1986)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997)
22. Hiroaki Minoura, Tsubasa Hirakawa, T.Y., Fujiyoshi, H.: Path predictions using object attributes and semantic environment. In: International Conference on Computer Vision Theory and Applications. (2019)
23. Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: Conference on Computer Vision and Pattern Recognition. (2019)
24. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations. (2014)
25. Deo, N., Trivedi, M.M.: Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. Intelligent Vehicles Symposium (2018)
26. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: International Conference on Computer Vision. (2019)
27. Rhinehart, N., McAllister, R., Kitani, K., Levine, S.: Precog: Prediction conditioned on goals in visual multi-agent settings. In: International Conference on Computer Vision. (2019)
28. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Neural Information Processing Systems. (2014)
29. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In: Conference on Computer Vision and Pattern Recognition Workshop. (2019)
30. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning. (2015)
31. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Neural Information Processing Systems. (2017)
32. Rehder, E., Kloeden, H.: Goal-directed pedestrian prediction. In: International Conference on Computer Vision Workshop. (2015)
33. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
34. Best, G., Fitch, R.: Bayesian intention inference for trajectory prediction with an unknown goal destination. In: International Conference on Intelligent Robots and Systems. (2015)
35. Li, J., Ma, H., Tomizuka, M.: Conditional generative neural system for probabilistic trajectory prediction. In: International Conference on Intelligent Robots and Systems. (2019)
36. Bhattacharyya, A., Hanselmann, M., Fritz, M., Schiele, B., Straehle, C.N.: Conditional flow variational autoencoders for structured sequence prediction. In: Neural Information Processing Systems. (2019)
37. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv e-prints (2020) arXiv:2001.00735

38. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer Assisted Intervention. (2015)
39. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: International Conference on Computer Vision. (2016)
40. Pellegrini, S., Ess, A., Gool, L.V.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: European Conference on Computer Vision. (2010)
41. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. Comput. Graph. Forum (2007)
42. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)
43. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Conference on Computer Vision and Pattern Recognition. (2016)
44. Thiede, L.A., Brahma, P.P.: Analyzing the variety loss in the context of probabilistic trajectory prediction. In: International Conference on Computer Vision. (2019)

# MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction

**MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction**
**Patrick Dendorfer***, Sven Elflein*, and Laura Leal-Taixé
*International Conference on Computer Vision (ICCV)*. 2021.
(* denotes equal contribution)

Following the IEEE reuse permissions, we include the *accepted* version of the publication.

# MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction

Patrick Dendorfer* Sven Elflein* Laura Leal-Taixé
Technical University Munich
{patrick.dendorfer,sven.elflein,leal.taixe}@tum.de

## Abstract

*Pedestrian trajectory prediction is challenging due to its uncertain and multimodal nature. While generative adversarial networks can learn a distribution over future trajectories, they tend to predict out-of-distribution samples when the distribution of future trajectories is a mixture of multiple, possibly disconnected modes. To address this issue, we propose a multi-generator model for pedestrian trajectory prediction. Each generator specializes in learning a distribution over trajectories routing towards one of the primary modes in the scene, while a second network learns a categorical distribution over these generators, conditioned on the dynamics and scene input. This architecture allows us to effectively sample from specialized generators and to significantly reduce the out-of-distribution samples compared to single generator methods.*

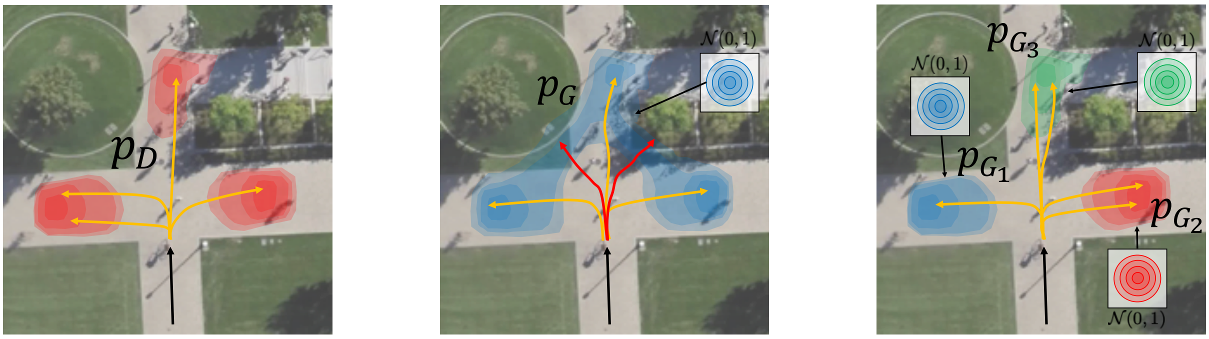
## 1. Introduction

To safely navigate through crowded scenes, intelligent agents such as autonomous vehicles or social robots need to anticipate human motion. Predicting human trajectories is particularly difficult because future actions are multimodal: given a past trajectory, there exist several plausible future paths, depending on the scene layout and social interactions among pedestrians. Recent methods leverage conditional generative adversarial networks (GANs) [14, 16, 34, 22] to learn a distribution over trajectories. These methods present significant improvements over deterministic models [1, 18]. However, they suffer from limitations observed in the context of GANs [38, 20] that manifest in mode collapse or prediction of undesired out-of-distribution (OOD) samples, effectively yielding non-realistic trajectories. Mode collapse can be tackled with best-of-many sampling [6] or regularizations of the latent space [22, 2] but the problem of OOD samples remains unsolved. These OOD samples are particularly problematic in real-world applications where high precision of predictions matters. Imagine an autonomous vehicle driving through crowded environments and interacting with pedestrians. To ensure the safety of pedestrians, the vehicle needs to anticipate their future motion and react accordingly, *e.g.*, brake or turn. As a consequence, unrealistic predictions may lead to sudden reactions that pose danger to other traffic participants.

To understand why OOD samples are produced by state-of-the-art GAN methods, we need to understand the underlying geometry of the problem. Consider a pedestrian reaching the junction in Figure 1a. There are three plausible main directions that the pedestrian can take, namely, going straight, left, or right. Furthermore, there exist several paths that route towards these directions. While all recent works agree that such trajectory distribution is inherently multimodal, we further observe that the distribution consists of several disconnected modes. Each mode is shown in Figure 1c in different colors, and as we can observe, the three modes are disconnected in space. Existing GAN models do not consider this property, and hence generate undesirable OOD samples in between modes, visualized as red trajectories in Figure 1b. This is an inherent problem of single-generator GANs, as they cannot learn a mapping from a continuous latent space to a disconnected, multimodal target distribution [38].

In this paper, we address this issue and explicitly focus on learning such disconnected multimodal distributions for pedestrian trajectory prediction. To this end, we propose a novel multi-generator GAN that treats the multimodal target distribution as a mixture of multiple continuous trajectory distributions by optimizing a continuous generator for each mode. Unlike previous multi-generator models [19, 7], our model needs to adapt to the selection of generators to different scenes, *e.g.*, two- and three-way junctions. For this, we employ a fixed number of generators and allow the model to learn the necessary number of modes directly from visual scene information. Towards this end, we train a second module estimating the categorical probability distribution over the individual generators, conditioned on the input observations. At test time, we first select a specific gener-

*Equal contribution.



(a) Target Distribution $p_D$

(b) Single Generator distribution $p_G$

(c) Multi-generator distribution

Figure 1: The figure illustrates a pedestrian reaching a junction (black) including (a) the multimodal target distribution of future paths, (b) learned future trajectory distribution by a single generator GAN predicting out-of-distribution samples (red), and (c) learned trajectory distribution of multi-generator mixture model.

ator based on its categorical probability and sample then trajectories specialized to that particular mode present in the scene. For measuring the quality of the predictions, we extend the concept of traditional $L_2$ error measures with a precision and recall metric [36, 23]. Our experimental evaluation shows that our proposed model overcomes state-of-the-art and single-generator methods when comparing the behavior of predicting OOD samples.

We summarize our **main contributions** as follows: (i) we discuss the limitations of single generator GANs and propose a novel multi-generator method that learns a multimodal distribution over future trajectories, conditioned on the visual input. To this end, we (ii) present a model that estimates a conditional distribution over the generators and elaborate a training scheme that allows us to jointly train our model end-to-end. Finally, (iii) we introduce recall and precision metrics for pedestrian trajectory prediction to measure the quality of the entire predictive distribution, and in particular OOD samples. We demonstrate our method's efficiency and robustness through extensive ablations. The source code of the model and experiments is available: https://github.com/selflein/MG-GAN.

## 2. Related Work

**Trajectory Forecasting.** Since its inception, the field of pedestrian trajectory prediction has moved from handcrafted [18] to data-driven [1] methods. While the first learning methods used deterministic LSTM encoder-decoder architectures (S-LSTM [1]), deep generative models [16, 34, 22, 2, 12, 8] quickly emerged as state-of-the-art prediction methods. This development enabled the shift from predicting a single future trajectory to producing a distribution of possible future trajectories. S-GAN [16] establishes a conditional Generative Adversarial Networks [14] to learn the ground-truth trajectory distribution and S-GAN-P [16]and SoPhie [34] extend S-GAN with visual and social interaction components. Further, S-BiGAT [22] increases the diversity of the samples by leveraging bicycle GAN training [42] that encourages the connection between the output and the latent code to be invertible. Goal-GAN [8] circumvents the problem of mode collapse by conditioning the decoder on a goal position estimated based on the topology of the scene.

GANs [14] have well-known issues with mode collapse, this is why many models [16, 34] use an $L_2$ variety loss [6] or modify the GAN objective [2] to encourage diversity of the samples. While producing highly diverse samples ensures coverage of all modes in the distribution, we also obtain many unrealistic out-of-distribution samples. The problem of OOD samples has been remained unnoticed partially due to the evaluation metrics used in the field which only measure the minimum $L_2$ distance between the set of predictions and the ground truth, namely the recall. Nonetheless, the realism of predicted trajectories, equivalent to a precision metric, is seldomly evaluated. We advocate that trajectory prediction methods should be evaluated concerning both of the aforementioned aspects.

Other work uses conditional variational autoencoders (VAE) [21] for multimodal pedestrian trajectory prediction [24, 35, 26, 5]. More recently, Trajectron++ [37] uses a VAE and represents agents' trajectories in a graph-structured recurrent neural network. PECNet [28] proposes goal-conditioned trajectory forecasting. Similar to GANs, VAEs are also continuous transformations and suffer from the limitations of generating distributions on disconnected manifolds [32].

Lastly, P2TIRL [9] learns a grid-based policy with maximum entropy inverse reinforcement learning. In summary, existing methods pay little attention to the resulting emergence of out-of-distribution samples and do not discuss the topological limitation in learning a distribution on disconnected supports.

**Generation of Disconnected Manifolds.** Understanding the underlying geometry of the problem is important when training deep generative models [11]. More precisely, learning disconnected manifolds requires disconnectedness within the model. A single generator preserves the topology of the continuous latent space and cannot exclusively predict samples on disconnected manifolds [38].

For image generation, the problem of multimodal learning is well-known and widely studied. Addressing this issue, [38] proposes a rejection sampling method based on the norm of the generator's Jacobian. InfoGAN [7] discretizes the latent space by introducing extra dimensions. Other works use mixtures of generators [40, 39, 41, 17, 13, 4] to construct a discontinuous function. However, these models assume either a uniform or unconditional probability for their discrete latent code or generators. As a result, these methods are unable to adapt to different scenes and thus unsuitable for the trajectory prediction task.

Our research is the first to address the problem of learning disconnected manifolds using multiple generators for the task of pedestrian trajectory prediction by modeling a conditional distribution over the generators.

## 3. Problem Definition

In this work, we tackle the problem of jointly predicting future trajectories for all pedestrians in the scene. For each pedestrian $i$, we generate a set of $K$ future trajectories $\{\hat{Y}_i^k\}_{k=1,...,K}$ with $t \in [t_{obs}+1, t_{pred}]$ for a given input trajectory $X_i$ with $t \in [t_1, t_{obs}]$. This implies learning the true distribution of trajectories conditioned on the input trajectories and scene layout.

In many real-world scenarios such as in Figure 1, the target distribution $p_D$ is multimodal and composed of disconnected modes.

**Why do Single Generator GANs produce OOD Samples?** State-of-the-art methods use the standard conditional GAN architecture [14] and its modifications [29, 3] to learn a distribution over future trajectories. These models learn a continuous mapping $G : \mathcal{X} \times \mathcal{Z} \rightarrow \mathcal{Y}$ from the latent space $\mathcal{Z}$ combined with the observations' space $\mathcal{X}$ to the space of future trajectories $\mathcal{Y}$. The probability prior $p(z)$ on $\mathcal{Z}$ is mainly a standard multivariate normal distribution with $z \sim \mathcal{N}(0, 1)$. When modeling $G$ with a neural network, the mapping is continuous and preserves the topology of the space. Hence, the transformation $G(x, \mathcal{Z})$ of the support of the probability distribution $\mathcal{Z}$ is connected in the output space [38]. Therefore, theoretic work [38, 20] discusses that learning a distribution on disconnected manifolds is impossible; we also observe this phenomenon in our experiments.

**Why are OOD Samples problematic?** Real world-applications relying on trajectory predictions, *e.g.* autonomous vehicles, have to treat every prediction as a possible future scenario and need to adjust their actions accordingly. Thus, not only missed but also unrealistic predictions may crucially hurt the performance of those applications. As OOD samples without support in the ground-truth distribution are likely to be unrealistic, we aim to keep their number small while still covering all modes.

**How can we prevent OOD Samples?** All single generator models will predict OOD if the target distribution lies on disconnected manifolds. Theoretically, there are only two ways to achieve disconnectedness in $\mathcal{Y}$: making $\mathcal{Z}$ disconnected or making the generator mapping $G : \mathcal{X} \times \mathcal{Z} \rightarrow \mathcal{Y}$ discontinuous. We discuss both approaches in our paper but find the latter to be more effective.

**How to measure OOD Samples?** Best-of-many $L_2$ distance metrics focus on minimizing the error between a single sample out of a set of predictions without assessing the quality of the remaining trajectories. Therefore, we compare our model on both, recall and precision [36, 23], which are commonly used to assess the quality of generative models. While existing distance measures highly correlate with recall, we are equally interested in precision that correlates with the number of OOD samples.

## 4. Method

In this section, we present our multi-generator framework for pedestrian trajectory prediction. Our model learns a discontinuous function as a mixture of distributions modeled by multiple generators (Section 4.1).

To adapt to different scenes, we train a second network estimating the categorical distribution over generators (Section 4.2) for new unseen scenes.

### 4.1. MG-GAN

**Visual and Trajectory Encoders.** We outline the architecture of our model in Figure 2. First, the feature encoders extract visual and dynamic features $d_i$ from the input sequences $X_i$ and scene image patches $I_i$ of each pedestrian $i$. The attention modules combine these encodings to compute the physical attention [34] features $v_i$ and social attention [2] features $s_i$. After the encoding and attention, we concatenate the dynamic $d_i$, physical $v_i$, and social $s_i$ features to $c_i = [d_i, v_i, s_i]$. In the following, we omit the index indicating individual pedestrians to avoid notation clutter. Note that we leverage established modules to model physical and social interactions [34, 2, 16], as our contribution is the multi-generator framework. We provide more details on these components in the supplementary.

**Multi-generator Model.** In our model, we leverage $n_G$ different generators $\{G_g\}$, where each generator specializes in learning a different trajectory distribution conditioned on

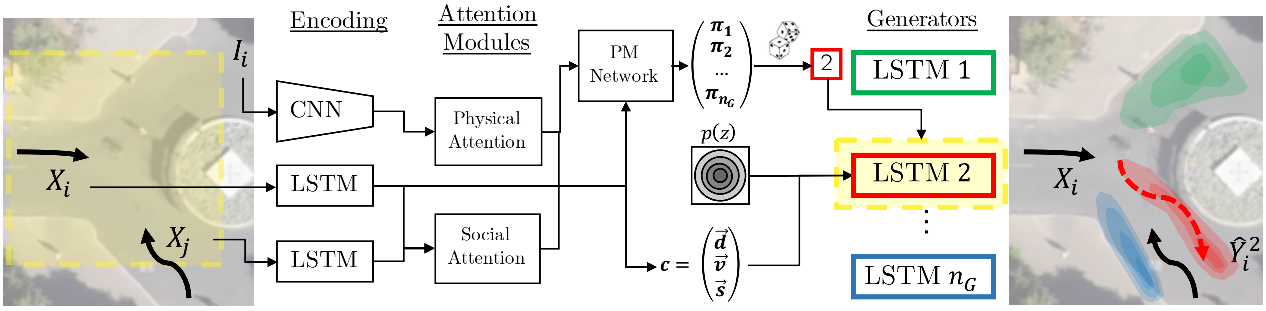


Figure 2: Architecture of MG-GAN. The scene image $I_i$ and observed trajectories $X$ are encoded and passed to the physical and social attention modules. The $n_G$ generators can predict different conditional trajectory distributions for the given scene observation. The PM-Net estimates probabilities $\boldsymbol{\pi}$ for the generators. The model samples or selects a generator from $\boldsymbol{\pi}$ and predicts a trajectory $\hat{Y}$ conditioned on the features $c$ and the noise vector $z$.

the input $c$. All generators share the same network architecture, however, they do not share weights. The generator architecture consists of a LSTM decoder, initialized with the features $c$ and a random noise vector $z \sim \mathcal{N}(0, 1)$ as the initial hidden state $h^0$. The final trajectory $\hat{Y}$ is then predicted recurrently:

$$\Delta\hat{Y}^t = \text{LSTM}_g\left(\Delta X^{t-1}, h^{t-1}\right). \tag{1}$$

Existing multi-generator modules proposed in the context of image generation assume the distribution over the generators to be constant [19, 17]. However, in the case of trajectory prediction, the number of modes is unknown a priori. Therefore, we train a module that adapts to the scene by activating specific generators, conditioned on the observations and interactions $c$.

## 4.2. Path Mode Network (PM-Net)

The Path Mode Network (PM-Net) parameterizes a distribution over the indices of the generators $p(g|c) = [\pi_1, \cdots, \pi_{n_G}]$ conditioned on the features $c$ and is modelled with a multi-layer perceptron MLP$(c)$. The outputs $\{\pi_g\}$ assign probabilities to each of the $n_G$ generators. During inference, we can sample different generators based on the predicted distribution. Note, that this provides a major advantage over existing methods [19, 20], where the distribution is fixed and cannot adjust to different scenes. In comparison, our PM-Net is capable of selecting the relevant generators for a given scene while deactivating unsuitable ones.

## 4.3. Model Training

We now present a training algorithm that jointly optimizes the distribution over generators parameterized by PM-Net and the multi-generator GAN model. For this, we propose an alternating training scheme, inspired by expectation-maximization [15, 20].

### 4.3.1 GAN Training

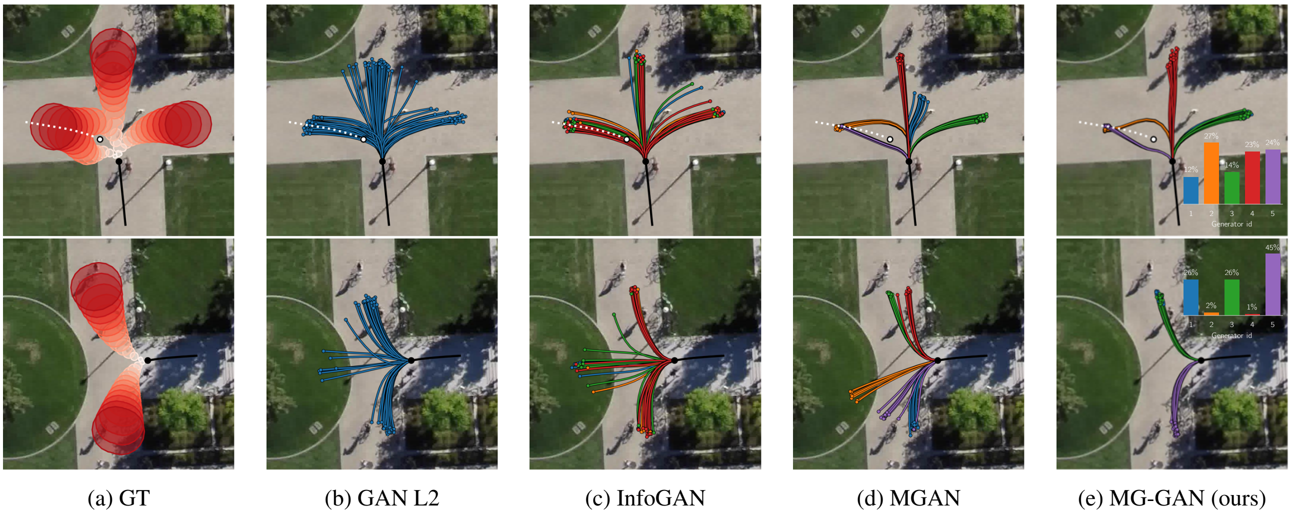
We train our model using a conditional generator, discriminator network $D$ [14] that distinguishes between real and fake trajectories and a classifier $C$ [19] learning to identify which generator predicted a given trajectory. More details on these networks' architectures can be found in the supplementary.

**Adversarial Loss.** We define each generator $G_g$ as $\hat{Y}_{g,z} = G_g(c, z)$ inducing an implicit distribution $p_{G_g}(\hat{Y}|c)$. All $n_G$ generators together describe a joint probability distribution $\sum_{g=1}^{n_G} \pi_g p_{G_g}(\hat{Y}|c)$, thus the established results [14] for GANs hold. We use the original adversarial loss $\mathcal{L}_{Adv}$ [14]. The discriminator $D$ learns to distinguish between real samples $Y$ and samples $\hat{Y}$ generated by the model encouraging realism of the predictions. However, $D$ by itself does not prevent the generators from collapsing to the same mode.

**Classification Loss.** To incentivize the generators to cover different, possibly distinct modes occupying different regions of the output space, we follow [19] and introduce a classifier $C$ which aims to identify the generator index $g$ that generated a sample $\hat{Y}_{g,z}$. The cross-entropy loss $\mathcal{L}_{Cl}$ between the classifier output and the true generator label of the predicted trajectory encourages the generators to model non-overlapping distributions and drives the trajectories of different generators spatially apart. This behavior is regularized through the adversarial loss $\mathcal{L}_{Adv}$ that constrains the samples to be realistic and not diverge from the real distribution. Overall, the training object reads as follows

$$\min_G \max_D \mathcal{L}_{Adv} + \lambda_{Traj}\mathcal{L}_{Traj} + \lambda_{Cl}L_{Cl}, \tag{2}$$

where we additionally apply a $L_2$ best-of-many loss [6, 16] $\mathcal{L}_{Traj}$ with $q$ samples to increase the diversity of predicted trajectories. $\lambda_{Traj}$ and $\lambda_{Cl}$ are weighting hyperparameters.



(a) GT (b) GAN L2 (c) InfoGAN (d) MGAN (e) MG-GAN (ours)

Figure 3: Predicted trajectories for two scenarios in the synthetic dataset. The upper row contains scene on a junction with 3 modes and an interacting pedestrian (white). The lower row shows a scenario with two modes. Figures (a) represent the support of the conditional multimodal ground-truth distributions for these scenes. Figures (e) of MG-GAN also show the probabilities $\boldsymbol{\pi}$ of the PM Network. We visualize trajectories of one generator/discrete latent variable in the same color.

#### 4.3.2 PM-Net Training

To train PM-Net, we approximate the likelihood of a particular generator distribution $p_{G_g}$ supporting the trajectory $Y$ by the generated trajectories $\hat{Y}_{g,c,z_i} = G_g(c, z_i)$ as:

$$p(Y|c,g) \propto \frac{1}{l}\sum_{i=1}^{l} \exp\left(\frac{-\left\|\hat{Y}_{g,c,z_i} - Y\right\|_2^2}{2\sigma}\right). \tag{3}$$

Here, we marginalize the GAN noise $z$ and assume a normally distributed and additive error $\epsilon \sim N(0, \sigma I)$ between $\hat{Y}$ and $Y$ as common for regression tasks [10]. We obtain the conditional probability over generators by applying Bayes' rule:

$$p(g|c,Y) = \frac{p(Y|c,g)}{\sum_{g'}^{n_G} p(Y|c,g')}. \tag{4}$$

Finally, we optimize the PM-Net with the approximated likelihood minimizing the cross entropy loss:

$$\mathcal{L}_\Pi = H(p(g|c,Y), \Pi(c)). \tag{5}$$

Intuitively, the network is trained to weigh the generator that generates trajectories closest to the ground-truth sample the highest. We provide the full derivation of the objective in the supplementary.

#### 4.3.3 Alternating Training Scheme

Our training scheme consists of two alternating steps similar to an expectation-maximization algorithm [15]:

**1. PM-Net Training Step:** We sample $l$ trajectories for each generator and optimize the parameters of PM-Net using Equation (5) while keeping the rest of the network's parameters fixed.

**2. Generator Training Step:** In the generator training step, we use PM-Net to generate probabilities $\boldsymbol{\pi}$ and sample $q$ generators predicting trajectories. With these predictions, we update the model excluding PM-Net optimizing Equation (2). We provide pseudo-code detailing our training procedure in the supplementary.

### 4.4. Trajectory Sampling

We can use the estimated probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{n_G}]$ generated by the PM-Net to establish different mechanisms to sample trajectories from the multiple generators. This helps us to cover all modes present in the scene with as-few-as-possible predictions. In single generator models [22, 34] the relation between regions in the Gaussian latent space and different modes in the output space is implicit and unknown. However, for MG-GAN we can use the estimated probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{n_G}]$ from the PM-Net to control and to cover predictions for all modes present in the scene. Next to randomly sampling $k$ trajectories (*Random*) from $\boldsymbol{\pi}$ we introduce an additional strategy (*Expectation*) where we compute the expected number of samples for each generator as $n_g = k \cdot \pi_g$. We round all $n_g$ to the nearest integer and adjust the number of the generator with the highest score to ensure that all numbers sum up to $k$.

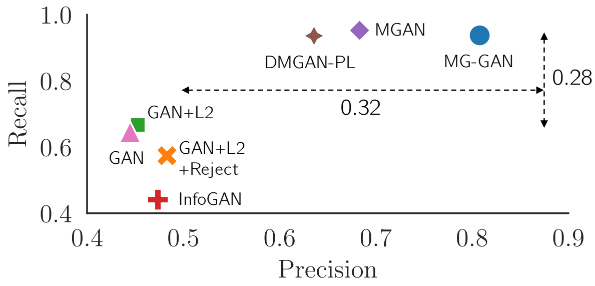


Figure 4: Precision vs. Recall on synthetic dataset.

## 5. Experimental Evaluation

We evaluate our model on four publicly available datasets [30, 25, 31, 27] for pedestrian trajectory prediction and compare our results with state-of-the-art methods. Furthermore, we conduct experiments on synthetic datasets. Compared to real data, synthetic data provides access to the ground-truth trajectory distribution which enables us to identify OOD samples by comparing ground-truth and generated trajectory distributions. Finally, we run an ablation on the individual components of MG-GAN and study the robustness of our model w.r.t. the number generators $n_G$.

### 5.1. Experimental Setup

We follow prior work [31, 1] and observe 8 past time steps (3.2 seconds) and predict the future 12 time steps (4.8 seconds) for every pedestrian in the scene.

**Metrics.** We evaluate results using the following metrics: *Average Displacement Error (ADE)* is defined as a mean $L_2$ distance between the prediction and ground-truth trajectory. *Final Displacement Error (FDE)* is defined as the distance between the prediction and ground-truth trajectory position at time $t_{pred}$.
For both metrics, ADE and FDE, we follow the *Minimum over $k$* procedure [16, 34, 22] with $k = 20$. Note that this approach only considers a single prediction with the lowest ADE and FDE, but not the entirety of the set of $k$ generated output trajectories combined. Therefore, we include additional metrics commonly used in the GAN literature [36, 23], namely *recall* and *precision*. Recall measures the coverage of all ground-truth modes, while precision measures the ratio of generated samples in the support of the ground truth distribution. Hence, the precision is directly related to the number of OOD samples. We also compute the $F1$ score, combining recall and precision.

**Datasets.** We perform the evaluation using the following datasets. ETH [30] and UCY datasets [25] contain five sequences (ETH: ETH and HOTEL, UCY: UNIV, ZARA1, and ZARA2), recorded in four different scenarios. We follow the standard leave-one-out approach for training and testing, where we train on four datasets and test on the remaining one. The Stanford Drone Dataset (SDD) [31] consists of 20 video sequences captured from a top view at the Stanford University campus. In our experiments, we follow the train-test-split of [33] and focus solely on pedestrians. The recently proposed Forking Paths Dataset (FPD) [27] is a realistic 3D simulated dataset providing multi-future trajectories for a single input trajectory. To study the ability of our model to predict multimodal trajectories while preventing OOD samples, we create a synthetic dataset where we simulate multiple possible future paths for the same observation emerging due to the scene layout and social interactions. Detailed information on the generated dataset is provided in the supplementary material.

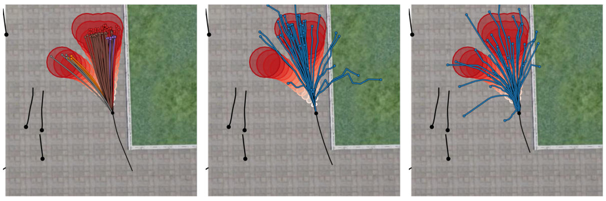


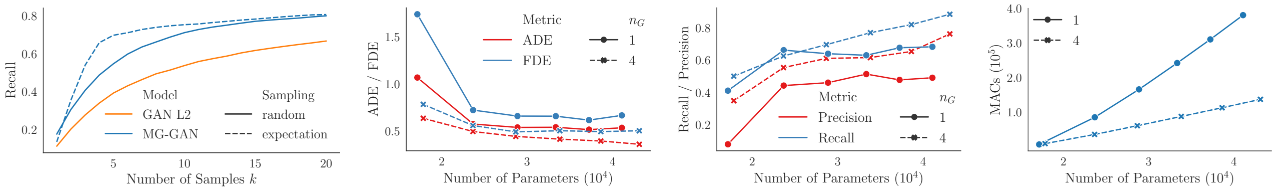
Figure 5: Generated samples of our MG-GAN, Trajectron++, and PECNet.

**Baselines.** We compare our method with several single and multi-generator GAN baselines. We evaluate a (i) vanilla *GAN* baseline, (ii) *GAN L2* trained with variety loss [6], (iii) *GAN L2 Reject* [38] that filters OOD samples based on gradients in the latent space, and (iv) InfoGAN [7] with discrete random latent variable. Furthermore, we compare MG-GAN to multi-generator models MGAN [19] and DMGAN-PL [20], proposed in the context of image generation, that we adapt for the task of trajectory prediction. To ensure comparability, all models use the same base model following SoPhie [34] with attention modules as described in Section 4.1. For qualitative comparison, we evaluate our method against state-of-the-art prediction models presented in Section 2 on the standard benchmarks for trajectory forecasting.

### 5.2. Experiments on Synthetic data

We first study our model on a synthetic dataset in which we have access to the ground-truth distribution of the future trajectories. In this experiment, we show that MG-GAN achieves better performance in learning a multimodal trajectory distribution with disconnected support and is more efficient than the baselines.

**Results.** The results in Figure 4 show that MG-GAN outperforms the single-generator baselines and increases Recall by 0.28 and Precision by 0.32. To this end, we find that all multi-generator methods have a similar recall but MG-GAN achieves a 15% higher Precision corresponding to a lower number of OOD samples.



(a) (b) (c) (d)

Figure 6: Comparison between single generator model *GAN+L2* and MG-GAN. (a) recall for different number of samples $k$ and sampling methods. (b) - (c) compares ADE/FDE, recall/precision, and MACs (Multiply–accumulate operations) for varying total number of model parameters.

**Visual Results.** In Figure 3, we visualize predicted trajectories for two different scenarios where the white trajectory represents another interacting pedestrian. The support of the ground-truth distribution for each timestep is shown as a red circle in Figure 3a. A model achieves low precision in Figure 4 if many trajectory points lie outside the corresponding red circle for a particular timestep. Similarly, a model has high recall if its samples cover most of the area of red circles. Single generator models, GAN+L2 (Figure 3b), and InfoGAN (Figure 3c) produce many OOD samples leading to low precision. In particular, we find that InfoGAN is not able to learn the correspondence between the discrete latent space and the modes in the trajectory space. While theoretically plausible, these results indicate that a discretized latent space is not well-suited for learning distribution on disconnected support. Contrarily, MGAN can learn the distribution but is incapable to adjust generators resulting in OOD samples in Figure 3d when the number of modes does not match the number of generators. Finally, our MG-GAN is able to adjust to both scenarios in Figure 3e as the PM-Net deactivates generators which are unsuitable and prevents OOD samples explaining the high Precision in Figure 4.

**Effective Mode Covering.** Figure 6a shows the recall depending on the number of samples $k$. Our method covers more modes of the ground-truth distribution than the single generator model for the same number of samples as indicated by the higher recall. Additionally, we observe significant improvements compared to random sampling by using expectation sampling leveraging PM-Net as described in Section 4.4, especially for fewer samples.

**Number of Parameters and Computational Cost.** In this experiment, we show that our MG-GAN does not require more resources w.r.t. parameters or computations compared to a single generator baseline. For this, we compare MG-GAN using four generators with the single-generator baseline while keeping the total number of parameters of both models fixed by only using approx. $1/4$ of the parameters for each generator. As can be seen in Figures 6b and 6c, MG-GAN outperforms the single generator GAN w.r.t. to ADE/FDE (50%) and recall/precision (30%) using the same number of total parameters across various parameter budgets. In Figure 6a, the computational cost measured by MACs for the prediction of a trajectory is always lower for MG-GAN compared to the baseline. The model only runs one selected generator with $1/4$ amount of parameters during the forward pass while the cost of running PM-Net is negligible.

| Dataset | S-LSTM [1] | S-GAN [16] | SoPhie [34] | S-BiGAT [22] | CGNS [26] | GoalGAN [8] | PECNet [28] | Trajectron++ [37] | MG-GAN (Ours) |
|---|---|---|---|---|---|---|---|---|---|
| **ETH** | 1.09/2.35 | 0.81/1.52 | 0.70/1.43 | 0.69/1.29 | 0.62/1.40 | 0.59/1.18 | 0.54/0.87 | **0.39/0.83** | 0.47/0.91 |
| **HOTEL** | 0.79/1.76 | 0.72/1.61 | 0.76/1.67 | 0.49/1.01 | 0.70/0.93 | 0.19/0.35 | 0.18/0.24 | **0.12/0.21** | 0.14/0.24 |
| **UNIV** | 0.67/1.40 | 0.60/1.26 | 0.54/1.24 | 0.55/1.32 | 0.48/1.22 | 0.60/1.19 | 0.35/0.60 | **0.20/0.44** | 0.54/1.07 |
| **ZARA1** | 0.47/1.00 | 0.34/0.69 | 0.30/0.63 | 0.30/0.62 | 0.32/0.59 | 0.43/0.87 | 0.22/0.39 | **0.15/0.33** | 0.36/0.73 |
| **ZARA2** | 0.56/1.17 | 0.42/0.84 | 0.38/0.78 | 0.36/0.75 | 0.35/0.71 | 0.32/0.65 | 0.17/0.30 | **0.11/0.25** | 0.29/0.60 |
| **AVG** | 0.72/1.54 | 0.58/1.18 | 0.54/1.15 | 0.48/1.00 | 0.49/0.97 | 0.43/0.85 | 0.29/0.48 | **0.19/0.41** | 0.36/0.71 |

Table 1: Quantitative results on ETH [30] and UCY [25]. We report ADE (↓) /FDE (↓) in meters. Underlined results denote the second best.

| | S-LSTM [1] | S-GAN [16] | CAR-NET [35] | DESIRE [24] | SoPhie [34] | CGNS [26] | CF-VAE [5] | P2TIRL [9] | GoalGAN [8] | PECNet [28] | MG-GAN (4) (Ours) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **ADE** | 57.0 | 27.3 | 25.7 | 19.3 | 16.3 | 15.6 | 12.6 | 12.6 | 12.2 | **10.0** | 13.6 |
| **FDE** | 31.2 | 41.4 | 51.8 | 34.1 | 29.4 | 28.2 | 22.3 | 22.1 | 22.1 | **15.9** | 25.8 |

Table 2: Quantitative results on Stanford Drone Dataset (SDD) [31]. We report ADE and FDE in pixels.

