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 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

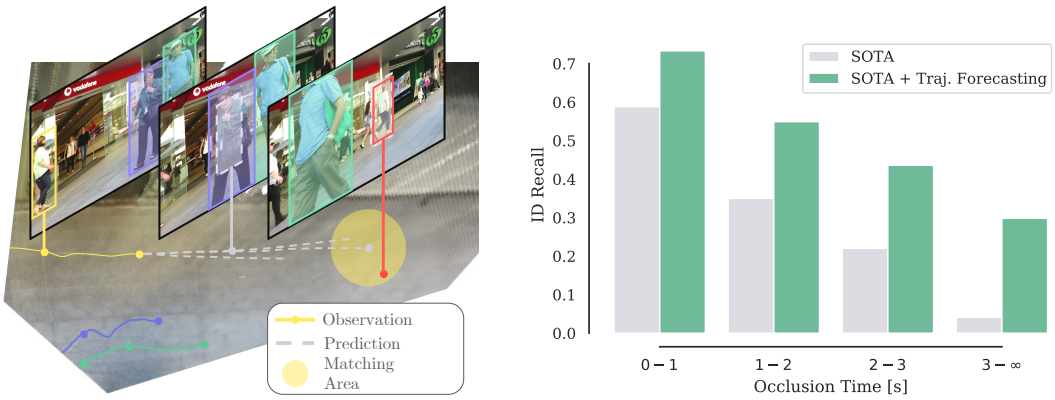
Multi-object tracking (MOT) is a long-standing research problem with applications ranging from real-time dynamic situational awareness for robot navigation [21, 15, 76, 43, 61, 10], traffic monitoring [69], studying animal behavior [51] and monitoring biological phenomena [3].

State-of-the-art MOT methods [75, 8, 4, 81, 74, 64] combine regression [75, 4] and combinatorial optimization [8] in conjunction with identity re-identification (ReID) models [32, 58, 8, 75, 65, 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, 76], 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 [53]. 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 [54, 37, 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 [79] 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 [40] 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 [78, 27, 36, 71, 48] and hand-crafting visual and motion-based descriptors [47, 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, 81] 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, 77], 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, 49, 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 [76] 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 [52, 35, 37, 54] 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].

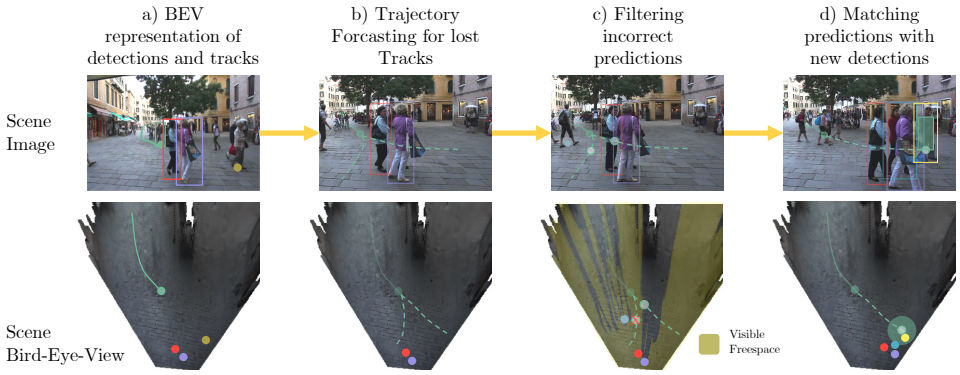


Figure 2: **Our method:** we bridge long-term occlusions by (a) localizing object tracks in BEV via estimated homography and (b) forecasting future trajectories for *lost* tracks. We (d) continually aim to match these *inactive* track predictions with new object detections and remove incorrect predictions 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 [55, 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, 55, 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, 42, 41] and estimating multimodal posterior distributions [37, 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. Given a set of 2D object detections represented as bounding boxes localized in the image plane, we

