BEAM: Bilevel Evaluation and Analysis of Multi-Object-Tracking under Latency

Erik Bauer[1], Yue Yin[2,3], Mohamed Ayeb[3], Christoph Krause[2], Ludwig Brabetz[3]

***Abstract*— Environmental perception, particularly multi-object-tracking, is a crucial part of autonomous vehicles. However, system-level disturbances like latencies and jittering sampling rates can highly degrade the performance of multi-object-tracking (MOT) systems, which in turn compromises the functionality and safety of the vehicle. In this work, we propose BEAM, a novel framework for evaluating MOT performance with respect to system-level disturbances such as perception latency. BEAM utilizes a bilevel evaluation scheme: first, a tracking state error distribution is computed for both the disturbed and undisturbed system. The relative change of the two distributions is measured using the Jensen-Shannon-divergence, from which we distill an easily interpretable evaluation score. In extensive experiments, we evaluate a state-of-the-art MOT tracking system on a real-world dataset (KITTI) both with and without a spatio-temporal latency compensator, injecting different perception latencies. Comparing BEAM to current MOT evaluation metrics, we show that our proposed framework is able to provide meaningful evaluation scores under latency where other metrics begin to fail. With our work, we present a novel disturbance-focused evaluation framework which explicitly evaluates both state precision and robustness against adverse system-level conditions. By introducing BEAM, we aim to contribute to more robust, safer perception systems through disturbance-focused evaluation.**

I. Introduction

As we are progressing towards a future with autonomous cars roaming our streets, it is crucial that these autonomous systems can safely navigate through their environments and avoid collisions with other traffic participants. One of the critical tasks to achieve safe, collision-free navigation is detecting and tracking all other traffic participants (multi-object-tracking, or MOT) [35].

Research on MOT relies heavily on public large-scale datasets [8], [3], [41], [36] to evaluate proposed methods with metrics like HOTA (Higher Order Tracking Accuracy) [23] and CLEAR (CLassification of Events, Activities, and Relationships) [1]. These metrics provide valuable insights on the performance of object tracking and are integral to establishing a baseline performance evaluation for different tracking approaches. However, evaluating MOT systems is a complex task, and there is no one metric to rule them all, capturing all information about a system that is relevant for every potential use case. Instead, a landscape of different metrics has developed in the research community, going from general-purpose metrics like HOTA to task-centric metrics like PKL [29].

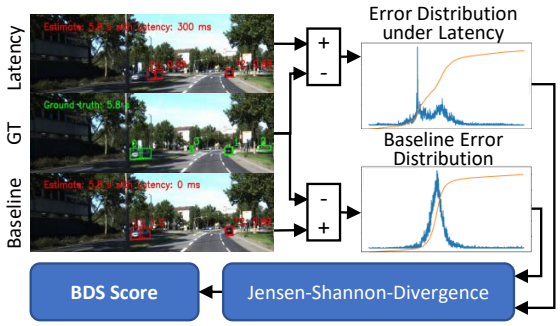


Fig. 1. BEAM evaluation methodology: first, multi-object-tracking is performed with and without system-level disturbance. The respective error distributions of the tracking results to the ground truth (GT) for both cases are computed by aggregation over objects and time and can be used for *analysis*. For comparative *evaluation*, the Jensen-Shannon-Distance (JSD) is computed in between the distributions as measure of performance degradation due to the disturbance. From the individual JSD values for each state dimension, a BEAM Disturbance Score (BDS) is computed, which serves as evaluation metric.

In this work, we aim to fill an empty spot in this landscape by introducing BEAM, a novel *disturbance-focused* evaluation framework for MOT systems (cf. Fig. 1). With this framework, we aim to complement existing metrics and offer a new perspective: systematically evaluating performance degradation with respect to disturbances, based on the statistical comparison of tracking state error distributions.

To evaluate the robustness of a MOT system with respect to a system-level disturbance, we introduce a bilevel evaluation scheme. First, we compute the state error distribution for true positive tracked objects, which is similar to the MOTP metric from CLEAR [1]. We compute these error distributions for both the undisturbed system and the disturbed system. Secondly, we introduce a comparative metric to evaluate the impact of the disturbance on the state error distributions. To this end, we compute the Jensen-Shannon-Divergence (JSD) [22] of the undisturbed and disturbed error distributions. From the computed divergences, we derive the *BEAM Disturbance Score (BDS* (cf. Fig. 1).

The BDS serves as simple indicator of robustness: the higher the score, the lower the disturbance-induced error and the more robust is the algorithm against the disturbance. We position BEAM with the BDS next to existing metrics as complementary evaluation framework, offering a new perspective specifically focused on robustness and precision.

[1]Department of Mechanical and Process Engineering, ETH Zurich. Work done during internship at BMW Group.
[2]BMW Group
[3]Department of Vehicle Systems and Fundamentals of Electrical Engineering, University of Kassel Germany
erbauer@ethz.ch, yue.yin@bmw.de, ayeb@uni-kassel.de, christoph.ck.krause@bmw.de, brabetz@uni-kassel.de

1) Contribution: Our contribution is twofold and is summarized as follows:

1) We propose BEAM, a novel bilevel methodology for evaluating the performance of MOT systems with respect to general system-level disturbances.
2) We motivate the consideration of robustness against latency as key performance indicator in evaluating MOT systems and show an exemplary application of BEAM with latency as disturbance on a real-world dataset, using different variations of a MOT system.

II. Related Work

MOT is a prominent problem in the computer vision research community. Consequently, the evaluation of MOT systems has been a research topic for considerable time. We provide an overview of selected MOT tracking methods and different evaluation paradigms and place our proposed methodology in the context of existing works.