## 5.3. Benchmark Results

In this section, we compare our method to the state-of-the-art on the standard benchmarks ETH [30], UCY [25], and SDD [31], as well as the recently proposed Forking Path Dataset (FPD) [27]. We report the performance of the model with the lowest validation error as we train our method with different numbers of generators $n_G \in \{2, \ldots, 8\}$. We discuss the robustness w.r.t. the number of generators in Section 5.4.

**ADE & FDE.** Our MG-GAN achieves competitive results for the ADE and FDE on the ETH/UCY and Stanford Drone Dataset (SDD) shown in Table 1 and Table 2, respectively. Even though our method does not achieve SOTA performance on the ADE and FDE metrics on these benchmarks, we still argue that our method provides significant improvement to the task. That is since the distance-based $L_2$ measures can be drastically reduced by increasing the variance of the predictions for the price of producing

|  | ADE ↓ | FDE ↓ | Precision ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|---|
| GAN+L2 | 28.81 | 58.37 | 0.55 | 0.87 | 0.67 |
| PECNet | **13.14** | **24.55** | 0.46 | 0.95 | 0.62 |
| Trajectron++ | 13.15 | 32.00 | 0.38 | **0.96** | 0.54 |
| MG-GAN (Ours) | 22.09 | 46.38 | **0.71** | 0.89 | **0.79** |

Table 3: Results on FPD. We report ADE/FDE in pixels.

more OOD samples. A visual comparison of the trajectories produced by Trajectron++ and PECNet in Figure 5 shows that these methods produce high variance predictions without accounting for any constraints in the scene. Contrarily, MG-GAN only predicts trajectories inside the ground-truth manifold (red). While covering all modes, our predictions remain in the support of the ground-truth distribution. To quantify this observation, we compute the recall and precision metrics.

**Precision & Recall.** As ADE and FDE do not consider the quality of the entire generated distribution, we add results using precision/recall metrics [36, 23] on the FPD dataset [27]. This is possible on FPD as it contains multiple feasible, human-annotated ground-truth trajectories.

In Table 3, MG-GAN outperforms *GAN+L2* by 29%, PECNet by 54% and Trajectron++ by 86% in terms of Precision, while the difference in Recall with 0.02, 0.06, and 0.07 points is small. Single generator models predict overly diverse trajectories, thus increasing Recall slightly and reducing ADE/FDE, but produce OOD samples leading to low Precision. These results confirm that MG-GAN is significantly more reliably at predicting paths that align well with the human-annotated future trajectories (high precision), while also covering a similar amount of modes in the scene (high recall). Overall, we conclude that MG-GAN does not match SOTA performance on traditional evaluation metrics in Table 1 and Table 2. However, studying precision and recall reveals that our model can lower the number of OOD and achieves an overall better *F1* than current SOTA methods.

## 5.4. Ablation Studies

In this section, we ablate the key modules of MG-GAN. We emphasize that the goal of the paper is to demonstrate the need and effectiveness of a conditional multi-generator framework for pedestrian trajectory prediction. Hence, the study of attention modules used within our model described in Section 4.1, is not the goal of this work and has been extensively done in prior work [16, 2, 34, 22].

**Effectiveness of Key Modules.** We perform the ablation on our synthetic dataset by removing key components from our final model: multiple generators, the classifier $C$, and the PM-Net in Table 4. Reducing the number of generators to 1 results in a significant drop in performance of almost 50% in recall and 31% in precision.

| M | C | PM | ADE ↓ | FDE ↓ | Precision ↑ | Recall ↑ |
|---|---|---|---|---|---|---|
|  |  |  | 0.94 | 1.58 | 0.46 | 0.48 |
| ✓ |  |  | 0.59 | 0.79 | 0.37 | 0.68 |
| ✓ |  | ✓ | 0.35 | 0.49 | 0.72 | 0.91 |
| ✓ | ✓ |  | 0.37 | 0.53 | 0.73 | 0.91 |
| ✓ | ✓ | ✓ | **0.32** | **0.44** | **0.77** | **0.95** |

Table 4: Ablation experiments: (M) Multi-generator, (C) Classifier, and (PM) Path Mode network.

|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Best |
|---|---|---|---|---|---|---|---|---|
| ADE | 0.37 | 0.38 | 0.38 | 0.39 | 0.37 | 0.36 | 0.37 | 0.36 |
| FDE | 0.72 | 0.74 | 0.75 | 0.76 | 0.71 | 0.71 | 0.72 | 0.70 |

Table 5: Results for $n_G \in \{2, \ldots, 8\}$ on ETH/UCY.

As described in Section 4.1, the classifier $C$ encourages individual generators to specialize and increases precision from 37% to 73%. Similarly, with PM-Net learning a distribution over generators, the precision increases from 37% to 72%. Finally, leveraging PM-Net and classifier $C$, combining the advantages of both, further improves the performance on all considered metrics.

**Robustness over the Number of Generators.** The multimodality over future trajectories depends on social interactions and the scene layout, imposing a significant challenge when choosing the number of generators $n_G$ at training time. To this end, we introduced the PM-Net that learns to activate generators depending on the observed scene features. As can be seen in Table 5, PM-Net successfully makes MG-GAN robust w.r.t. the choice of $n_G$ as results only deviate 7% from the best reported values at maximum.

## 6. Conclusion

In this paper, we addressed the issue of single-generator GAN models for pedestrian trajectory prediction. While existing generative networks learn a distribution over future trajectories, they are fundamentally incapable of learning a distribution consisting of multiple disconnected modes. To overcome this problem, our proposed MG-GAN leverages multiple generators that specialize in different modes and learns to sample from these generators conditioned on the scene observation. We demonstrated the efficacy of MG-GAN at reducing out-of-distribution samples in comparison to the existing state-of-the-art. Finally, we emphasized the importance of precision next to recall metrics and hope to encourage a discussion on preventing OOD in future work.

**Acknowledgements.** This project was funded by the Humboldt Foundation through the Sofja Kovalevskaja Award. We highly thank Aljoša Ošep for helpful discussions, constructive feedback, and proofreading.

## 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 *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 6, 7

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social Ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1, 2, 3, 8

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 2017. 3

[4] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and Equilibrium in Generative Adversarial Nets GANs. In *International Conference on Machine Learning*, 2017. 3

[5] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional Flow Variational Autoencoders for Structured Sequence Prediction. In *Neural Information Processing Systems*, 2019. 2, 7

[6] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a "Best of Many" sample objective. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 4, 6

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems*, 2016. 1, 3, 6

[8] Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé. Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. In *Asian Conference on Computer Vision*, 2020. 2, 7

[9] Nachiket Deo and Mohan M. Trivedi. Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans. *arXiv e-prints*, page arXiv:2001.00735, 2020. 2, 7

[10] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998. 5

[11] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the Manifold Hypothesis. *Journal of the American Mathematical Society*, 2013. 3

[12] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. GD-GAN: Generative Adversarial Networks for Trajectory Prediction and Group Detection in Crowds. In *Asian Conference on Computer Vision*, 2018. 2

[13] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. S. Torr, and P. K. Dokania. Multi-agent Diverse Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Neural Information Processing Systems*, 2014. 1, 2, 3, 4

[15] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Neural Information Processing Systems*, 2017. 4, 5

[16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 4, 6, 7, 8

[17] Hao He, Hao Wang, Guang-He Lee, and Yonglong Tian. Bayesian Modelling and Monte Carlo Inference for GAN. In *International Conference on Learning Representations*, 2019. 3, 4

[18] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51, 1995. 1, 2

[19] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Q. Phung. MGAN: Training Generative Adversarial Nets with Multiple Generators. In *International Conference on Learning Representations*, 2018. 1, 4, 6

[20] Mahyar Khayatkhoei, Maneesh K. Singh, and Ahmed Elgammal. Disconnected Manifold Learning for Generative Adversarial Networks. In *Neural Information Processing Systems*, 2018. 1, 3, 4, 6

[21] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2013. 2

[22] 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 *Neural Information Processing Systems*, 2019. 1, 2, 5, 6, 7, 8

[23] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *Neural Information Processing Systems*, 2019. 2, 3, 6, 8

[24] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2, 7

[25] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by Example. *Comput. Graph. Forum*, 2007. 6, 7

[26] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional Generative Neural System for Probabilistic Trajectory Prediction. In *International Conference on Intelligent Robots and Systems*, 2019. 2, 7

[27] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020. 6, 7, 8

[28] 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*, 2020. 2, 7

[29] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled Generative Adversarial Networks. In *International Conference on Learning Representations*, 2017. 3

[30] S. Pellegrini, Andreas Ess, and L. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision*, 2010. 6, 7

[31] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016. 6, 7

[32] Jason Tyler Rolfe. Discrete Variational Autoencoders. In *International Conference on Learning Representations*, Apr. 2017. 2

[33] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. TrajNet: Towards a Benchmark for Human Trajectory Prediction. *arXiv preprint*, 2018. 6

[34] 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 *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 5, 6, 7, 8

[35] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-Net: Clairvoyant attentive recurrent network. In *European Conference on Computer Vision*, 2018. 2, 7

[36] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Neural Information Processing Systems*, 2018. 2, 3, 6, 8

[37] T. Salzmann, B. Ivanovic, Punarjay Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, 2020. 2, 7

[38] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no GANs land. In *Proceedings of Machine Learning and Systems*, pages 6767–6776, 2020. 1, 3, 6

[39] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. AdaGAN: Boosting generative models. In *Neural Information Processing Systems*, 2017. 3

[40] Yaxing Wang, Lichao Zhang, and Joost Van De Weijer. Ensembles of generative adversarial networks. *arXiv preprint arXiv:1612.00991*, 2016. 3

[41] Peilin Zhong, Yuchen Mo, Chang Xiao, Pengyu Chen, and Changxi Zheng. Rethinking Generative Mode Coverage: A Pointwise Guaranteed Approach. In *Neural Information Processing Systems*, 2019. 3

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Neural Information Processing Systems*, 2018. 2

# MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking

**MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking**
**Patrick Dendorfer**, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé
*International Journal of Computer Vision (IJCV)*. 2020.

International Journal of Computer Vision (2021) 129:845–881
https://doi.org/10.1007/s11263-020-01393-0

# MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking

Patrick Dendorfer[1] · Aljoša Ošep[1] · Anton Milan[2] · Konrad Schindler[3] · Daniel Cremers[1] · Ian Reid[4] · Stefan Roth[5] · Laura Leal-Taixé[1]

Received: 1 May 2020 / Accepted: 14 October 2020 / Published online: 23 December 2020

**Abstract**
Standardized benchmarks have been crucial in pushing the performance of computer vision algorithms, especially since the advent of deep learning. Although leaderboards should not be over-claimed, they often provide the most objective measure of performance and are therefore important guides for research. We present *MOTChallenge*, a benchmark for single-camera Multiple Object Tracking (MOT) launched in late 2014, to collect existing and new data and create a framework for the standardized evaluation of multiple object tracking methods. The benchmark is focused on multiple people tracking, since pedestrians are by far the most studied object in the tracking community, with applications ranging from robot navigation to self-driving cars. This paper collects the first three releases of the benchmark: (i) *MOT15*, along with numerous state-of-the-art results that were submitted in the last years, (ii) *MOT16*, which contains new challenging videos, and (iii) *MOT17*, that extends *MOT16* sequences with more precise labels and evaluates tracking performance on three different object detectors. The second and third release not only offers a significant increase in the number of labeled boxes, but also provide labels for multiple object classes beside pedestrians, as well as the level of visibility for every single object of interest. We finally provide a categorization of state-of-the-art trackers and a broad error analysis. This will help newcomers understand the related work and research trends in the MOT community, and hopefully shed some light into potential future research directions.

**Keywords** Multi-object-tracking · Evaluation · MOTChallenge · Computer vision · MOTA

## 1 Introduction

Evaluating and comparing single-camera multi-target tracking methods is not trivial for numerous reasons (Milan et al. 2013). Firstly, unlike for other tasks, such as image denoising, the ground truth, i.e., the perfect solution one aims to achieve, is difficult to define clearly. Partially visible, occluded, or cropped targets, reflections in mirrors or windows, and objects that very closely resemble targets all impose intrinsic ambiguities, such that even humans may not agree on one particular ideal solution. Secondly, many different evaluation metrics with free parameters and ambiguous definitions often lead to conflicting quantitative results across

Communicated by Daniel Scharstein.

Anton Milan: Work done prior to joining Amazon.

Patrick Dendorfer
patrick.dendorfer@tum.de

Aljoša Ošep
aljosa.osep@tum.de

Anton Milan
antmila@amazon.com

Konrad Schindler
schindler@ethz.ch

Daniel Cremers
cremers@tum.de

Ian Reid
ian.reid@adelaide.edu.au

Stefan Roth
stefan.roth@vision.tu-darmstadt.de

Laura Leal-Taixé
leal.taixe@tum.de

[1] Technical University Munich, Munich, Germany
[2] Amazon Research, Tübingen, Germany
[3] ETH Zürich, Zurich, Switzerland
[4] The University of Adelaide, Adelaide, Australia
[5] Technical University of Darmstadt, Darmstadt, Germany

the literature. Finally, the lack of pre-defined test and training data makes it difficult to compare different methods fairly.

Even though multi-target tracking is a crucial problem in scene understanding, until recently it still lacked large-scale benchmarks to provide a fair comparison between tracking methods. Typically, methods are tuned for each sequence, reaching over 90% accuracy in well-known sequences like PETS (Ferryman and Ellis 2010). Nonetheless, the real challenge for a tracking system is to be able to perform well on a variety of sequences with different level of crowdedness, camera motion, illumination, etc., without overfitting the set of parameters to a specific video sequence.

To address this issue, we released the *MOTChallenge* benchmark in 2014, which consisted of three main components: (1) a (re-)collection of publicly available and new datasets, (2) a centralized evaluation method, and (3) an infrastructure that allows for crowdsourcing of new data, new evaluation methods and even new annotations. The first release of the dataset named *MOT15* consists of 11 sequences for training and 11 for testing, with a total of 11286 frames or 996 seconds of video. 3D information was also provided for 4 of those sequences. Pre-computed object detections, annotations (only for the training sequences), and a common evaluation method for all datasets were provided to all participants, which allowed for all results to be compared fairly.

Since October 2014, over 1,000 methods have been publicly tested on the *MOTChallenge* benchmark, and over 1833 users have registered, see Fig. 1. In particular, 760 methods have been tested on *MOT15*, 1,017 on *MOT16* and 692 on *MOT17*; 132, 213 and 190 (respectively) were published on the public leaderboard. This established *MOTChallenge* as the first standardized large-scale tracking benchmark for single-camera multiple people tracking.

Despite its success, the first tracking benchmark, *MOT15*, was lacking in a few aspects:

- The annotation protocol was not consistent across all sequences since some of the ground truth was collected from various online sources;
- the distribution of crowd density was not balanced for training and test sequences;
- some of the sequences were well-known (e.g., PETS09-S2L1) and methods were overfitted to them, which made them not ideal for testing purposes;
- the provided public detections did not show good performance on the benchmark, which made some participants switch to other pedestrian detectors.

To resolve the aforementioned shortcomings, we introduced the second benchmark, *MOT16*. It consists of a set of 14 sequences with crowded scenarios, recorded from different viewpoints, with/without camera motion, and it covers a diverse set of weather and illumination conditions. Most importantly, the annotations for *all* sequences were carried out by qualified researchers from scratch following a strict protocol and finally double-checked to ensure a high annotation accuracy. In addition to pedestrians, we also annotated classes such as vehicles, sitting people, and occluding objects. With this fine-grained level of annotation, it was possible to accurately compute the degree of occlusion and cropping of all bounding boxes, which was also provided with the benchmark.

For the third release, *MOT17*, we (1) further improved the annotation consistency over the sequences[1] and (2) proposed a new evaluation protocol with public detections. In *MOT17*, we provided 3 sets of public detections, obtained using three different object detectors. Participants were required to evaluate their trackers using all three detections sets, and results were then averaged to obtain the final score. The main idea behind this new protocol was to establish the robustness of the trackers when fed with detections of different quality. Besides, we released a separate subset for evaluating object detectors, *MOT17Det*.

In this work, we categorize and analyze 73 published trackers that have been evaluated on *MOT15*, 74 trackers on *MOT16*, and 57 on *MOT17*.[2] Having results on such a large number of sequences allows us to perform a thorough analysis of trends in tracking, currently best-performing methods, and special failure cases. We aim to shed some light on potential research directions for the near future in order to further improve tracking performance.

In summary, this paper has two main goals:

- To present the *MOTChallenge* benchmark for a fair evaluation of multi-target tracking methods, along with its first releases: *MOT15*, *MOT16*, and *MOT17*;
- to analyze the performance of 73 state-of-the-art trackers on *MOT15*, 74 trackers on *MOT16*, and 57 on *MOT17* to analyze trends in MOT over the years. We analyze the main weaknesses of current trackers and discuss promising research directions for the community to advance the field of multi-target tracking.

The benchmark with all datasets, ground truth, detections, submitted results, current ranking and submission guidelines can be found at:

http://www.motchallenge.net/.

[1] We thank the numerous contributors and users of MOTChallenge that pointed us to issues with annotations.

[2] In this paper, we only consider published trackers that were on the leaderboard on April 17th, 2020, and used the provided set of public detections. For this analysis, we focused on peer-reviewed methods, i.e., published at a conference or a journal, and excluded entries for which we could not find corresponding publications due to lack of information provided by the authors.

## 2 Related work

*Benchmarks and challenges* In the recent past, the computer vision community has developed centralized benchmarks for numerous tasks including object detection (Everingham et al. 2015), pedestrian detection (Dollár et al. 2009), 3D reconstruction (Seitz et al. 2006), optical flow (Baker et al. 2011; Geiger et al. 2012), visual odometry (Geiger et al. 2012), single-object short-term tracking (Kristan et al. 2014), and stereo estimation (Geiger et al. 2012; Scharstein and Szeliski 2002). Despite potential pitfalls of such benchmarks (Torralba and Efros 2011), they have proven to be extremely helpful to advance the state of the art in the respective area.

For single-camera multiple target tracking, in contrast, there has been very limited work on standardizing quantitative evaluation. One of the few exceptions is the well-known PETS dataset (Ferryman and Ellis 2010) addressing primarily surveillance applications. The 2009 version consists of 3 subsets S: S1 targeting person count and density estimation, S2 targeting people tracking, and S3 targeting flow analysis and event recognition. The simplest sequence for tracking (S2L1) consists of a scene with few pedestrians, and for that sequence, state-of-the-art methods perform extremely well with accuracies of over 90% given a good set of initial detections (Henriques et al. 2011; Milan et al. 2014; Zamir et al. 2012). Therefore, methods started to focus on tracking objects in the most challenging sequence, i.e., with the highest crowd density, but hardly ever on the complete dataset. Even for this widely used benchmark, we observe that tracking results are commonly obtained inconsistently, involving using different subsets of the available data, inconsistent model training that is often prone to overfitting, varying evaluation scripts, and different detection inputs. Results are thus not easily comparable. Hence, the questions that arise are: (i) are these sequences already too easy for current tracking methods?, (ii) do methods simply overfit?, and (iii) are existing methods poorly evaluated?

The PETS team organizes a workshop approximately once a year to which researchers can submit their results, and methods are evaluated under the same conditions. Although this is indeed a fair comparison, the fact that submissions are evaluated only once a year means that the use of this benchmark for high impact conferences like ICCV or CVPR remains challenging. Furthermore, the sequences tend to be focused only on surveillance scenarios and lately on specific tasks such as vessel tracking. Surveillance videos have a low frame rate, fixed camera viewpoint, and low pedestrian density. The ambition of *MOTChallenge* is to tackle more general scenarios including varying viewpoints, illumination conditions, different frame rates, and levels of crowdedness.

A well-established and useful way of organizing datasets is through standardized challenges. These are usually in the form of web servers that host the data and through which results are uploaded by the users. Results are then evaluated in a centralized way by the server and afterward presented online to the public, making a comparison with any other method immediately possible.

There are several datasets organized in this fashion: the Labeled Faces in the Wild (Huang et al. 2007) for unconstrained face recognition, the PASCAL VOC (Everingham et al. 2015) for object detection and the ImageNet large scale visual recognition challenge (Russakovsky et al. 2015).

The KITTI benchmark (Geiger et al. 2012) was introduced for challenges in autonomous driving, which includes stereo/flow, odometry, road and lane estimation, object detection, and orientation estimation, as well as tracking. Some of the sequences include crowded pedestrian crossings, making the dataset quite challenging, but the camera position is located at a fixed height for all sequences.

Another work that is worth mentioning is Alahi et al. (2014), in which the authors collected a large amount of data containing 42 million pedestrian trajectories. Since annotation of such a large collection of data is infeasible, they use a denser set of cameras to create the "ground-truth" trajectories. Though we do not aim at collecting such a large amount of data, the goal of our benchmark is somewhat similar: to push research in tracking forward by generalizing the test data to a larger set that is highly variable and hard to overfit.

DETRAC (Wen et al. 2020) is a benchmark for vehicle tracking, following a similar submission system to the one we proposed with *MOTChallenge*. This benchmark consists of a total of 100 sequences, 60% of which are used for training. Sequences are recorded from a high viewpoint (surveillance scenarios) with the goal of vehicle tracking.

*Evaluation* A critical question with any dataset is how to measure the performance of the algorithms. In the case of multiple object tracking, the CLEAR-MOT metrics (Stiefelhagen et al. 2006) have emerged as the standard measures. By measuring the intersection over union of bounding boxes and matching those from ground-truth annotations and results, measures of accuracy and precision can be computed. Precision measures how well the persons are localized, while accuracy evaluates how many distinct errors such as missed targets, ghost trajectories, or identity switches are made.

Alternatively, trajectory-based measures by Wu and Nevatia (2006) evaluate how many trajectories were mostly tracked, mostly lost, and partially tracked, relative to the track lengths. These are mainly used to assess track coverage. The IDF1 metric (Ristani et al. 2016) was introduced for MOT evaluation in a multi-camera setting. Since then it has been adopted for evaluation in the standard single-camera setting in our benchmark. Contrary to MOTA, the ground truth to predictions mapping is established at the level of entire tracks instead of on frame by frame level, and therefore, measures long-term tracking quality. In Sect. 7 we report IDF1 perfor-

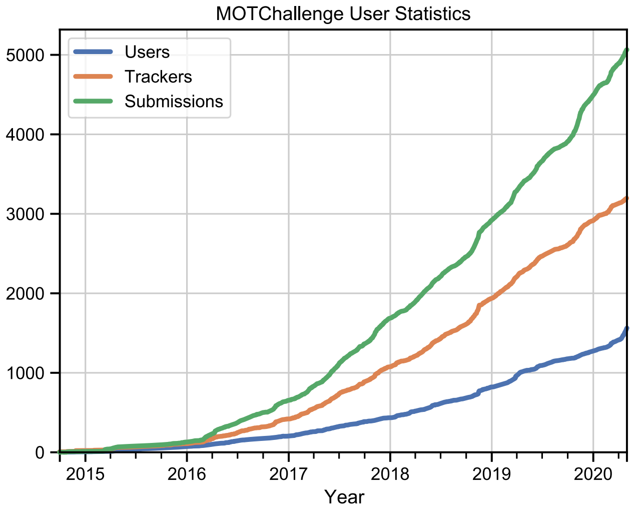


**Fig. 1** Evolution of *MOTChallenge* submissions, number of users registered and trackers created

mance in conjunction with MOTA. A detailed discussion on the measures can be found in Sect. 6.

A key parameter in both families of metrics is the intersection over union threshold which determines whether a predicted bounding box was matched to an annotation. It is fairly common to observe methods compared under different thresholds, varying from 25 to 50%. There are often many other variables and implementation details that differ between evaluation scripts, which may affect results significantly. Furthermore, the evaluation script is not the only factor. Recently, a thorough study (Mathias et al. 2014) on face detection benchmarks showed that annotation policies vary greatly among datasets. For example, bounding boxes can be defined tightly around the object, or more loosely to account for pose variations. The size of the bounding box can greatly affect results since the intersection over union depends directly on it.

Standardized benchmarks are preferable for comparing methods in a fair and principled way. Using the same ground-truth data and evaluation methodology is the only way to guarantee that the only part being evaluated is the tracking method that delivers the results. This is the main goal of the *MOTChallenge* benchmark.

## 3 History of MOTChallenge

The first benchmark was released in October 2014 and it consists of 11 sequences for training and 11 for testing, where the testing sequences have not been available publicly. We also provided a set of detections and evaluation scripts. Since its release, 692 tracking results were submitted to the benchmark, which has quickly become the standard for evaluating multiple pedestrian tracking methods in high impact conferences such as ICCV, CVPR, and ECCV. Together with the release of the new data, we organized the 1st Workshop on Benchmarking Multi-Target Tracking (BMTT) in conjunction with the IEEE Winter Conference on Applications of Computer Vision (WACV) in 2015.[3]

After the success of the first release of sequences, we created a 2016 edition, with 14 longer and more crowded sequences and a more accurate annotation policy which we describe in this manuscript (Sect. C.1). For the release of *MOT16*, we organized the second workshop[4] in conjunction with the European Conference in Computer Vision (ECCV) in 2016.

For the third release of our dataset, *MOT17*, we improved the annotation consistency over the *MOT16* sequences and provided three public sets of detections, on which trackers need to be evaluated. For this release, we organized a Joint Workshop on Tracking and Surveillance in conjunction with the Performance Evaluation of Tracking and Surveillance (PETS) (Ferryman and Ellis 2010; Ferryman and Shahrokni 2009) workshop and the Conference on Vision and Pattern Recognition (CVPR) in 2017.[5]

In this paper, we focus on the *MOT15*, *MOT16*, and *MOT17* benchmarks because numerous methods have already submitted their results to these challenges for several years that allow us to analyze these methods and to draw conclusions about research trends in multi-object tracking.

Nonetheless, work continues on the benchmark, with frequent releases of new challenges and datasets. The latest pedestrian tracking dataset was first presented at the 4th *MOTChallenge* workshop[6] (CVPR 2019), an ambitious tracking challenge with eight new sequences (Dendorfer et al. 2019). With the feedback of the workshop the sequences were revised and re-published as the *MOT20* (Dendorfer et al. 2020) benchmark. This challenge focuses on very crowded scenes, where the object density can reach up to 246 pedestrians per frame. The diverse sequences show indoor and outdoor scenes, filmed either during day or night. With more than 2*M* bounding boxes and 3833 tracks, *MOT20* constitutes a new level of complexity and challenges the performance of tracking methods in very dense scenarios. At the time of this article, only 11 submissions for *MOT20* had been received, hence a discussion of the results is not yet significant nor informative, and is left for future work.

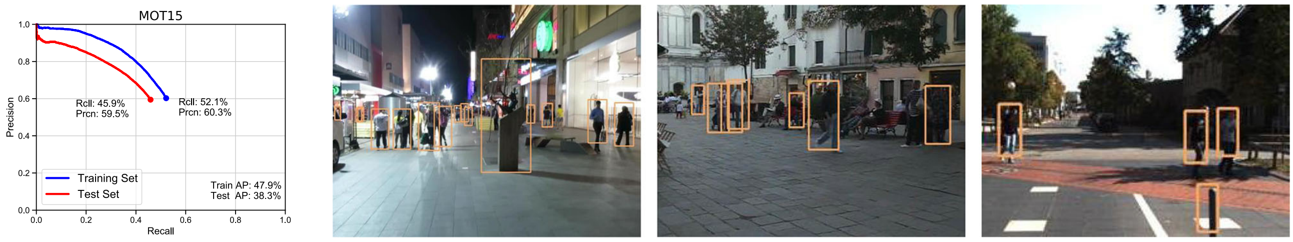
The future vision of *MOTChallenge* is to establish it as a general platform for benchmarking multi-object tracking, expanding beyond pedestrian tracking. To this end, we recently added a public benchmark for multi-camera 3D zebrafish tracking (Pedersen et al. 2020), and a benchmark

[3] https://motchallenge.net/workshops/bmtt2015/.

[4] https://motchallenge.net/workshops/bmtt2016/.

[5] https://motchallenge.net/workshops/bmtt-pets2017/.

[6] https://motchallenge.net/workshops/bmtt2019/.



**(a)** Detection performance of [Dollár et al., 2014] **(b)** ADL-Rundle-8 **(c)** Venice-1 **(d)** KITTI-16

**Fig. 2** **a** The performance of the provided detection bounding boxes evaluated on the training (blue) and the test (red) set. The circle indicates the operating point (i.e., the input detection set) for the trackers. **b**–**d** Exemplar detection results

for the large-scale Tracking any Object (TAO) dataset (Dave et al. 2020). This dataset consists of 2907 videos, covering 833 classes by 17,287 tracks.

In Fig. 1, we plot the evolution of the number of users, submissions, and trackers created since *MOTChallenge* was released to the public in 2014. Since our 2nd workshop was announced at ECCV, we have experienced steady growth in the number of users as well as submissions.

## 4 MOT15 Release

One of the key aspects of any benchmark is data collection. The goal of *MOTChallenge* is not only to compile yet another dataset with completely new data but rather to: (1) create a common framework to test tracking methods on, and (2) gather existing and new challenging sequences with very different characteristics (frame rate, pedestrian density, illumination, or point of view) in order to challenge researchers to develop more general tracking methods that can deal with all types of sequences. In Table 5 of the Appendix we show an overview of the sequences included in the benchmark.

### 4.1 Sequences

We have compiled a total of 22 sequences that combine different videos from several sources (Andriluka et al. 2010; Benfold and Reid 2011; Ess et al. 2008; Ferryman and Ellis 2010; Geiger et al. 2012) and new data collected from us. We use half of the data for training and a half for testing, and the annotations of the testing sequences are not released to the public to avoid (over)fitting of methods to specific sequences. Note, the test data contains over 10 min of footage and 61,440 annotated bounding boxes, therefore, it is hard for researchers to over-tune their algorithms on such a large amount of data. This is one of the major strengths of the benchmark.

We collected 6 new challenging sequences, 4 filmed from a static camera and 2 from a moving camera held at pedestrian's height. Three sequences are particularly challenging: a night sequence filmed from a moving camera and two outdoor sequences with a high density of pedestrians. The moving camera together with the low illumination creates a lot of motion blur, making this sequence extremely challenging. A smaller subset of the benchmark including only these six new sequences were presented at the 1st Workshop on Benchmarking Multi-Target Tracking,[7] where the top-performing method reached MOTA (tracking accuracy) of only 12.7%. This confirms the difficulty of the new sequences.[8]

### 4.2 Detections

To detect pedestrians in all images of the *MOT15* edition, we use the object detector of Dollár et al. (2014), which is based on aggregated channel features (ACF). We rely on the default parameters and the pedestrian model trained on the INRIA dataset (Dalal and Triggs 2005), rescaled with a factor of 0.6 to enable the detection of smaller pedestrians. The detector performance along with three sample frames is depicted in Fig. 2, for both the training and the test set of the benchmark. Recall does not reach 100% because of the non-maximum suppression applied.

We cannot (nor necessarily want to) prevent anyone from using a different set of detections. However, we require that this is noted as part of the tracker's description and is also displayed in the rating table.

### 4.3 Weaknesses of *MOT15*

By the end of 2015, it was clear that a new release was due for the *MOTChallenge* benchmark. The main weaknesses of *MOT15* were the following:

- *Annotations* we collected annotations online for the existing sequences, while we manually annotated the new sequences. Some of the collected annotations were not

[7] https://motchallenge.net/workshops/bmtt2015/.

[8] The challenge results are available at http://motchallenge.net/results/WACV_2015_Challenge/.

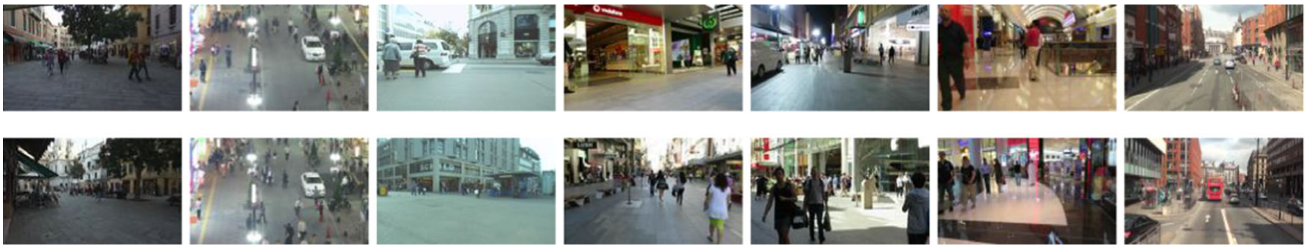
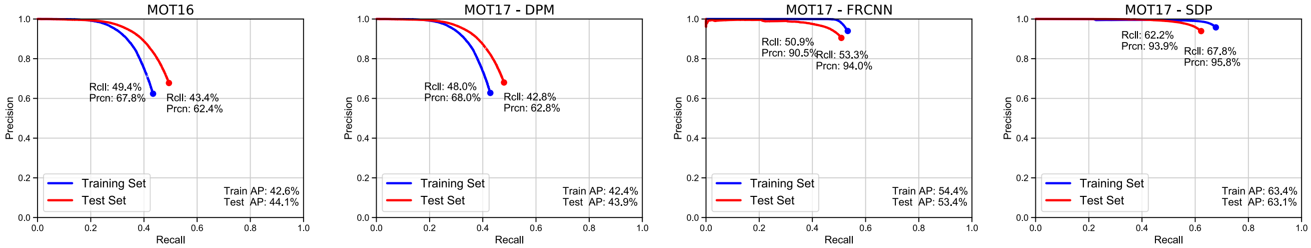


Fig. 3 An overview of the *MOT16*/*MOT17* dataset. Top: Training sequences. Bottom: test sequences (Color figure online)



**(a)** DPM v5 **(b)** DPM v5 **(c)** Faster-RCNN **(d)** SDP

Fig. 4 The performance of three popular pedestrian detectors evaluated on the training (blue) and the test (red) set. The circle indicates the operating point (i.e. the input detection set) for the trackers of *MOT16* and *MOT17* (Color figure online)

accurate enough, especially in scenes with moving cameras.

