2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

STT: Stateful Tracking with Transformers for Autonomous Driving

Longlong Jing*, Ruichi Yu*†, Xu Chen*, Zhengli Zhao, Shiwei Sheng,
Colin Graber, Qi Chen, Qinru Li, Shangxuan Wu, Han Deng, Sangjin Lee,
Chris Sweeney, Qiurui He, Wei-Chih Hung, Tong He, Xingyi Zhou‡,
Farshid Moussavi, James Guo, Yin Zhou, Mingxing Tan, Weilong Yang, Congcong Li
Waymo LLC, ‡Google Research

***Abstract*— Tracking objects in three-dimensional space is critical for autonomous driving. To ensure safety while driving, the tracker must be able to reliably track objects across frames and accurately estimate their states such as velocity and acceleration in the present. Existing works frequently focus on the association task while either neglecting the model's performance on state estimation or deploying complex heuristics to predict the states. In this paper, we propose STT, a Stateful Tracking model built with Transformers, that can consistently track objects in the scenes while also predicting their states accurately. STT consumes rich appearance, geometry, and motion signals through long term history of detections and is jointly optimized for both data association and state estimation tasks. Since the standard tracking metrics like MOTA and MOTP do not capture the combined performance of the two tasks in the wider spectrum of object states, we extend them with new metrics called S-MOTA and MOTP_S that address this limitation. STT achieves competitive real-time performance on the Waymo Open Dataset.**

I. Introduction

3D Multi-Object Tracking (3D MOT) plays a pivotal role in various robotics applications such as autonomous vehicles. To avoid collisions while driving, robotic cars must reliably track objects on the road and accurately estimate their motion states, such as speed and acceleration. While development of 3D MOT has made much progress in recent years, most methods [1], [2], [3] still use approximated object states as intermediate features for data association without explicitly optimizing model performance on state estimation. Although some tracking methods [4], [5], [6], [7] exist that predict motion states, they often do so by employing filter-based algorithms such as the Kalman filter (KF) with complex heuristic rules [1], [3], [8] to estimate object states and cannot easily utilize appearance features or raw sensor measurements in a data-driven fashion [9]. While there are machine learning-based methods [10] that add prediction heads to detection models to estimate motion states, they struggle to produce consistent tracks from long-term temporal information due to computational and memory limitations.

To address the limitations of existing approaches, we introduce STT, a Stateful Tracking model with Transformers, which combines data association and state estimation into a single model. At the core of our model architecture are a Track-Detection Interaction (TDI) module that performs data association by learning the interaction between a track and its surrounding detections and a Track State Decoder (TSD) that produces the state estimation of the tracks. All the modules are jointly optimized (Figure 2), which allows STT to obtain superior performance while simplifying the system complexity.

Existing tracking evaluation mainly use multi-object tracking accuracy (MOTA) and multi-object tracking precision (MOTP) [11] to measure the association and localization quality, but they do not take the quality of other states into account such as velocity and acceleration. To explicitly capture the full state estimation quality of the tracking performance, we extend the existing evaluation metric MOTA to Stateful MOTA (S-MOTA) which enforces accurate state estimation during label-prediction matching, and MOTP to MOTP_S which applies to arbitrary state variables so that we can assess the quality of the state estimation beyond position.

To demonstrate the effectiveness of our STT model, we conduct extensive experiments on the large-scale Waymo Open Dataset (WOD) [12]. Our model achieves competitive performance with 58.2 MOTA and state-of-the-art results in our extended S-MOTA and MOTP_S metrics. We conduct comprehensive ablation studies for STT, which allows us to better understand its performance.

The contributions of this work are summarized as follows:

1) We propose a 3D MOT tracker which tracks objects and estimates their motion states in a single trainable model.
2) We extend the existing evaluation metrics to S-MOTA and MOTP_S to evaluate tracking performance that explicitly considers the quality of the state estimation.
3) Our proposed model achieves improved performance over strong baselines with standard metrics and state-of-the-art results with the newly extended metrics on the Waymo Open Dataset.

II. Related Work

2D Multi-Object Tracking [14], [13], [15] aims to track objects in crowd scenes [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [10], [31], and the dominant methods follow a tracking-by-detection paradigm [32], [33], [34], [35], [36]. 2D MOT approaches rarely estimate the motion state of objects since it is challenging to perform 3D state estimation from 2D data and the

*Equal Contributions.
†Corresponding author.