1) MOT datasets: Research in MOT relies heavily on open datasets (KITTI [8], Argoverse [5], nuScenes [3], Waymo [36], Argoverse 2 [41]), which serve as benchmarks for measuring and comparing the performance of different MOT methods. For evaluation, some metrics like the CLEAR metrics are shared across datasets while other metrics such as HOTA are implemented only on single datasets.

In the context of latency, current datasets for MOT offer (near) perfectly synchronized data from different sensors. Temporal alignment of the data is done offline after recording. In contrast, this assumption of data synchronization and timely availability is easily violated in real-world applications. As we are reaching a state of increasing maturity for MOT systems under the assumptions given by current datasets, loosening these assumptions to explore robustness naturally presents the next challenge for the research community.

2) Multi-object-tracking: MOT considers the problem of detecting and subsequently tracking objects across frames [24], [14]. The two key problems are detection and association. In MOT research, detection is usually done on visual observations, using learning-based object detectors on LiDAR data (light detection and ranging) [26], [37], [34], [43], [39], or camera data [31], [30], [4]. In the automotive industry, radar sensors are often used, allowing for precise velocity sensing using the micro-Doppler effect [28]. The observations obtained from detectors is often formulated as detection list containing the hypothetical detections and their attributes (position, dimension, ...).

Association for online trackers is most commonly framed as an N-to-M-assignment problem, where N detections are associated to M tracks from frame $k-1$ to k, subject to some matching cost function [42], [11], [40], [17]. For evaluation, a wide variety of paradigms can be considered [16]. In the following, we will give a short overview of the most relevant metrics.

3) HOTA and CLEAR metrics: The HOTA [23] and CLEAR [1] metric families are considered the de-facto standard for MOT evaluation. Scores are computed based on the classification of tracking results as true positive (TP) or false positive (FP). To assign tracks as TP or FP, the intersection of union (IoU) of image-space bounding boxes is used: given the bounding box of an estimated track and a ground truth track, the value of the IoU is used as quality factor of the potential match, higher IoU indicating a better match. Then, the Hungarian algorithm [19] can be used to establish a maximum-quality matching of estimated to ground truth tracks. Having established this matching, classification metrics such as accuracy (HOTA/MOTA) and spatial errors like the mean localization error of TP tracks (MOTP) can be computed. Both of HOTA and CLEAR metrics are agnostic of the downstream task, instead aiming for highly general performance assessments.

4) Task-centric metrics: Task-centric metrics consider the effect of the MOT system on downstream tasks such as autonomous driving. This family of metrics evaluates a task-specific performance given a perception system. Then, different perception systems can be compared with respect to their impact on task-specific performance.

For the downstream task of autonomous driving, special consideration is placed upon criteria such as safety [25], [38] or the predicted trajectory [29], [13]. Particularly, Philion et al. [29] introduce the idea of comparing metrics obtained via ground truth observations to metrics obtained with imperfect perception systems using the Kullback-Leibler-Divergence (KL-Divergence), resulting in the PKL (Planning-KL-Divergence) metric for 3D object detection. Gog et al. [9] build upon the CARLA simulator [6] to present a platform for investigating the impact of latency on algorithms for autonomous driving.

Li et al. [21] introduce the term of streaming perception and propose a metric to evaluate the realtime capabilities of MOT systems, given simulated online time constraints. Our work shares their focus on evaluating MOT systems in a simulated real-time scenario under latency. However, we differentiate our work in two ways: an explicitly probabilistic error formulation, where we view the disturbance as conditioning and the introduction of the Jensen-Shannon-Divergence to compute an evaluation metric.

III. Methodology

In the following, we will briefly give a background on latency in real-world scenarios and define the MOT problem in the context of disturbance-focused evaluation. Then, we introduce the two levels of the BEAM framework: the computation of error distributions and the subsequent use of their divergence as indicator for performance degradation. Finally, we consider latency as concrete example of a disturbance.

A. Background: Latency in Real-World Scenarios

Our proposed evaluation framework is designed to support various types of disturbances. In this work, we choose to place our focus on the issue of online perception, as it is a traditionally underrepresented research field, but has a high impact on real-world applications. Particularly for safety-critical downstream tasks like autonomous driving, it

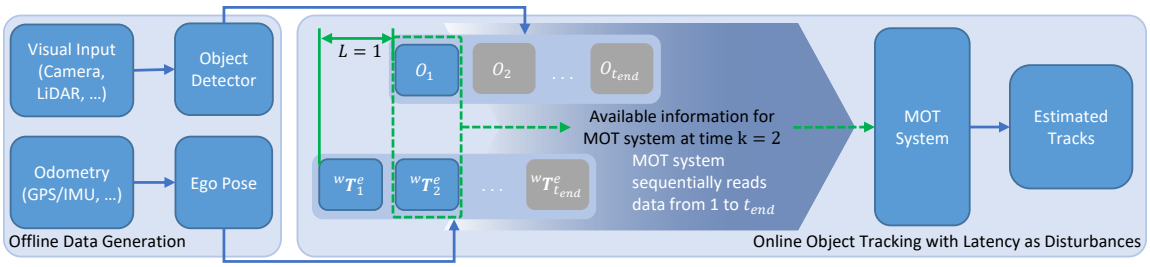


Fig. 2. A flowchart for the dataflow of the typical MOT system with latency $L = 1$ discrete timestep (equivalent to 100 ms for a frequency of $f = 10$ Hz). We introduce a latency L on the visual observations, modelling the high latency that object detectors can incur. Green dotted box indicates the available information for MOT system at timestep k.

is imperative that systems can be systematically evaluated under adverse conditions. In the scope of this work, we focus on latency. At the junction of non-realtime computing with real-time control systems, it can have a significant impact on both the performance and safety of the overall system [7], [35]. Furthermore, Li et al. [21] find that online perception, and thereby exposure to issues such as latency, poses a significantly harder challenge than the offline perception tasks most current benchmarks rely on. Yet, nearly all real-world applications of MOT systems have strong requirements for accurate online perception systems.