– *Difficulty* generally, we wanted to include some well-known sequences, e.g., PETS2009, in the *MOT15* benchmark. However, these sequences have turned out to be too simple for state-of-the-art trackers why we concluded to create a new and more challenging benchmark.

To overcome these weaknesses, we created *MOT16*, a collection of all-new challenging sequences (including our new sequences from *MOT15*) and creating annotations following a more strict protocol (see Sect. C.1 of the Appendix).

## 5 MOT16 and MOT17 Releases

Our ambition for the release of *MOT16* was to compile a benchmark with new and more challenging sequences compared to *MOT15*. Figure 3 presents an overview of the benchmark training and test sequences (detailed information about the sequences is presented in Table 9 in the Appendix).

*MOT17* consists of the same sequences as *MOT16*, but contains two important changes: (i) the annotations are further improved, i.e., increasing the accuracy of the bounding boxes, adding missed pedestrians, annotating additional occluders, following the comments received by many anonymous benchmark users, as well as the second round of sanity checks, (ii) the evaluation system significantly differs from *MOT17*, including the evaluation of tracking methods using three different detectors in order to show the robustness to varying levels of noisy detections.

### 5.1 MOT16 Sequences

We compiled a total of 14 sequences, of which we use half for training and a half for testing. The annotations of the testing sequences are not publicly available. The sequences can be classified according to moving/static camera, viewpoint, and illumination conditions (Fig. 11 in Appendix). The new data contains almost 3 times more bounding boxes for training and testing than *MOT15*. Most sequences are filmed in high resolution, and the mean crowd density is 3 times higher when compared to the first benchmark release. Hence, the new sequences present a more challenging benchmark than *MOT15* for the tracking community.

### 5.2 Detections

We evaluate several state-of-the-art detectors on our benchmark, and summarize the main findings in Fig. 4. To evaluate the performance of the detectors for the task of tracking, we evaluate them using all bounding boxes considered for the tracking evaluation, including partially visible or occluded objects. Consequently, the recall and average precision (AP) is lower than the results obtained by evaluating solely on visible objects, as we do for the detection challenge.

*MOT16 Detections* We first train the deformable part-based model (DPM) v5 (Felzenszwalb and Huttenlocher 2006) and find that it outperforms other detectors such as Fast-

RNN (Girshick 2015) and ACF (Dollár et al. 2014) for the task of detecting persons on *MOT16*. Hence, for that benchmark, we provide DPM detections as public detections.

*MOT17 Detections* For the new *MOT17* release, we use Faster-RCNN (Ren et al. 2015) and a detector with scale-dependent pooling (SDP) (Yang et al. 2016), both of which outperform the previous DPM method. After a discussion held in one of the *MOTChallenge* workshops, we agreed to provide all three detections as public detections, effectively changing the way *MOTChallenge* evaluates trackers. The motivation is to challenge trackers further to be more general and work with detections of varying quality. These detectors have different characteristics, as can be seen in in Fig. 4. Hence, a tracker that can work with all three inputs is going to be inherently more robust. The evaluation for *MOT17* is, therefore, set to evaluate the output of trackers on all three detection sets, averaging their performance for the final ranking. A detailed breakdown of detection bounding box statistics on individual sequences is provided in Table 10 in the Appendix.

# 6 Evaluation

*MOTChallenge* is also a platform for a fair comparison of state-of-the-art tracking methods. By providing authors with standardized ground-truth data, evaluation metrics, scripts, as well as a set of precomputed detections, all methods are compared under the same conditions, thereby isolating the performance of the tracker from other factors. In the past, a large number of metrics for quantitative evaluation of multiple target tracking have been proposed (Bernardin and Stiefelhagen 2008; Li et al. 2009; Schuhmacher et al. 2008; Smith et al. 2005; Stiefelhagen et al. 2006; Wu and Nevatia 2006). Choosing "the right" one is largely application dependent and the quest for a unique, general evaluation measure is still ongoing. On the one hand, it is desirable to summarize the performance into a single number to enable a direct comparison between methods. On the other hand, one might want to provide more informative performance estimates by detailing the types of errors the algorithms make, which precludes a clear ranking.

Following a recent trend (Bae and Yoon 2014; Milan et al. 2014; Wen et al. 2014), we employ three sets of tracking performance measures that have been established in the literature: (i) the frame-to-frame based *CLEAR-MOT* metrics proposed by Stiefelhagen et al. (2006), (ii) track quality measures proposed by Wu and Nevatia (2006), and (iii) trajectory-based IDF1 proposed by Ristani et al. (2016).

These evaluation measures give a complementary view on tracking performance. The main representative of CLEAR-MOT measures, Multi-Object Tracking Accuracy (MOTA), is evaluated based on frame-to-frame matching between track predictions and ground truth. It explicitly penalizes identity switches between consecutive frames, thus evaluating tracking performance only locally. This measure tends to put more emphasis on object detection performance compared to temporal continuity. In contrast, track quality measures (Wu and Nevatia 2006) and IDF1 Ristani et al. (2016), perform prediction-to-ground-truth matching on a trajectory level and over-emphasize the temporal continuity aspect of the tracking performance. In this section, we first introduce the matching between predicted track and ground-truth annotation before we present the final measures. All evaluation scripts used in our benchmark are publicly available.[9]

## 6.1 Multiple Object Tracking Accuracy

MOTA summarizes three sources of errors with a single performance measure:

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t}, \tag{1}$$

where $t$ is the frame index and $GT$ is the number of ground-truth objects. where $FN$ are the false negatives, i.e., the number of ground truth objects that were not detected by the method. $FP$ are the false positives, i.e., the number of objects that were falsely detected by the method but do not exist in the ground-truth. $IDSW$ is the number of identity switches, i.e., how many times a given trajectory changes from one ground-truth object to another. The computation of these values as well as other implementation details of the evaluation tool are detailed in Appendix Sect. D. We report the percentage MOTA $(-\infty, 100]$ in our benchmark. Note, that MOTA can also be negative in cases where the number of errors made by the tracker exceeds the number of all objects in the scene.

**Justification** We note that MOTA has been criticized in the literature for not having different sources of errors properly balanced. However, to this day, MOTA is still considered to be the most expressive measure for single-camera MOT evaluation. It was widely adopted for ranking methods in more recent tracking benchmarks, such as PoseTrack (Andriluka et al. 2018), KITTI tracking (Geiger et al. 2012), and the newly released Lyft (Kesten et al. 2019), Waymo (Sun et al. 2020), and ArgoVerse (Chang et al. 2019) benchmarks. We adopt MOTA for ranking, however, we recommend taking alternative evaluation measures (Ristani et al. 2016; Wu and Nevatia 2006) into the account when assessing the tracker's performance.

**Robustness** One incentive behind compiling this benchmark was to reduce dataset bias by keeping the data as diverse as possible. The main motivation is to challenge state-of-the-art

[9] http://motchallenge.net/devkit.

approaches and analyze their performance in unconstrained environments and on unseen data. Our experience shows that most methods can be heavily overfitted on one particular dataset, and may not be general enough to handle an entirely different setting without a major change in parameters or even in the model.

### 6.2 Multiple Object Tracking Precision

The Multiple Object Tracking Precision is the average dissimilarity between all true positives and their corresponding ground-truth targets. For bounding box overlap, this is computed as:

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \tag{2}$$

where $c_t$ denotes the number of matches in frame $t$ and $d_{t,i}$ is the bounding box overlap of target $i$ with its assigned ground-truth object in frame $t$. MOTP thereby gives the average overlap of $t_d$ between all correctly matched hypotheses and their respective objects and ranges between $t_d := 50\%$ and $100\%$.

It is important to point out that MOTP is a measure of localisation precision, *not* to be confused with the *positive predictive value* or *relevance* in the context of precision / recall curves used, e.g., in object detection.

In practice, it quantifies the localization precision of the detector, and therefore, it provides little information about the actual performance of the tracker.

### 6.3 Identification Precision, Identification Recall, and F1 Score

CLEAR-MOT evaluation measures provide event-based tracking assessment. In contrast, the IDF1 measure (Ristani et al. 2016) is an identity-based measure that emphasizes the track identity preservation capability over the entire sequence. In this case, the predictions-to-ground-truth mapping is established by solving a bipartite matching problem, connecting pairs with the largest temporal overlap. After the matching is established, we can compute the number of True Positive IDs (IDTP), False Negative IDs (IDFN), and False Positive IDs (IDFP), that generalise the concept of per-frame TPs, FNs and FPs to tracks. Based on these quantities, we can express the Identification Precision (IDP) as:

$$IDP = \frac{IDTP}{IDTP + IDFP}, \tag{3}$$

and Identification Recall (IDR) as:

$$IDR = \frac{IDTP}{IDTP + IDFN}. \tag{4}$$

Note that IDP and IDR are the fraction of computed (ground-truth) detections that are correctly identified. IDF1 is then expressed as a ratio of correctly identified detections over the average number of ground-truth and computed detections and balances identification precision and recall through their harmonic mean:

$$IDF1 = \frac{2 \cdot IDTP}{2 \cdot IDTP + IDFP + IDFN}. \tag{5}$$

### 6.4 Track Quality Measures

The final measures that we report on our benchmark are qualitative, and evaluate the percentage of the ground-truth trajectory that is recovered by a tracking algorithm. Each ground-truth trajectory can be consequently classified as mostly tracked (MT), partially tracked (PT), and mostly lost (ML). As defined in Wu and Nevatia (2006), a target is mostly tracked if it is successfully tracked for at least 80% of its life span, and considered lost in case it is covered for less than 20% of its total length. The remaining tracks are considered to be partially tracked. A higher number of MT and a few ML is desirable. Note, that it is irrelevant for this measure whether the ID remains the same throughout the track. We report MT and ML as a ratio of mostly tracked and mostly lost targets to the total number of ground-truth trajectories.

In certain situations, one might be interested in obtaining long, persistent tracks without trajectory gaps. To that end, the number of track fragmentations (FM) counts how many times a ground-truth trajectory is interrupted (untracked). A fragmentation event happens each time a trajectory changes its status from tracked to untracked and is resumed at a later point. Similarly to the ID switch ratio (c.f. Sect. D.1), we also provide the relative number of fragmentations as FM/Recall.

## 7 Analysis of State-of-the-Art Trackers

We now present an analysis of recent multi-object tracking methods that submitted to the benchmark. This is divided into two parts: (i) categorization of the methods, where our goal is to help young scientists to navigate the recent MOT literature, and (ii) error and runtime analysis, where we point out methods that have shown good performance on a wide range of scenes. We hope this can eventually lead to new promising research directions.

We consider all valid submissions to all three benchmarks that were published before April 17th, 2020, and used the provided set of public detections. For this analysis, we focus on methods that are peer-reviewed, i.e., published at a conference or a journal. We evaluate a total of 101 (public) trackers; 73 trackers were tested on *MOT15*, 74 on *MOT16* and 57 on

**Table 1** The *MOT15* leaderboard

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MPNTrack (Brasó and Leal-Taixé 2020) | 51.54 | 58.61 | 76.05 | 1.32 | 225 | 187 | 7620 | 21,780 | 375 | 872 | 5.81 | 13.51 |
| Tracktor++v2 (Bergmann et al. 2019) | 46.60 | 47.57 | 76.36 | 0.80 | 131 | 201 | 4624 | 26,896 | 1290 | 1702 | 22.94 | 30.27 |
| TrctrD15 (Xu et al. 2020) | 44.09 | 45.99 | 75.26 | 1.05 | 124 | 192 | 6085 | 26,917 | 1347 | 1868 | 23.97 | 33.24 |
| Tracktor++ (Bergmann et al. 2019) | 44.06 | 46.73 | 75.03 | 1.12 | 130 | 189 | 6477 | 26,577 | 1318 | 1790 | 23.23 | 31.55 |
| KCF (Chu et al. 2019) | 38.90 | 44.54 | 70.56 | 1.27 | 120 | 227 | 7321 | 29,501 | 720 | 1440 | 13.85 | 27.70 |
| AP_HWDPL_p (Long et al. 2017) | 38.49 | 47.10 | 72.56 | 0.69 | 63 | 270 | 4005 | 33,203 | 586 | 1263 | 12.75 | 27.48 |
| STRN (Xu et al. 2019) | 38.06 | 46.62 | 72.06 | 0.94 | 83 | 241 | 5451 | 31,571 | 1033 | 2665 | 21.25 | 54.82 |
| AMIR15 (Sadeghian et al. 2017) | 37.57 | 46.01 | 71.66 | 1.37 | 114 | 193 | 7933 | 29,397 | 1026 | 2024 | 19.67 | 38.81 |
| JointMC (Keuper et al. 2018) | 35.64 | 45.12 | 71.90 | 1.83 | 167 | 283 | 10,580 | 28,508 | 457 | 969 | 8.53 | 18.08 |
| RAR15pub (Fang et al. 2018) | 35.11 | 45.40 | 70.94 | 1.17 | 94 | 305 | 6771 | 32,717 | 381 | 1523 | 8.15 | 32.58 |
| HybridDAT (Yang et al. 2017) | 34.97 | 47.72 | 72.57 | 1.46 | 82 | 304 | 8455 | 31,140 | 358 | 1267 | 7.26 | 25.69 |
| INARLA (Wu et al. 2019) | 34.69 | 42.06 | 70.72 | 1.71 | 90 | 216 | 9855 | 29,158 | 1112 | 2848 | 21.16 | 54.20 |
| STAM (Chu et al. 2017) | 34.33 | 48.26 | 70.55 | 0.89 | 82 | 313 | 5154 | 34,848 | 348 | 1463 | 8.04 | 33.80 |
| QuadMOT (Son et al. 2017) | 33.82 | 40.43 | 73.42 | 1.37 | 93 | 266 | 7898 | 32,061 | 703 | 1430 | 14.70 | 29.91 |
| NOMT (Choi 2015) | 33.67 | 44.55 | 71.94 | 1.34 | 88 | 317 | 7762 | 32,547 | 442 | 823 | 9.40 | 17.50 |
| DCCRF (Zhou et al. 2018a) | 33.62 | 39.08 | 70.91 | 1.02 | 75 | 271 | 5917 | 34,002 | 866 | 1566 | 19.39 | 35.07 |
| TDAM (Yang and Jia 2016) | 33.03 | 46.05 | 72.78 | 1.74 | 96 | 282 | 10,064 | 30,617 | 464 | 1506 | 9.25 | 30.02 |
| CDA_DDALpb (Bae and Yoon 2018) | 32.80 | 38.79 | 70.70 | 0.86 | 70 | 304 | 4983 | 35,690 | 614 | 1583 | 14.65 | 37.77 |
| MHT_DAM (Kim et al. 2015) | 32.36 | 45.31 | 71.83 | 1.57 | 115 | 316 | 9064 | 32,060 | 435 | 826 | 9.10 | 17.27 |
| LFNF (Sheng et al. 2017) | 31.64 | 33.10 | 72.03 | 1.03 | 69 | 301 | 5943 | 35,095 | 961 | 1106 | 22.41 | 25.79 |
| GMPHD_OGM (Song et al. 2019) | 30.72 | 38.82 | 71.64 | 1.13 | 83 | 275 | 6502 | 35,030 | 1034 | 1351 | 24.05 | 31.43 |
| PHD_GSDL (Fu et al. 2018) | 30.51 | 38.82 | 71.20 | 1.13 | 55 | 297 | 6534 | 35,284 | 879 | 2208 | 20.65 | 51.87 |
| MDP (Xiang et al. 2015) | 30.31 | 44.68 | 71.32 | 1.68 | 94 | 277 | 9717 | 32,422 | 680 | 1500 | 14.40 | 31.76 |
| MCF_PHD (Wojke and Paulus 2016) | 29.89 | 38.18 | 71.70 | 1.54 | 86 | 317 | 8892 | 33,529 | 656 | 989 | 14.44 | 21.77 |
| CNNTCM (Wang et al. 2016) | 29.64 | 36.82 | 71.78 | 1.35 | 81 | 317 | 7786 | 34,733 | 712 | 943 | 16.38 | 21.69 |
| RSCNN (Mahgoub et al. 2017) | 29.50 | 36.97 | 73.07 | 2.05 | 93 | 262 | 11,866 | 30,474 | 976 | 1176 | 19.36 | 23.33 |
| TBSS15 (Zhou et al. 2018b) | 29.21 | 37.23 | 71.28 | 1.05 | 49 | 316 | 6068 | 36,779 | 649 | 1508 | 16.17 | 37.57 |
| SCEA (Yoon et al. 2016) | 29.08 | 37.15 | 71.11 | 1.05 | 64 | 341 | 6060 | 36,912 | 604 | 1182 | 15.13 | 29.61 |
| SiameseCNN (Leal-Taixe et al. 2016) | 29.04 | 34.27 | 71.20 | 0.89 | 61 | 349 | 5160 | 37798 | 639 | 1316 | 16.61 | 34.20 |

**Table 1** continued

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HAM_INTP15 (Yoon et al. 2018a) | 28.62 | 41.45 | 71.13 | 1.30 | 72 | 317 | 7485 | 35,910 | 460 | 1038 | 11.07 | 24.98 |
| GMMA_intp (Song et al. 2018) | 27.32 | 36.59 | 70.92 | 1.36 | 47 | 311 | 7848 | 35,817 | 987 | 1848 | 23.67 | 44.31 |
| oICF (Kieritz et al. 2016) | 27.08 | 40.49 | 69.96 | 1.31 | 46 | 351 | 7594 | 36,757 | 454 | 1660 | 11.30 | 41.32 |
| TO (Manen et al. 2016) | 25.66 | 32.74 | 72.17 | 0.83 | 31 | 414 | 4779 | 40,511 | 383 | 600 | 11.24 | 17.61 |
| LP_SSVM (Wang and Fowlkes 2016) | 25.22 | 34.05 | 71.68 | 1.45 | 42 | 382 | 8369 | 36,932 | 646 | 849 | 16.19 | 21.28 |
| HAM_SADF (Yoon et al. 2018a) | 25.19 | 37.80 | 71.38 | 1.27 | 41 | 420 | 7330 | 38,275 | 357 | 745 | 9.47 | 19.76 |
| ELP (McLaughlin et al. 2015) | 24.99 | 26.21 | 71.17 | 1.27 | 54 | 316 | 7345 | 37,344 | 1396 | 1804 | 35.60 | 46.00 |
| AdTobKF (Loumponias et al. 2018) | 24.82 | 34.50 | 70.78 | 1.07 | 29 | 375 | 6201 | 39,321 | 666 | 1300 | 18.50 | 36.11 |
| LINF1 (Fagot-Bouquet et al. 2016) | 24.53 | 34.82 | 71.33 | 1.01 | 40 | 466 | 5864 | 40,207 | 298 | 744 | 8.62 | 21.53 |
| TENSOR (Shi et al. 2018) | 24.32 | 24.13 | 71.58 | 1.15 | 40 | 336 | 6644 | 38,582 | 1271 | 1304 | 34.16 | 35.05 |
| TFMOT (Boragule and Jeon 2017) | 23.81 | 32.30 | 71.35 | 0.78 | 35 | 447 | 4533 | 41,873 | 404 | 792 | 12.69 | 24.87 |
| JPDA_m (Rezatofighi et al. 2015) | 23.79 | 33.77 | 68.17 | 1.10 | 36 | 419 | 6373 | 40,084 | 365 | 869 | 10.50 | 25.00 |
| MotiCon (Leal-Taixé et al. 2014) | 23.07 | 29.38 | 70.87 | 1.80 | 34 | 375 | 10,404 | 35,844 | 1018 | 1061 | 24.44 | 25.47 |
| DEEPDA_MOT (Yoon et al. 2019a) | 22.53 | 25.92 | 70.92 | 1.27 | 46 | 447 | 7346 | 39,092 | 1159 | 1538 | 31.86 | 42.28 |
| SegTrack (Milan et al. 2015) | 22.51 | 31.48 | 71.65 | 1.36 | 42 | 461 | 7890 | 39,020 | 697 | 737 | 19.10 | 20.20 |
| EAMTTpub (Sanchez-Matilla et al. 2016) | 22.30 | 32.84 | 70.79 | 1.37 | 39 | 380 | 7924 | 38,982 | 833 | 1485 | 22.79 | 40.63 |
| SAS_MOT15 (Maksai and Fua 2019) | 22.16 | 27.15 | 71.10 | 0.97 | 22 | 444 | 5591 | 41,531 | 700 | 1240 | 21.60 | 38.27 |
| OMT_DFH (Ju et al. 2017a) | 21.16 | 37.34 | 69.94 | 2.29 | 51 | 335 | 13,218 | 34,657 | 563 | 1255 | 12.92 | 28.79 |
| MTSTracker (Nguyen Thi Lan Anh et al. 2017) | 20.64 | 31.87 | 70.32 | 2.62 | 65 | 266 | 15,161 | 32,212 | 1387 | 2357 | 29.16 | 49.55 |
| TC_SIAMESE (Yoon et al. 2018b) | 20.22 | 32.59 | 71.09 | 1.06 | 19 | 487 | 6127 | 42,596 | 294 | 825 | 9.59 | 26.90 |
| DCO_X (Milan et al. 2016) | 19.59 | 31.45 | 71.39 | 1.84 | 37 | 396 | 10,652 | 38,232 | 521 | 819 | 13.79 | 21.68 |
| CEM (Milan et al. 2014) | 19.30 | N/A | 70.74 | 2.45 | 61 | 335 | 14,180 | 34,591 | 813 | 1023 | 18.60 | 23.41 |
| RNN_LSTM (Milan et al. 2017) | 18.99 | 17.12 | 70.97 | 2.00 | 40 | 329 | 11,578 | 36,706 | 1490 | 2081 | 37.01 | 51.69 |
| RMOT (Yoon et al. 2015) | 18.63 | 32.56 | 69.57 | 2.16 | 38 | 384 | 12,473 | 36,835 | 684 | 1282 | 17.08 | 32.01 |
| TSDA_OAL (Ju et al. 2017b) | 18.61 | 36.07 | 69.68 | 2.83 | 68 | 305 | 16,350 | 32,853 | 806 | 1544 | 17.32 | 33.18 |
| GMPHD_15 (Song and Jeon 2016) | 18.47 | 28.38 | 70.90 | 1.36 | 28 | 399 | 7864 | 41,766 | 459 | 1266 | 14.33 | 39.54 |
| SMOT (Dicle et al. 2013) | 18.23 | 0.00 | 71.23 | 1.52 | 20 | 395 | 8780 | 40,310 | 1148 | 2132 | 33.38 | 61.99 |
| ALExTRAC (Bewley et al. 2016b) | 16.95 | 17.30 | 71.18 | 1.60 | 28 | 378 | 9233 | 39,933 | 1859 | 1872 | 53.11 | 53.48 |
| TBD (Geiger et al. 2014) | 15.92 | 0.00 | 70.86 | 2.58 | 46 | 345 | 14,943 | 34,777 | 1939 | 1963 | 44.68 | 45.23 |
| GSCR (Fagot-Bouquet et al. 2015) | 15.78 | 27.90 | 69.38 | 1.31 | 13 | 440 | 7597 | 43,633 | 514 | 1010 | 17.73 | 34.85 |
| TC_ODAL (Bae and Yoon 2014) | 15.13 | 0.00 | 70.53 | 2.24 | 23 | 402 | 12,970 | 38,538 | 637 | 1716 | 17.09 | 46.04 |
| DP_NMS (Pirsiavash et al. 2011) | 14.52 | 19.69 | 70.76 | 2.28 | 43 | 294 | 13,171 | 34,814 | 4537 | 3090 | 104.69 | 71.30 |

Performance of several trackers according to different metrics

**Table 2** The *MOT16* leaderboard

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MPNTrack (Brasó and Leal-Taixé 2020) | 58.56 | 61.69 | 78.88 | 0.84 | 207 | 258 | 4949 | 70,252 | 354 | 684 | 5.76 | 11.13 |
| Tracktor++v2 (Bergmann et al. 2019) | 56.20 | 54.91 | 79.20 | 0.40 | 157 | 272 | 2394 | 76,844 | 617 | 1068 | 10.66 | 18.46 |
| TrctrD16 (Xu et al. 2020) | 54.83 | 53.39 | 77.47 | 0.50 | 145 | 281 | 2955 | 78,765 | 645 | 1515 | 11.36 | 26.67 |
| Tracktor++ (Bergmann et al. 2019) | 54.42 | 52.54 | 78.22 | 0.55 | 144 | 280 | 3280 | 79,149 | 682 | 1480 | 12.05 | 26.15 |
| NOTA_16 (Chen et al. 2019) | 49.83 | 55.33 | 74.49 | 1.22 | 136 | 286 | 7248 | 83,614 | 614 | 1372 | 11.34 | 25.34 |
| HCC (Ma et al. 2018b) | 49.25 | 50.67 | 79.00 | 0.90 | 135 | 303 | 5333 | 86,795 | 391 | 535 | 7.46 | 10.21 |
| eTC (Wang et al. 2019) | 49.15 | 56.11 | 75.49 | 1.42 | 131 | 306 | 8400 | 83,702 | 606 | 882 | 11.20 | 16.31 |
| KCF16 (Chu et al. 2019) | 48.80 | 47.19 | 75.66 | 0.99 | 120 | 289 | 5875 | 86,567 | 906 | 1116 | 17.25 | 21.25 |
| LMP (Tang et al. 2017) | 48.78 | 51.26 | 79.04 | 1.12 | 138 | 304 | 6654 | 86,245 | 481 | 595 | 9.13 | 11.29 |
| TLMHT (Sheng et al. 2018a) | 48.69 | 55.29 | 76.43 | 1.12 | 119 | 338 | 6632 | 86,504 | 413 | 642 | 7.86 | 12.22 |
| STRN_MOT16 (Xu et al. 2019) | 48.46 | 53.90 | 73.75 | 1.53 | 129 | 265 | 9038 | 84,178 | 747 | 2919 | 13.88 | 54.23 |
| GCRA (Ma et al. 2018a) | 48.16 | 48.55 | 77.50 | 0.86 | 98 | 312 | 5104 | 88,586 | 821 | 1117 | 15.97 | 21.73 |
| FWT (Henschel et al. 2018) | 47.77 | 44.28 | 75.51 | 1.50 | 145 | 290 | 8886 | 85,487 | 852 | 1534 | 16.04 | 28.88 |
| MOTDT (Long et al. 2018) | 47.63 | 50.94 | 74.81 | 1.56 | 115 | 291 | 9253 | 85,431 | 792 | 1858 | 14.90 | 34.96 |
| NLLMPa (Levinkov et al. 2017) | 47.58 | 47.34 | 78.51 | 0.99 | 129 | 307 | 5844 | 89,093 | 629 | 768 | 12.30 | 15.02 |
| EAGS16 (Sheng et al. 2018b) | 47.41 | 50.13 | 75.95 | 1.41 | 131 | 324 | 8369 | 86,931 | 575 | 913 | 10.99 | 17.45 |
| JCSTD (Tian et al. 2019) | 47.36 | 41.10 | 74.43 | 1.36 | 109 | 276 | 8076 | 86,638 | 1266 | 2697 | 24.12 | 51.39 |
| ASTT  (Tao et al. 2018) | 47.24 | 44.27 | 76.08 | 0.79 | 124 | 316 | 4680 | 90,877 | 633 | 814 | 12.62 | 16.23 |
| eHAF16 (Sheng et al. 2018c) | 47.22 | 52.44 | 75.69 | 2.13 | 141 | 325 | 12,586 | 83,107 | 542 | 787 | 9.96 | 14.46 |
| AMIR (Sadeghian et al. 2017) | 47.17 | 46.29 | 75.82 | 0.45 | 106 | 316 | 2681 | 92,856 | 774 | 1675 | 15.77 | 34.14 |
| JointMC (MCjoint) (Keuper et al. 2018) | 47.10 | 52.26 | 76.27 | 1.13 | 155 | 356 | 6703 | 89,368 | 370 | 598 | 7.26 | 11.73 |
| YOONKJ16 (Yoon et al. 2020) | 46.96 | 50.05 | 75.76 | 1.33 | 125 | 317 | 7901 | 88,179 | 627 | 945 | 12.14 | 18.30 |
| NOMT_16 (Choi 2015) | 46.42 | 53.30 | 76.56 | 1.65 | 139 | 314 | 9753 | 87,565 | 359 | 504 | 6.91 | 9.70 |
| JMC (Tang et al. 2016) | 46.28 | 46.31 | 75.68 | 1.08 | 118 | 301 | 6373 | 90,914 | 657 | 1114 | 13.10 | 22.22 |
| DD_TAMA16 (Yoon et al. 2019b) | 46.20 | 49.43 | 75.42 | 0.87 | 107 | 334 | 5126 | 92,367 | 598 | 1127 | 12.12 | 22.84 |
| DMAN_16 (Zhu et al. 2018) | 46.08 | 54.82 | 73.77 | 1.34 | 132 | 324 | 7909 | 89,874 | 532 | 1616 | 10.49 | 31.87 |
| STAM16 (Chu et al. 2017) | 45.98 | 50.05 | 74.92 | 1.16 | 111 | 331 | 6895 | 91,117 | 473 | 1422 | 9.46 | 28.43 |
| RAR16pub (Fang et al. 2018) | 45.87 | 48.77 | 74.84 | 1.16 | 100 | 318 | 6871 | 91,173 | 648 | 1992 | 12.96 | 39.85 |
| MHT_DAM_16 (Kim et al. 2015) | 45.83 | 46.06 | 76.34 | 1.08 | 123 | 328 | 6412 | 91,758 | 590 | 781 | 11.88 | 15.72 |
| MTDF (Fu et al. 2019) | 45.72 | 40.07 | 72.63 | 2.03 | 107 | 276 | 12,018 | 84,970 | 1987 | 3377 | 37.21 | 63.24 |
| INTERA_MOT (Lan et al. 2018) | 45.40 | 47.66 | 74.41 | 2.27 | 137 | 294 | 13,407 | 85,547 | 600 | 930 | 11.30 | 17.52 |
| EDMT (Chen et al. 2017a) | 45.34 | 47.86 | 75.94 | 1.88 | 129 | 303 | 11,122 | 87,890 | 639 | 946 | 12.34 | 18.27 |
| DCCRF16 (Zhou et al. 2018a) | 44.76 | 39.67 | 75.63 | 0.95 | 107 | 321 | 5613 | 94,133 | 968 | 1378 | 20.01 | 28.49 |
| TBSS (Zhou et al. 2018b) | 44.58 | 42.64 | 75.18 | 0.70 | 93 | 333 | 4136 | 96,128 | 790 | 1419 | 16.71 | 30.01 |

**Table 2** continued

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OTCD_1_16 (Liu et al. 2019) | 44.36 | 45.62 | 75.36 | 0.97 | 88 | 361 | 5759 | 94,927 | 759 | 1787 | 15.83 | 37.28 |
| QuadMOT16 (Son et al. 2017) | 44.10 | 38.27 | 76.40 | 1.08 | 111 | 341 | 6388 | 94775 | 745 | 1096 | 15.52 | 22.83 |
| CDA_DDALv2 (Bae and Yoon 2018) | 43.89 | 45.13 | 74.69 | 1.09 | 81 | 337 | 6450 | 95,175 | 676 | 1795 | 14.14 | 37.55 |
| LFNF16 (Sheng et al. 2017) | 43.61 | 41.62 | 76.63 | 1.12 | 101 | 347 | 6616 | 95,363 | 836 | 938 | 17.53 | 19.67 |
| oICF_16 (Kieritz et al. 2016) | 43.21 | 49.33 | 74.31 | 1.12 | 86 | 368 | 6651 | 96,515 | 381 | 1404 | 8.10 | 29.83 |
| MHT_bLSTM6 (Kim et al. 2018) | 42.10 | 47.84 | 75.85 | 1.97 | 113 | 337 | 11,637 | 93,172 | 753 | 1156 | 15.40 | 23.64 |
| LINF1_16 (Fagot-Bouquet et al. 2016) | 41.01 | 45.69 | 74.85 | 1.33 | 88 | 389 | 7896 | 99,224 | 430 | 963 | 9.43 | 21.13 |
| PHD_GSDL16 (Fu et al. 2018) | 41.00 | 43.14 | 75.90 | 1.10 | 86 | 315 | 6498 | 99,257 | 1810 | 3650 | 39.73 | 80.11 |
| GMPHD_ReId Baisa (2019b) | 40.42 | 49.71 | 75.25 | 1.11 | 85 | 329 | 6572 | 101,266 | 792 | 2529 | 17.81 | 56.88 |
| AM_ADM (Lee et al. 2018) | 40.12 | 43.79 | 75.45 | 1.44 | 54 | 351 | 8503 | 99,891 | 789 | 1736 | 17.45 | 38.40 |
| EAMTT_pub (Sanchez-Matilla et al. 2016) | 38.83 | 42.43 | 75.15 | 1.37 | 60 | 373 | 8114 | 102,452 | 965 | 1657 | 22.03 | 37.83 |
| OVBT (Ban et al. 2016) | 38.40 | 37.82 | 75.39 | 1.95 | 57 | 359 | 11,517 | 99,463 | 1321 | 2140 | 29.07 | 47.09 |
| GMMCP (Dehghan et al. 2015) | 38.10 | 35.50 | 75.84 | 1.12 | 65 | 386 | 6607 | 105,315 | 937 | 1669 | 22.18 | 39.51 |
| LTTSC-CRF (Le et al. 2016) | 37.59 | 42.06 | 75.94 | 2.02 | 73 | 419 | 11,969 | 101,343 | 481 | 1012 | 10.83 | 22.79 |
| JCmin_MOT (Boragule and Jeon 2017) | 36.65 | 36.16 | 75.86 | 0.50 | 57 | 413 | 2936 | 111,890 | 667 | 831 | 17.27 | 21.51 |
| HISP_T (Baisa 2018) | 35.87 | 28.93 | 76.07 | 1.08 | 59 | 380 | 6412 | 107,918 | 2594 | 2298 | 63.56 | 56.31 |
| LP2D_16 (Leal-Taixé et al. 2014) | 35.74 | 34.18 | 75.84 | 0.86 | 66 | 385 | 5084 | 111,163 | 915 | 1264 | 23.44 | 32.39 |
| GM_PHD_DAL (Baisa 2019a) | 35.13 | 26.58 | 76.59 | 0.40 | 53 | 390 | 2350 | 111,886 | 4047 | 5338 | 104.75 | 138.17 |
| TBD_16 (Geiger et al. 2014) | 33.74 | 0.00 | 76.53 | 0.98 | 55 | 411 | 5804 | 112,587 | 2418 | 2252 | 63.22 | 58.88 |
| GM_PHD_N1T (Baisa and Wallace 2019) | 33.25 | 25.47 | 76.84 | 0.30 | 42 | 425 | 1750 | 116,452 | 3499 | 3594 | 96.85 | 99.47 |
| CEM_16 (Milan et al. 2014) | 33.19 | N/A | 75.84 | 1.16 | 59 | 413 | 6837 | 114,322 | 642 | 731 | 17.21 | 19.60 |
| GMPHD_HDA (Song and Jeon 2016) | 30.52 | 33.37 | 75.42 | 0.87 | 35 | 453 | 5169 | 120,970 | 539 | 731 | 16.02 | 21.72 |
| SMOT_16 (Dicle et al. 2013) | 29.75 | N/A | 75.18 | 2.94 | 40 | 362 | 17,426 | 107,552 | 3108 | 4483 | 75.79 | 109.32 |
| JPDA_m_16 (Rezatofighi et al. 2015) | 26.17 | N/A | 76.34 | 0.62 | 31 | 512 | 3689 | 130,549 | 365 | 638 | 12.85 | 22.47 |
| DP_NMS_16 (Pirsiavash et al. 2011) | 26.17 | 31.19 | 76.34 | 0.62 | 31 | 512 | 3689 | 130,557 | 365 | 638 | 12.86 | 22.47 |