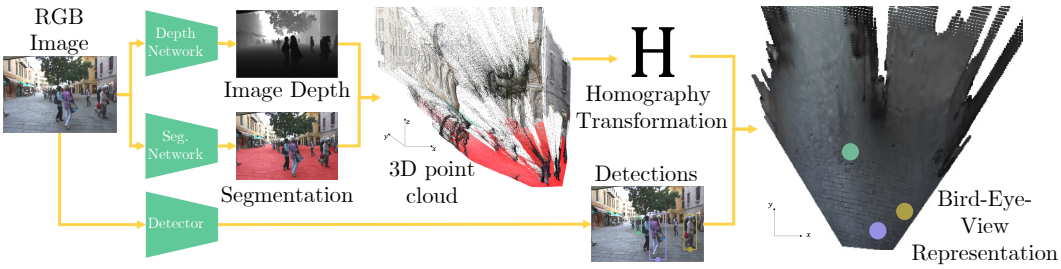


Figure 3: We estimate the homography H for a sequence by reconstructing a 3D point cloud using a monocular depth estimator. We obtain ground image-to-point-cloud correspondences using a semantic segmentation model that masks ground pixels as needed to estimate the homography matrix. With the estimated homography matrix, we transform the bottom points of bounding boxes to 2D BEV coordinates.

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.

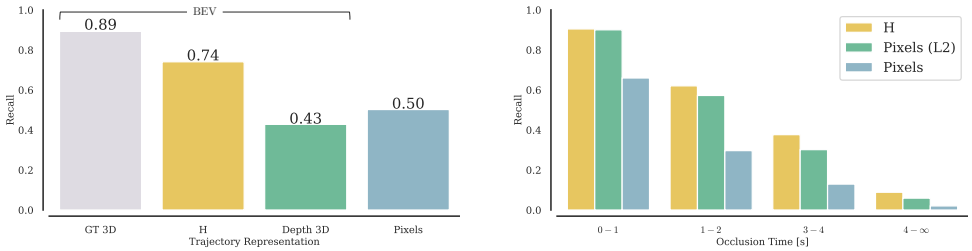
3.2 Forecasting

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 internally learns an observation-conditioned distribution of future trajectories, from which we can sample.



(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 Δ_{L_2} distance < $2m$. We also project the prediction back to the image for forecasts in bird's-eye view. The model *Pixel (L2)* predicts motion in pixel space and projects the endpoint into BEV for matching.

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 [55], or graph neural networks [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 [63]. 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. 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 τ_{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 τ_{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 τ_{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 ≥ 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 τ_{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 [80, 68, 38, 5].

This boils down to computing association costs c_{ij} between the predictions of an inactive track i and new un-associated detections j:

$$c_{ij} = (\Delta_{\text{IoU}} + \max(\tau_{L_2} - \Delta_{L_2}, 0)) \cdot (\Delta_{\text{App}} \geq \tau_{\text{App}} \text{ and } \Delta_{\text{IoU}} \geq \tau_{\text{IoU}}), \tag{1}$$

where Δ_{IoU} is the IoU score between the two bounding boxes, Δ_{L_2} is the Euclidean distance between the prediction i and a detection j in BEV, and Δ_{App} represents the cosine distance between visual features f_i and f_j. $\tau_{L_2}, \tau_{\text{IoU}}$, and τ_{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 τ_{App} and minimum IoU overlap of the bounding boxes τ_{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* [46] 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 [82, 79, 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) [40], 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^{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 [79], JDE [68], CSTrack [82], FairMOT [80], TraDes [66], QDTrack [50], CenterTrack [81] and TransTrack [62] for an evaluation 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.

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					
						$\text{FDE}_S \downarrow$	$\text{FDE}_L \downarrow$	HOTA ↑	AssA ↑	AssRe ↑	AssPr ↑	$\text{ID}_S^{\text{lost}} \downarrow$	$\text{ID}_L^{\text{lost}} \downarrow$
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** %

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 [81] 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.

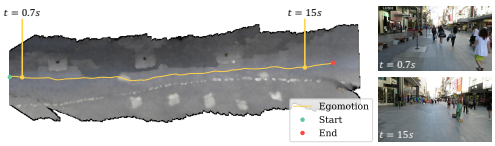


Figure 5: Visualization of BEV reconstruction for moving camera sequence and egomotion estimation.

Scores L_2	Scores IoU	Threshold τ_{IoU}	Threshold τ_{App}	HOTA ↑	AssA↑	AssRe ↑	AssPr ↑	$\text{ID}^{\text{lost}} \downarrow$
✓				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 τ 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 leads to the lowest association precision (71.31) and HOTA score (53.81) *wrt.* tracking performance.