In the following, we provide a brief characterization of latency in real-world use cases and its potential impact.

1) Latency in autonomous systems: In most autonomous systems, there are various sources of latencies which can differ by order of magnitudes depending on the sensors and algorithms used [7], [10]. For applications such as autonomous driving, LiDAR sensors are commonly used for environmental perception [2], [32]. On modern desktop computers, inference times for object detection algorithms for LiDAR data average around 100 ms [26], and on mobile platforms with resource-constrained computing hardware, latencies tend to only increase. For sensor fusion, latency could be mitigated by matching timestamps from data coming from different sensors. However, for systems with hard real-time constraints, control performance and safety can potentially be put at risk due to consequently increased response times.

2) Latency in dynamic maneuvers: Autonomous systems are the most vulnerable to latency during highly dynamic maneuvers. For example, an AEB (Automated Emergency Braking) system in cars is designed to trigger a braking maneuver to avoid or mitigate collisions in safety-critial driving situations. Given the hard real-time constraints present in these scenarios, latency in perception, planning and actuation leads to performance degradations or even safety-relevant failures. For instance, the CCRs-scenario (Car-to-Car Rear Stationary) in NCAP (New Car Assessment Program) for AEB Inter-Urban requires the ego vehicle to drive towards a stationary object at a speed of 80 km/h [33]. A 300 ms latency in environmental perception without any countermeasure results in a reduction of approximately 6.67 m of distance available for the ego vehicle to decelerate and stop safely - significantly more than the length of an average car.

B. Methodology Outline

An architectural outline of our evaluation architecture is shown in Figure 1. For a given tracking system, we compute our proposed BDS metric through comparing the error distributions of the undisturbed baseline system with the error distribution of the disturbed system. We present our methodology by defining a standard MOT system, followed by the disturbance-informed computation of the error distributions. Finally, we derive the computation of the BDS evaluation metric given the two error distributions.

C. Defining the MOT System

We consider an online MOT system ϕ which at discrete timestep k takes in $M_k \in \mathbb{N}_0$ observations $\mathbf{O}_k = (\mathbf{o}_k^1, \mathbf{o}_k^2, \ldots, \mathbf{o}_t^{M_k})$ and ego pose matrix ${}^w\mathbf{T}_k^e \in SE(3)$ ($SE(3)$: 3D Special Euclidean Group [20]) and maps them to $N_k \in \mathbb{N}_0$ tracks $\hat{\mathbf{S}}_k = (\hat{\mathbf{s}}_k^a, \hat{\mathbf{s}}_k^b, \ldots, \hat{\mathbf{s}}_k^c)$ where $a, b, \ldots, c \in \mathbb{N}_0$ are the IDs assigned to the different tracks. The ground truth for estimated track i is denoted without diacritical mark (i.e. bar or hat) as $\mathbf{s}_k^i$, if it exists.

All collected observations at time index k are $\mathbf{O}_{1:k} = (\mathbf{O}_1, \ldots, \mathbf{O}_k)$. Similarly, we denote all collected ego poses and tracks as ${}^w\mathbf{T}_{1:k}^e$ and $\hat{\mathbf{S}}_{1:k}$, respectively. The superscripts w and e denote the transformation from world to ego reference frame. The system ϕ generates tracking state estimates as follows:

$$\hat{\mathbf{S}}_k = \phi\left(\mathbf{O}_{1:k}, {}^w\mathbf{T}_{1:k}^e\right) \tag{1}$$

We assume the outputs of the MOT system $\hat{\mathbf{s}}_k^i \in \mathbb{R}^7$ to be constructed from standard 3D bounding boxes which are defined by 3D position $\mathbf{p}_k^i \in \mathbb{R}^3$, dimensions $\mathbf{d}_k^i \in \mathbb{R}^3$ and a rotation angle $\psi_k^i \in \mathbb{R}$ normal to the ground:

$$\hat{\mathbf{s}}_k^i = \begin{pmatrix} \hat{\mathbf{p}}_k^i & \hat{\mathbf{d}}_k^i & \hat{\psi}_k^i \end{pmatrix}^T \tag{2}$$

Furthermore, observations and tracks taken at time k are defined in the corresponding ego reference frame ${}^w\mathbf{T}_k^e$. Additionally, for each estimated track, we have access to a confidence score. For simplicity, we will assume a constant frequency. The final timestep is denoted as t_{end}.

D. Applying the BEAM Framework

To evaluate the performance degradation a MOT system experiences with respect to the disturbance $D_{1:t_{end}}$, we consider the state error of the estimated tracks with respect to their ground truth. Aggregating state errors for true positive estimated tracks, we compute the error distribution with no disturbance, which serves as baseline. Then, in a similar fashion, we compute the error distribution with a disturbance applied, and finally consider the JSD (Jensen-Shannon-Distance [22]) of the two error distributions.