Performance of several trackers according to different metrics

**Table 3** The *MOT17* leaderboard

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MPNTrack (Brasó and Leal-Taixé 2020) | 58.85 | 61.75 | 78.62 | 0.98 | 679 | 788 | 17,413 | 213,594 | 1185 | 2265 | 19.07 | 36.45 |
| Tracktor++v2 (Bergmann et al. 2019) | 56.35 | 55.12 | 78.82 | 0.50 | 498 | 831 | 8866 | 235,449 | 1987 | 3763 | 34.10 | 64.58 |
| TrctrD17 (Xu et al. 2020) | 53.72 | 53.77 | 77.23 | 0.66 | 458 | 861 | 11,731 | 247,447 | 1947 | 4792 | 34.68 | 85.35 |
| Tracktor++ (Bergmann et al. 2019) | 53.51 | 52.33 | 77.98 | 0.69 | 459 | 861 | 12,201 | 248,047 | 2072 | 4611 | 36.98 | 82.28 |
| JBNOT (Henschel et al. 2019) | 52.63 | 50.77 | 77.12 | 1.78 | 465 | 844 | 31,572 | 232,659 | 3050 | 3792 | 51.90 | 64.53 |
| FAMNet (Chu and Ling 2019) | 52.00 | 48.71 | 76.48 | 0.80 | 450 | 787 | 14,138 | 253,616 | 3072 | 5318 | 55.80 | 96.60 |
| eTC17 (Wang et al. 2019) | 51.93 | 58.13 | 76.34 | 2.04 | 544 | 836 | 36,164 | 232,783 | 2288 | 3071 | 38.95 | 52.28 |
| eHAF17 (Sheng et al. 2018c) | 51.82 | 54.72 | 77.03 | 1.87 | 551 | 893 | 33,212 | 236,772 | 1834 | 2739 | 31.60 | 47.19 |
| YOONKJ17 (Yoon et al. 2020) | 51.37 | 53.98 | 77.00 | 1.64 | 500 | 878 | 29,051 | 243,202 | 2118 | 3072 | 37.23 | 53.99 |
| FWT_17 (Henschel et al. 2018) | 51.32 | 47.56 | 77.00 | 1.36 | 505 | 830 | 24,101 | 247,921 | 2648 | 4279 | 47.24 | 76.33 |
| NOTA (Chen et al. 2019) | 51.27 | 54.46 | 76.68 | 1.13 | 403 | 833 | 20,148 | 252,531 | 2285 | 5798 | 41.36 | 104.95 |
| JointMC (jCC) (Keuper et al. 2018) | 51.16 | 54.50 | 75.92 | 1.46 | 493 | 872 | 25,937 | 247,822 | 1802 | 2984 | 32.13 | 53.21 |
| STRN_MOT17 (Xu et al. 2019) | 50.90 | 55.98 | 75.58 | 1.42 | 446 | 797 | 25,295 | 249,365 | 2397 | 9363 | 42.95 | 167.78 |
| MOTDT17 (Long et al. 2018) | 50.85 | 52.70 | 76.58 | 1.36 | 413 | 841 | 24,069 | 250,768 | 2474 | 5317 | 44.53 | 95.71 |
| MHT_DAM_17 (Kim et al. 2015) | 50.71 | 47.18 | 77.52 | 1.29 | 491 | 869 | 22,875 | 252,889 | 2314 | 2865 | 41.94 | 51.92 |
| TLMHT_17 (Sheng et al. 2018a) | 50.61 | 56.51 | 77.65 | 1.25 | 415 | 1022 | 22,213 | 255,030 | 1407 | 2079 | 25.68 | 37.94 |
| EDMT17 (Chen et al. 2017a) | 50.05 | 51.25 | 77.26 | 1.82 | 509 | 855 | 32,279 | 247,297 | 2264 | 3260 | 40.31 | 58.04 |
| GMPHDOGM17 (Song et al. 2019) | 49.94 | 47.15 | 77.01 | 1.35 | 464 | 895 | 24,024 | 255,277 | 3125 | 3540 | 57.07 | 64.65 |
| MTDF17 (Fu et al. 2019) | 49.58 | 45.22 | 75.48 | 2.09 | 444 | 779 | 37,124 | 241,768 | 5567 | 9260 | 97.41 | 162.03 |
| PHD_GM (Sanchez-Matilla and Cavallaro 2019) | 48.84 | 43.15 | 76.74 | 1.48 | 449 | 830 | 26,260 | 257,971 | 4407 | 6448 | 81.19 | 118.79 |

**Table 3** continued

| Method | MOTA | IDF1 | MOTP | FAR | MT | ML | FP | FN | IDSW | FM | IDSWR | FMR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OTCD_1_17 (Liu et al. 2019) | 48.57 | 47.90 | 76.91 | 1.04 | 382 | 970 | 18,499 | 268,204 | 3502 | 5588 | 66.75 | 106.51 |
| HAM_SADF17 (Yoon et al. 2018a) | 48.27 | 51.14 | 77.22 | 1.18 | 402 | 981 | 20,967 | 269,038 | 1871 | 3020 | 35.76 | 57.72 |
| DMAN (Zhu et al. 2018) | 48.24 | 55.69 | 75.69 | 1.48 | 454 | 902 | 26,218 | 263,608 | 2194 | 5378 | 41.18 | 100.94 |
| AM_ADM17 (Lee et al. 2018) | 48.11 | 52.07 | 76.69 | 1.41 | 316 | 934 | 25,061 | 265,495 | 2214 | 5027 | 41.82 | 94.95 |
| PHD_GSDL17 (Fu et al. 2018) | 48.04 | 49.63 | 77.15 | 1.31 | 402 | 838 | 23,199 | 265,954 | 3998 | 8886 | 75.63 | 168.09 |
| MHT_bLSTM (Kim et al. 2018) | 47.52 | 51.92 | 77.49 | 1.46 | 429 | 981 | 25,981 | 268,042 | 2069 | 3124 | 39.41 | 59.51 |
| MASS (Karunasekera et al. 2019) | 46.95 | 45.99 | 76.11 | 1.45 | 399 | 856 | 25,733 | 269,116 | 4478 | 11,994 | 85.62 | 229.31 |
| GMPHD_Rd17 (Baisa 2019b) | 46.83 | 54.06 | 76.41 | 2.17 | 464 | 784 | 38,452 | 257,678 | 3865 | 8097 | 71.14 | 149.03 |
| IOU17 (Bochinski et al. 2017) | 45.48 | 39.40 | 76.85 | 1.13 | 369 | 953 | 19,993 | 281,643 | 5988 | 7404 | 119.56 | 147.84 |
| LM_NN_17 (Babaee et al. 2019) | 45.13 | 43.17 | 78.93 | 0.61 | 348 | 1088 | 10,834 | 296,451 | 2286 | 2463 | 48.17 | 51.90 |
| FPSN (Lee and Kim 2019) | 44.91 | 48.43 | 76.61 | 1.90 | 388 | 844 | 33,757 | 269,952 | 7136 | 14,491 | 136.82 | 277.84 |
| HISP_T17 (Baisa 2019c) | 44.62 | 38.79 | 77.19 | 1.43 | 355 | 913 | 25,478 | 276,395 | 10,617 | 7487 | 208.12 | 146.76 |
| GMPHD_DAL (Baisa 2019a) | 44.40 | 36.23 | 77.42 | 1.08 | 350 | 927 | 19,170 | 283,380 | 11,137 | 13,900 | 223.74 | 279.25 |
| SAS_MOT17 (Maksai and Fua 2019) | 44.24 | 57.18 | 76.42 | 1.66 | 379 | 1044 | 29,473 | 283,611 | 1529 | 2644 | 30.74 | 53.16 |
| GMPHD_SHA (Song and Jeon 2016) | 43.72 | 39.17 | 76.53 | 1.46 | 276 | 1012 | 25,935 | 287,758 | 3838 | 5056 | 78.33 | 103.18 |
| SORT17 (Bewley et al. 2016a) | 43.14 | 39.84 | 77.77 | 1.60 | 295 | 997 | 28,398 | 287,582 | 4852 | 7127 | 98.96 | 145.36 |
| EAMTT_17 (Sanchez-Matilla et al. 2016) | 42.63 | 41.77 | 76.03 | 1.73 | 300 | 1006 | 30,711 | 288,474 | 4488 | 5720 | 91.83 | 117.04 |
| GMPHD_N1Tr (Baisa and Wallace 2019) | 42.12 | 33.87 | 77.66 | 1.03 | 280 | 1005 | 18,214 | 297,646 | 10,698 | 10,864 | 226.43 | 229.94 |
| GMPHD_KCF (Kutschbach et al. 2017) | 39.57 | 36.64 | 74.54 | 2.87 | 208 | 1019 | 50,903 | 284,228 | 5811 | 7414 | 117.10 | 149.40 |
| GM_PHD (Eiselein et al. 2012) | 36.36 | 33.92 | 76.20 | 1.34 | 97 | 1349 | 23,723 | 330,767 | 4607 | 11,317 | 111.34 | 273.51 |

Performance of several trackers according to different metrics

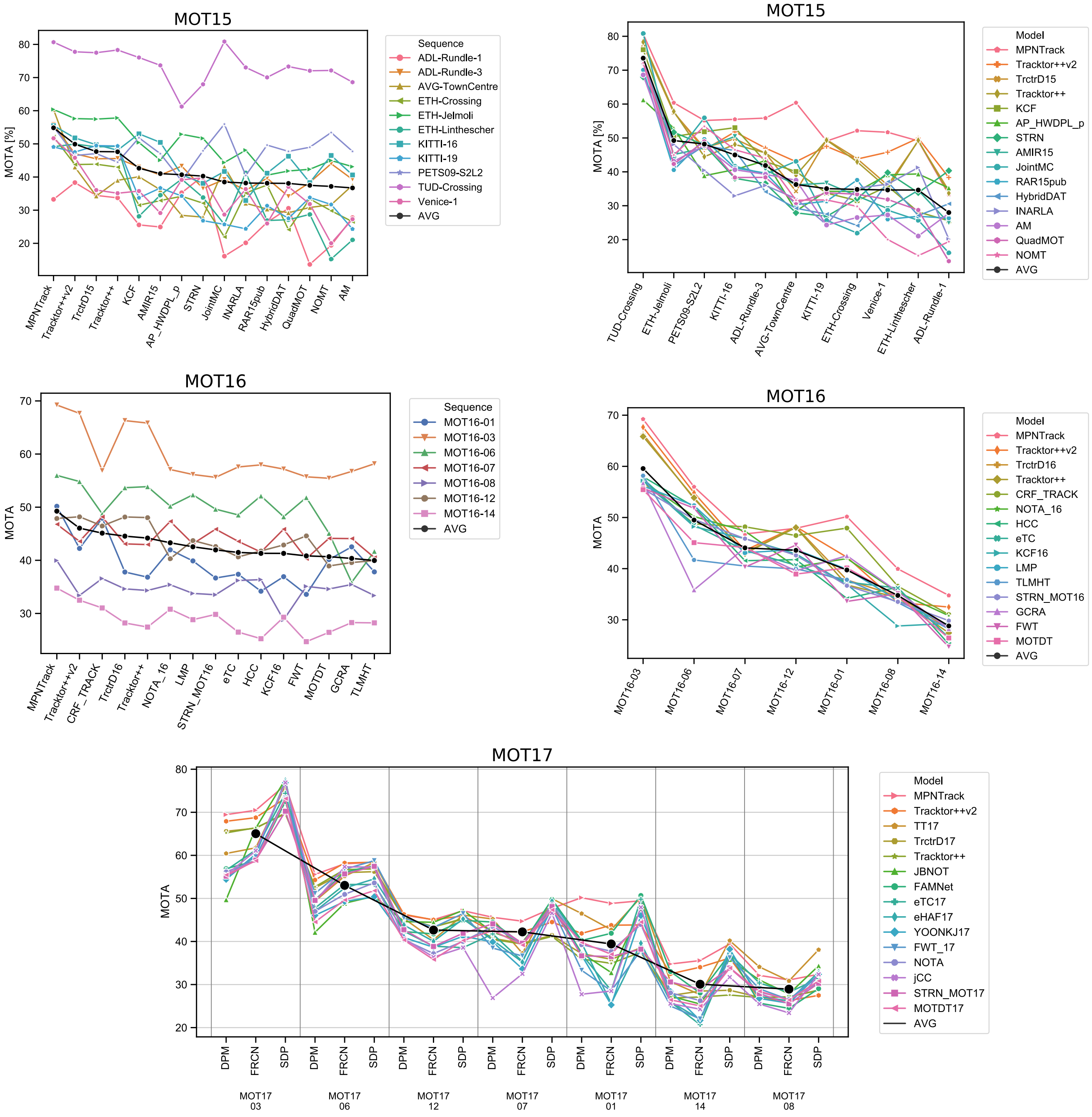


**Fig. 5** Graphical overview of the top 15 trackers of all benchmarks. The entries are ordered from easiest sequence/best performing method, to hardest sequence/poorest performance, respectively. The mean performance across all sequences/submissions is depicted with a thick black line

*MOT17*. A small subset of the submissions[10] were done by the benchmark organizers and not by the original authors of the respective method. Results for *MOT15* are summarized in Table 1, for *MOT16* in Table 2 and for *MOT17* in Table 3. The performance of the top 15 ranked trackers is demonstrated in Fig. 5.

### 7.1 Trends in Tracking

**Global optimization** The community has long used the paradigm of tracking-by-detection for MOT, i.e., dividing the task into two steps: (i) object detection and (ii) data association, or temporal linking between detections. The data association problem could be viewed as finding a set of disjoint paths in a graph, where nodes in the graph represent object detections, and links hypothesize feasible associa-

[10] The methods DP_NMS, TC_ODAL, TBD, SMOT, CEM, DCO_X, and LP2D were taken as baselines for the benchmark.

tions. Detectors usually produce multiple spatially-adjacent detection hypotheses, that are usually pruned using heuristic non-maximum suppression (NMS).

Before 2015, the community mainly focused on finding strong, preferably globally optimal methods to solve the data association problem. The task of linking detections into a consistent set of trajectories was often cast as, e.g., a graphical model and solved with k-shortest paths in DP_NMS (Pirsiavash et al. 2011), as a linear program solved with the simplex algorithm in LP2D (Leal-Taixé et al. 2011), as a Conditional Random Field in DCO_*X* (Milan et al. 2016), SegTrack (Milan et al. 2015), LTTSC-CRF (Le et al. 2016), and GMMCP (Dehghan et al. 2015), using joint probabilistic data association filter (JPDA) (Rezatofighi et al. 2015) or as a variational Bayesian model in OVBT (Ban et al. 2016).

A number of tracking approaches investigate the efficacy of using a Probability Hypothesis Density (PHD) filter-based tracking framework (Baisa 2019a; Baisa 2019b; Baisa and Wallace 2019; Fu et al. 2018; Sanchez-Matilla et al. 2016; Song and Jeon 2016; Song et al. 2019; Wojke and Paulus 2016). This family of methods estimate states of multiple targets and data association simultaneously, reaching 30.72% MOTA on *MOT15* (GMPHD_OGM), 41% and 40.42% on *MOT16* (PHD_GSDL and GMPHD_ReId, respectively) and 49.94% (GMPHD_OGM) on *MOT17*.

Newer methods (Tang et al. 2015) bypassed the need to pre-process object detections with NMS. They proposed a multi-cut optimization framework, which finds the connected components in a graph that represent feasible solutions, clustering all detections that correspond to the same target. This family of methods (JMC (Tang et al. 2016), LMP (Tang et al. 2017), NLLMPA (Levinkov et al. 2017), JointMC (Keuper et al. 2018), HCC (Ma et al. 2018b)) achieve 35.65% MOTA on *MOT15* (JointMC), 48.78% and 49.25% (LMP and HCC, respectively) on *MOT16* and 51.16% (JointMC) on *MOT17*.

**Motion Models** A lot of attention has also been given to motion models, used as additional association affinity cues, e.g., SMOT (Dicle et al. 2013), CEM (Milan et al. 2014), TBD (Geiger et al. 2014), ELP (McLaughlin et al. 2015) and MotiCon (Leal-Taixé et al. 2014). The pairwise costs for matching two detections were based on either simple distances or simple appearance models, such as color histograms. These methods achieve around 38% MOTA on *MOT16* (see Table 2) and 25% on *MOT15* (see Table 1).

**Hand-Crafted Affinity Measures** After that, the attention shifted towards building robust pairwise similarity costs, mostly based on strong appearance cues or a combination of geometric and appearance cues. This shift is clearly reflected in an improvement in tracker performance and the ability for trackers to handle more complex scenarios. For example, LINF1 (Fagot-Bouquet et al. 2016) uses sparse appearance models, and oICF (Kieritz et al. 2016) use appearance models based on integral channel features. Top-performing methods of this class incorporate long-term interest point trajectories, e.g., NOMT (Choi 2015), and, more recently, learned models for sparse feature matching JMC (Tang et al. 2016) and JointMC (Keuper et al. 2018) to improve pairwise affinity measures. As can be seen in Table 1, methods incorporating sparse flow or trajectories yielded a performance boost – in particular, NOMT is a top-performing method published in 2015, achieving MOTA of 33.67% on MOT15 and 46.42% on MOT16. Interestingly, the first methods outperforming NOMT on *MOT16* were published only in 2017 (AMIR (Sadeghian et al. 2017) and NLLMP (Levinkov et al. 2017)).

**Towards Learning** In 2015, we observed a clear trend towards utilizing learning to improve MOT.

LP_SSVM (Wang and Fowlkes 2016) demonstrates a significant performance boost by learning the parameters of linear cost association functions within a network flow tracking framework, especially when compared to methods using a similar optimization framework but hand-crafted association cues, e.g. Leal-Taixé et al. (2014). The parameters are learned using structured SVM (Taskar et al. 2003). MDP (Xiang et al. 2015) goes one step further and proposes to learn track management policies (birth/death/association) by modeling object tracks as Markov Decision Processes (Thrun et al. 2005). Standard MOT evaluation measures (Stiefelhagen et al. 2006) are not differentiable. Therefore, this method relies on reinforcement learning to learn these policies. As can be seen in Table 1, this method outperforms the majority of methods published in 2015 by a large margin and surpasses 30% MOTA on *MOT15*.

In parallel, methods start leveraging the representational power of deep learning, initially by utilizing transfer learning. MHT_DAM (Kim et al. 2015) learns to adapt appearance models online using multi-output regularized least squares. Instead of weak appearance features, such as color histograms, they extract base features for each object detection using a pre-trained convolutional neural network. With the combination of the powerful MHT tracking framework (Reid 1979) and online-adapted features used for data association, this method surpasses MDP and attains over 32% MOTA on *MOT15* and 45% MOTA on *MOT16*. Alternatively, JMC (Tang et al. 2016) and JointMC (Keuper et al. 2018) use a pre-learned deep matching model to improve the pairwise affinity measures. All aforementioned methods leverage pre-trained models.

**Learning Appearance Models** The next clearly emerging trend goes in the direction of learning appearance models for data association in end-to-end fashion directly on the target (i.e., *MOT15*, *MOT16*, *MOT17*) datasets. SiameseCNN (Leal-Taixe et al. 2016) trains a siamese convolutional neural network to learn spatio-temporal embeddings based on object appearance and estimated optical flow using contrastive loss (Hadsell et al. 2006). The learned embeddings

**Table 4** *MOT15*, *MOT16*, *MOT17* trackers and their characteristics

| Method | Box–box affinity | App. | Opt. | Extra inputs | OA | TR | ON |
|---|---|---|---|---|---|---|---|
| MPNTrack (Brasó and Leal-Taixé 2020) | Appearance, geometry (L) | ✓ | MCF, LP | MC | ✗ | ✓ | ✗ |
| DeepMOT (Xu et al. 2020) | Re-id (L) | ✓ | – | MC | ✗ | ✓ | ✓ |
| TT17 (Zhang et al. 2020) | Appearance, geometry (L) | ✓ | MHT/MWIS | ✗ | ✗ | ✗ | ✗ |
| CRF_TRACK (Xiang et al. 2020) | Appearance, geometry (L) | ✓ | CRF | re-id | ✗ | ✗ | ✗ |
| Tracktor (Bergmann et al. 2019) | Re-id (L) | ✓ | – | MC, re-id | ✗ | ✓ | ✓ |
| KCF (Chu et al. 2019) | Re-id (L) | ✓ | Multicut | re-id | ✓ | ✓ | ✓ |
| STRN (Xu et al. 2019) | Geometry, appearance (L) | ✓ | Hungarian algorithm | – | ✗ | ✗ | ✓ |
| JBNOT (Henschel et al. 2019) | Joint, body distances | ✗ | Frank–Wolfe algortihm | Body joint det. | ✗ | ✗ | ✗ |
| FAMNet (Chu and Ling 2019) | (L) | ✓ | Rank-1 tensor approx. | – | ✗ | ✓ | ✓ |
| MHT_bLSTM (Kim et al. 2018) | Appearance, motion (L) | ✓ | MHT/MWIS | Pre-trained CNN | ✓ | ✗ | ✓ |
| JointMC (Keuper et al. 2018) | DeepMatching (L), geometric | ✓ | Multicut | OF, non-nms dets | ✗ | ✗ | ✗ |
| RAR (Fang et al. 2018) | Appearance, motion (L) | ✓ | Hungarian algorithm | – | ✗ | ✗ | ✓ |
| HCC (Ma et al. 2018b) | Re-id (L) | ✓ | Multicut | External re-id | ○ | ✗ | ✗ |
| FWT (Henschel et al. 2018) | DeepMatching, geometric | ✓ | Frank-Wolfe algorithm | Head detector | ✗ | ✗ | ✗ |
| DMAN (Zhu et al. 2018) | Appearance (L), geometry | ✓ | – | – | ✓ | ✓ | ✓ |
| eHAF (Zhu et al. 2018) | Appearance, motion | ✓ | MHT/MWIS | Super-pixels, OF | ✗ | ✗ | ✗ |
| QuadMOT (Son et al. 2017) | Re-id (L), motion | ✓ | Min-max label prop. | – | ✓ | ✗ | ✓ |
| STAM (Chu et al. 2017) | Appearance (L), motion | ✓ | – | – | ✓ | ✓ | ✓ |
| AMIR (Sadeghian et al. 2017) | Motion, appearance, interactions (L) | ✓ | Hungarian algortihm | – | ✗ | ✗ | ✓ |
| LMP (Tang et al. 2017) | Re-id (L) | ✓ | Multicut | Non-nms det., re-id | ✓ | ✗ | ✗ |
| NLLMPa (Levinkov et al. 2017) | DeepMatching | ✓ | Multicut | Non-NMS dets | ✗ | ✗ | ✗ |
| LP_SSVM (Wang and Fowlkes 2016) | Appearance, motion (L) | ✓ | MCF, greedy | – | ✗ | ✗ | ✗ |
| SiameseCNN (Leal-Taixe et al. 2016) | Appearance (L), geometry, motion | ✓ | MCF, LP | OF | ✗ | ✗ | ✗ |
| SCEA (Yoon et al. 2016) | Appearance, geometry | ✓ | Clustering | – | ✗ | ✗ | ✓ |
| JMC (Tang et al. 2016) | DeepMatching | ✓ | Multicut | Non-NMS dets | ✗ | ✗ | ✗ |
| LINF1 (Fagot-Bouquet et al. 2016) | Sparse representation | ✓ | MCMC | – | ✗ | ✗ | ✗ |
| EAMTTpub (Sanchez-Matilla et al. 2016) | 2D distances | ✗ | Particle Filter | Non-NMS dets | ✗ | ✗ | ✓ |
| OVBT (Ban et al. 2016) | Dynamics from flow | ✓ | Variational EM | OF | ✗ | ✗ | ✓ |
| LTTSC-CRF (Le et al. 2016) | SURF | ✓ | CRF | SURF | ✗ | ✗ | ✗ |
| GMPHD_HDA (Song and Jeon 2016) | HoG similarity, color histogram | ✓ | GM-PHD filter | HoG | ✗ | ✗ | ✓ |
| DCO_X (Milan et al. 2016) | Motion, geometry | ✓ | CRF | – | ✗ | ✗ | ✗ |
| ELP (McLaughlin et al. 2015) | Motion | ✗ | MCF, LP | – | ✗ | ✗ | ✗ |
| GMMCP (Dehghan et al. 2015) | Appearance, motion | ✓ | GMMCP/CRF | – | ✗ | ✗ | ✗ |
| MDP (Xiang et al. 2015) | Motion (flow), geometry, appearance | ✓ | Hungarian algorithm | OF | ✗ | ✗ | ✓ |
| MHT_DAM (Kim et al. 2015) | (L) | ✓ | MHT/MWIS | – | ✓ | ✗ | ✓ |

**Table 4** continued

| Method | Box–box affinity | App. | Opt. | Extra inputs | OA | TR | ON |
|---|---|---|---|---|---|---|---|
| NOMT (Choi 2015) | Interest point traj. | ✓ | CRF | OF | ✗ | ✗ | ✗ |
| JPDA_m (Rezatofighi et al. 2015) | Mahalanobis distance | ✗ | LP | – | ✗ | ✗ | ✗ |
| SegTrack (Milan et al. 2015) | Shape, geometry, motion | ✓ | CRF | OF, super-pixels | ✗ | ✗ | ✗ |
| TBD (Geiger et al. 2014) | IoU + NCC | ✓ | Hungarian algorithm | – | ✗ | ✗ | ✗ |
| CEM (Milan et al. 2014) | Motion | ✗ | Greedy sampling | – | ✗ | ✗ | ✗ |
| MotiCon (Leal-Taixé et al. 2014) | Motion descriptors | ✗ | MCF, LP | OF | ✗ | ✗ | ✗ |
| SMOT (Dicle et al. 2013) | Target dynamics | ✗ | Hankel Least Squares | – | ✗ | ✗ | ✗ |
| DP_NMS (Pirsiavash et al. 2011) | 2D image distances | ✗ | k-shortest paths | – | ✗ | ✗ | ✗ |
| LP2D (Leal-Taixé et al. 2011) | 2D image distances, IoU | ✗ | MCF, LP | – | ✗ | ✗ | ✗ |

*App.* appearance model, *OA* online target appearance adaptation, *TR* target regression, *ON* online method, *(L)* learned. Components: *MC* motion compensation module, *OF* optical flow, *Re-id* learned re-identification module, *HoG* histogram of oriented gradients, *NCC* normalized cross-correlation, *IoU* intersection over union. **Association:** *GMMCP* Generalized maximum multi-clique problem, *MCF* Min-cost flow formulation (Zhang et al. 2008), *LP* linear programming, *MHT* multi-hypothesis tracking (Reid 1979), *MWIS* maximum independent set problem, *CRF* conditional random field formulation are then combined with contextual cues for robust data association. This method uses similar linear programming based optimization framework (Zhang et al. 2008) compared to LP_SSVM (Wang and Fowlkes 2016), however, it surpasses it significantly performance-wise, reaching 29% MOTA on *MOT15*. This demonstrates the efficacy of fine-tuning appearance models directly on the target dataset and utilizing convolutional neural networks. This approach is taken a step further with QuadMOT (Son et al. 2017), which similarly learns spatio-temporal embeddings of object detections. However, they train their siamese network using quadruplet loss (Chen et al. 2017b) and learn to place embedding vectors of temporally-adjacent detections instances closer in the embedding space. These methods reach 33.42% MOTA in *MOT15* and 41.1% on *MOT16*.

The learning process, in this case, is supervised. Different from that, HCC (Ma et al. 2018b) learns appearance models in an unsupervised manner. To this end, they train their method using object trajectories obtained from the test set using offline correlation clustering-based tracking framework (Levinkov et al. 2017). TO (Manen et al. 2016), on the other hand, proposes to mine detection pairs over consecutive frames using single object trackers to learn affinity measures which are plugged into a network flow optimization tracking framework. Such methods have the potential to keep improving affinity models on datasets for which ground-truth labels are not available.

**Online Appearance Model Adaptation** The aforementioned methods only learn general appearance embedding vectors for object detection and do not adapt the tracking target appearance models online. Further performance is gained by methods that perform such adaptation online (Chu et al. 2017; Kim et al. 2015, 2018; Zhu et al. 2018). MHT_bLSTM (Kim et al. 2018) replaces the multi-output regularized least-squares learning framework of MHT_DAM (Kim et al. 2015) with a bi-linear LSTM and adapts both the appearance model as well as the convolutional filters in an online fashion. STAM (Chu et al. 2017) and DMAN (Zhu et al. 2018) employ an ensemble of single-object trackers (SOTs) that share a convolutional backbone and learn to adapt the appearance model of the targets online during inference. They employ a spatio-temporal attention model that explicitly aims to prevent drifts in appearance models due to occlusions and interactions among the targets. Similarly, KCF (Chu et al. 2019) employs an ensemble of SOTs and updates the appearance model during tracking. To prevent drifts, they learn a tracking update policy using reinforcement learning. These methods achieve up to 38.9% MOTA on *MOT15*, 48.8% on MOT16 (KCF), and 50.71% on *MOT17* (MHT_DAM). Surprisingly, MHT_DAM out-performs its bilinear-LSTM variant (MHT_bLSTM achieves a MOTA of 47.52%) on *MOT17*.

**Learning to Combine Association Cues** A number of methods go beyond learning only the appearance model. Instead, these approaches learn to encode and combine heterogeneous association cues. SiameseCNN (Leal-Taixe et al. 2016) uses gradient boosting to combine learned appearance embeddings with contextual features. AMIR (Sadeghian et al. 2017) leverages recurrent neural networks in order to encode appearance, motion, pedestrian interactions and learns to combine these sources of information. STRN (Xu et al. 2019) proposes to leverage relational neural networks to learn to combine association cues, such as appearance, motion, and geometry. RAR (Fang et al. 2018) proposes recurrent autoregressive networks for learning a generative appearance and motion model for data association. These methods achieve 37.57% MOTA on *MOT15* and 47.17% on *MOT16*.

**Fine-Grained Detection** A number of methods employ additional fine-grained detectors and incorporate their outputs into affinity measures, e.g., a head detector in the case of FWT (Henschel et al. 2018), or a body joint detectors in JBNOT (Henschel et al. 2019), which are shown to help significantly with occlusions. The latter attains 52.63% MOTA on MOT17, which places it as the second-highest scoring method published in 2019.

**Tracking-by-Regression** Several methods leverage ensembles of (trainable) single-object trackers (SOTs), used to regress tracking targets from the detected objects, utilized in combination with simple track management (birth/death) strategies. We refer to this family of models as MOT-by-SOT or tracking-by-regression. We note that this paradigm for MOT departs from the traditional view of the multi-object tracking problem in computer vision as a generalized assignment problem (or multi-dimensional assignment problem), i.e. the problem of grouping object detections into a discrete set of tracks. Instead, methods based on target regression bring the focus back to the target state estimation. We believe the reasons for the success of these methods is two-fold: (i) rapid progress in learning-based SOT (Held et al. 2016; Li et al. 2018) that effectively leverages convolutional neural networks, and (ii) these methods can effectively utilize image evidence that is not covered by the given detection bounding boxes. Perhaps surprisingly, the most successful tracking-by-regression method, Tracktor (Bergmann et al. 2019), does not perform online appearance model updates (c.f., STAM, DMAN (Chu et al. 2017; Zhu et al. 2018) and KCF (Chu et al. 2019)). Instead, it simply re-purposes the regression head of the Faster R-CNN (Ren et al. 2015) detector, which is interpreted as the target regressor. This approach is most effective when combined with a motion compensation module and a learned re-identification module, attaining 46% MOTA on

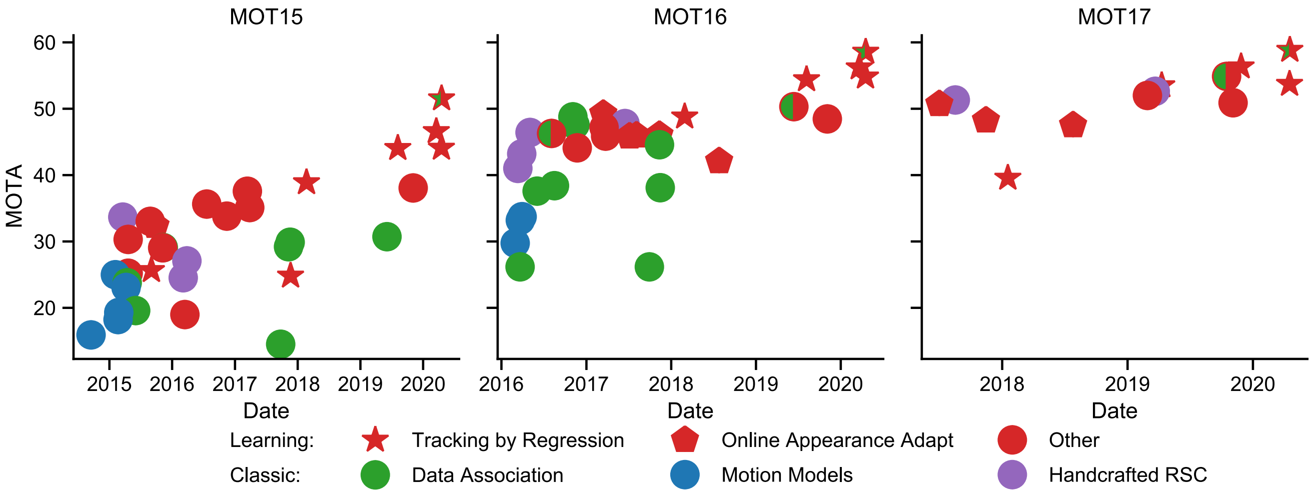


Fig. 6 Overview of tracker performances measured by their date of submission time and model type category

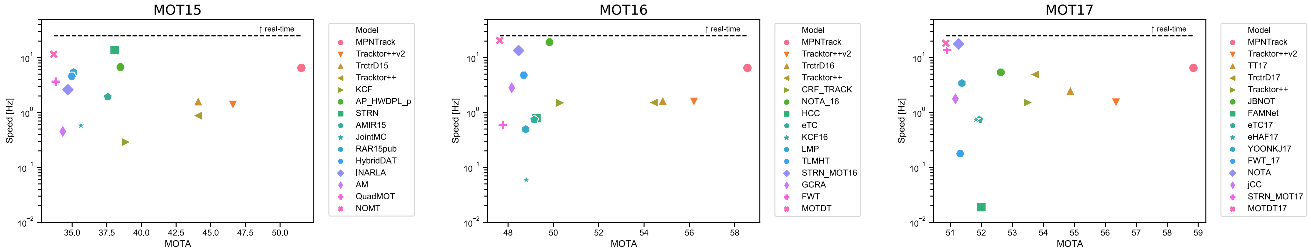


Fig. 7 Tracker performance measured by MOTA versus processing efficiency in frames per second for *MOT15*, *MOT16*, and *MOT17* on a log-scale. The latter is only indicative of the true value and has not been measured by the benchmark organizers. See text for details

MOT15 and 56% on MOT16 and MOT17, outperforming methods published in 2019 by a large margin.