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 [79]	CenterTrack [81]	CSTrack [82]	FairMOT [80]	JDE [68]	TraDeS [66]	TransTrack [62]	QDTrack [50]	BYTE [79]	CenterTrack [81]
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)

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 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% (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% ID_L^{lost}) is on-par with vanilla GAN (8.57% 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% 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 τ leads to the highest decrease in terms of ID^{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 τ_{App} and IoU τ_{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 ([81]) 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 [79] 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[79] 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 [79] 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 [79]	CenterTrack [81]	CSTrack [82]	FairMOT [80]	JDE [68]	TraDeS [66]	TransTrack [62]	QDTrack [50]
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 [60]	59.73	71.99	77.01	2853	57.08
CrowdTrack [67]	60.26	73.62	75.61	2544	59.26
TLR [80]	60.72	73.58	76.48	3369	58.88
MAA [59]	61.98	75.88	79.36	1452	60.16
ByteTrack [79]	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 [80]	54.42	68.44	59.57	1881	56.6
CrowdTrack [67]	54.95	68.24	70.68	3198	52.57
MAA [59]	57.28	71.15	73.90	1331	55.14
ReMOT [60]	61.15	73.14	77.42	1789	58.68
ByteTrack [79]	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 [79] 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 the 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*, June 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 booktitle 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. *arXiv:2003.09003[cs]*, 2020. arXiv: 2003.09003.

[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 *Conference on Computer Vision and Pattern Recognition*, 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] 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.

[38] 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.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[40] 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 booktitle of Computer Vision*, 2020.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. In *Journal of Photogrammetry and Remote Sensing*, 2018.

[46] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, 2016. arXiv: 1603.00831.

[47] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. In *Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] Donald B Reid. An algorithm for tracking multiple targets. In *Transactions on Automatic Control*, 1979.

[54] 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.

[55] 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.

[56] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018. arXiv: 1708.07120.

[57] 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.

[58] 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.

[59] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *Winter Conference on Applications of Computer Vision*, 2022.

[60] 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 Search form Search*, 2021.

[61] 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.

[62] 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.

[63] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no gans land. In *Proceedings of Machine Learning and Systems*, 2020.

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

[65] 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.

[66] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *International Conference on Computer Vision*, 2021.

[67] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *International Conference on Robotics and Automation*, 2020.

[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 booktitle 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] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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 booktitle of Computer Vision*, 2021.

[81] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.

[82] Zijie Zhuang, Longhui Wei, Lingxi Xie, Tianyu Zhang, Hengheng Zhang, Haozhe Wu, Haizhou Ai, and Qi Tian. Rethinking the distribution gap of person re-identification with camera-based batch normalization. In *European Conference on Computer Vision*, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See 5
   (c) Did you discuss any potential negative social impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] `https://github.com/dendorferpatrick/QuoVadis`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See 4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See 3.1
   (b) Did you mention the license of the assets? [No] The dataset is freely available for research purposes
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] No new assets were introduced
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

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.

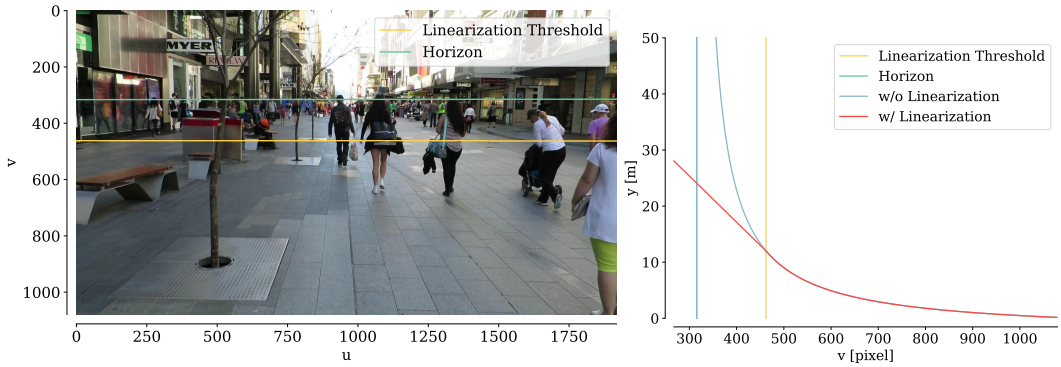
To prevent this hyperbolic growth for image points closer to the horizon, we linerarize Equation (4) around p_y^T and apply the linear transformation for all $p_y \leq p_y^T$ as shown in Figure 6b. Thus, we stabilize the distance between two points to prevent very unrealistic velocities, which would 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 $\{\hat{Y}_i^k\}_{k=1,\dots,K}$ 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 [57].