1) Computing track error distributions: Given the matching from detection to ground truth (cf. Sec. III-E.1), we compute the baseline error distribution with no disturbance. We can compute the individual state errors as follows:

$$\mathbf{e}_k^i = \hat{\mathbf{s}}_k^i - \mathbf{s}_k^i \tag{3}$$

For the estimated tracks with the disturbance introduced (denoted as $\tilde{\mathbf{s}}_k^i$), we proceed identically, using the similarly obtained assignments:

$$\tilde{\mathbf{e}}_k^i = \tilde{\mathbf{s}}_k^i - \mathbf{s}_k^i \tag{4}$$

We recall that the state error is a 7-dimensional vector, containing the error of the 3D centroid, the dimensions and the error of the rotation angle normal to the ground at time k. To then obtain the baseline error distribution and disturbance-affected error distribution for evaluation, we aggregate all computed state errors through time, giving us sets of errors in the j-th state dimension: $\mathbf{E}_j = \{\mathbf{e}_k^{i,j}\}_{k=1,i=1}^{k=t_{end},i=N_k}$ and $\tilde{\mathbf{E}}_j = \{\tilde{\mathbf{e}}_k^{i,j}\}_{k=1,i=1}^{k=t_{end},i=N_k}$. From these sets of errors, we define the baseline error distributions as $p(\mathbf{E}_j)$ and the error distributions conditioned on disturbance $D_{1:t_{end}}$ as $p(\tilde{\mathbf{E}}_j \mid D_{1:t_{end}})$. We compute the Jensen-Shannon-Distance (JSD) [22] of the individual error distributions as measure of difference in between the probability distribution functions:

$$\mathrm{BDS}_j = 1 - \mathrm{JSD}(p(\tilde{\mathbf{E}}_j \mid D_{1:t_{end}}) \,||\, p(\mathbf{E}_j)) \tag{5}$$

Here, we use the JSD over the KL-Divergence as it is less sensitive to outliers and (using base 2 for logarithms) bound to the interval of $[0, 1]$. It follows that the BEAM Disturbance Score (BDS) is bound to the same interval. Intuitively, the lower the BDS, the larger the deviation from the baseline error distribution is and the worse the performance is. Finally, to obtain a single, simply interpretable metric over all states, we take the mean of the individual scores for all 7 state dimensions:

$$\mathrm{BDS} = \frac{1}{7}\sum_{j=1}^{7} \mathrm{BDS}_j \tag{6}$$

E. Introducing Latency as Disturbance

We introduce desynchronization in between the visual observations and the ego pose from odometry by imposing a latency L on the visual observations (cf. Fig. 2). This latency is a sum of different potential delays such as measurement latency, processing latency or communication latency. L is assumed to be known and for now, constant. The new output tracks of the MOT system under latency will be denoted by $\tilde{\mathbf{S}}_k$. Concretely, imposing L, we can write:

$$\tilde{\mathbf{S}}_k = \phi(\mathbf{O}_{1:k-L}, {}^w\mathbf{T}_{1:k}^e) \tag{7}$$

At each timestep, we obtain estimated tracks $\tilde{\mathbf{S}}_k$. These tracks are obtained by combining delayed observations up to $\mathbf{O}_{k-L}$ with pose information up to ${}^w\mathbf{T}_k^e$ (cf. Fig. 2). Given this definition of our disturbed system, we can move forward by applying BEAM to evaluate the performance degradation that latency L incurs.

1) Associating estimated tracks to ground truth: As a preliminary to computing differences in between estimated tracks $\hat{\mathbf{S}}_k$ (or $\tilde{\mathbf{S}}_k$) and ground truth tracks $\mathbf{S}_k$, we need to find an assignment in between them. We first compute an assignment of observations (object detections) to ground truth tracks with no disturbance in effect. Concretely, we find a matching from true positive observations in $\mathbf{O}_k$ to the ground truth $\mathbf{S}_k$.

Then, we perform tracking with ϕ using the observations $\mathbf{O}_k$ or in the case of a latency-disturbed system, $\mathbf{O}_{k-L}$ to produce estimated tracks $\hat{\mathbf{S}}_k$ (respectively $\tilde{\mathbf{S}}_k$). For the undisturbed system, we compute an assignment from $\hat{\mathbf{S}}_k$ to $\mathbf{O}_k$, by which we find an assignment from $\hat{\mathbf{S}}_k$ to $\mathbf{S}_k$.

For the latency-disturbed system (Eq. (7)), we match from $\tilde{\mathbf{S}}_k$ to $\mathbf{O}_{k-L}$. For each true positive estimated track in $\tilde{\mathbf{S}}_k$, the matching gives us the ID of the corresponding ground truth track in $\mathbf{S}_{k-L}$. However, we aim to compute the error to the ground truth at timestep k. Taking advantage of persistent ground truth IDs, we can make a valid association to $\mathbf{S}_k$ if the ground truth ID from $k-L$ is also present at k, otherwise, we discard the match as invalid.

To find each assignment, we filter out false positive observations using the detection confidence (threshold 0.8) and centroid distance to the nearest ground truth ($1.5\,\mathrm{m}$) as simple thresholds. We then use the Hungarian algorithm [19] for graph-based matching with Euclidean centroid distances as costs.

IV. Experiments on Real-World Data

To illustrate the use of BEAM, we apply it with latency as disturbance on the KITTI Tracking dataset [8]. The KITTI dataset provides data at $10\,\mathrm{Hz}$, which allows us to inject latencies in increments of $100\,\mathrm{ms}$ as shown in Fig. 2. Other modern datasets such as the nuScenes dataset [3] only provide data at $2\,\mathrm{Hz}$, which is insufficient for our purposes. The issue of application on other datasets with lower framerates will be left for future work, where interpolation techniques as shown in by Li et al. [21] could be used.