**Towards End-to-End Learning** Even though tracking-by-regression methods brought substantial improvements, they are not able to cope with larger occlusions gaps. To combine the power of graph-based optimization methods with learning, MPNTrack (Brasó and Leal-Taixé 2020) proposes a method that leverages message-passing networks (Battaglia et al. 2016) to directly learn to perform data association via edge classification. By combining the regression capabilities of Tracktor (Bergmann et al. 2019) with a learned discrete neural solver, MPNTrack establishes a new state of the art, effectively using the best of both worlds—target regression and discrete data association. This method is the first one to surpass MOTA above 50% on *MOT15*. On the *MOT16* and *MOT17* it attains a MOTA of 58.56% and 58.85%, respectively. Nonetheless, this method is still not fully end-to-end trained, as it requires a projection step from the solution given by the graph neural network to the set of feasible solutions according to the network flow formulation and constraints.

Alternatively, (Xiang et al. 2020) uses MHT framework (Reid 1979) to link tracklets, while iteratively re-evaluating appearance/motion models based on progressively merged tracklets. This approach is one of the top on *MOT17*, achieving 54.87% MOTA.

In the spirit of combining optimization-based methods with learning, Zhang et al. (2020) revisits CRF-based tracking models and learns unary and pairwise potential functions in an end-to-end manner. On *MOT16*, this method attains MOTA of 50.31%.

We do observe trends towards learning to perform end-to-end MOT. To the best of our knowledge, the first method attempting this is RNN_LSTM (Milan et al. 2017), which jointly learns motion affinity costs and to perform bi-partite detection association using recurrent neural networks (RNNs). FAMNet (Chu and Ling 2019) uses a single network to extract appearance features from images, learns association affinities, and estimates multi-dimensional assignments of detections into object tracks. The multi-dimensional assignment is performed via a differentiable network layer that computes rank-1 estimation of the assignment tensor, which allows for back-propagation of the gradient. They perform learning with respect to binary cross-entropy loss between predicted assignments and ground-truth.

All aforementioned methods have one thing in common—they optimize network parameters with respect to proxy losses that do not directly reflect tracking quality, most commonly measured by the CLEAR-MOT evaluation measures (Stiefelhagen et al. 2006). To evaluate MOTA, the assignment between track predictions and ground truth needs to be established; this is usually performed using the Hungarian algorithm (Kuhn and Yaw 1955), which contains non-differentiable operations. To address this discrepancy DeepMOT (Xu et al. 2020) proposes the missing link—a differentiable matching layer that allows expressing a soft, differentiable variant of MOTA and MOTP.

**Conclusion** In summary, we observed that after an initial focus on developing algorithms for discrete data association (Dehghan et al. 2015; Le et al. 2016; Pirsiavash et al. 2011; Zhang et al. 2008), the focus shifted towards hand-crafting powerful affinity measures (Choi 2015; Kieritz et al. 2016; Leal-Taixé et al. 2014), followed by large improvements brought by learning powerful affinity models (Leal-Taixe et al. 2016; Son et al. 2017; Wang and Fowlkes 2016; Xiang et al. 2015).

In general, the major outstanding trends we observe in the past years all leverage the representational power of deep learning for learning association affinities, learning to adapt appearance models online (Chu et al. 2019, 2017; Kim et al. 2018; Zhu et al. 2018) and learning to regress tracking targets (Bergmann et al. 2019; Chu et al. 2019, 2017; Zhu et al. 2018). Figure 6 visualizes the promise of deep learning for tracking by plotting the performance of submitted models over time and by type.

The main common components of top-performing methods are: (i) learned single-target regressors (single-object trackers), such as (Held et al. 2016; Li et al. 2018), and (ii) re-identification modules (Bergmann et al. 2019). These methods fall short in bridging large occlusion gaps. To this end, we identified Graph Neural Network-based methods (Brasó and Leal-Taixé 2020) as a promising direction for future research. We observed the emergence of methods attempting to learn to track objects in end-to-end fashion instead of training individual modules of tracking pipelines (Chu and Ling 2019; Milan et al. 2017; Xu et al. 2020). We believe this is one of the key aspects to be addressed to further improve performance and expect to see more approaches leveraging deep learning for that purpose.

### 7.2 Runtime Analysis

Different methods require a varying amount of computational resources to track multiple targets. Some methods may require large amounts of memory while others need to be executed on a GPU. For our purpose, we ask each benchmark participant to provide the number of seconds required to produce the results on the entire dataset, regardless of the computational resources used. It is important to note that the resulting numbers are therefore only indicative of each approach and are not immediately comparable to one another.

Figure 7 shows the relationship between each submission's performance measured by MOTA and its efficiency in terms of frames per second, averaged over the entire dataset. There are two observations worth pointing out. First, the majority of methods are still far below real-time performance, which is assumed at 25 Hz. Second, the average processing rate $\sim$ 5 Hz does not differ much between the different sequences, which suggests that the different object densities (9 ped./fr. in *MOT15* and 26 ped./fr. in *MOT16*/*MOT17*) do not have a large impact on the speed the models. One explanation is that novel learning methods have an efficient forward computation, which does not vary much depending on the number of objects. This is in clear contrast to classic methods that relied on solving complex optimization problems at inference, which increased computation significantly as the pedestrian density increased. However, this conclusion has to be taken with caution because the runtimes are reported by the users on a trust base and cannot be verified by us.

### 7.3 Error Analysis

As we now, different applications have different requirements, e.g., for surveillance it is critical to have few false negatives, while for behavior analysis, having a false positive can mean computing wrong motion statistics. In this section, we take a closer look at the most common errors made by the tracking approaches. This simple analysis can guide researchers in choosing the best method for their task. In Fig. 8, we show the number of false negatives (FN, blue) and false positives (FP, red) created by the trackers on average with respect to the number of FN/FP of the object detector, used as an input. A ratio below 1 indicates that the trackers have improved in terms of FN/FP over the detector. We show the performance of the top 15 trackers, averaged over sequences. We order them according to MOTA from left to right in decreasing order.

We observe all top-performing trackers reduce the amount of FPs and FNs compared to the public detections. While the trackers reduce FPs significantly, FNs are decreased only slightly. Moreover, we can see a direct correlation between the FN and tracker performance, especially for *MOT16* and *MOT17* datasets, since the number of FNs is much larger than the number of FPs. The question is then, why are methods not focusing on reducing FNs? It turns out that "filling the gaps" between detections, what is commonly thought trackers should do, is not an easy task.

It is not until 2018 that we see methods drastically decreasing the number of FNs, and as a consequence, MOTA performance leaps forward. As shown in Fig. 6, this is due to the appearance of learning-based tracking-by-regression methods (Bergmann et al. 2019; Brasó and Leal-Taixé 2020;

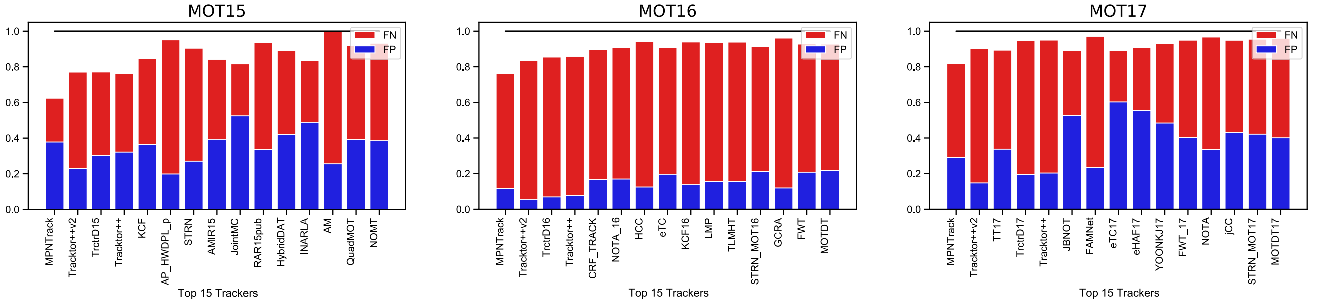


**Fig. 8** Detailed error analysis. The plots show the error ratios for trackers w.r.t detector (taken at the lowest confidence threshold), for two types of errors: false positives (FP) and false negatives (FN). Values above 1 indicate a higher error count for trackers than for detectors. Note that most trackers concentrate on removing false alarms provided by the detector at the cost of eliminating a few true positives, indicated by the higher FN count

Chu et al. 2017; Zhu et al. 2018). Such methods decrease the number of FNs the most by effectively using image evidence not covered by detection bounding boxes and regressing targets to areas where they are visible but missed by detectors. This brings us back to the common wisdom that trackers should be good at “filling the gaps“ between detections.

Overall, it is clear that *MOT17* still presents a challenge both in terms of detection as well as tracking. It will require significant further future efforts to bring performance to the next level. In particular, the next challenge that future methods will need to tackle is bridging large occlusion gaps, which can not be naturally resolved by methods performing target regression, as these only work as long as the target is (partially) visible.

## 8 Conclusion and Future Work

We have introduced *MOTChallenge*, a standardized benchmark for a fair evaluation of single-camera multi-person tracking methods. We presented its first two data releases with about 35,000 frames of footage and almost 700,000 annotated pedestrians. Accurate annotations were carried out following a strict protocol, and extra classes such as vehicles, sitting people, reflections, or distractors were also annotated in the second release to provide further information to the community.

We have further analyzed the performance of 101 trackers; 73 *MOT15*, 74 *MOT16*, and 57 on *MOT17* obtaining several insights. In the past, at the center of vision-based MOT were methods focusing on global optimization for data association. Since then, we observed that large improvements were made by hand-crafting strong affinity measures and leveraging deep learning for learning appearance models, used for better data association. More recent methods moved towards directly regressing bounding boxes, and learning to adapt target appearance models online. As the most promising recent trends that hold a large potential for future research, we identified the methods that are going in the direction of learning to track objects in an end-to-end fashion, combining optimization with learning.

We believe our Multiple Object Tracking Benchmark and the presented systematic analysis of existing tracking algorithms will help identify the strengths and weaknesses of the current state of the art and shed some light into promising future research directions.

**Acknowledgements** We would like to specially acknowledge Siyu Tang, Sarah Becker, Andreas Lin, and Kinga Milan for their help in the annotation process. We thank Bernt Schiele for helpful discussions and important insights into benchmarking. IDR gratefully acknowledges the support of the Australian Research Council through FL130100102. LLT acknowledges the support of the Sofja Kovalevskaja Award from the Humboldt Foundation, endowed by the Federal Ministry of Education and Research. DC acknowledges the support of the ERC Consolidator Grant *3D Reloaded*.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

# Appendices

## A Benchmark Submission

Our benchmark consists of the database and evaluation server on one hand, and the website as the user interface on the other. It is open to everyone who respects the submission policies (see next section). Before participating, every user is required to create an account, providing an institutional and not a generic e-mail address.[11]

After registering, the user can create a new tracker with a unique name and enter all additional details. It is mandatory to indicate:

- the full name and a brief description of the method
- a reference to the publication of the method, if already existing,
- whether the method operates online or on a batch of frames and whether the source code is publicly available,
- whether only the provided or also external training and detection data were used.

After creating all details of a new tracker, it is possible to assign open challenges to this tracker and submit results to the different benchmarks. To participate in a challenge the user has to provide the following information for each challenge they want to submit to:

- name of the challenge in which the tracker will be participating,
- a reference to the publication of the method, if already existing,
- the *total* runtime in seconds for computing the results for the test sequences and the hardware used, and
- whether only provided data was used for training, or also data from other sources were involved.

The user can then submit the results to the challenge in the format described in Sect. B.1. The tracking results are automatically evaluated and appear on the user's profile. The results are *not* automatically displayed in the public ranking table. The user can decide at any point in time to make the results public. Results can be published anonymously, e.g., to enable a blind review process for a corresponding paper. In this case, we ask to provide the venue and the paper ID or a similar unique reference. We request that a proper reference to the method's description is added upon acceptance of the paper. Anonymous entries are hidden from the benchmark after six months of inactivity.

The trackers and challenge meta information such as description, project page, runtime, or hardware can be edited at any time. Visual results of all public submissions, as well as annotations and detections, can be viewed and downloaded on the individual result pages of the corresponding tracker.

### A.1 Submission Policy

The main goal of this benchmark is to provide a platform that allows for objective performance comparison of multiple target tracking approaches on real-world data. Therefore, we introduce a few simple guidelines that must be followed by all participants.

*Training* Ground truth is only provided for the training sequences. It is the participant's own responsibility to find the best setting using *only* the training data. The use of additional training data must be indicated during submission and will be visible in the public ranking table. The use of ground truth labels on the test data is strictly forbidden. This or any other misuse of the benchmark will lead to the deletion of the participant's account and their results.

*Detections* We also provide a unique set of detections (see Sect. 4.2) for each sequence. We expect all tracking-by-detection algorithms to use the given detections. In case the user wants to present results with another set of detections or is not using detections at all, this should be clearly stated during submission and will also be displayed in the results table.

*Submission Frequency* Generally, we expect one single submission for a particular method per benchmark. If for any reason the user needs to re-compute and re-submit the results (e.g. due to a bug discovered in the implementation), they may do so after a waiting period of 72 h after the last submission to submit to the same challenge with any of their trackers. This policy should discourage the use of the benchmark server for training and parameter tuning on the test data. The number of submissions is counted and displayed for each method. We allow a maximum number of 4 submissions per tracker and challenge. We allow a user to create several tracker instances for different tracking models. However, a user can only create a new tracker every 30 days. Under *no* circumstances must anyone create a second account and attempt to re-submit in order to bypass the waiting period. Such behavior will lead to the deletion of the accounts and exclusion of the user from participating in the benchmark.

### A.2 Challenges and Workshops

We have two modalities for submission: the general open-end challenges and the special challenges. The main challenges, 2D MOT 2015, 3D MOT 2015, MOT16, and MOT17 are

[11] For accountability and to prevent abuse by using several email accounts.

always open for submission and are nowadays the standard evaluation platform for multi-target tracking methods submitting to computer vision conferences such as CVPR, ICCV or ECCV.

Special challenges are similar in spirit to the widely known PASCAL VOC series (Everingham et al. 2015), or the ImageNet competitions (Russakovsky et al. 2015). Each special challenge is linked to a workshop. The first edition of our series was the WACV 2015 Challenge that consisted of six outdoor sequences with both moving and static cameras, followed by the 2nd edition held in conjunction with ECCV 2016 on which we evaluated methods on the new *MOT16* sequences. The *MOT17* sequences were presented in the Joint Workshop on Tracking and Surveillance in conjunction with the Performance Evaluation of Tracking and Surveillance (PETS) (Ferryman and Ellis 2010; Ferryman and Shahrokni 2009) benchmark at the Conference on Vision and Pattern Recognition (CVPR) in 2017. The results and winning methods were presented during the respective workshops. Submission to those challenges is open only for a short period of time, i.e., there is a fixed submission deadline for all participants. Each method must have an accompanying paper presented at the workshop. The results of the methods are kept hidden until the date of the workshop itself when the winning method is revealed and a prize is awarded.

# B MOT 15

We have compiled a total of 22 sequences, of which we use half for training and half for testing. The annotations of the testing sequences are not released in order to avoid (over)fitting of the methods to the specific sequences. Nonetheless, the test data contains over 10 minutes of footage and 61,440 annotated bounding boxes, therefore, it is hard for researchers to over-tune their algorithms on such a large amount of data. This is one of the major strengths of the benchmark. We classify the sequences according to:

We classify the sequences according to:

- *Moving or static camera* the camera can be held by a person, placed on a stroller (Ess et al. 2008) or on a car (Geiger et al. 2012), or can be positioned fixed in the scene.
- *Viewpoint* the camera can overlook the scene from a high position, a medium position (at pedestrian's height), or at a low position.

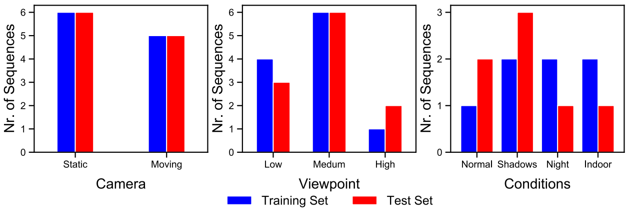


**Fig. 9** Comparison histogram between training and testing sequences of *static* versus *moving* camera, camera *viewpoint*: low, medium or high, *conditions*: normal, shadows, night or indoor

- *Weather* the illumination conditions in which the sequence was taken. Sequences with strong shadows and saturated parts of the image make tracking challenging, while night sequences contain a lot of motion blur, which is often a problem for detectors. Indoor sequences contain a lot of reflections, while the sequences classified as normal do not contain heavy illumination artifacts that potentially affect tracking.

We divide the sequences into training and testing to have a balanced distribution, as shown in Fig. 9.

## B.1 Data Format

All images were converted to JPEG and named sequentially to a 6-digit file name (e.g. 000001.jpg). Detection and annotation files are simple comma-separated value (CSV) files. Each line represents one object instance, and it contains 10 values as shown in Table 6.

The first number indicates in which frame the object appears, while the second number identifies that object as belonging to a trajectory by assigning a unique ID (set to −1 in a detection file, as no ID is assigned yet). Each object can be assigned to only one trajectory. The next four numbers indicate the position of the bounding box of the pedestrian in 2D image coordinates. The position is indicated by the top-left corner as well as the width and height of the bounding box. This is followed by a single number, which in the case of detections denotes their confidence score. The last three numbers indicate the 3D position in real-world coordinates of the pedestrian. This position represents the feet of the person. In the case of 2D tracking, these values will be ignored and can be left at −1.

**Table 5** Overview of the sequences currently included in the *MOT15* benchmark

| Name | FPS | Resolution | Length | Tracks | Boxes | Density | 3D | Camera | Viewpoint | Conditions |
|---|---|---|---|---|---|---|---|---|---|---|
| *Training sequences* | | | | | | | | | | |
| TUD-Stadtmitte (Andriluka et al. 2010) | 25 | 640 × 480 | 179 (00:07) | 10 | 1156 | 6.5 | Yes | Static | Medium | Normal |
| TUD-Campus (Andriluka et al. 2010) | 25 | 640 × 480 | 71 (00:03) | 8 | 359 | 5.1 | No | Static | Medium | Normal |
| PETS09-S2L1 (Ferryman and Ellis 2010) | 7 | 768 × 576 | 795 (01:54) | 19 | 4476 | 5.6 | Yes | Static | High | Normal |
| ETH-Bahnhof (Ess et al. 2008) | 14 | 640 × 480 | 1000 (01:11) | 171 | 5415 | 5.4 | Yes | Moving | Low | Normal |
| ETH-Sunnyday(Ess et al. 2008) | 14 | 640 × 480 | 354 (00:25) | 30 | 1858 | 5.2 | Yes | Moving | Low | Shadows |
| ETH-Pedcross2(Ess et al. 2008) | 14 | 640 × 480 | 840 (01:00) | 133 | 6263 | 7.5 | No | Moving | Low | Shadows |
| ADL-Rundle-6 (new) | 30 | 1920 × 1080 | 525 (00:18) | 24 | 5009 | 9.5 | No | Static | Low | Indoor |
| ADL-Rundle-8 (new) | 30 | 1920 × 1080 | 654 (00:22) | 28 | 6783 | 10.4 | No | Moving | Medium | Night |
| KITTI-13 (Geiger et al. 2012) | 10 | 1242 × 375 | 340 (00:34) | 42 | 762 | 2.2 | No | Moving | Medium | Shadows |
| KITTI-17 (Geiger et al. 2012) | 10 | 1242 × 370 | 145 (00:15) | 9 | 683 | 4.7 | No | Static | Medium | Shadows |
| Venice-2 (new) | 30 | 1920 × 1080 | 600 (00:20) | 26 | 7141 | 11.9 | No | Static | Medium | Normal |
| Total training | | | 5503 (06:29) | 500 | 39,905 | 7.3 | | | | |
| *Testing sequences* | | | | | | | | | | |
| TUD-Crossing (Andriluka et al. 2018) | 25 | 640 × 480 | 201 (00:08) | 13 | 1102 | 5.5 | No | Static | Medium | Normal |
| PETS09-S2L2 (Ferryman and Ellis 2010) | 7 | 768 × 576 | 436 (01:02) | 42 | 9641 | 22.1 | Yes | Static | High | Normal |
| ETH-Jelmoli (Ess et al. 2008) | 14 | 640 × 480 | 440 (00:31) | 45 | 2537 | 5.8 | Yes | Moving | Low | Shadows |
| ETH-Linthescher (Ess et al. 2008) | 14 | 640 × 480 | 1194 (01:25) | 197 | 8930 | 7.5 | Yes | Moving | Low | Shadows |
| ETH-Crossing (Ess et al. 2008) | 14 | 640 × 480 | 219 (00:16) | 26 | 1003 | 4.6 | No | Moving | Low | Normal |
| AVG-TownCentre (Benfold and Reid 2011) | 2.5 | 1920 × 1080 | 450 (03:45) | 226 | 7148 | 15.9 | Yes | Static | High | Normal |
| ADL-Rundle-1 (new) | 30 | 1920 × 1080 | 500 (00:17) | 32 | 9306 | 18.6 | No | Moving | Medium | Normal |
| ADL-Rundle-3 (new) | 30 | 1920 × 1080 | 625 (00:21) | 44 | 10166 | 16.3 | No | Static | Medium | Shadows |
| KITTI-16 (Geiger et al. 2012) | 10 | 1242 × 370 | 209 (00:21) | 17 | 1701 | 8.1 | No | Static | Medium | Shadows |
| KITTI-19 (Geiger et al. 2012) | 10 | 1242 × 374 | 1059 (01:46) | 62 | 5343 | 5.0 | No | Moving | Medium | Shadows |
| Venice-1 (new) | 30 | 1920 × 1080 | 450 (00:15) | 17 | 4563 | 10.1 | No | Static | Medium | Normal |
| Total testing | | | 5783 (10:07) | 721 | 61,440 | 10.6 | | | | |

An example of such a detection 2D file is:

```
1, -1, 794.2, 47.5, 71.2, 174.8, 67.5, -1, -1,
                          -1
1, -1, 164.1, 19.6, 66.5, 163.2, 29.4, -1, -1,
                          -1
1, -1, 875.4, 39.9, 25.3, 145.0, 19.6, -1, -1,
                          -1
2, -1, 781.7, 25.1, 69.2, 170.2, 58.1, -1, -1,
                          -1
```

For the ground truth and results files, the 7th value (confidence score) acts as a flag whether the entry is to be considered. A value of 0 means that this particular instance is ignored in the evaluation, while a value of 1 is used to mark it as active. An example of such an annotation 2D file is:

```
1, 1, 794.2, 47.5, 71.2, 174.8, 1, -1, -1, -1
1, 2, 164.1, 19.6, 66.5, 163.2, 1, -1, -1, -1
1, 3, 875.4, 39.9, 25.3,  35.0, 0, -1, -1, -1
2, 1, 781.7, 25.1, 69.2, 170.2, 1, -1, -1, -1
```

In this case, there are 2 pedestrians in the first frame of the sequence, with identity tags 1, 2. The third pedestrian is too small and therefore not considered, which is indicated with a flag value (7th value) of 0. In the second frame, we can see that pedestrian 1 remains in the scene. Note, that since this is a 2D annotation file, the 3D positions of the pedestrians are ignored and therefore are set to -1. All values including the bounding box are 1-based, i.e. the top left corner corresponds to (1, 1).

To obtain a valid result for the entire benchmark, a separate CSV file following the format described above must be created for each sequence and called `"Sequence-Name.txt"`. All files must be compressed into a single zip file that can then be uploaded to be evaluated.

## C MOT16 and MOT17 Release

Table 9 presents an overview of the *MOT16* and *MOT17* dataset.

### C.1 Annotation Rules

We follow a set of rules to annotate every moving person or vehicle within each sequence with a bounding box as accurately as possible. In this section, we define a clear protocol that was obeyed throughout the entire dataset annotations of *MOT16* and *MOT17* to guarantee consistency.

#### C.1.1 Target Class

In this benchmark, we are interested in tracking moving objects in videos. In particular, we are interested in evaluating multiple people tracking algorithms. Therefore, people will be the center of attention of our annotations. We divide the pertinent classes into three categories:

(i) *moving* or *standing* pedestrians;
(ii) people that are *not in an upright position* or artificial representations of humans; and
(iii) *vehicles* and *occluders*.

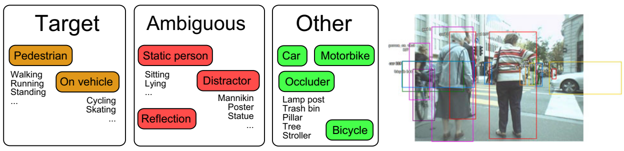


**Fig. 10** Left: An overview of annotated classes. The classes in orange will be the central ones to evaluate on. The red classes include ambiguous cases such that neither recovering nor missing will be penalized in the evaluation. The classes in green are annotated for training purposes and for computing the occlusion level of all pedestrians. Right: An exemplar of an annotated frame. Note how partially cropped objects are also marked outside of the frame. Also note that the bounding box encloses the entire person but not e.g. the white bag of Pedestrian 1 (bottom left)

In the first group, we annotate all moving or standing (upright) pedestrians that appear in the field of view and can be determined as such by the viewer. People on bikes or skateboards will also be annotated in this category (and are typically found by modern pedestrian detectors). Furthermore, if a person *briefly* bends over or squats, e.g. to pick something up or to talk to a child, they shall remain in the standard *pedestrian* class. The algorithms that submit to our benchmark are expected to track these targets.

In the second group, we include all people-like objects whose exact classification is ambiguous and can vary depending on the viewer, the application at hand, or other factors. We annotate all static people that are not in an upright position, e.g. sitting, lying down. We also include in this category any artificial representation of a human that might fire a detection response, such as mannequins, pictures, or reflections. People behind glass should also be marked as distractors. The idea is to use these annotations in the evaluation such that an algorithm is neither penalized nor rewarded for tracking, e.g., a sitting person or a reflection.

In the third group, we annotate all moving vehicles such as cars, bicycles, motorbikes and non-motorized vehicles (e.g. strollers), as well as other potential occluders. These annotations will not play any role in the evaluation, but are provided to the users both for training purposes and for computing the level of occlusion of pedestrians. Static vehicles (parked cars, bicycles) are not annotated as long as they do not occlude any pedestrians. The rules are summarized in Table 7, and in Fig. 10 we present a diagram of the classes of objects we annotate, as well as a sample frame with annotations.

**Table 6** Data format for the input and output files, both for detection and annotation files

| Position | Name | Description |
|---|---|---|
| 1 | Frame number | Indicate at which frame the object is present |
| 2 | Identity number | Each pedestrian trajectory is identified by a unique ID ($-1$ for detections) |
| 3 | Bounding box left | Coordinate of the top-left corner of the pedestrian bounding box |
| 4 | Bounding box top | Coordinate of the top-left corner of the pedestrian bounding box |
| 5 | Bounding box width | Width in pixels of the pedestrian bounding box |
| 6 | Bounding box height | Height in pixels of the pedestrian bounding box |
| 7 | Confidence score | Indicates how confident the detector is that this instance is a pedestrian. For the ground truth and results, it acts as a flag whether the entry is to be considered. |
| 8 | $x$ | 3D $x$ position of the pedestrian in real-world coordinates ($-1$ if not available) |
| 9 | $y$ | 3D $y$ position of the pedestrian in real-world coordinates ($-1$ if not available) |
| 10 | $z$ | 3D $z$ position of the pedestrian in real-world coordinates ($-1$ if not available) |

**Table 7** Annotation rules

| | |
|---|---|
| What? | *Targets*: all upright people including |
| | + walking, standing, running pedestrians |
| | + cyclists, skaters |
| | *Distractors*: static people or representations |
| | + people not in upright position (sitting, lying down) |
| | + reflections, drawings or photographs of people |
| | + human-like objects like dolls, mannequins |
| | *Others*: moving vehicles and other occluders |
| | + Cars, bikes, motorbikes |
| | + Pillars, trees, buildings |
| When? | Start as early as possible |
| | End as late as possible. |
| | Keep ID as long as the person is inside the field of view and its path can be determined unambiguously |
| How? | The bounding box should contain all pixels belonging to that person and at the same time be as tight as possible |
| Occlusions | Always annotate during occlusions if the position can be determined unambiguously |
| | If the occlusion is very long and it is not possible to determine the path of the object using simple reasoning (e.g. constant velocity assumption), the object will be assigned a new ID once it reappears |

### C.1.2 Bounding Box Alignment

The bounding box is aligned with the object's extent as accurately as possible. It should contain all object pixels belonging to that instance and at the same time be as tight as possible. This implies that a walking side-view pedestrian will typically have a box whose width varies periodically with the stride, while a front view or a standing person will maintain a more constant aspect ratio over time. If the person is partially occluded, the extent is estimated based on other available information such as expected size, shadows, reflections, previous and future frames and other cues. If a person is cropped by the image border, the box is estimated beyond the original frame to represent the entire person and to estimate the level of cropping. If an occluding object cannot be accurately enclosed in one box (e.g. a tree with branches or an escalator may require a large bounding box where most of the area does not belong to the actual object), then several boxes may be used to better approximate the extent of that object.

Persons on vehicles are only annotated separately from the vehicle when clearly visible. For example, children inside strollers or people inside cars are not annotated, while motorcyclists or bikers are.

### C.1.3 Start and End of Trajectories

The box (track) appears as soon as the person's location and extent can be determined precisely. This is typically the case when ≈ 10% of the person becomes visible. Similarly, the track ends when it is no longer possible to pinpoint the exact location. In other words, the annotation starts as early and ends as late as possible such that the accuracy is not forfeited. The box coordinates may exceed the visible area. A person leaving the field of view and re-appearing at a later point is assigned a new ID.

### C.1.4 Minimal Size

Although the evaluation will only take into account pedestrians that have a minimum height in pixels, annotations contain all objects of all sizes as long as they are distinguishable by the annotator. In other words, all targets are annotated independently of their sizes in the image.

### C.1.5 Occlusions

There is no need to explicitly annotate the level of occlusion. This value is be computed automatically using the annotations. We leverage the assumption that for two or more overlapping bounding boxes the object with the lowest y-value of the bounding box is closest to the camera and therefore occlude the other object behind it. Each target is fully annotated through occlusions as long as its extent and location can be determined accurately. If a target becomes completely occluded in the middle of a sequence and does not become visible later, the track is terminated (marked as 'outside of view'). If a target reappears after a prolonged period such that its location is ambiguous during the occlusion, it is assigned a new ID.

### C.1.6 Sanity Check

Upon annotating all sequences, a "sanity check" is carried out to ensure that no relevant entities are missed. To that end, we run a pedestrian detector on all videos and add all high-confidence detections that correspond to either humans or distractors to the annotation list.

## C.2 Data Format

All images were converted to JPEG and named sequentially to a 6-digit file name (e.g. 000001.jpg). Detection and annotation files are simple comma-separated value (CSV) files. Each line represents one object instance and contains 9 values as shown in Table 11.

The first number indicates in which frame the object appears, while the second number identifies that object as belonging to a trajectory by assigning a unique ID (set to $-1$ in a detection file, as no ID is assigned yet). Each object can be assigned to only one trajectory. The next four numbers indicate the position of the bounding box of the pedestrian in 2D image coordinates. The position is indicated by the top-left corner as well as the width and height of the bounding box. This is followed by a single number, which in the case of detections denotes their confidence score. The last two numbers for detection files are ignored (set to -1).

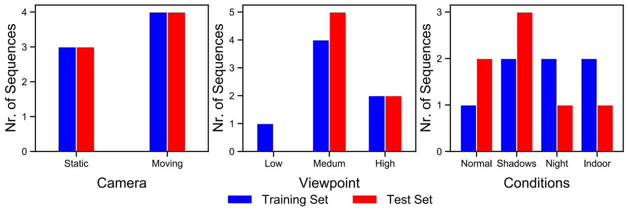


**Fig. 11** Comparison histogram between training and testing sequences of *MOT16*/*MOT17*: camera: static vs. moving camera, viewpoint: low, medium or high, conditions: normal, shadows, night or indoor

An example of such a 2D detection file is:

```
1, -1, 794.2, 47.5, 71.2, 174.8, 67.5, -1, -1
1, -1, 164.1, 19.6, 66.5, 163.2, 29.4, -1, -1
1, -1, 875.4, 39.9, 25.3, 145.0, 19.6, -1, -1
2, -1, 781.7, 25.1, 69.2, 170.2, 58.1, -1, -1
```

For the ground truth and result files, the 7th value (confidence score) acts as a flag whether the entry is to be considered. A value of 0 means that this particular instance is ignored in the evaluation, while a value of 1 is used to mark it as active. The 8th number indicates the type of object annotated, following the convention of Table 12. The last number shows the visibility ratio of each bounding box. This can be due to occlusion by another static or moving object, or to image border cropping.
An example of such an annotation 2D file is:

```
1, 1, 794.2, 47.5, 71.2, 174.8, 1, 1, 0.8
1, 2, 164.1, 19.6, 66.5, 163.2, 1, 1, 0.5
2, 4, 781.7, 25.1, 69.2, 170.2, 0, 12, 1.
```

In this case, there are 2 pedestrians in the first frame of the sequence, with identity tags 1, 2. In the second frame, we can see a reflection (class 12), which is to be considered by the evaluation script and will neither count as a false negative nor as a true positive, independent of whether it is correctly recovered or not. All values including the bounding box are 1-based, i.e. the top left corner corresponds to (1, 1).

To obtain a valid result for the entire benchmark, a separate CSV file following the format described above must be created for each sequence and called `"Sequence-Name.txt"`. All files must be compressed into a single ZIP file that can then be uploaded to be evaluated.