B.3 Depth Estimator

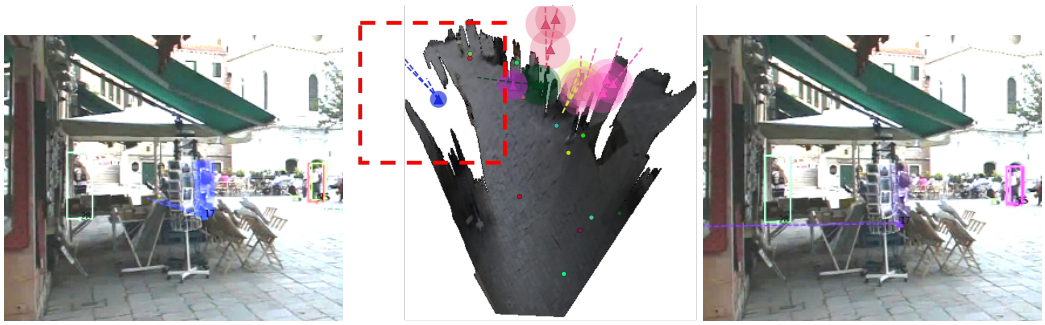
Depth estimation is a crucial part of the BEV estimation in our model. Therefore, we use a vision transformer-based [18] network [6] for monocular dense depth estimation.

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 640×480 to 960×576 we grow the transformer positional embedding vector size from 500 to 1000.

We trained the model using AdamW optimizer [39] with weight decay 10^{-2}. Following [56], the maximum learning rate λ_{max} was set to 3.5×10^{-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 960×576 on a training split of MOTSynth dataset with the batch size of 8 on 4 $RTX8000$ for one week. Then, we trained the model for 30 epochs with an image resolution of 960×576 on the full MOTSynth dataset with a batch size of 8 on 4 $RTX8000$ 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: `pavement`, `road`, `platform`, `floor`, `floor − wood`, `grass`, `sand`, `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, 44, 9, 45, 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. Additionally 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 over 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

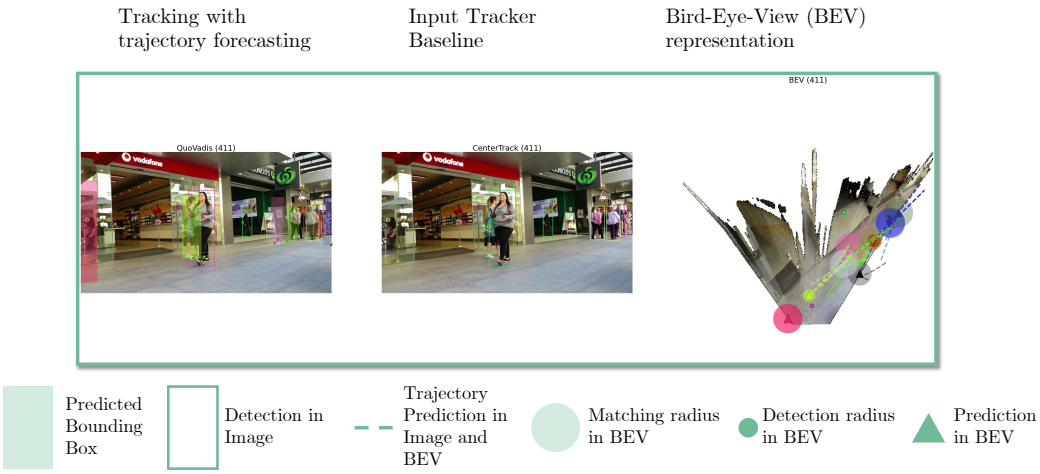


Figure 8: Description of appended sequence videos

to get connected components and to smooth the deviations of the visibility flag values. The rolling 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.