For performing MOT, we use the provided LiDAR readings, the pose data and the ground truth annotations for cars. With these experiments, we show 3 key insights:

1) Existing MOT metrics are insufficient to evaluate systems under disturbances.
2) Generating a probabilistic overview of the error yields valuable and interpretable results to understand the MOT system behavior and limitations.

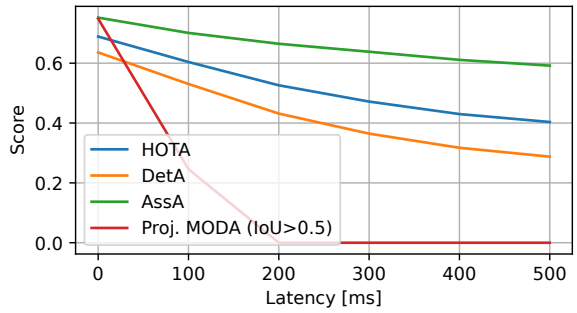


Fig. 3. Average HOTA, detection accuracy (DetA) and association accuracy (AssA) scores [23] for evaluation on the KITTI Tracking dataset using no compensator. Furthermore, we show an adaptation of the MODA score from the CLEAR metrics [1] which uses the IoU of bounding boxes projected on the ground plane instead of the image plane, which goes to zero after 200 ms. In Section IV-B, we describe the degradation of these classical metrics under latency.

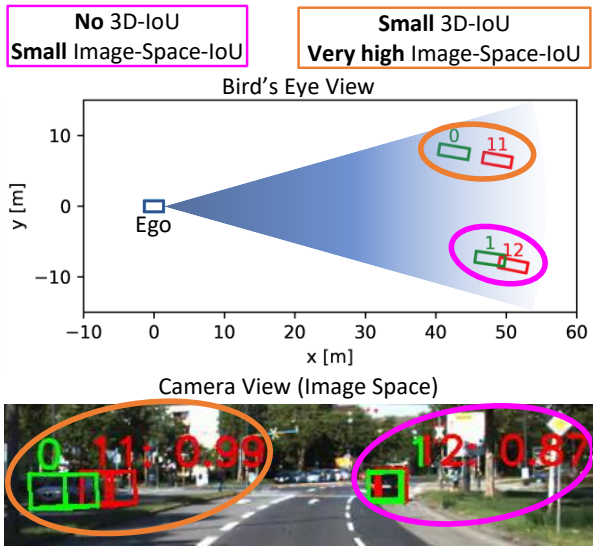


Fig. 4. A crop of frame 53 of sequence 0002 of the KITTI Tracking dataset with 300 ms latency. The ground truth is shown in green, estimated tracks in red. We can observe the inconsistency of image-space IoU-based matching under latency: For the pair 0-11, the image-space IoU is small, while for the pair 1-12, it is very high. From the bird's eye view, however, the 3D-IoU of both pairs is small or zero. In contrast to our method, both matching methods fail to *consistently* identify TP matches under latency.

3) Considering the divergence from a baseline error distribution is a powerful way to distill the probabilistic error perspective into interpretable scores.

A. Experimental Setup

To demonstrate our key insights, the examples we show in this work were all obtained with the combination of the Voxel-RCNN detector [39] pretrained on the KITTI 3D Object Detection dataset and 3D-MOT tracker [42], which are a state-of-the-art detector and tracker with open-source code available for easy use. In addition to evaluating the barebones tracker, we propose a spatio-temporal latency compensator module (cf. Sec. IV-D) which can be inserted in between the detector and tracker in order to compensate for latency through kinematic predictions.

For evaluation, we use all 21 training sequences of the KITTI Tracking dataset, which provide hand-labelled ground truth annotations for vehicles. We investigate the error behavior for latencies ranging from 0 ms to 500 ms.

B. Evaluation with Classical Metrics

The evaluation with classical metrics such as the HOTA [23] family of metrics shows us that we can see a clear performance degradation with increasing latency (cf. Fig. 3). However, this can be attributed to classification of more tracks as false positives and gives us little intuition of the nature of the performance degradation.

1) Degradation of TP/FP classification: We can observe that the ability to consistently match a track to its ground truth is quickly diminished under disturbances such as latency (cf. Fig. 4) through either small or no image-space IoU. As more latency incurs larger spatial shifts, traditional methods like HOTA and CLEAR tend to misclassify tracks as false positives. This incurs a large performance loss in classification metrics (cf. Fig. 3).

For metrics like MOTP, which consider the mean localization error of tracks to their ground truth, FP tracks are discarded. As the number of FP tracks rises, the total number of samples from which the MOTP is computed decreases, which in turn diminishes the power of the metric. Comparing image-space IoU to the 3D-IoU (as introduced by Weng et al. [40]), the number of TP tracks with correctly associated ground truth still remains higher as the image-space IoU is less sensitive to spatial shifts (cf. Fig. 4).

In general, disturbance-unaware evaluation metrics that depend on a matching to ground truth ("oracle-dependent" metrics [12]) will experience similar degradations. While performance in such metrics visibly decreases, their results lose their interpretability by simply classifying most tracks as false positives.