**Table 8** Overview of the types of annotations currently found in the *MOT16*/*MOT17* benchmark

| Sequence | Pedestrian | Person on vehicle | Car | Bicycle | Motorbike | Vehicle (non-mot.) | Static person | Distractor | Occluder (ground) | Occluder (full) | Refl. | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOT16/17-01 | 6395/6450 | 346 | 0/0 | 341 | 0 | 0 | 4790/5230 | 900 | 3150/4050 | 0 | 0/0 | 15,922/17,317 |
| MOT16/17-02 | 17,833/18,581 | 1549 | 0/0 | 1559 | 0 | 0 | 5271/5271 | 1200 | 1781/1843 | 0 | 0/0 | 29,193/30,003 |
| MOT16/17-03 | 104,556/104,675 | 70 | 1500/1500 | 12,060 | 1500 | 0 | 6000/6000 | 0 | 24,000/24,000 | 13,500 | 0/0 | 163,186/163,305 |
| MOT16/17-04 | 47,557/47,557 | 0 | 1050/1050 | 11,550 | 1050 | 0 | 4798/4798 | 0 | 23,100/23,100 | 18,900 | 0/0 | 108,005/108,005 |
| MOT16/17-05 | 6818/6917 | 315 | 196/196 | 315 | 0 | 11 | 0/0 | 16 | 0/235 | 0 | 0/0 | 7671/8013 |
| MOT16/17-06 | 11,538/11,784 | 150 | 0/0 | 118 | 0 | 0 | 269/269 | 238 | 109/109 | 0 | 0/299 | 12,422/12,729 |
| MOT16/17-07 | 16,322/16,893 | 0 | 0/0 | 0 | 0 | 0 | 2,023/2,023 | 0 | 1920/2420 | 0 | 0/131 | 20,265/21,504 |
| MOT16/17-08 | 16,737/21,124 | 0 | 0/0 | 0 | 0 | 0 | 1715/3535 | 2719 | 6875/6875 | 0 | 0/0 | 28,046/34,253 |
| MOT16/17-09 | 5257/5325 | 0 | 0/0 | 0 | 0 | 0 | 0/514 | 1575 | 1050/1050 | 0 | 948/1947 | 8830/10,411 |
| MOT16/17-10 | 12,318/12,839 | 0 | 25/25 | 0 | 0 | 0 | 1376/1376 | 470 | 2740/2740 | 0 | 0/0 | 16,929/17,450 |
| MOT16/17-11 | 9174/9436 | 0 | 0/0 | 0 | 0 | 0 | 0/82 | 306 | 596/596 | 0 | 0/181 | 10,076/10,617 |
| MOT16/17-12 | 8295/8667 | 0 | 0/0 | 0 | 0 | 0 | 1012/1036 | 763 | 1394/1710 | 0 | 0/953 | 11,464/13,272 |
| MOT16/17-13 | 11,450/11,642 | 0 | 4484/4918 | 103 | 0 | 0 | 0/0 | 4 | 2542/2733 | 680 | 0/122 | 19,263/20,202 |
| MOT16/17-14 | 18,483/18,483 | 0 | 1563/1563 | 0 | 0 | 0 | 712/712 | 47 | 4062/4062 | 393 | 0/0 | 25,260/25,294 |
| Total | 292,733/300,373 | 2430 | 8818/9252 | 26,046 | 2550 | 11 | 27,966/30,846 | 8238 | 73,319/75,523 | 33,473 | 948/3633 | 476,532/492,375 |

**Table 9** Overview of the sequences currently included in the *MOT16*/*MOT17* benchmark

| Name | FPS | Resolution | Length | Tracks | Boxes | Density | Camera | Viewpoint | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| *Training sequences* | | | | | | | | | |
| MOT16/17-02 (new) | 30 | 1920 × 1080 | 600 (00:20) | 54/62 | 17,833/18,581 | 29.7/31.0 | Static | Medium | Cloudy |
| MOT16/17-04 (new) | 30 | 1920 × 1080 | 1050 (00:35) | 83/83 | 47,557/47557 | 45.3/45.3 | Static | High | Night |
| MOT16/17-05 (Ess et al. 2008) | 14 | 640 × 480 | 837 (01:00) | 125/133 | 6818/6917 | 8.1/8.3 | Moving | Medium | Sunny |
| MOT16/17-09 (new) | 30 | 1920 × 1080 | 525 (00:18) | 25/26 | 5257/5325 | 10.0/10.1 | Static | Low | Indoor |
| MOT16/17-10 (new) | 30 | 1920 × 1080 | 654 (00:22) | 54/57 | 12,31812,839 | 18.8/19.6 | Moving | Medium | Night |
| MOT16/17-11 (new) | 30 | 1920 × 1080 | 900 (00:30) | 69/75 | 9174/9436 | 10.2/10.5 | Moving | Medium | Mndoor |
| MOT16/17-13 (new) | 25 | 1920 × 1080 | 750 (00:30) | 107/110 | 11,450/11,642 | 15.3/15.5 | Moving | High | Sunny |
| Total training | | | 5316 (03:35) | 517/546 | 110,407/112,297 | 20.8/21.1 | | | |
| *Testing sequences* | | | | | | | | | |
| MOT16/17-01 (new) | 30 | 1920 × 1080 | 450 (00:15) | 23/24 | 6,395/6,450 | 14.2/14.3 | Static | Medium | Cloudy |
| MOT16/17-03 (new) | 30 | 1920 × 1080 | 1,500 (00:50) | 148/148 | 104,556/104,675 | 69.7/69.8 | Static | High | Night |
| MOT16/17-06 (Ess et al. 2008) | 14 | 640 × 480 | 1,194 (01:25) | 221/222 | 11,538/11,784 | 9.7/9.9 | Moving | Medium | Sunny |
| MOT16/17-07 (new) | 30 | 1920 × 1080 | 500 (00:17) | 54/60 | 16,322/16.893 | 32.6/33.8 | Moving | Medium | Shadow |
| MOT16/17-08 (new) | 30 | 1920 × 1080 | 625 (00:21) | 63/76 | 16,737/21,124 | 26.8/33.8 | Static | Medium | Sunny |
| MOT16/17-12 (new) | 30 | 1920 × 1080 | 900 (00:30) | 86/91 | 8,295/8,667 | 9.2/9.6 | Moving | Medium | Indoor |
| MOT16/17-14 (new) | 25 | 1920 × 1080 | 750 (00:30) | 164/164 | 18,483/18,483 | 24.6/24.6 | Moving | High | Sunny |
| Total testing | | | 5919 (04:08) | 759/785 | 182,326/188,076 | 30.8/31.8 | | | |

**Table 10** Detection bounding box statistics

| Seq | MOT16 | | MOT17 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | DPM | | DPM | | FRCNN | | SDP | |
| | nDet. | nDet./fr. | nDet. | nDet./fr. | nDet. | nDet./fr. | nDet. | nDet./fr. |
| MOT16/17-01 | 3775 | 8.39 | 3775 | 8.39 | 5514 | 12.25 | 5837 | 12.97 |
| MOT16/17-02 | 7267 | 12.11 | 7267 | 12.11 | 8186 | 13.64 | 11,639 | 19.40 |
| MOT16/17-03 | 85,854 | 57.24 | 85,854 | 57.24 | 65,739 | 43.83 | 80,241 | 53.49 |
| MOT16/17-04 | 39,437 | 37.56 | 39,437 | 37.56 | 28,406 | 27.05 | 37,150 | 35.38 |
| MOT16/17-05 | 4333 | 5.20 | 4333 | 5.20 | 3848 | 4.60 | 4767 | 5.70 |
| MOT16/17-06 | 7851 | 6.58 | 7851 | 6.58 | 7809 | 6.54 | 8283 | 6.94 |
| MOT16/17-07 | 11,309 | 22.62 | 11,309 | 22.62 | 9377 | 18.75 | 10,273 | 20.55 |
| MOT16/17-08 | 10,042 | 16.07 | 10,042 | 16.07 | 6921 | 11.07 | 8118 | 12.99 |
| MOT16/17-09 | 5976 | 11.38 | 5976 | 11.38 | 3049 | 5.81 | 3607 | 6.87 |
| MOT16/17-10 | 8832 | 13.50 | 8832 | 13.50 | 9701 | 14.83 | 10,371 | 15.86 |
| MOT16/17-11 | 8590 | 9.54 | 8590 | 9.54 | 6007 | 6.67 | 7509 | 8.34 |
| MOT16/17-12 | 7764 | 8.74 | 7764 | 8.74 | 4726 | 5.32 | 5440 | 6.09 |
| MOT16/17-13 | 5355 | 7.22 | 5355 | 7.22 | 8442 | 11.26 | 7744 | 10.41 |
| MOT16/17-14 | 8781 | 11.71 | 8781 | 11.71 | 10,055 | 13.41 | 10,461 | 13.95 |
| Total | 215,166 | 19.19 | 215,166 | 19.19 | 177,780 | 15.84 | 211,440 | 18.84 |

**Table 11** Data format for the input and output files, both for detection (DET) and annotation/ground truth (GT) files

| Position | Name | Description |
|---|---|---|
| 1 | Frame number | Indicate at which frame the object is present |
| 2 | Identity number | Each pedestrian trajectory is identified by a unique ID ($-1$ for detections) |
| 3 | Bounding box left | Coordinate of the top-left corner of the pedestrian bounding box |
| 4 | Bounding box top | Coordinate of the top-left corner of the pedestrian bounding box |
| 5 | Bounding box width | Width in pixels of the pedestrian bounding box |
| 6 | Bounding box height | Height in pixels of the pedestrian bounding box |
| 7 | Confidence score | DET: Indicates how confident the detector is that this instance is a pedestrian.<br>GT: It acts as a flag whether the entry is to be considered (1) or ignored (0). |
| 8 | Class | GT: Indicates the type of object annotated |
| 9 | Visibility | GT: Visibility ratio, a number between 0 and 1 that says how much of that object is visible. Can be due to occlusion and due to image border cropping |

## D Implementation Details of the Evaluation

In this section, we detail how to compute false positives, false negatives, and identity switches, which are the basic units for the evaluation metrics presented in the main paper. We also explain how the evaluation deals with special non-target cases: people behind a window or sitting people.

### D.1 Tracker-to-Target Assignment

There are two common prerequisites for quantifying the performance of a tracker. One is to determine for each hypothesized output, whether it is a true positive (TP) that describes an actual (annotated) target, or whether the output is a false alarm (or false positive, FP). This decision is typically made by thresholding based on a defined distance (or dissimilarity) measure $d$ between the coordinates of the true and predicted box placed around a target (see Sect. D.2). A target that is missed by any hypothesis is a false negative

**Table 12** Label classes present in the annotation files and ID appearing in the 7th column of the files as described in Table 11

| Label | ID |
|---|---|
| Pedestrian | 1 |
| Person on vehicle | 2 |
| Car | 3 |
| Bicycle | 4 |
| Motorbike | 5 |
| Non motorized vehicle | 6 |
| Static person | 7 |
| Distractor | 8 |
| Occluder | 9 |
| Occluder on the ground | 10 |
| Occluder full | 11 |
| Reflection | 12 |

(FN). A good result is expected to have as few FPs and FNs as possible. Next to the absolute numbers, we also show the false positive ratio measured by the number of false alarms per frame (FAF), sometimes also referred to as false positives per image (FPPI) in the object detection literature.

The same target may be covered by multiple outputs. The second prerequisite before computing the numbers is then to establish the correspondence between all annotated and hypothesized objects under the constraint that a true object should be recovered at most once, and that one hypothesis cannot account for more than one target.

For the following, we assume that each ground-truth trajectory has one unique start and one unique endpoint, i.e., that it is not fragmented. Note that the current evaluation procedure does not explicitly handle target re-identification. In other words, when a target leaves the field-of-view and then reappears, it is treated as an unseen target with a new ID. As proposed in Stiefelhagen et al. (2006), the optimal matching is found using Munkres (a.k.a. Hungarian) algorithm. However, dealing with video data, this matching is not performed independently for each frame, but rather considering a temporal correspondence. More precisely, if a ground-truth object $i$ is matched to hypothesis $j$ at time $t-1$ *and* the distance (or dissimilarity) between $i$ and $j$ in frame $t$ is below $t_d$, then the correspondence between $i$ and $j$ is carried over to frame $t$ even if there exists another hypothesis that is closer to the actual target. A mismatch error (or equivalently an identity switch, IDSW) is counted if a ground-truth target $i$ is matched to track $j$ and the last known assignment was $k \neq j$. Note that this definition of ID switches is more similar to (Li et al. 2009) and stricter than the original one (Stiefelhagen et al. 2006). Also note that, while it is certainly desirable to keep the number of ID switches low, their absolute number alone is not always expressive to assess the overall performance, but should rather be considered concerning the number of recovered targets. The intuition is that a method that finds twice as many trajectories will almost certainly produce more identity switches. For that reason, we also state the relative number of ID switches, which is computed as IDSW / Recall.

These relationships are illustrated in Fig. 12. For simplicity, we plot ground-truth trajectories with dashed curves, and the tracker output with solid ones, where the color represents a unique target ID. The grey areas indicate the matching threshold (see Sect. D.3). Each true target that has been successfully recovered in one particular frame is represented with a filled black dot with a stroke color corresponding to its matched hypothesis. False positives and false negatives are plotted as empty circles. See figure caption for more details.

After determining true matches and establishing correspondences it is now possible to compute the metrics. We do so by concatenating all test sequences and evaluating the entire benchmark. This is in general more meaningful than averaging per-sequences figures because of the large variation on the number of targets per sequence.

## D.2 Distance Measure

The relationship between ground-truth objects and a tracker output is established using bounding boxes on the image plane. Similar to object detection (Everingham et al. 2015), the intersection over union (a.k.a. the Jaccard index) is usually employed as the similarity criterion, while the threshold $t_d$ is set to 0.5 or 50%.

## D.3 Target-Like Annotations

People are a common object class present in many scenes, but should we track all people in our benchmark? For example, should we track static people sitting on a bench? Or people on bicycles? How about people behind a glass? We define the target class of *MOT16* and *MOT17* as *all upright people, standing or walking, that are reachable along the viewing ray without a physical obstacle.* For instance, reflections or people behind a transparent wall or window are excluded. We also exclude from our target class people on bicycles (riders) or other vehicles.

For all these cases where the class is very similar to our target class (see Fig. 13), we adopt a similar strategy as in (Mathias et al. 2014). That is, a method is neither penalized nor rewarded for tracking or not tracking those similar classes. Since a detector is likely to fire in those cases, we do not want to penalize a tracker with a set of false positives for properly following that set of detections, i.e., of a person on a bicycle. Likewise, we do not want to penalize with false negatives a tracker that is based on motion cues and therefore does not track a sitting person.

To handle these special cases, we adapt the tracker-to-target assignment algorithm to perform the following steps:

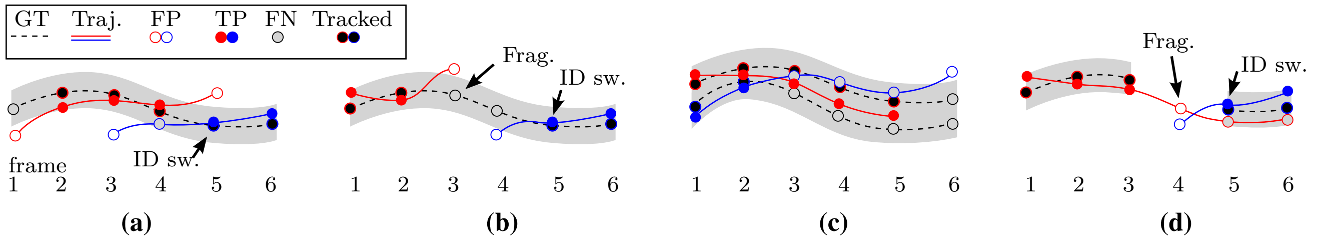


**Fig. 12** Four cases illustrating tracker-to-target assignments. **a** An ID switch occurs when the mapping switches from the previously assigned red track to the blue one. **b** A track fragmentation is counted in frame 3 because the target is tracked in frames 1–2, then interrupts, and then reacquires its 'tracked' status at a later point. A new (blue) track hypothesis also causes an ID switch at this point. **c** Although the tracking results are reasonably good an optimal single-frame assignment in frame 1 is propagated through the sequence, causing 5 missed targets (FN) and 4 false positives (FP). Note that no fragmentations are counted in frames 3 and 6 because tracking of those targets is not resumed at a later point. **d** A degenerate case illustrating that target re-identification is not handled correctly. An interrupted ground-truth trajectory will typically cause a fragmentation. Also note the less intuitive ID switch, which is counted because blue is the closest target in frame 5 that is not in conflict with the mapping in frame 4

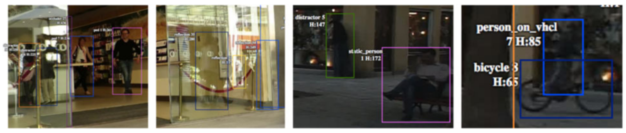


**Fig. 13** The annotations include different classes of objects similar to the target class, a pedestrian in our case. We consider these special classes (distractor, reflection, static person and person on vehicle) to be so similar to the target class that a tracker should neither be penalized nor rewarded for tracking them in the sequence (Color figure online)

1. At each frame, all bounding boxes of the result file are matched to the ground truth via the Hungarian algorithm.
2. All result boxes that overlap more than the matching threshold ($> 50\%$) with one of these classes (distractor, static person, reflection, person on vehicle) excluded from the evaluation.
3. During the final evaluation, *only* those boxes that are annotated as *pedestrians* are used.

## References

Alahi, A., Ramanathan, V., & Fei-Fei, L. (2014). Socially-aware large-scale crowd forecasting. In *Conference on computer vision and pattern recognition*.

Andriluka, M., Roth, S., & Schiele, B. (2010). Monocular 3D pose estimation and tracking by detection. In *Conference on computer vision and pattern recognition*.

Andriluka, M., Iqbal, U., Insafutdinov, E., Pishchulin, L., Milan, A., Gall, J., & Schiele, B. (2018). Posetrack: A benchmark for human pose estimation and tracking. In *Conference on computer vision and pattern recognition*.

Babaee, M., Li, Z., & Rigoll, G. (2019). A dual CNN-RNN for multiple people tracking. *Neurocomputing*, *368*, 69–83.

Bae, S.-H., & Yoon, K.-J. (2014). Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Conference on computer vision and pattern recognition*.

Bae, S.-H., & Yoon, K.-J. (2018). Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 595–610.

Baisa, N. L. (2018). Online multi-target visual tracking using a HISP filter. In *International joint conference on computer vision, imaging and computer graphics theory and applications*.

Baisa, N. L. (2019a). Online multi-object visual tracking using a GM-PHD filter with deep appearance learning. In *International conference on information fusion*.

Baisa, N. L. (2019b). Occlusion-robust online multi-object visual tracking using a GM-PHD filter with a CNN-based re-identification. arXiv preprint arXiv:1912.05949.

Baisa, N. L. (2019c). Robust online multi-target visual tracking using a HISP filter with discriminative deep appearance learning. arXiv preprint arXiv:1908.03945.

Baisa, N. L., & Wallace, A. (2019). Development of a n-type GM-PHD filter for multiple target, multiple type visual tracking. *Journal of Visual Communication and Image Representation*, *59*, 257–271.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, *92*(1), 1–31.

Ban, Y., Ba, S., Alameda-Pineda, X., & Horaud, R. (2016). Tracking multiple persons based on a variational Bayesian model. In *European conference on computer vision workshops*.

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. (2016). Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*.

Benfold, B., & Reid, I. (2011). Unsupervised learning of a scene-specific coarse gaze estimator. In *International conference on computer vision*.

Bergmann, P., Meinhardt, T., & Leal-Taixé, L. (2019). Tracking without bells and whistles. In *International conference on computer vision*.

Bernardin, K., & Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*,. https://doi.org/10.1155/2008/246309.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016a). Simple online and realtime tracking. In *International conference on image processing*.

Bewley, A., Ott, L., Ramos, F., & Upcroft, B. (2016b). Alextrac: Affinity learning by exploring temporal reinforcement within association chains. In *International conference on robotics and automation*.

Bochinski, E., Eiselein, V., & Sikora, T. (2017). High-speed tracking-by-detection without using image information. In *International conference on advanced video and signal based surveillance*.

Boragule, A., & Jeon, M. (2017). Joint cost minimization for multi-object tracking. *International conference on advanced video and signal based surveillance*.

Brasó, G., & Leal-Taixé, L. (2020). Learning a neural solver for multiple object tracking. In *Conference on computer vision and pattern recognition*.

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., & Hays, J. (2019). Argoverse: 3D tracking and forecasting with rich maps. In *Conference on computer vision and pattern recognition*.

Chen, J., Sheng, H., Zhang, Y., & Xiong, Z. (2017a). Enhancing detection model for multiple hypothesis tracking. In *Conference on computer vision and pattern recognition workshops*.

Chen, L., Ai, H., Chen, R., & Zhuang, Z. (2019). Aggregate tracklet appearance features for multi-object tracking. *Signal Processing Letters*, *26*(11), 1613–1617.

Chen, W., Chen, X., Zhang, J., & Huang, K. (2017b). Beyond triplet loss: A deep quadruplet network for person re-identification. In *Conference on computer vision and pattern recognition*.

Choi, W. (2015). Near-online multi-target tracking with aggregated local flow descriptor. In *International conference on computer vision*.

Chu, P., Fan, H., Tan, C. C., & Ling, H. (2019). Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *Winter conference on applications of computer vision*.

Chu, P., & Ling, H. (2019). FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *International conference on computer vision*.

Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., & Yu, N. (2017). Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *International conference on computer vision*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Conference on computer vision and pattern recognition workshops*.

Dave, A., Khurana, T., Tokmakov, P., Schmid, C., & Ramanan, D. (2020) Tao: A large-scale benchmark for tracking any object. In *European conference on computer vision*.

Dehghan, A., Assari, S. M., & Shah, M. (2015) GMMCP-tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Conference on computer vision and pattern recognition workshops*.

Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-Taixe, L. (2019). Cvpr19 tracking and detection challenge: How crowded can it get? arXiv preprint arXiv:1906.04567.

Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-Taixé, L. (2020). MOT20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003.

Dicle, C., Camps, O., & Sznaier, M. (2013) The way they move: Tracking targets with similar appearance. In *International conference on computer vision*.

Dollár, P., Appel, R., Belongie, S., & Perona, P. (2014). Fast feature pyramids for object detection. *Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1532–1545.

Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2009) Pedestrian detection: A benchmark. In *Conference on computer vision and pattern recognition workshops*.

Eiselein, V., Arp, D., Pätzold, M., & Sikora, T. (2012). Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *International conference on advanced video and signal-based surveillance*.

Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *Conference on computer vision and pattern recognition*.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The Pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.

Fagot-Bouquet, L., Audigier, R., Dhome, Y., & Lerasle, F. (2015). Online multi-person tracking based on global sparse collaborative representations. In *International conference on image processing*.

Fagot-Bouquet, L., Audigier, R., Dhome, Y., & Lerasle, F. (2016). Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *European conference on computer vision workshops*.

Fang, K., Xiang, Y., Li, X., & Savarese, S. (2018). Recurrent autoregressive networks for online multi-object tracking. In *Winter conference on applications of computer vision*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2006) Efficient belief propagation for early vision. In *Conference on computer vision and pattern recognition*.

Ferryman, J., & Ellis, A. (2010) PETS2010: Dataset and challenge. In *International conference on advanced video and signal based surveillance*.

Ferryman, J., & Shahrokni, A. (2009). PETS2009: Dataset and challenge. In *International workshop on performance evaluation of tracking and surveillance*.

Fu, Z., Angelini, F., Chambers, J., & Naqvi, S. M. (2019). Multi-level cooperative fusion of GM-PHD filters for online multiple human tracking. *Transactions on Multimedia*, *21*(9), 2277–2291.

Fu, Z., Feng, P., Angelini, F., Chambers, J. A., & Naqvi, S. M. (2018). Particle PHD filter based multiple human tracking using online group-structured dictionary learning. *Access*, *6*, 14764–14778.

Geiger, A., Lauer, M., Wojek, C., Stiller, C., & Urtasun, R. (2014). 3D traffic scene understanding from movable platforms. *Transactions on Pattern Analysis and Machine Intelligence*, *36*(5), 1012–1025.

Geiger, A., Lenz, P., & Urtasun, R. (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on computer vision and pattern recognition*.

Girshick, R. (2015). Fast R-CNN. In *International conference on computer vision*.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Conference on computer vision and pattern recognition*.

Held, D., Thrun, S., & Savarese, S. (2016). Learning to track at 100 fps with deep regression networks. In *European conference on computer vision*.

Henriques, J. a., Caseiro, R., & Batista, J. (2011). Globally optimal solution to multi-object tracking with merged measurements. In *International conference on computer vision*.

Henschel, R., Leal-Taixé, L., Cremers, D., & Rosenhahn, B. (2018). Fusion of head and full-body detectors for multi-object tracking. In *Conference on computer vision and pattern recognition workshops*.

Henschel, R., Zou, Y., & Rosenhahn, B. (2019). Multiple people tracking using body and joint detections. In *Conference on computer vision and pattern recognition workshops*.

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachussetts, Amherst.

Ju, J., Kim, D., Ku, B., Han, D., & Ko, H. (2017a). Online multi-object tracking with efficient track drift and fragmentation handling. *Journal of the Optical Society of America A*, *34*(2), 280–293.

Ju, J., Kim, D., Ku, B., Han, D. K., & Ko, H. (2017b). Online multi-person tracking with two-stage data association and online appearance model learning. *IET Computer Vision*, *11*(1), 87–95.

Karunasekera, H., Wang, H., & Zhang, H. (2019). Multiple object tracking with attention to appearance, structure, motion and size. *Access*,. https://doi.org/10.1109/ACCESS.2019.2932301.

Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., et al. (2019) Lyft level 5 av dataset 2019. https://level5.lyft.com/dataset/.

Keuper, M., Tang, S., Andres, B., Brox, T., & Schiele, B. (2018). Motion segmentation and multiple object tracking by correlation co-clustering. *Transactions on Pattern Analysis and Machine Intelligence*,. https://doi.org/10.1109/TPAMI.2018.2876253.

Kieritz, H., Becker, S., Häbner, W., & Arens, M. (2016). Online multi-person tracking using integral channel features. In *International conference on advanced video and signal based surveillance*.

Kim, C., Li, F., Ciptadi, A., & Rehg, J. M. (2015). Multiple hypothesis tracking revisited. In *International conference on computer vision*.

Kim, C., Li, F., & Rehg, J. M. (2018). Multi-object tracking with neural gating using bilinear LSTM. In *European conference on computer vision*.

Kristan, M., et al. (2014). The visual object tracking VOT2014 challenge results. In *European conference on computer vision workshops*.

Kuhn, H. W., & Yaw, B. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–97.

Kutschbach, T., Bochinski, E., Eiselein, V., & Sikora, T. (2017). Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *International conference on advanced video and signal based surveillance*.

Lan, L., Wang, X., Zhang, S., Tao, D., Gao, W., & Huang, T. S. (2018). Interacting tracklets for multi-object tracking. *Transactions on Image Processing*, *27*(9), 4585–4597.

Le, N., Heili, A., & Odobez, J.-M. (2016). Long-term time-sensitive costs for CRF-based tracking by detection. In *European conference on computer vision workshops*.

Leal-Taixe, L., Canton-Ferrer, C., & Schindler, K. (2016). Learning by tracking: Siamese CNN for robust target association. In *Conference on computer vision and pattern recognition workshops*.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., & Savarese, S. (2014). Learning an image-based motion context for multiple people tracking. In *Conference on computer vision and pattern recognition*.

Leal-Taixé, L., Pons-Moll, G., & Rosenhahn, B. (2011). Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *International conference on computer vision workshops*.

Lee, S., & Kim, E. (2019). Multiple object tracking via feature pyramid Siamese networks. *Access*, *7*, 8181–8194.

Lee, S.-H., Kim, M.-Y., & Bae, S.-H. (2018). Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures. *Access*, *6*, 67316–67328.

Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., & Andres, B. (2017). Joint graph decomposition and node labeling: Problem, algorithms, applications. In *Conference on computer vision and pattern recognition*.

Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with Siamese region proposal network. In *Conference on computer vision and pattern recognition*.

Li, Y., Huang, C., & Nevatia, R. (2009). Learning to associate: Hybrid boosted multi-target tracker for crowded scene. In *Conference on computer vision and pattern recognition*.

Liu, Q., Liu, B., Wu, Y., Li, W., & Yu, N. (2019). Real-time online multi-object tracking in compressed domain. *Access*, *7*, 76489–76499.

Long, C., Haizhou, A., Chong, S., Zijie, Z., & Bo, B. (2017). Online multi-object tracking with convolutional neural networks. In *International conference on image processing*.

Long, C., Haizhou, A., Zijie, Z., & Chong, S. (2018) Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *International conference on multimedia and expo*.

Loumponias, K., Dimou, A., Vretos, N., & Daras, P. (2018). Adaptive tobit Kalman-based tracking. In *International conference on signal-image technology & internet-based systems*.

Ma, C., Yang, C., Yang, F., Zhuang, Y., Zhang, Z., Jia, H., & Xie, X. (2018a). Trajectory factory: Tracklet cleaving and re-connection by deep Siamese bi-GRU for multiple object tracking. In *International conference on multimedia and expo*.

Ma, L., Tang, S., Black, M. J., & Van Gool, L. (2018b). Customized multi-person tracker. In *Asian conference on computer vision*.

Mahgoub, H., Mostafa, K., Wassif, K. T., & Farag, I. (2017). Multi-target tracking using hierarchical convolutional features and motion cues. *International Journal of Advanced Computer Science & Applications*, *8*(11), 217–222.

Maksai, A., & Fua, P. (2019). Eliminating exposure bias and metric mismatch in multiple object tracking. In *Conference on computer vision and pattern recognition*.

Manen, S., Timofte, R., Dai, D., & Gool, L. V. (2016). Leveraging single for multi-target tracking using a novel trajectory overlap affinity measure. In *Winter conference on applications of computer vision*.

Mathias, M., Benenson, R., Pedersoli, M., & Gool, L. V. (2014). Face detection without bells and whistles. In *European conference on computer vision workshops*.

McLaughlin, N., Martinez Del Rincon, J., Miller, P. (2015). Enhancing linear programming with motion modeling for multi-target tracking. In *Winter conference on applications of computer vision*.

Milan, A., Leal-Taixé, L., Schindler, K., & Reid, I. (2015). Joint tracking and segmentation of multiple targets. In *Conference on computer vision and pattern recognition*.

Milan, A., Rezatofighi, S. H., Dick, A., Reid, I., & Schindler, K. (2017). Online multi-target tracking using recurrent neural networks. In *Conference on artificial on intelligence*.

Milan, A., Roth, S., & Schindler, K. (2014). Continuous energy minimization for multitarget tracking. *Transactions on Pattern Analysis and Machine Intelligence*, *36*(1), 58–72.

Milan, A., Schindler, K., & Roth, S. (2013). Challenges of ground truth evaluation of multi-target tracking. In *Conference on computer vision and pattern recognition workshops*.

Milan, A., Schindler, K., & Roth, S. (2016). Multi-target tracking by discrete-continuous energy minimization. *Transactions on Pattern Analysis and Machine Intelligence*, *38*(10), 2054–2068.

Nguyen Thi Lan Anh, F. N., Khan, Furqan, & Bremond, F. (2017). Multi-object tracking using multi-channel part appearance representation. In *International conference on advanced video and signal based surveillance*.

Pedersen, M., Haurum, J. B., Bengtson, S. H., & Moeslund, T. B. (June 2020). 3D-ZEF: A 3D zebrafish tracking benchmark dataset. In *Conference on computer vision and pattern recognition*.

Pirsiavash, H., Ramanan, D., & Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *Conference on computer vision and pattern recognition*.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *Transactions on Automatic Control*, *24*(6), 843–854.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*.

Rezatofighi, H., Milan, A., Zhang, Z., Shi, Q., Dick, A., & Reid, I. (2015). Joint probabilistic data association revisited. In *International conference on computer vision*.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., & Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Sadeghian, A., Alahi, A., Savarese, S. (2017). Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *International conference on computer vision*.

Sanchez-Matilla, R., Cavallaro, A. (2019). A predictor of moving objects for first-person vision. In *International conference on image processing*.

Sanchez-Matilla, R., Poiesi, F., & Cavallaro, A. (2016). Online multi-target tracking with strong and weak detections. In *European conference on computer vision workshops*.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1), 7–42.

Schuhmacher, D., Vo, B.-T., & Vo, B.-N. (2008). A consistent metric for performance evaluation of multi-object filters. *Transactions on Signal Processing*, *56*(8), 3447–3457.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Conference on computer vision and pattern recognition*.

Sheng, H., Chen, J., Zhang, Y., Ke, W., Xiong, Z., & Yu, J. (2018a). Iterative multiple hypothesis tracking with tracklet-level association. *Transactions on Circuits and Systems for Video Technology*, *29*(12), 3660–3672.

Sheng, H., Hao, L., Chen, J., et al. (2017). Robust local effective matching model for multi-target tracking. In *Advances in multimedia information processing* (Vol. 127, No. 8).

Sheng, H., Zhang, X., Zhang, Y., Wu, Y., & Chen, J. (2018b). Enhanced association with supervoxels in multiple hypothesis tracking. *Access*, *7*, 2107–2117.

Sheng, H., Zhang, Y., Chen, J., Xiong, Z., & Zhang, J. (2018c). Heterogeneous association graph fusion for target association in multiple object tracking. *Transactions on Circuits and Systems for Video Technology*, *29*(11), 3269–3280.

Shi, X., Ling, H., Pang, Y. Y., Hu, W., Chu, P., & Xing, J. (2018). Rank-1 tensor approximation for high-order association in multi-target tracking. *International Journal of Computer Vision*, *127*, 1063–1083.

Smith, K., Gatica-Perez, D., Odobez, J.-M., & Ba, S. (2005). Evaluating multi-object tracking. In *Workshop on empirical evaluation methods in computer vision*.

Son, J., Baek, M., Cho, M., & Han, B. (2017). Multi-object tracking with quadruplet convolutional neural networks. In *Conference on computer vision and pattern recognition*.

Song, Y., & Jeon, M. (2016). Online multiple object tracking with the hierarchically adopted GM-PHD filter using motion and appearance. In *International conference on consumer electronics*.

Song, Y., Yoon, Y., Yoon, K., & Jeon, M. (2018). Online and real-time tracking with the GMPHD filter using group management and relative motion analysis. In *International conference on advanced video and signal based surveillance*.

Song, Y., Yoon, K., Yoon, Y., Yow, K., & Jeon, M. (2019). Online multi-object tracking with GMPHD filter and occlusion group management. *Access*, *7*, 165103–165121.

Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J. S., Mostefa, D., & Soundararajan, P. (2006). The clear 2006 evaluation. In *Multimodal technologies for perception of humans*.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., & Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *Conference on computer vision and pattern recognition*.

Tang, S., Andres, B., Andriluka, M., & Schiele, B. (2015). Subgraph decomposition for multi-target tracking. In *Conference on computer vision and pattern recognition*.

Tang, S., Andres, B., Andriluka, M., & Schiele, B. (2016). Multi-person tracking by multicuts and deep matching. In *European conference on computer vision workshops*.

Tang, S., Andriluka, M., Andres, B., & Schiele, B. (2017). Multiple people tracking with lifted multicut and person re-identification. In *Conference on computer vision and pattern recognition*.

Tao, Y., Chen, J., Fang, Y., Masaki, I., & Horn, B. K. (2018). Adaptive spatio-temporal model based multiple object tracking in video sequences considering a moving camera. In *International conference on universal village*.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin Markov networks. In *Advances in neural information processing systems*.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics (intelligent robotics and autonomous agents)*. Cambridge: The MIT Press.

Tian, W., Lauer, M., & Chen, L. (2019). Online multi-object tracking using joint domain information in traffic scenarios. *Transactions on Intelligent Transportation Systems*, *21*(1), 374–384.

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *Conference on computer vision and pattern recognition*.

Wang, B., Wang, L., Shuai, B., Zuo, Z., Liu, T., et al. (2016). Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In *Conference on computer vision and pattern recognition*.

Wang, G., Wang, Y., Zhang, H., Gu, R., & Hwang, J.-N. (2019). Exploit the connectivity: Multi-object tracking with trackletnet. In *International conference on multimedia*.

Wang, S., & Fowlkes, C. (2016). Learning optimal parameters for multi-target tracking with contextual interactions. *International Journal of Computer Vision*, *122*(3), 484–501.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., et al. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, *193*, 102907.

Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., & Li, S. Z. (2014). Multiple target tracking based on undirected hierarchical relation hypergraph. In *Conference on computer vision and pattern recognition*.

Wojke, N., & Paulus, D. (2016). Global data association for the probability hypothesis density filter using network flows. *International conference on robotics and automation*.

Wu, B., & Nevatia, R. (2006). Tracking of multiple, partially occluded humans based on static body part detection. In *Conference on computer vision and pattern recognition*.

Wu, H., Hu, Y., Wang, K., Li, H., Nie, L., & Cheng, H. (2019). Instance-aware representation learning and association for online multi-person tracking. *Pattern Recognition*, *94*, 25–34.

Xiang, J., Xu, G., Ma, C., & Hou, J. (2020). End-to-end learning deep CRF models for multi-object tracking. *Transactions on Circuits and Systems for Video Technology*,. https://doi.org/10.1109/TCSVT.2020.2975842.

Xiang, Y., Alahi, A., & Savarese, S. (2015). Learning to track: Online multi-object tracking by decision making. In *International conference on computer vision*.

Xu, J., Cao, Y., Zhang, Z., & Hu, H. (2019). Spatial-temporal relation networks for multi-object tracking. In *International conference on computer vision*.

Xu, Y., Osep, A., Ban, Y., Horaud, R., Leal-Taixe, L., & Alameda-Pineda, X. (2020). How to train your deep multi-object tracker. In *Conference on computer vision and pattern recognition*.

Yang, F., Choi, W., & Lin, Y. (2016). Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *Conference on computer vision and pattern recognition*.

Yang, M., & Jia, Y. (2016). Temporal dynamic appearance modeling for online multi-person tracking. *Computer Vision and Image Understanding*,. https://doi.org/10.1016/j.cviu.2016.05.003.

Yang, M., Wu, Y., & Jia, Y. (2017). A hybrid data association framework for robust online multi-object tracking. *Transactions on Image Processing*,. https://doi.org/10.1109/TIP.2017.2745103.

Yoon, J., Yang, H., Lim, J., & Yoon, K. (2015). Bayesian multi-object tracking using motion context from multiple objects. In *Winter conference on applications of computer vision*.

Yoon, J. H., Lee, C. R., Yang, M. H., & Yoon, K. J. (2016). Online multi-object tracking via structural constraint event aggregation. In *International conference on computer vision and pattern recognition*.

Yoon, K., Gwak, J., Song, Y., Yoon, Y., & Jeon, M. (2020). OneShotDa: Online multi-object tracker with one-shot-learning-based data association. *Access*, *8*, 38060–38072.

Yoon, K., Kim, D. Y., Yoon, Y.-C., & Jeon, M. (2019a). Data association for multi-object tracking via deep neural networks. *Sensors*, *19*, 559.

Yoon, Y., Boragule, A., Song, Y., Yoon, K., & Jeon, M. (2018a). Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. In *International conference on advanced video and signal based surveillance*.

Yoon, Y., Kim, D. Y., Yoon, K., Song, Y., & Jeon, M. (2019b). Online multiple pedestrian tracking using deep temporal appearance matching association. arXiv preprint arXiv:1907.00831.

Yoon, Y.-C., Song, Y.-M., Yoon, K., & Jeon, M. (2018). Online multi-object tracking using selective deep appearance matching. In *International conference on consumer electronics Asia*.

Zamir, A. R., Dehghan, A., & Shah, M. (2012). GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In *European conference on computer vision*.

Zhang, L., Li, Y., & Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *Conference on computer vision and pattern recognition*.

Zhang, Y., Sheng, H., Wu, Y., Wang, S., Lyu, W., Ke, W., et al. (2020). Long-term tracking with deep tracklet association. *Transactions on Image Processing*, *29*, 6694–6706.

Zhou, H., Ouyang, W., Cheng, J., Wang, X., & Li, H. (2018). Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *Transactions on Circuits and Systems for Video Technology*,. https://doi.org/10.1109/TCSVT.2018.2825679.

Zhou, X., Jiang, P., Wei, Z., Dong, H., & Wang, F. (2018b). Online multi-object tracking with structural invariance constraint. In *British machine vision conference*.

Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., & Yang, M.-H. (2018). Online multi-object tracking with dual matching attention networks. In *European conference on computer vision workshops*.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?

**Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?**
**Patrick Dendorfer**, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé
*Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

# Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?

**Patrick Dendorfer   Vladimir Yugay   Aljoša Ošep   Laura Leal-Taixé**

Technical University of Munich

{patrick.dendorfer, vladimir.yugay, aljosa.osep, leal.taixe}@tum.de

## Abstract

Recent developments in monocular multi-object tracking have been very successful in tracking visible objects and bridging short occlusion gaps, mainly relying on data-driven appearance models. While we have significantly advanced short-term tracking performance, bridging longer occlusion gaps remains elusive: state-of-the-art object trackers only bridge less than $10\%$ of occlusions longer than three seconds. We suggest that the missing key is reasoning about future trajectories over a longer time horizon. Intuitively, the longer the occlusion gap, the larger the search space for possible associations. In this paper, we show that even a small yet diverse set of trajectory predictions for moving agents will significantly reduce this search space and thus improve long-term tracking robustness. Our experiments suggest that the crucial components of our approach are reasoning in a bird's-eye view space and generating a small yet diverse set of forecasts while accounting for their localization uncertainty. This way, we can advance state-of-the-art trackers on the *MOTChallenge* dataset and significantly improve their long-term tracking performance. This paper's source code and experimental data are available at `https://github.com/dendorferpatrick/QuoVadis`.

## 1 Introduction

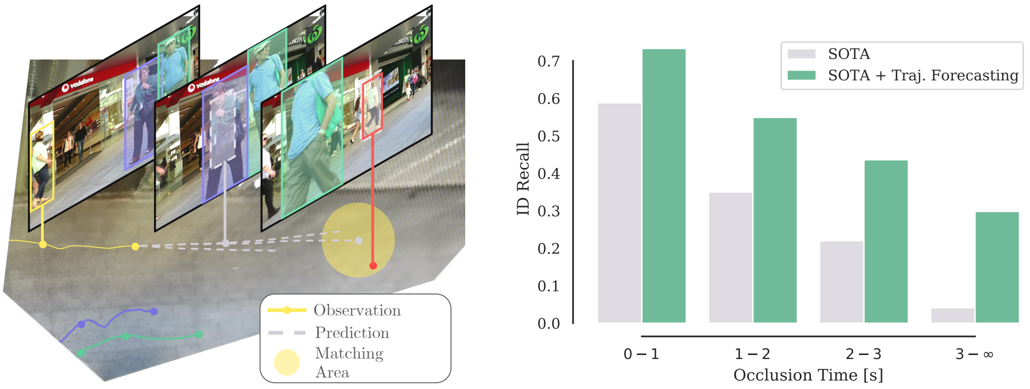
Multi-object tracking (MOT) is a long-standing research problem with applications ranging from real-time dynamic situational awareness for robot navigation [21, 15, 77, 44, 62, 10], traffic monitoring [69], studying animal behavior [52] and monitoring biological phenomena [3].

State-of-the-art MOT methods [75, 8, 4, 82, 74, 65] combine regression [75, 4] and combinatorial optimization [8] in conjunction with identity re-identification (ReID) models [32, 59, 8, 75, 66, 4] to track objects in the image space. Such approaches have been very successful for tracking visible objects and bridging *short-term* occlusions. However, as can be seen in Figure 1b, *long-term tracking* remains an open challenge: state-of-the-art methods successfully bridge $50\%$ of occlusions within one second, falling below $10\%$ when the occlusion extends for more than 3 seconds. This is often not reflected in standard benchmarks [15, 21, 69, 77], as long-term occlusions are statistically rare.

In the past, combining ReID models with simple motion models has been immensely helpful [15] for short-term tracking. Nonetheless, as the occlusion time becomes longer, the set of possible associations grows exponentially with the increasing gap length [54]. This combinatorial complexity hinders the ability of visual-based ReID models to disambiguate between objects. Consequently, we believe that ReID models are insufficient to resolve long-term occlusions. However, continued efforts to develop stronger appearance models will remain an important research direction in vision-based MOT. Tracking moving pedestrians during occlusions is challenging, and simple linear motion models fail since human motion is complex and driven by non-observable factors such as goals, intent, or simply preferences. Therefore, we propose an alternative in this work: using long-term trajectory forecasting in order to prune down the combinatorial search space of feasible trajectory continuations.

As the main contribution of this paper, we carefully study *what is needed* to leverage trajectory forecasting for multi-object tracking, as we have recently witnessed rapid progress in learning-based

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Illustration of our method.

(b) ID recall.

Figure 1: State-of-the-art methods for vision-based MOT can successfully track visible objects and bridge short occlusion gaps; however, they fail at long-term tracking. (a) To bridge occlusion gaps, we lift monocular 2D detections to 3D world space, in which we reason about their possible future locations. This transformation allows us to reconnect detections that undergo long occlusions. As **yellow track** becomes occluded, our method predicts a small set of plausible future locations in 3D. In turn, we correctly associate **red detection** to the **yellow track** by accounting for the forecast uncertainty area. (b) As can be seen from the ratio of correct track association after different occlusion time lengths for the **prior work** and **our method**, this approach allows us to significantly improve long-term tracking capabilities and gap longer occlusion gaps. *Best seen in color.*

trajectory forecasting [55, 38, 23, 1]. However, these methods operate in a fully-observed, metric bird's-eye view (BEV) space, effectively disentangling the effect of the perspective projection on reasoning about motion. By contrast, monocular MOT methods only observe a projection of the visible portion of our 3D space. Our analysis reveals that we can bridge this gap by localizing trajectories in BEV-space, but crucially, the localization of 2D bounding boxes in BEV must be *temporally coherent*. We achieve this by estimating a single homography per sequence in a data-driven manner.

Forecasting methods can reason beyond simple linear extrapolations, predict multiple possible future outcomes, and account for social interactions. But are these all necessary ingredients for bridging complex and long-term occlusions? Our study suggests that the key ingredient is to estimate a set of forecasts that can possibly cover several diverging future paths with only a handful of samples and account for prediction uncertainty.

Our *trajectory forecasting* approach can be applied to improve the long-term tracking capabilities of existing object tracking methods. In particular, by applying our framework on top of the state-of-the-art method [80] of the *MOTChallenge* benchmark, we improve the performance on HOTA on *MOT17* by 0.09pp and *MOT20* by 0.10pp and further decrease the number of IDSW by 93 and 36, respectively. We hope our conclusions will encourage the community to continue investigating how 3D reconstruction and trajectory forecasting improve single-camera long-term tracking.

We summarize our **main contributions** as follows: we (i) present a study on how we can reconcile two related fields of research on vision-based trajectory forecasting and monocular multi-object tracking. Our study reveals that the core component of this interplay is temporally coherent reasoning about motion in 3D space. We (ii) utilize a synthetic MOT dataset to study how to localize objects in 3D BEV space in a manner that facilitates robust reasoning about plausible future motion and which are the core forecasting components needed to bridge longer occlusion gaps; Finally, (iii) we demonstrate that we can generalize our conclusions from synthetic sandbox to real-world monocular *MOTChallenge* sequences and demonstrate that our recipe can be used to improve long-term tracking performance for several object trackers.

## 2 Preliminaries

This section discusses the fundamentals of vision-based multi-object tracking and trajectory forecasting, the current state-of-the-art, and analyzes failure cases.

### 2.1 Multi-object Tracking

Monocular multi-object tracking (MOT) is the task of localizing objects as bounding boxes in image sequences and assigning them an identity-preserving unique ID. State-of-the-art methods decompose the problem into object detection and detection association.

**Quantifying tracking errors.** The *detection* aspect of the task is commonly quantified by counting per-frame detection errors over the sequence. To quantify *association* errors, we count *identity switches* (IDSW) (*i.e.*, wrong ID swaps or re-initializing a ground-truth track with a different tracking ID) and *identity transfers* (IDTR) (*i.e.*, incorrectly linking two different objects with the same tracking ID). While a successful association over occlusion gaps decreases the number of IDSW, a wrong association between tracklets leads to an IDTR instead. Recently introduced HOTA [41] metric separately evaluates object detection and temporal association aspects of the tracking task. Temporal association is quantified via *association accuracy* (AssA) term, that quantifies *association recall* (AssRe) and *association precision* (AssPr). The AssRe term accounts for IDSW errors, while AssPr accounts for IDTR errors.

**Are all identity errors created equal?** Correct track association of objects that undergo longer occlusion gaps is especially challenging because the appearance and position of an object may drastically change. In MOT datasets [15, 21] the majority of occlusions are short occlusions ($\leq 2s$). Hence, solving rare long occlusions ($> 2s$) does not significantly impact the model performance. As a result, long-term tracking is commonly overlooked in the literature. This can be seen in Figure 1b: state-of-the-art methods bridge less than $10\%$ gaps beyond three-second-long occlusions.

**Prior work.** Early tracking methods focus on combinatorial optimization [79, 27, 36, 71, 49] and hand-crafting visual and motion-based descriptors [48, 33, 11], especially beneficial in the era of unreliable object detectors. State-of-the-art methods for monocular visual MOT are data-driven and primarily rely on appearance. Regression-based methods [4, 75, 82] can localize objects even when object detections are missing, often used in conjunction with ReID models to bridge short occlusions. However, regression models fail when an occluded person appears at a distant image position. For solving long-term occlusion, discrete optimization methods, combined with end-to-end learning based on graph neural networks [8, 70, 78], construct large graphs stretching over multiple seconds leading to high computational costs and complexity. Motion has always played an essential role in visual tracking [5, 20, 4], especially beneficial in 3D where it is dis-entangled from projective distortion [34, 50, 26]. The interplay between reasoning in 3D for monocular pedestrian tracking and linear motion models was first investigated in [29].

Identity preservation is important in several applications, ranging from video editing, safety camera analysis, and social robots interacting with humans to autonomous driving. We only have access to a single RGB camera in several application scenarios. Exceptions are autonomous driving datasets [21, 10] that generally provide 3D sensory data, together with 3D track information. However, only a handful of object tracks contain occlusion gaps longer than $2s$: $0.6\%$ in BDD100K [77] and $4\%$ in widely-used KITTI tracking [21] dataset. Therefore, autonomous driving datasets are, at the moment, not well suited for studying long-term tracking. Instead we conduct our experiments and analysis using *MOTChallenge* [15] dataset, where $19.4\%$ of tracks undergo *long* ($> 2s$) occlusions gaps.

We hypothesize that bridging long-term gaps requires understanding the projection geometry and motion models that can reason about plausible diverging future paths and non-linear motion.

### 2.2 Trajectory Forecasting

Pedestrian trajectory forecasting has been studied independently of the closely related task of object tracking. Forecasting is challenging because (i) human behavior and, therefore, future motion underlies the effect of complex social and scene interactions and latent navigating intent. Moreover, (ii) entire scene geometry is usually not directly visible to the observer, and in general, it is difficult to localize past trajectories precisely. To this end, existing models use standard datasets [53, 35, 38, 55] and study forecasting in idealized conditions: given an accurate bird's-eye view of the scene and perfectly-localized past trajectories to predict trajectory continuations in metric space.

**Quantifying forecasting accuracy.** Forecasting performance is measured in metric space as $L_2$ distance between the prediction and ground-truth trajectory (as final displacement error, FDE, or average displacement error, ADE) *wrt.* top-k forecasts (commonly $k = 20$). We note that this approach mainly incentivizes high forecasting recall and neglects forecasting precision which is important for the application of forecasting methods [13].

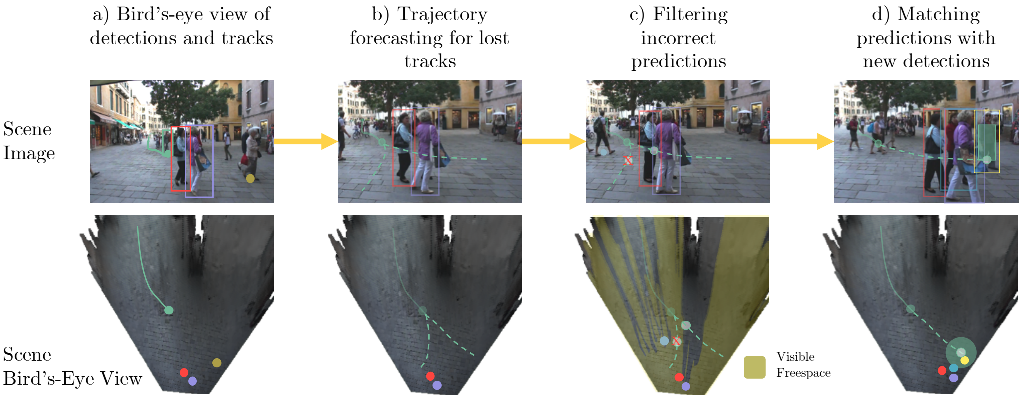

a) Bird's-eye view of detections and tracks
b) Trajectory forecasting for lost tracks
c) Filtering incorrect predictions
d) Matching predictions with new detections
Scene Image
Scene Bird's-Eye View
Visible Freespace

Figure 2: **Our method:** we bridge long-term occlusions by (a) localizing object tracks in bird's-eye view via the estimated homography and (b) forecasting future trajectories for *lost* tracks. We (d) continually aim to match these *inactive* track predictions with new object detections and remove incorrect predictions under a visibility constraint (c).

**Prior work.** Early forecasting methods were deterministic, firstly based on physical models [25], and later on data-driven LSTM-based encoder-decoder networks [1] methods, focusing on modeling social [23, 1, 2] and scene [56, 31] interactions. The forecasting task is inherently uncertain, and we need to express the stochasticity in the model. With learning a distribution of possible future trajectories, generative models [23, 56, 31, 2, 14, 13] have emerged as state-of-the-art prediction methods. Recent efforts have been explicitly focusing on conditioning forecasting on estimated pedestrian goal/intent [14, 43, 42] and estimating multimodal posterior distributions [38, 13] that yield diverse trajectories that cover different plausible directions. These deep neural network approaches can model complex and non-linear trajectories beyond simple linear models.

Can we bring the *two worlds* together, and if so, *how*? Furthermore, which of the aforementioned aspects of forecasting methods (*i.e.*, stochasticity, non-linearity, multimodality, diversity, accounting for interactions) are *important* in the context of multi-object tracking? These are the questions we discuss in the following sections.

## 3 Methodology

In this section, we present our method for long-term multi-object tracking based on trajectory forecasting in bird's-eye view (BEV) scene representation. Simply applying trajectory prediction to multi-object tracking is not trivially possible, as object trajectories observed in the image space break multiple assumptions of real-world trajectory prediction. While trajectory prediction works in bird's-eye view coordinates, the motion and size of objects in image space depend on the camera's intrinsic parameters, orientation, and position. In addition, we face temporal (limited length of observation), association (association errors along with observation), and measurement (imprecise localization of objects) uncertainties of the trajectories. Contrarily, objects are represented as bounding boxes in the image instead of single 2D positions for the object tracking task. To bridge the gap between prediction and tracking, we must find a transformation from the image to the real space. We assume objects move on a planar ground to formulate such a transformation. Thus, the bottom-center points of detection bounding boxes $p$ can be mapped to a 2D BEV coordinate $x$ via an initially unknown homography transformation $H$ that relates the homogeneous coordinates as $x \propto H \cdot p$.

**Overview.** Given a monocular video sequence captured from a stationary camera from *arbitrary* viewpoint, we first estimate the homography $H$, which maps the image plane to the 3D world ground-plane for the whole sequence (Section 3.1). Then, we incorporate our model into an online tracker that takes a monocular tracker output and localizes tracks and detections in BEV space (Figure 2*a*) using the estimated homography. Next, we forecast lost tracks in BEV space (Figure 2*b*) using our trajectory forecasting network (described in Section 3.2). Finally, we integrate forecasts into the online tracker (Section 3.3) while accounting for the uncertainty in estimated forecasts, and match new detections to existing tracks to resolve short- and long-term occlusions (Figure 2*d*).

### 3.1 Data-driven Homography Estimation

For combining monocular object tracking and forecasting, we first need to transform object detections and tracks from the image sequence into points and trajectories in a bird's-eye view representation.

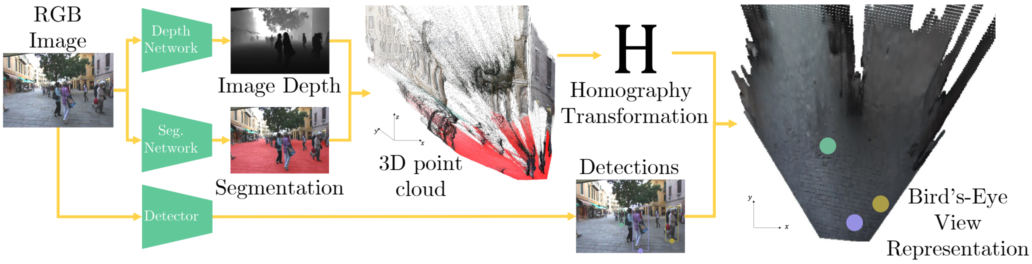


Figure 3: We estimate the homography $H$ for a sequence by reconstructing a 3D point cloud using a monocular depth estimator. We obtain ground image-to-point-cloud correspondences using a semantic segmentation model that masks ground pixels as needed to estimate the homography matrix. With the estimated homography matrix, we transform the bottom points of bounding boxes to 2D BEV coordinates.

Given a set of 2D object detections represented as bounding boxes localized in the image plane, we aim to find a homography $H$ that maps their bottom-center positions to their corresponding 2D BEV coordinates. In Figure 3, we outline our homography estimation method. We first train a monocular depth estimator [6] on a synthetic dataset [19] to reconstruct a 3D point cloud (with estimated or known intrinsics) of the first frame of a static sequence. Then, we leverage the semantic segmentation network [73] to mask, select, and fit a plane to the ground pixels. We estimate the normal vector of the ground plane in 3D and align the plane to the $XY$ plane. Then, we project ground points along the $z$-axis, leaving us with a pairwise correspondence between ground pixels in the image and a 2D position in BEV, as needed to estimate the homography between the two planes. We also linearize the homography transformation for pixel positions close to the plane's horizon to prevent the transformation from diverging (for more information, see Appendix A.1).

**Static camera.** We compute the homography only for the first frame of the sequence and use it throughout the sequence, making our pedestrian localization robust to temporal fluctuations of the depth estimator.

**Moving camera.** For moving camera sequences, we also need to account for the egomotion of the camera, which we estimate between consecutive frames as follows. First, we compute a frame-dependent homography $H_t$ for each frame. Then, we compute pairwise pixel-correspondences between (masked) ground pixels using optical flow [12] and compute a translation vector between the two point sets (lifted to 3D via $H_t$).

Empirically, we observe that estimating only translation (without rotation) yields more robust egomotion estimates.

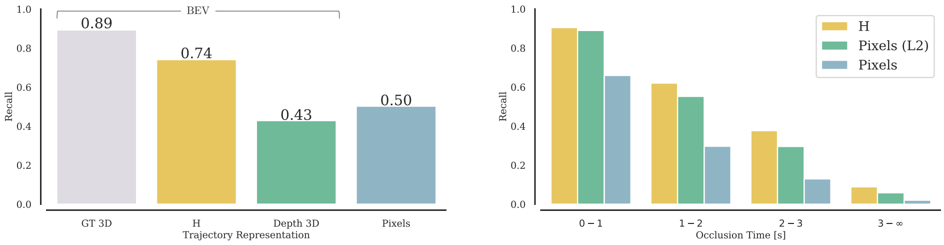
### 3.2 Forecasting

Localization of object tracks in BEV enables us to leverage data-driven forecasting models beyond simple linear motion to reason for future trajectories. However, these models expect ground-truth fixed-size past trajectory observations, while our projected trajectories are noisy and of varying lengths. As discussed in Section 2.2, prediction models are optimized for metrics that incentivize a large number of predictions and minimize $L_2$ distance to ground-truth trajectories. It is thus unclear how different proposed concepts translate into real-world tracking scenarios. We, therefore, identify the main design patterns proposed in the forecasting community and verify their impact *directly* in the context of *forecasting to track*.

**Preprocessing.** Forecasting models encode trajectories using an LSTM encoder-decoder [1] architecture, which takes a fixed-size observed trajectory as input and predicts a future trajectory. We construct input trajectories from temporally consecutive detections of the same track ID localized in BEV. To account for localization noise, we smooth the noisy observations using the Kalman filter and linearly extrapolate trajectories into the past to get trajectories of the required fixed-size input length.

**Trajectory forecasting design patterns.** In our experimental setup, we build on the LSTM encoder-decoder architecture [1] and include the following key design patterns recently emerging in the forecasting community.

*Stochasticity.* Stochastic trajectory predictors enable us to sample multiple plausible future trajectories to account for the uncertainty in future positions. We follow the approach by [23] and learn a generative GAN [22] model and train it with a best-of-many [7] loss. As a result, the network



(a) Overall recall (BEV and pixel-space).

(b) Recall *wrt.* different occlusion lengths.

Figure 4: Endpoint matching recall of predictions and GT trajectories using a linear motion predictor. A prediction is successfully matched when $\Delta_{\text{IoU}} > 0.5$ or $\Delta_{L_2}$ distance < $2m$. We also project the prediction back to the image for forecasts in the bird's-eye view. The model *Pixel (L2)* predicts motion in pixel space and projects the endpoint into BEV for matching.

internally learns an observation-conditioned distribution of future trajectories, from which we can sample.

*Social Interactions.* Social interactions impact future motion: pedestrians adapt their trajectories on-the-fly to avoid collisions. Several methods [23, 1, 2, 31] account for interactions in the forecasting literature. These methods leverage pooling [23], attention [56], or graph neural netwchecklorks [31] to provide social context (*i.e.*, trajectories of surrounding agents) to the trajectory decoder. To answer whether modeling social interactions matters for tracking, we implement Social GAN (S-GAN) [23], which uses pairwise interaction features between neighboring pedestrians by using max-pooling before they are passed to the decoder.

*Multimodality and Diversity.* While the aforementioned generative models learn a distribution over trajectories, they need to sample many trajectories to cover all modes present in the scene, as learning a multimodal posterior with a single GAN is difficult [64]. To predict the scene's main modes with as few samples as possible, we implement a multi-generator GAN network [13], extending the presented GAN architecture by training multiple decoder heads, where each decoder learns to focus on a particular model. As a result, we get a set of plausible but maximally separated predictions by sampling from these different generators.

### 3.3 Tracking via Forecasting

We assume we have an online object tracker capable of tracking visible objects (*e.g.*, bounding box regression-based tracker [4]). As long as tracks are being updated with new detections, we consider them *active* and keep them in the active set $\mathcal{S}_A$. Once we cannot associate a detection, a track becomes *inactive* and is stored in the set of inactive tracks $\mathcal{S}_I$. For each frame $t$ the tracker outputs a set of tracks $\mathcal{O} = (o_1, \dots, o_M)$ with $o_i = (\texttt{ID}_i, b_i, f_i)$ where $\texttt{ID} \in \mathbb{N}^+$ represents the track identity, $b \in \mathbb{R}^4$ denotes a bounding box in pixel space (see Figure 2a), and $f \in \mathbb{R}^D$ represents a $D$-dim feature vector encoding the appearance information obtained from a pre-trained convolutional network [24]. We localize bounding boxes in BEV coordinates $x \in \mathbb{R}^2$ using our estimated homography $H$.

**Quo Vadis?** If an object track becomes inactive (*i.e.*, temporally lost), we move the active track into the memory bank and predict $k$ trajectories of length $\tau_{max}$ in BEV space using the trajectory forecasting model as described in Section 3.2. As long as the track is inactive and not yet matched, we move along the predicted trajectory and do not predict an entirely new trajectory in each frame.

**Filtering and removing predictions.** *No prediction can live forever.* When we use stochastic trajectory predictors with multiple predictions, we need to limit the number of inaccurate or obsolete predictions to decrease the chance of false re-association. In practice, we limit the lifetime of a prediction to a maximal lifetime of $\tau_{max}$ and try to filter out *unlikely* forecasts. We use spatial and social context to determine the *freespace* [29] in which objects should be visible to the camera. We assume that visible objects eventually are detected and, therefore, remove prediction branches that should be visible for more than $\tau_{vis}$ frames.

We consider an object as *visible* if neither scene nor other pedestrians occlude the object. In particular, this means that the predicted BEV position lies in an area of the projected ground mask (shown in Figure 2c) and has no bounding box overlap $\geq 0.25$ with any other object detection, closer to the camera. Relative order can be determined based on bottom bounding box coordinates for amodal detections. If all predictions from an inactive track are removed even before $\tau_{max}$, we also remove the entire track.

**Matching predictions with new detections.** Given the trajectory forecasts, we match them with new detections via bi-partite matching, following the standard practice in tracking [81, 68, 39, 5]. This boils down to computing association costs $c_{ij}$ between the predictions of an inactive track $i$ and new un-associated detections $j$:

$$c_{ij} = (\Delta_{\text{IoU}} + \max\left(\tau_{L_2} - \Delta_{L_2}, 0\right)) \cdot \left(\Delta_{\text{App}} \geq \tau_{\text{App}} \,\text{and}\, \Delta_{\text{IoU}} \geq \tau_{\text{IoU}}\right), \quad (1)$$

where $\Delta_{\text{IoU}}$ is the IoU score between the two bounding boxes, $\Delta_{L_2}$ is the Euclidean distance between the prediction $i$ and a detection $j$ in BEV, and $\Delta_{\text{App}}$ represents the cosine distance between visual features $f_i$ and $f_j$. $\tau_{L_2}, \tau_{\text{IoU}}$, and $\tau_{\text{App}}$ denote thresholds for the matching. Therefore, we determine an association in BEV metric space and in the image domain using IoU bounding box overlap between the forecast and detected box.

Matching tracks based on spatial distance in real space leads to high recall and reduces the number of ID switches (IDSW) but may also lead to an increase in ID transfer errors (IDTR), especially in crowded scenes with many new detections close to each other. While forecasting significantly narrows combinatorial search space for associations, verifying potential associations with an appearance model is still beneficial in practice. To decrease the number of wrong associations, we require a minimal visual similarity $\tau_{\text{App}}$ and minimum IoU overlap of the bounding boxes $\tau_{\text{IoU}}$ for close objects. These thresholds serve as a filter of visually incompatible matches but do not add to the value of the cost function for the matching. In essence, we obtain a pre-selection of potential matching candidates by using the trajectory forecast and filter those if the appearance drastically deviates between the last observation and the new detection.

## 4 Experimental Evaluation

In this section, we first discuss our evaluation test-bed, followed by an experimental study on bird's-eye-view (BEV) trajectory reconstruction (Section 4.1). Then, we analyze different forecasting design patterns applied to the domain of object tracking in BEV space and discuss the relevance of different model modules for tracking (Section 4.2). Afterward, we demonstrate how our approach can be used to improve several vision-based MOT methods on static sequences and to justify our design decisions. Finally, we show that our forecasting model can be used to establish new state-of-the-art on the real-world *MOT17* and *MOT20* datasets (Section 4.4). For visualization of our tracking method, we refer the reader to Appendix C.

**Datasets.** We evaluate our trajectory prediction framework on different publicly available pedestrian tracking datasets, namely synthetic *MOTSynth* [19] and two real-world *MOT17* and *MOT20* datasets [15]. *MOTSynth* is a large synthetic dataset for multi-object tracking. It provides $764$ diverse sequences with various viewpoints, lighting, and weather conditions. Importantly, it provides ground-truth depth information and 3D key points for pedestrians, allowing us to study the suitability of different methods for BEV trajectory reconstruction. *MOT17* [47] and *MOT20* [16] are real-world tracking datasets commonly used to benchmark pedestrian tracking models. We use these datasets to evaluate our method on real-world recordings. For our experiments, we utilize the commonly used split of the *MOT17* training set, where the first half of each sequence is used for training and the second half for the evaluation [37, 80, 72].

**Metrics.** For measuring the quality of the bird's-eye view reconstruction, we indirectly evaluate the quality of the reconstruction by evaluating the forecasting and tracking performance.

To compare different models for trajectory forecasting, we report the standard $L_2$ final displacement error (FDE) for top-k predictions for $2s$ and $4s$ prediction horizons (see Section 2.2).

For multi-object tracking evaluation, we report higher-order tracking accuracy (HOTA) [41], with a focus on the association aspect of the task. To this end, we also report AssA, AssPr, and the number of ID switches IDSWs. Additionally, we report IDSW when the tracker loses an object and re-initiates a new track for the same object when it re-appears. We call these errors as ID$^{\text{lost}}$. For metric discussion, we refer to Section 2.1. Metrics labeled with either $S$ (short) or $L$ (long) only consider prediction or occlusion lengths shorter or longer than $2s$, respectively

**Hyperparameters.** For all experiments, we use the following parameters for the matching of detections with inactive tracks: $\tau_{L_2} = 2.5m, \tau_{\text{App}} = 0.8, \tau_{\text{IoU}} = 0.2$ The maximal lifetime of prediction is $\tau_{max} = 6s$ and maximal visibility $\tau_{\text{vis}} = 1s$ before it is removed. We refer the reader to Appendix B for further information on implementation details.

**Object trackers.** We study and ablate our method on eight high-ranked state-of-the-art trackers of *MOTChallenge* and refer to them as *baseline*. We use BYTE [80], JDE [68], CSTrack [37], FairMOT [81], TraDes [72], QDTrack [51], CenterTrack [82] and TransTrack [63] for an evaluation

Table 1: Which forecasting modules matter for tracking? Evaluated on *MOT17* validation set.

| Model | Nr. Samples | Deter - ministic | Stoch- astic | Social | Multi- modal | Prediction | | Tracking | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | FDE$_S$ ↓ | FDE$_L$ ↓ | HOTA ↑ | AssA ↑ | AssRe ↑ | AssPr ↑ | ID$_S^{\text{lost}}$ ↓ | ID$_L^{\text{lost}}$ ↓ |
| Baseline | | | | | | – | – | 50.71 | 46.87 | 51.80 | **78.11** | 0 % | 0 % |
| Static | 1 | ✓ | | | | 1.59 | 2.09 | 53.84 | 53.51 | 60.04 | 72.95 | -14.77 % | -8.40 % |
| Kalman Filter (pixel) | 1 | ✓ | | | | – | – | 54.08 | 54.02 | 60.45 | 72.81 | -**22.37** % | -8.99 % |
| Kalman Filter | 1 | ✓ | | | | 0.69 | 1.23 | 54.11 | 54.04 | 60.75 | 71.73 | -19.50 % | -16.07 % |
| GAN | 3 | | ✓ | | | 0.85 | 1.26 | 54.43 | 54.61 | 61.11 | 73.21 | -17.99 % | -8.64 % |
| GAN | 20 | | ✓ | | | **0.65** | **0.99** | 53.81 | 53.40 | 60.45 | 71.31 | -18.03 % | -15.63 % |
| S-GAN | 3 | | ✓ | ✓ | | 0.87 | 1.21 | **54.52** | 54.78 | 61.22 | 73.28 | -16.92 % | -8.57 % |
| MG-GAN | 3 | | ✓ | | ✓ | 0.67 | 1.03 | **54.52** | **54.80** | **61.35** | 73.13 | -21.19 % | -**17.43** % |

on the *MOT17* validation set and BYTE and CenterTrack on the *MOT20* training dataset. These trackers use ReID similarity and/or simple motion cues for bridging short-term occlusions.