2) Our approach: In contrast, BEAM is a disturbance-aware evaluation framework. Given sufficient knowledge about the disturbance, we can *uphold matching quality* and thereby extract meaningful information about the performance degradation. To extract meaningful information, we use state error distributions. They allow for a fine-grained overview of the state errors (for true positive objects) instead of the binary classification used by metrics like HOTA or MOTA. In comparison to MOTP, our probabilistic approach allows for divergence-based comparisons using the JSD. The remaining tradeoff is that BEAM is not considering classification metrics: false positive detections are discarded (as they have no ground truth to be compared to) and have no impact on the score. This underlines that BEAM is designed to complement existing metrics and provide additional in-depth information about the precision and robustness of the evaluated MOT system.

C. Evaluation With BEAM

In Fig. 5, we can observe the BDS scores for latencies up to 500 ms. For brevity, we only show the longitudinal error

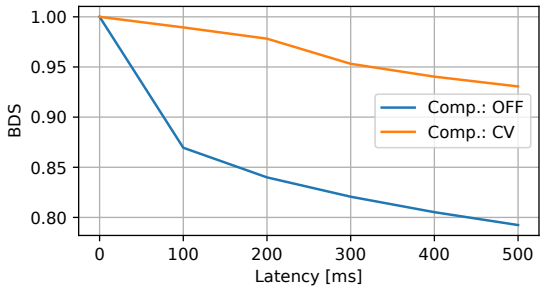


Fig. 5. BDS scores for latencies from 0 ms to 500 ms for both the tracker with constant velocity (CV) spatio-temporal compensator and without (OFF). The increased BDS score using the compensator clearly reflects the improved error distribution shown for $L = 3$ (equivalent to 300 ms) in Fig. 6c.

distributions in Fig. 6, which have the highest impact on the final scores.

We recall that high BDS scores (e.g. close to the maximum of 1) are indicative of robust performance under latency. Following expectations, scores achieved with the compensator introduced in Sec. IV-D are significantly higher than uncompensated scores. With the compensator (Fig. 6c), we can observe nearly a return to the baseline error distribution despite 300 ms latency. Further discussion follows in Sec. IV-E.

D. Compensating for Known Latency

In the following, we propose a spatio-temporal compensator Γ for a known latency which relies on simple kinematic models. To illustrate the use of BEAM in evaluating and comparing different systems, we will use the previously shown tracker with the added latency compensator.

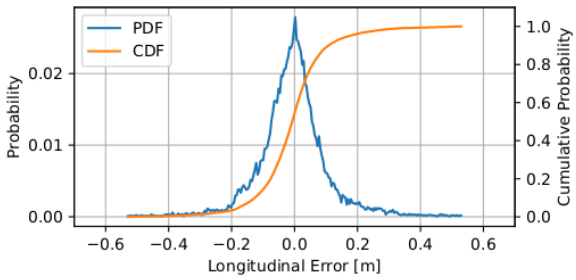
1) Compensator definition: Let Γ be the compensator, which takes in the output of the disturbed MOT system ϕ together with latency L and maps to a corrected tracking output $\bar{\mathbf{S}}_k$. We want to compensate for the displacement $\delta_{k|k-L}$ which a tracked vehicle $\hat{\mathbf{s}}^i_k$ experiences. As we assume that we have access to ego poses, this largely amounts to predicting the movement of the tracks under latency $\hat{\mathbf{S}}_{k-L}$ from timestep $k-L$ to timestep k to compute the spatio-temporally-aligned tracks $\bar{\mathbf{S}}_{k|k-L}$:

$$\bar{\mathbf{S}}_{k|k-L} = \Gamma\left(\phi(\mathbf{O}_{1:k-L}, {}^w\mathbf{T}^e_{1:k}), L\right) \tag{8}$$

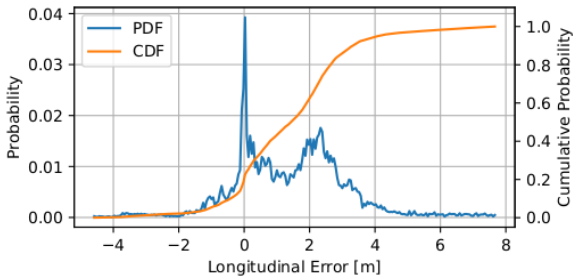
$$= \Gamma\left(\hat{\mathbf{S}}_{1:k-L}, {}^w\mathbf{T}^e_{1:k}, L\right) \tag{9}$$

For our approach and the limited scope of this work, we propose a simple first-order spatio-temporal compensator that is based upon a constant velocity (CV) assumption. We define Γ through two operations: first, we perform *temporal* alignment and transform each tracked object $\hat{\mathbf{s}}^i_{k-L}$ into the current ego reference frame ${}^w\mathbf{T}^e_k$. Let $\hat{\mathbf{p}}^i_{k-L}$ be the position of the tracked object and $(\hat{\mathbf{p}}^i_{k-L})^*$ its position in the updated reference frame ${}^w\mathbf{T}^e_k$:

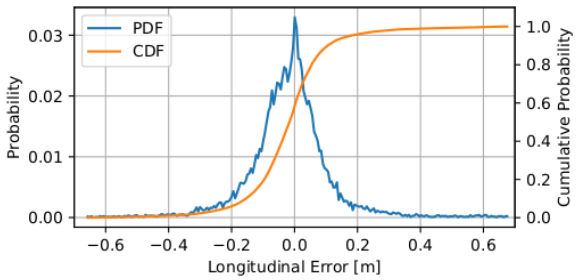
$$(\hat{\mathbf{p}}^i_{k-L})^* = ({}^w\mathbf{T}^e_k)^{-1}\, {}^w\mathbf{T}^e_{k-L}\hat{\mathbf{p}}^i_{k-L} \tag{10}$$



(a) Error distribution without latency present ($L = 0$).