### 4.1 Bird's-Eye View Estimation

This section discusses different approaches to obtaining scene BEV representations of detected objects in the image for forecasting. We work with static sequences of the *MOTSynth* dataset (that provides depth maps used for evaluating and training a monocular depth estimator). With this, we test a linear motion model to gap occlusions of different durations, which we obtain by running a CenterTrack [82] baseline tracker. We evaluate the ratio of successful matches between the target and the linear prediction and count a match to be successful if the IoU of the predicted bounding box is larger than $0.5$ or the $L_2$ distance in metric space is lower than $2m$. We get the predicted bounding box by translating the last observed bounding box by the predicted displacement in the image.

**Baselines.** We compare motion in (i) BEV and (ii) pixel space. We evaluate different approaches to localize trajectories: (a) using ground truth (GT) 3D coordinates orthographically projected to BEV (oracle), (b) the proposed homography estimation as described in Section 3.1, and (c) directly using learned monocular depth estimates and resulting point clouds, followed by orthogonal projection of points these representing an object.

**Conclusions.** As seen in Figure 4, GT 3D (oracle) based motion estimates solve $89.3\%$ of the gaps, suggesting that the motion in the synthetic dataset is dominantly linear. Our proposed data-driven homography estimation approach only drops by $15\%$ compared to using ground-truth 3D keypoints. By contrast, estimating linear motion in pixels space only resolves $50.2\%$, and using per-frame monocular depth estimates $43.1\%$ of the occlusion gaps. This is likely because such depth estimates are not temporally stable. As can be seen in Figure 4b, this performance is especially apparent for longer occlusion gaps. Furthermore, we forecast motion in pixel space but transform the prediction into BEV for matching. While increasing performance compared to exclusive forecasting and matching in pixel space, we find that the results are inferior to predictions in BEV due to the distortion of projecting real motion into the image plane. We conclude that our proposed homography transformation is suitable for forecasting.

### 4.2 Trajectory Prediction Models

In this section, we evaluate different forecasting models and components (as discussed in Section 3.2). We compare a constant-velocity model (Kalman filter) in BEV and pixel space, an identity (static) model, and a stochastic GAN predictor generating $k = 3$ and $k = 20$ samples. Furthermore, we test predicting social interactions with GAN and the multimodal trajectories with MG-GAN in BEV space.

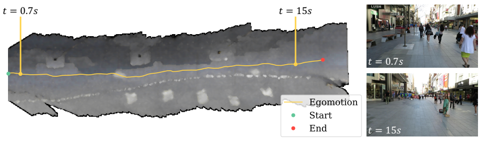


Figure 5: Visualization of BEV reconstruction for moving camera sequence and egomotion estimation.

| Scores | | Threshold | | HOTA ↑ | AssA↑ | AssRe ↑ | AssPr ↑ | ID$^{\text{lost}}$ ↓ |
|---|---|---|---|---|---|---|---|---|
| $L_2$ | IoU | $\tau_{\text{IoU}}$ | $\tau_{\text{App}}$ | | | | | |
| ✓ | | | | 53.89 | 53.56 | 60.43 | 72.21 | -16.18 % |
| ✓ | | | ✓ | 53.89 | 53.57 | 60.51 | 71.69 | -16.26 % |
| ✓ | | ✓ | ✓ | 54.10 | 53.92 | 60.43 | **73.36** | -16.84 % |
| | ✓ | | | 54.13 | 54.01 | 60.97 | 72.00 | -24.06 % |
| ✓ | ✓ | | | 53.75 | 53.35 | **61.17** | 69.27 | -**28.02**% |
| ✓ | ✓ | ✓ | | 53.97 | 53.75 | 61.08 | 70.73 | -26.93 % |
| ✓ | ✓ | | ✓ | 54.06 | 53.92 | 61.07 | 71.01 | -21.40 % |
| ✓ | ✓ | ✓ | ✓ | **54.27** | **54.29** | 61.08 | 72.36 | $-20.53\%$ |

Table 2: Ablation of matching prediction and effect of different thresholds $\tau$ on different tracking metrics.

**Forecasting.** We observe in Table 1 that the linear model performs well for short-term windows ($0.69$ FDE$_S$), suggesting that linear motion is suitable for short occlusions. We also do not find a significant difference between GAN w/o social module (S-GAN). While FDE error suggests vanilla GAN ($k = 20$) yields the best forecasting results ($0.65$ FDE$_S$ and $0.99$ FDE$_L$), but this configuration

Table 3: We improve tracking results of *all* top-8 state-of-the-art models (MOT17 validation set and *MOT20* training set). Differences to the baseline performance are shown in $(\cdot)$.

| | MOT17 (val, static scenes) | | | | | | | | MOT20 (train) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | BYTE [80] | CenterTrack [82] | CSTrack [37] | FairMOT [81] | JDE [68] | TraDeS [72] | TransTrack [63] | QDTrack [51] | BYTE [80] | CenterTrack [82] |
| HOTA | 71.36 (+0.21) | 61.78 (+3.56) | 61.60 (+0.43) | 58.42 (+0.09) | 51.06 (+0.20) | 62.45 (+0.67) | 60.68 (-0.23) | 58.87 (+0.54) | 56.85 (+0.06) | 32.71 (+0.62) |
| AssA | 73.96 (+0.49) | 66.18 (+7.54) | 63.84 (+0.8) | 59.21 (+0.37) | 54.36 (+0.45) | 67.41 (+1.6) | 63.47 (-0.49) | 60.14 (+1.22) | 53.97 (+0.20) | 28.94 (+1.34) |
| AssRe | 79.21 (+0.66) | 69.66 (+8.38) | 69.15 (+1.07) | 64.31 (+0.5) | 60.82 (+0.89) | 73.1 (+2.43) | 69.19 (+0.02) | 65.31 (+2.06) | 59.89 (+0.4) | 34.34 (+5.12) |
| AssPr | 83.11 (-0.67) | 81.75 (-5.47) | 77.79 (-2.28) | 74.45 (-1.71) | 68.9 (-2.35) | 80.0 (-1.91) | 79.53 (-1.68) | 77.4 (-2.98) | 68.65 (-5.24) | 52.37 (-21.06) |
| IDSW | 84 (-3) | 137 (-146) | 269 (-28) | 198 (-12) | 316 (-19) | 106 (-32) | 112 (-3) | 219 (-34) | 1815 (-78) | 5240 (-2700) |
| MOTA | 80.09 (+0.01) | 70.77 (+0.39) | 71.31 (+0.05) | 71.82 (+0.05) | 59.57 (+0.06) | 70.93 (+0.09) | 69.5 (+0.01) | 69.61 (+0.08) | 73.38 (+0.0) | 47.57 (+0.24) |
| IDF1 | 82.92 (+0.42) | 74.46 (+7.13) | 74.16 (+0.95) | 73.93 (+0.59) | 65.01 (+1.27) | 76.36 (+1.21) | 71.46 (+0.02) | 70.41 (+0.77) | 72.47 (+0.37) | 45.85 (+4.13) |
| IDR | 78.61 (+0.39) | 65.25 (+6.25) | 67.53 (+0.87) | 66.23 (+0.53) | 56.08 (+1.09) | 67.12 (+1.06) | 61.39 (+0.01) | 62.17 (+0.68) | 66.44 (+0.34) | 35.87 (+3.23) |
| IDP | 87.72 (+0.44) | 86.71 (+8.3) | 82.23 (+1.05) | 83.65 (+0.67) | 77.31 (+1.51) | 88.55 (+1.4) | 85.47 (+0.02) | 81.17 (+0.89) | 79.7 (+0.41) | 63.53 (+5.72) |

leads to the lowest association precision $(71.31)$ and HOTA score $(53.81)$ *wrt.* tracking performance. This result suggests a misalignment of evaluation metrics used in forecasting and tracking; a better forecaster in terms of ADE/FDE does not necessarily lead to a better tracker. This is a known drawback of ADE/FDE metrics, which essentially measure only recall and not the precision of the forecasting output. Furthermore, this shows the careful trade-off between the number of predictions $k$ and the recall/precision of the predictions and tracking results.

**Tracking.** To investigate the effect on long-term occlusions, we focus on the change of $\text{ID}_L^{lost}$ for short $(t_{occl} \leq 2s)$ and long $(t_{occl} > 2s)$ occlusion gaps. As can be seen in Table 1, even the static motion model solves $8.4\%$ $(\text{ID}_L^{lost})$, as many occluded objects do not move. By modeling linear motion (Kalman filter in pixel space), we can improve short-term re-association for $0.59pp$ (long-term IDSW) over the static model. We focus the discussion on long occlusion gaps. In terms of the generative model, we observe that interaction-aware S-GAN $(8.64\%\ \text{ID}_L^{lost})$ is on-par with vanilla GAN $(8.57\%\ \text{ID}_L^{lost})$ for $k = 3$; interestingly, both are below linear Kalman filter (BEV) performance, suggesting that these models suffer from low precision. Only MG-GAN, explicitly trained to generate multimodal trajectories, outperforms the linear model $(17.43\%\ \text{ID}_L^{lost})$ and significantly outperforms vanilla GAN with only three samples. These conclusions generalize to tracking metrics.

### 4.3 Tracking Evaluation

In this section, we study the impact of forecasting models on the valuation set's tracking performance. First, we discuss the impact of different design decisions on matching strategy, as explained in Section 3.3.

**Trajectory matching.** First, we ablate the matching cost function (Equation (1)). As can be seen in Table 2, we find that a combination of $L_2$ and IoU without any threshold $\tau$ leads to the highest decrease in terms of $\text{ID}^{\text{lost}}$ $(-28.02\%)$ and overall highest association recall (AssRe) $(61.17)$. However, this is at the cost of decreasing association precision (AssPr) $(-4.09)$. We obtain the highest AssPr $(73.36)$ by only relying on $L2$ matching and thresholding, however, at the loss of AssRe $(-0.74)$. Adding appearance-based $\tau_{\text{App}}$ and IoU $\tau_{\text{IoU}}$ thresholds provide the best trade-off and overall highest AssA $(54.29)$ and HOTA score $(54.27)$ while still recovering $21\%$ of lost trajectories.

**Validation results.** In Table 3, we present the performance of different state-of-the-art trackers on the *static sequences* of *MOT17*-val and *MOT20*-train (trained on *MOT17*), equipped with our trajectory forecasting model. As can be seen, our model brings stable improvements over all the key metrics: HOTA, AssA, and IDSW. Our trajectory prediction model consistently reduces IDSW for all models. This is also shown in Figure 1b where we demonstrate that our forecasting model improves ID recall significantly for occlusion times $> 1s$.

While our focus was on sequences with stationary viewpoints, we show that our model is also applicable to sequences with moving cameras by estimating the camera's egomotion as described in Section 3.1. In Table 4, we present results on the *moving sequences* of the MOT17 validation set (excluding sequence MOT17-05 for which the quality and consistency of our depth estimator was too low to construct time-consistent homographies). As can be seen, we improve 6 out of 8 trackers *wrt.* HOTA score and even improve CenterTrack [82] by 3.07 pp. We visualize the traversed BEV map of sequence MOT17-07 in Figure 5.

### 4.4 Benchmark Evaluation

In this section, we apply our method to state-of-the-art tracker ByteTrack [80] and, by improving its long-term tracking capabilities, establish a new state-of-the-art on the *MOT17* & *MOT20* benchmarks. We evaluate our method in the *private detection* regime, as these trackers use private detectors.

In Table 5, we compare our *QuoVadis* to the base tracker ByteTrack [80] and compare both to *MOT17* benchmark published top-performers. We improve performance on key metrics overall top methods. Notably, we reduce the number of identity switches by 93 compared to [80] and establish a new

Table 4: Results of top-8 state-of-the-art models on dynamic scenes of the MOT17 validation set excluding MOT17-05. Differences in the baseline performance are shown in (·)

| | MOT17 (val, moving scenes) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | BYTE [80] | CenterTrack [82] | CSTrack [37] | FairMOT [81] | JDE [68] | TraDeS [72] | TransTrack [63] | QDTrack [51] |
| HOTA | 60.08 (+0.02) | 51.77 (+3.07) | 54.51 (0.0) | 56.1 (0.0) | 52.14 (+1.47) | 53.36 (+1.27) | 52.7 (+0.28) | 52.28 (+0.76) |
| AssA | 60.44 (+0.03) | 53.18 (+6.49) | 59.04 (+0.0) | 61.15 (0.0) | 55.32 (+3.06) | 54.08 (+2.44) | 51.99 (+0.54) | 53.71 (+1.6) |
| AssRe | 66.53 (-0.0) | 58.21 (+8.32) | 63.35 (+0.0) | 65.87 (0.0) | 61.52 (+3.75) | 59.86 (+3.1) | 59.13 (+0.79) | 61.8 (+3.82) |
| AssPr | 78.29 (+0.09) | 76.98 (-4.66) | 80.46 (-0.0) | 79.21 (0.0) | 73.5 (-0.98) | 75.52 (-3.08) | 72.45 (-0.32) | 72.53 (-4.76) |
| IDSW | 54 (+1) | 131 (-62) | 97 (-5) | 86 (0) | 122 (-10) | 99 (-11) | 120 (-5) | 71 (-7) |
| MOTA | 72.54 (-0.01) | 59.46 (+0.46) | 60.68 (+0.04) | 63.78 (0.0) | 60.52 (+0.07) | 64.13 (+0.08) | 63.64 (+0.04) | 60.21 (+0.05) |
| IDF1 | 73.11 (0.0) | 63.48 (+5.76) | 70.69 (0.0) | 73.1 (0.0) | 68.23 (+1.85) | 67.72 (+2.29) | 64.08 (+0.79) | 65.81 (+2.86) |

Table 5: Comparison under the "private detector" protocol on *MOT17* test set.

| Tracker | HOTA | IDF1 | MOTA | IDSW | AssA |
|---|---|---|---|---|---|
| ReMOT [76] | 59.73 | 71.99 | 77.01 | 2853 | 57.08 |
| CrowdTrack [61] | 60.26 | 73.62 | 75.61 | 2544 | 59.26 |
| TLR [67] | 60.72 | 73.58 | 76.48 | 3369 | 58.88 |
| MAA [60] | 61.98 | 75.88 | 79.36 | 1452 | 60.16 |
| ByteTrack [80] | 63.05 | 77.30 | 80.25 | 2196 | 61.97 |
| QuoVadis (Ours) | **63.14** | **77.71** | **80.27** | **2103** | **62.07** |

Table 6: Comparison under the "private detector" protocol on *MOT20* test set.

| Tracker | HOTA | IDF1 | MOTA | IDSW | AssA |
|---|---|---|---|---|---|
| FairMOT [81] | 54.42 | 68.44 | 59.57 | 1881 | 56.6 |
| CrowdTrack [61] | 54.95 | 68.24 | 70.68 | 3198 | 52.57 |
| MAA [60] | 57.28 | 71.15 | 73.90 | 1331 | 55.14 |
| ReMOT [76] | 61.15 | 73.14 | 77.42 | 1789 | 58.68 |
| ByteTrack [80] | 61.34 | 75.20 | 77.76 | 1223 | 59.55 |
| QuoVadis (Ours) | **61.48** | **75.70** | **77.77** | **1187** | **59.87** |

state-of-the-art in terms of HOTA (63.14). We observe similar trends on *MOT20*, where we improve over the base tracker ByteTrack [80] by $+0.5$ in terms of IDF1 and reduce the number of identity switches by 36, similarly establishing a new state-of-the-art (61.48 HOTA).

## 5 Remarks and Limitations

The paper primarily focuses on the conceptual work of building an entire pipeline from video to tracks studying different forecasting paradigms, and showing the benefit of leveraging trajectory forecasting in BEV for the tracking task. Nevertheless, we want to outline further remarks and limitations of our work.

**Model complexity.** Our model is complex, consisting of multiple sub-modules, as constructing object trajectories in BEV space based on a monocular video and trajectory forecasting are challenging problems. We foresee that future work will improve the end-to-end integration and efficiency of the algorithms.

**Bird's-eye view reconstruction.** A vital part of our approach is an accurate homography transformation that allows us to project the objects in the image into the 3D ground plane. However, the homography transformation depends on the quality of depth estimates, which makes the overall approach sensitive to errors in 3D localization. Future work will benefit from further development of time-consistent depth estimators. Furthermore, the presented trajectory prediction models do not yet account for the BEV localization uncertainty, which results from errors in the transformation or the simplified assumption of the ground plane. These limitations show the need to develop trajectory forecasting models that account for the localization uncertainties of the upstream tasks.

## 6 Conclusion

This paper presents a study on how to bridge the gap between real-world trajectory prediction and single-camera tracking. Throughout our paper, we identified challenges and solutions to leveraging real-world trajectory prediction to benefit single-camera tracking. In particular, we focus on resolving the re-identification of objects after long-term occlusions. Here, we start from the first principles, questioning motion representation in pixel space and using a combination of models to construct a more accurate BEV representation of the scene. We find that the key component is a forecasting approach reasoning about multiple feasible future directions with a small set of multimodal forecasts. We can substantiate our conclusion by achieving new state-of-the-art performance on the *MOT17* and *MOT20* datasets.

Ultimately, we have showcased a novel way of combining state-of-the-art trajectory prediction models and multi-object tracking task. We have outlined a new way of thinking about motion prediction in tracking and motivating the beneficial symbiosis of both tasks. We hope that both fields start moving towards each other and incorporate the requirements and needs of each other.

**Acknowledgments.** This research was partially funded by the Humboldt Foundation through the Sofja Kovalevskaja Award.

## 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 *Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social Ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[3] Samreen Anjum and Danna Gurari. CTMC: Cell tracking with mitosis detection dataset challenge. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *International Conference on Computer Vision*, 2019.

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *International Conference on Image Processing*, 2016.

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[7] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a “Best of Many” sample objective. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[8] Guillem Braso and Laura Leal-Taixe. Learning a neural solver for multiple object tracking. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, 2012.

[10] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *International Conference on Computer Vision*, 2015.

[12] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. `https://github.com/open-mmlab/mmflow`, 2021.

[13] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. MG-GAN: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *International Conference on Computer Vision*, 2021.

[14] Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé. Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. In *Asian Conference on Computer Vision*, 2020.

[15] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. MOTChallenge: A benchmark for single-camera multiple target tracking. In *International Journal of Computer Vision*, 2020.

[16] 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. In *arXiv preprint arXiv:2003.09003*, 2020.

[17] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *International Conference on Computer Vision*, 2015.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[19] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. MOTSynth: How can synthetic data help pedestrian detection and tracking? In *International Conference on Computer Vision*, 2021.

[20] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3D traffic scene understanding from movable platforms. In *Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems*, 2014.

[23] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[25] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. In *Physical Review E*, 1995.

[26] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint Monocular 3D Vehicle Detection and Tracking. In *International Conference on Computer Vision*, 2018.

[27] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision*, 2008.

[28] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard I. Hartley. Learning to estimate hidden motions with global motion aggregation. In *International Conference on Computer Vision*, 2021.

[29] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. In *International Conference on Computer Vision*, 2021.

[30] Daniel Kondermann, Rahul Nair, Stephan Meister, Wolfgang Mischler, Burkhard Güssefeld, Katrin Honauer, Sabine Hofmann, Claus Brenner, and Bernd Jähne. Stereo ground truth with error bars. In *Asian Conference on Computer Vision*, 2014.

[31] 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 *Conference on Neural Information Processing Systems*, 2019.

[32] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[33] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[34] Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. In *Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[35] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by Example. In *Comput. Graph. Forum*, 2007.

[36] Yuan Li, Chang Huang, and Ramkat Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[37] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, Shuyuan Zhu, and Weiming Hu. Rethinking the competition between detection and rReID in multiobject tracking. In *Transactions on Image Processing*, 2022.

[38] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[39] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. GSM: Graph similarity model for multi-object tracking. In *International Joint Conferences on Artificial Intelligence*, 2020.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[41] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. In *International Journal of Computer Vision*, 2020.

[42] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *International Conference on Computer Vision*, 2021.

[43] 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*, 2020.

[44] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments. In *Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[45] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[46] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. In *Journal of Photogrammetry and Remote Sensing*, 2018.

[47] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. In *arXiv preprint arXiv:1603.00831*, 2016.

[48] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. In *Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[49] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[50] Aljoša Ošep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image- and world-space tracking in traffic scenes. In *International Conference on Robotics and Automation*, 2017.

[51] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[52] Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B Moeslund. 3D-ZeF: A 3D zebrafish tracking benchmark dataset. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[53] S. Pellegrini, Andreas Ess, and L. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision*, 2010.

[54] Donald B Reid. An algorithm for tracking multiple targets. In *Transactions on Automatic Control*, 1979.

[55] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016.

[56] 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 *Conference on Computer Vision and Pattern Recognition*, 2019.

[57] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *arXiv preprint arXiv:1708.07120*, 2018.

[58] Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018.

[59] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[60] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *Winter Conference on Applications of Computer Vision*, 2022.

[61] Daniel Stadler and Jürgen Beyerer. On the performance of crowd-specific detectors in multi-pedestrian tracking. In *International Conference on Advanced Video and Signal-Based Surveillance*, 2021.

[62] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[63] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. In *arXiv preprint arXiv: 2012.15460*, 2020.

[64] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no gans land. In *Proceedings of Machine Learning and Systems*, 2020.

[65] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[66] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, B.B.G Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[67] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[68] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, 2020.

[69] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. In *Computer Vision and Image Understanding*, 2015.

[70] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris Kitani. GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[71] Bo Wu and Ramkat Nevatia. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In *International Journal of Computer Vision*, 2009.

[72] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[73] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[74] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. In *arXiv preprint arXiv:2103.15145*, 2021.

[75] Yihong Xu, Aljoša Ošep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[76] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. ReMOT: A model-agnostic refinement for multiple object tracking. In *Image and Vision Computing*, 2020.

[77] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[78] Jan-Nico Zaech, Alexander Liniger, Dengxin Dai, Martin Danelljan, and Luc Van Gool. Learnable Online Graph Representations for 3D Multi-Object Tracking. In *Robotics and Automation Society*, 2022.

[79] Li Zhang, Li Yuan, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[80] 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 *European Conference on Computer Vision*, 2022.

[81] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. In *International Journal of Computer Vision*, 2021.

[82] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.

# Quo Vadis: Supplementary Material

The supplementary material complements our work with additional information on the bird's-eye-view reconstruction in Appendix A. Furthermore, we provide implementation and training details on different components and networks used in our method in Appendix B. Finally, we present visual examples as visualizations and videos in Appendix C.

## A Information on Bird's-Eye View Reconstruction

The paper presents our approach to constructing a bird's-eye-view (BEV) representation for a static tracking sequence. Here, we extend the explanation by adding a description of moving cameras and how we linearize the homography transformation for farther objects to avoid enormous distances and unrealistic velocities.

### A.1 Linearization of Homography

To get a bird's-eye-view (BEV) representation of the tracking scene, we estimate the homography $H$ between the image and the ground plane. Hence, the homogenous pixel positions transform accordingly to Equation (2) as follows:

$$s \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}. \tag{2}$$

This approach assumes that objects move on a perfect plane and object's position in the image is represented as the bottom mid-point of the object's bounding box. Depending on the perspective transformation of the camera, we find that minor changes in pixel value lead to enormous distances in BEV. Given a homography matrix:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}, \tag{3}$$

the BEV coordinate $y$ is computed as:

$$y = \frac{h_{21} \cdot p_x + h_{22} \cdot p_y + h_{23}}{h_{31} \cdot p_x + h_{32} \cdot p_y + h_{33}}. \tag{4}$$

As the denominator in Equation (4) is approaching zero, the $y$-coordinate grows hyperbolically. This behavior is undesired for trajectory prediction because these large jumps in the object's position result in unrealistic velocities for the object. Therefore, we define a threshold for which we linearly extrapolate the transformation such that the transformed distance between two neighboring pixel points is maximal $0.2m$ as shown as a red line in Figure 6b. This formulates the condition as

$$\|\frac{h_{21} \cdot p_x + h_{22} \cdot p_y + h_{23}}{h_{31} \cdot p_x + h_{32} \cdot p_y + h_{33}} - \frac{h_{21} \cdot p_x + h_{22} \cdot (p_y + 1) + h_{23}}{h_{31} \cdot p_x + h_{32} \cdot (p_y + 1) + h_{33}}\| \leq 0.2m \tag{5}$$

We call the $p_y$ value for which the inequality Equation (5) is equal, the linearization threshold $p_y^T$. The pixel point where the denominator of Equation (4) becomes 0 is called the horizon because no point on the plane is projected on a lower point in the image.

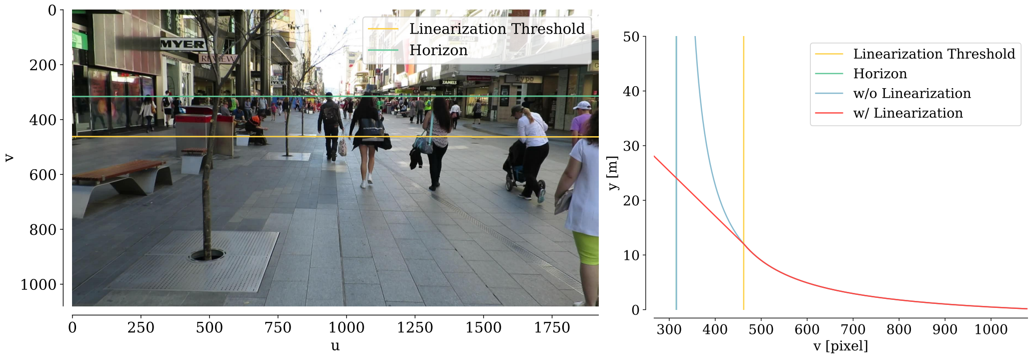
To prevent this hyperbolic growth for image points closer to the horizon, we linerarize Equation (4) around $p_y^T$ and apply the linear transformation for all $p_y \leq p_y^T$ as shown in Figure 6b. Thus, we stabilize the distance between two points to prevent very unrealistic velocities, which would make the transformed values pointless. To transform from pixel space to BEV and back, we also inverse the linearized transformation to get a one-to-one mapping.

## B Implementation Details

In this section, we provide additional information on the implementation of our method and its key components. The source code is available at `https://github.com/dendorferpatrick/QuoVadis`.

### B.1 Synthetic Training Data

For training the trajectory predictor (Appendix B.2) and the depth estimator (Appendix B.3) we use the MOTSynth dataset [19] which provides ground truth 3D positions of objects and image depth information. We use the split suggested by Fabbri *et al.* [19] with 576 sequences in the training set and 192 in the validation set.



(a) Horizon and Linear Threshold in Scene.

(b) Linearization of BEV coordinates.

Figure 6: Demonstration of horizon and linearization threshold for sequence image. Linearization of homography transformation is necessary to prevent enormous distances in the transformed coordinates and unrealistic velocities.

## B.2 Trajectory Predictor

For our trajectory model, we use the implementation of MG-GAN [13]. For studying the effect of modeling social interactions on tracking, we implement a social max-pooling module following S-GAN [23].

**Model.** The trajectory prediction model generates a set of $K$ future trajectories $\{\hat{Y}_i^k\}_{k=1,\ldots,K}$ with $t \in [t_{obs}+1, t_{pred}]$ given the input trajectory $X_i$ with $t \in [t_1, t_{obs}]$ for each pedestrian $i$. We use $t_{obs} = 8$ observation steps and $t_{pred} = 12$ prediction steps as default for training the model. However, input and output length can vary depending on the observed tracks during inference for the tracking model.

For the multimodal MG-GAN implementation, we use $n_G = 3$ generators from which we sample one prediction from each generator during inference.

**Training.** We construct trajectories of the MOTSynth data with 8 observation and 12 prediction steps, each step being $0.4s$. The entire model is trained in a GAN framework using a prediction model and a discriminator network. We train the network on the entire train dataset over 200 epochs, with a learning rate $\lambda = 10^{-3}$, and using a batch size scheduler [58].

## B.3 Depth Estimator

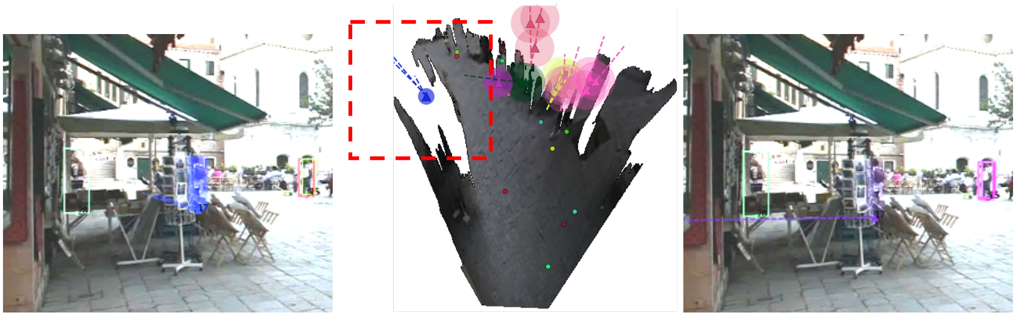
Depth estimation is a crucial part of the BEV estimation in our model. Therefore, we use a vision transformer-based [18] network [6] for monocular dense depth estimation.

**Model.** The transformer-based model regresses the depth prediction as a linear combination of depth range bins of adaptive size. The network encoder-decoder extracts visual features from the image, which are passed to the mVit block. mVit is a lightweight vision transformer based on [18]. The model applies an MLP on top of the mVit's output, predicting the size of the bins for the depth range. The encoder computes the weights of the bins by passing the features through multiple convolutional layers with a final softmax non-linear activation function.

**Training.** The network trains on the synthetic MOTsynth dataset to leverage a large number of tracking scenes of varying perspectives, weather, and light conditions. To better generalize to real-world data, we augment the scenes with ground reflections by mirroring surfaces in the image. This results in better performance, especially for the indoor MOT sequences, with ground reflections. We find the model trained on synthetic data performs well on real data even without fine-tuning.

To increase the default model depth map resolution from $640 \times 480$ to $960 \times 576$ we grow the transformer positional embedding vector size from 500 to 1000.

We trained the model using AdamW optimizer [40] with weight decay $10^{-2}$. Following [57], the maximum learning rate $\lambda_{\text{max}}$ was set to $3.5 \times 10^{-4}$ with linear warm-up from $\frac{1}{4}\lambda_{\text{max}} \rightarrow \lambda_{\text{max}}$ for the first 30% of the iterations followed by cosine annealing to $\frac{3}{4}\lambda_{\text{max}}$. We trained the model for 20 epochs with an image resolution of $960 \times 576$ on a training split of MOTSynth dataset with the batch size of 8 on 4 $RTX$8000 for one week. Then, we trained the model for 30 epochs with an image resolution of $960 \times 576$ on the full MOTSynth dataset with a batch size of 8 on 4 $RTX$8000 for ten days.



(a) Projected MG-GAN prediction (b) MG-GAN predictions in BEV (c) Prediction of Kalman Filter in Pixel Space

Figure 7: Demonstration of prediction for MG-GAN in BEV and Kalman filter in pixel space.

### B.4 Image Segmentation

We run a pre-trained Detectron2 [73] segmentation network to get the segmentation masks of the tracking scene images. Explicitly, we use the pre-trained COCO Panoptic Segmentation model with Panoptic FPN [73]. The model outputs semantic labels for 134 COCO classes and panoptic object ids, which are irrelevant to our task.

In our model, we use segmentation labels to mask ground pixels of the scene. Therefore, we combine the following COCO classes to our ground class: $\texttt{pavement}, \texttt{road}, \texttt{platform}, \texttt{floor}, \texttt{floor} - \texttt{wood}, \texttt{grass}, \texttt{sand}, \texttt{dirt}$.

#### B.4.1 Optical Flow

We estimate the optical flow using the implementation [12] of an attention-based GMA model[28]. We use the standard MMFlow configuration for the GMA pre-trained model on a mix of the datasets [28, 17, 45, 9, 46, 30]. While the model was pre-trained on images with size (768, 368), we resized the MOTChallenge images to (960, 540) at test time.

## C Visual and Qualitative Results

This section shows a visual example of the difference between a BEV and a 2D image space prediction. Furthermore, we want to point to the additional scene videos also provided in the Supplementary material.

**2D versus 3D.** In Figure 7 we show the trajectory prediction of our MG-GAN projected into the image (Figure 7a), the prediction in BEV (Figure 7b), and trajectory prediction in pixel space using a Kalman Filter (Figure 7c). We find the problem of the model reasoning in pixel space and cannot account for the effect of the camera perspective. As a consequence, this results in unrealistic motion in image space.

In contrast, we see in Figure 7a that our model predicting in BEV understands the spatial structure of the scene and is, therefore, able to predict the correct trajectory for the object and resolves the long-term occlusion.

**Example Videos.** In addition to the written supplementary material, we provide brief video clips of different MOT17 validation and test sequences with ByteTrack and Center Tracks.

In Figure 8, we give a brief description of the format of the provided video sequences. We show our predictor output on the left side, the baseline tracker output in the middle, and the online BEV prediction and reasoning on the right side. For our model, we show the tracker detection and predictions in BEV, including their projection in the image.

## D Information on computation of ID Recall

In the introduction, we present the performance of the baseline trackers compared to our trajectory forecasting model on how well they can re-associate tracks after occlusion from occlusions. We measure the performance as a fraction of ground-truth tracks detected and assigned correctly before and after occlusion.

As the first step, we need to identify the ground-truth occlusion regions for every sequence. We use the visibility scores of objects and threshold those into a binary visibility flag, stating whether an object is visible in a given frame. Then, we apply a minimum rolling window on the visibility flags to get connected components and to smooth the deviations of the visibility flag values. The rolling

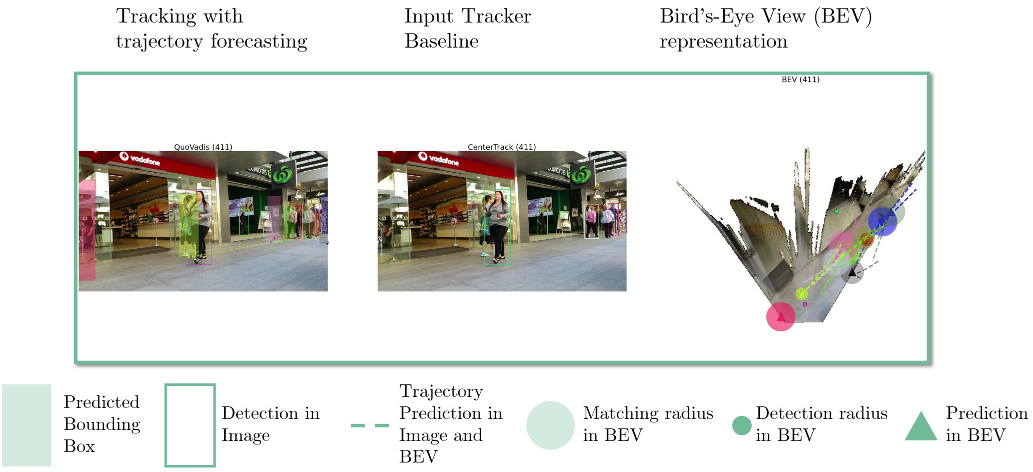


Figure 8: Description of supplementary sequence videos.

window also includes visible frames before and after the occlusion, where the actual IDSW may happen. We compute the frame ids where an occlusion starts and ends by extracting all connected components, with the visibility flag being $0$. We only consider components where the object is visible before and after the occlusion.

Finally, we check for every tracker if the tracker detected an object at the start and the end of the occlusion component and if the track ids between the beginning and start match. We use $\tau_{\text{vis}} = 0.1$ as visibility threshold and a temporal window size $\texttt{ws} = 5$ as hyperparameters.