(b) Error distribution with 300 ms latency ($L = 3$).



(c) Error distribution with 300 ms latency ($L = 3$) and spatio-temporal compensator.

Fig. 6. Longitudinal error distributions with and without latency, shown with probability distribution function (PDF) and cumulative distribution function (CDF). Notably in Figure 6b, the distribution appears to become multi-modal with one significant peak around zero and one peak around 2 m. Using the compensator, we can see in Fig. 6c that the error distribution returns to its original shape with slightly wider tails.

Similarly, $(\hat{\mathbf{p}}^i_{k-L-1})^*$ is the previous position expressed in ${}^w\mathbf{T}^e_k$. Then, we approximate the velocity at time $k-L$ through backwards-differences:

$$\frac{\partial}{\partial t}(\hat{\mathbf{p}}^i_{k-L})^* = \frac{1}{T}\left[(\hat{\mathbf{p}}^i_{k-L})^* - (\hat{\mathbf{p}}^i_{k-L-1})^*\right] \tag{11}$$

with T denoting the duration of a timestep. Finally, we perform *spatial* alignment by correcting the object position with

$$\bar{\mathbf{p}}^i_{k|k-L} \approx (\hat{\mathbf{p}}^i_{k-L})^* + L\frac{\partial}{\partial t}(\hat{\mathbf{p}}^i_{k-L})^* \tag{12}$$

If the object was not detected in the last timestep, we use the predicted position of the object given by the tracker.

2) Computing derivatives of detected objects: In most modern cars, radar sensors are commonly used to obtain highly precise information about velocities of detected objects [2], [32]. However, radar sensors are not commonly

Method	Mean [m]	Std. [m]	99th-perc. [m]
Baseline (Fig. 6a)	0.00	0.13	0.54
No comp. (Fig. 6b)	1.48	1.82	7.71
CV comp. (Fig. 6c)	0.00	0.15	0.68

TABLE I

ERROR STATISTICS FOR FIG. 6.

used as modality in MOT datasets and therefore, we rely on state estimation. We find that for the scope of this work, using backwards-differences presents a simple solution to illustrate the potential of the CV model. Alternatively, common solutions for state estimation that could be used are Kalman filters [15], differentiable filters [18] or learning-based velocity estimation [27]. However, these methods require extensive tuning or large datasets, which position them outside the scope of this work.

3) Experiments with compensator: We repeat previous experiments with the added compensator. Evidently, this relatively simple method performs well for the given scenarios as shown by comparing the error distributions in Fig. 6a and Fig. 6c and considering the reduced BDS in Fig. 5. Despite increased tail errors, we still see a reduction in the order of a magnitude of the 99-th percentile error compared to the uncompensated case (cf. Tab. I), underlining the potential of employing even simple models.

E. Discussion

In this section, we showcased the methodology behind our proposed BEAM framework, differentiating it from current MOT evaluation metrics [23], [1], [40] (cf. Sec. IV-B) and showing its capabilities using the KITTI Tracking dataset with latency as disturbance (cf. Sec. IV-C). Evaluating a state-of-the-art MOT system with and without a latency compensator (cf. Sec. IV-D), we show the suitability of BEAM for effective error analysis and evaluation of system robustness. In the following, we will discuss the obtained results with the proposed methods.

1) Spatio-temporal compensator: Using the BEAM framework, we observe only a small performance degradation in the BDS (cf. Fig. 5) and statistical error measures when using the compensator (cf. Tab. I). It appears the constant velocity assumption during each timestep is sufficiently robust for most scenarios in the KITTI Tracking dataset to keep the error distribution close to the baseline. Nevertheless, the CV assumption would fail in scenarios with more pronounced acceleration. In more dynamic scenarios, a constant acceleration model or using a mixture of learning-based and model-based compensator may prove to be more robust: however, such scenarios are not considered in the KITTI Tracking dataset. We present this as future research opportunity to apply our evaluation method to more dynamic scenarios where different compensators may be used to cope with the problem.

2) BEAM versus classical metrics: While we can see a performance degradation with classical metrics (cf. Sec. IV-B), we can attribute these performance degradations to simply classifying most estimated tracks as false positives instead of correctly associating them to their ground truth and then considering the occuring error. To eliminate this limitation, we introduce disturbance-aware ground truth matching, which allows us to consistently match tracks to their ground truth, even under disturbances like latency.

Being able to consistently match tracks to their ground truth allows us to perform reliable, meaningful error analysis that can give us rich insights into the failure modes of the system under test: in Fig. 6, we can see how the error distribution degrades from undisturbed baseline to multi-modal distribution with significantly increased tail errors. In turn, we can also see how the error distribution of the system with compensator closely resembles the original distribution.

These changes in the distribution are clearly reflected in the BDS (cf. Fig. 5), which we can leverage for simple comparative evaluation of the performance degradation with respect to an undisturbed baseline. For comparative evaluation, the BDS distills the more complex probabilistic perspective into an easy-to-interpret score in between zero and one. The applications of the BDS are flexible: either comparing the impact of different disturbances on a system or comparing different systems under the same disturbance. This could allow the BDS to be used alongside different standard metrics for benchmarking different perception systems.

V. CONCLUSION

With technology becoming increasingly complex and advanced, we need to push the boundaries of evaluation and certifiable safety. Especially for autonomous driving tasks, only the most robust systems will gain the trust and acceptance of their users. With this work, we aim to showcase BEAM, a method of systematically challenging the assumptions we are traditionally operating on in perception benchmarks and how we can evaluate current MOT systems with respect to system-level disturbances. Focusing on latency as disturbance, we applied BEAM to analyse its impact on MOT systems and showed how we can counteract latency with a simple compensator module for state forecasting.

However, there are many more possible disturbances than latency: there is a vast number of adverse factors that can impact MOT systems that we lay out as future work to investigate: jittering sampling rates, message loss and many more. The formulation of BEAM is flexible by design to accomodate future developments aiming to build a catalogue of disturbances we can evaluate against. With BEAM, we take a step towards a unified framework to evaluate perception robustness, facilitating the future development of more robust and subsequently safer systems.

REFERENCES

[1] Keni Bernardin and Rainer Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[2] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C-emerging Technologies*, 89:384–406, 2018.

[3] 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, June 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[5] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator, 13–15 Nov 2017.

[7] Davide Falanga, Suseong Kim, and Davide Scaramuzza. How Fast Is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid. *IEEE Robotics and Automation Letters*, 4:1884–1891, 2019.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[9] Ionel Gog, Sukrit Kalra, Peter Schafhalter, Matthew A. Wright, Joseph Gonzalez, and Ion Stoica. Pylot: A Modular Platform for Exploring Latency-Accuracy Tradeoffs in Autonomous Vehicles. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8806–8813, 2021.

[10] Roelof Hamberg, Teun Hendriks, and Tjerk Bijlsma. Temporal Performance of Advanced Driver Assistance Systems vis-á-vis Human Driving Behavior in Dense Traffic. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1292–1297, 2015.

[11] Jiawei He, Chun yan Fu, and Xiyang Wang. 3D Multi-Object Tracking Based on Uncertainty-Guided Data Association. *ArXiv*, abs/2303.01786, 2023.

[12] Michael Hoss. Checklist to Transparently Define Test Oracles for TP, FP, and FN Objects in Automated Driving. *ArXiv*, abs/2308.07106, 2023.

[13] B. Ivanovic and Marco Pavone. Injecting Planning-Awareness into Prediction and Detection Evaluation. *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 821–828, 2021.

[14] Diego M. Jiménez-Bravo, Álvaro Lozano Murciego, André Sales Mendes, Héctor Sánchez San Blas, and Javier Bajo. Multi-object tracking in traffic environments: A systematic literature review. *Neurocomputing*, 494:43–55, 2022.

[15] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*, volume 82, pages 35–45. 03 1960.

[16] Ch.Revathi Ramesh Karri, José Machado Da Silva, and Miguel Velhote Correia. Key Indicators to Assess the Performance of LiDAR-Based Perception Algorithms: A Literature Review. *IEEE Access*, 11:109142–109168, 2023.

[17] Aleksandr Kim, Aljosa Osep, and Laura Leal-Taixé. EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11315–11321, 2021.

[18] Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, 45:561 – 578, 2020.

[19] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 1955.

[20] Steven M. LaValle. *Planning algorithms*. 2006.

[21] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception, 2020.

[22] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory*, 37:145–151, 1991.

[23] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and B. Leibe. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *International Journal of Computer Vision*, 129:548 – 578, 2020.

[24] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021.

[25] Maria Lyssenko, Christoph Gladisch, Christian Heinzemann, Matthias Woehrle, and Rudolph Triebel. Towards Safety-Aware Pedestrian Detection in Autonomous Systems. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 293–300, 2022.

[26] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3D Object Detection for Autonomous Driving: A Comprehensive Survey. *International Journal of Computer Vision*, 131:1909 – 1963, 2022.

[27] Robert McCraith, Lukás Neumann, and Andrea Vedaldi. Real Time Monocular Vehicle Velocity Estimation using Synthetic Data. *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1406–1412, 2021.

[28] Andre Pearce, J. Andrew Zhang, Richard Xu, and Kai Wu. Multi-Object Tracking with mmWave Radar: A Review. *Electronics*, 12(2), 2023.

[29] Jonah Philion, Amlan Kar, and Sanja Fidler. Learning to Evaluate Perception Models Using Planner-Centric Metrics. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14052–14061, 2020.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection, June 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[32] Francisca Rosique, Pedro J. Navarro, Carlos Fernández, and Antonio Padilla. A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research. *Sensors (Basel, Switzerland)*, 19, 2019.

[33] Richard Schram, Aled Williams, and Michiel R. van Ratingen. Implementation of Autonomous Emergency Braking (AEB), the Next Step in Euro NCAP's Safety Assessment. 2013.

[34] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10526–10535, 2019.

[35] Chen Sun, Ruihe Zhang, Yukun Lu, Yaodong Cui, Zejian Deng, Dongpu Cao, and Amir Khajepour. Toward Ensuring Safety for Autonomous Driving Perception: Standardization Progress, Research Advances, and Perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] OpenPCDet Development Team. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet, 2020.

[38] Georg Volk, Jörg Gamerdinger, Alexander von Bernuth, and Oliver Bringmann. A Comprehensive Safety Metric to Evaluate Perception in Autonomous Systems. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2020.

[39] Hai Wang, Zhiyu Chen, Yingfeng Cai, Long Chen, Yicheng Li, Miguel Ángel Sotelo, and Zhixiong Li. Voxel-RCNN-Complex: An Effective 3-D Point Cloud Object Detector for Complex Traffic Conditions. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

[40] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366, 2019.

[41] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[42] Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5668–5677, 2022.

[43] Qiang Zhou and Chaohui Yu. Point RCNN: An Angle-Free Framework for Rotated Object Detection. *Remote Sensing*, 14(11), 2022.