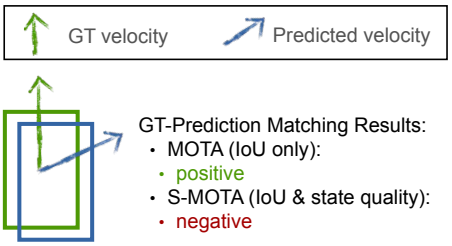

Fig. 1: **Illustration of S-MOTA metric.** MOTA [13] only considers IoUs in label-prediction matching, and does not reveal state errors (e.g., velocity error shown in the figure). This limitation is addressed by S-MOTA via an additional thresholding step to assess the accuracy of predicted state.

motion states estimated from a perspective view are often not informative for downstream modules in autonomous driving.

3D Multi-Object Tracking is a popular problem in autonomous driving [37], [38], [39], [40], [41], [42]. Compared to 2D tracking, this problem space is less explored. Prior works in 3D tracking have primarily relied on Kalman Filters [2], [43], [3], as seen in numerous state-of-the-art methods on the Waymo Open Dataset. Other works explore learning-based solutions [44], [45]. Unlike these works that either ignore or separate the state estimation task from association task, our STT model can learn these two tasks together.

State Estimation is a problem domain where the goal is to predict the state of an object including its dynamic attributes (e.g., speed, acceleration) and semantic attributes (e.g., object type, appearance). Existing tracking solutions primarily focus on the dynamic attributes for state estimation, as these are highly correlated with tracking performance. Common practices include predicting them using a motion filter that smooths estimations over time [2], [3] and including them as an output in an object detection model [10], [46]. Compared to these methods, our approach has a dedicated machine learning module that can encode the temporal features from a detection model and predict accurate object state.

In **Multi-Object Tracking Evaluation**, the most commonly used metric [12], [47] is the MOTA [11], [13]. It captures both the detection box quality and tracking performance. However, it only explicitly evaluates the position result and does not directly evaluate other object states. MOTP [11] also only considers the localization error of the positive matches in MOTA. The stateful metrics we propose consider a wider range of state estimates jointly with association, and thus better reflect the overall tracking quality. While MOTA can be combined with other standalone metrics for assessing the state estimation [47], S-MOTA uses a single unified metric that highlights the estimation quality across all states and MOTP_S offers fine-grained evaluation on any generic state. Other tracking metrics like IDF1 [48] and HOTA [49] put more emphasis on data association quality and are complementary to our proposed metrics.

III. Methodology

In this section, we will first formalize the tracking problem and then describe the architecture of our STT model. We will cover its training and inference process and discuss our new tracking metrics that cover a wide spectrum of the object states. An overview of STT is shown in Figure 2.

A. The Tracking Problem

The goal of the tracking problem discussed in this paper is to maintain a set of tracks $\vec{\tau}_1^t, \vec{\tau}_2^t, \ldots, \vec{\tau}_{N^t}^t$ for the N^t objects in a scene at time t, where each tracklet $\vec{\tau}_n^t = [S_n^{t_k}, \ldots, S_n^t]$ consists of a list of state vectors S_n^t from t_k to the current time t. The state vector S_n^t is defined as $S_n^t = [\{s\}|_{s\in\mathcal{S}}]$, where $s \in \mathbb{R}^{d_s}$ is a d_s-dimensional vector representing state type s, $\mathcal{S}$ is the set of state types being considered, and $[\cdot]$ is the concatenation operation. In this work, we model states $S_n^t = [\mathbf{x}, \mathbf{v}, \mathbf{a}] \in \mathbb{R}^6$, the concatenation of position $\mathbf{x} \in \mathbb{R}^2$, velocity $\mathbf{v} \in \mathbb{R}^2$, and acceleration $\mathbf{a} \in \mathbb{R}^2$. Each state type is defined over the XY plane, as objects on the road rarely move alone the Z direction. Nevertheless, the problem can be easily generalized to the Z direction.

Assume that the tracks are given as $\vec{\tau}_1^{t-1}, \vec{\tau}_2^{t-1}, \ldots, \vec{\tau}_{N^{t-1}}^{t-1}$ at time $t-1$, and a new set of 3D detection are given at time t as $p_1, p_2, \ldots, p_{N^t}$, where $p_i = (b_i, o_i, f_i)$ with bounding box b_i, appearance features o_i, and confidence score $f_i \in [0, 1]$. The box $b_i \in \mathbb{R}^7$ contains the position (x, y, z), sizes (width, length, height), and heading. The tracking problem is then defined as computing the tracks $\vec{\tau}_1^t, \ldots, \vec{\tau}_{N^t}^t$ and their states $S_1^t, \ldots, S_{N^t}^t$ at time t. Note that N^t can be different from N^{t-1}, as new tracks can be created and the existing tracks can be deleted due to the lack of observations.

B. Modeling

1) Detection Encoder and Temporal Fusion: As a tracking model, STT can interact with arbitrary 3D detection models. To ensure that STT can learn a descriptive embedding that captures the geomtry, appearance, and motion features of the detection, we design a Detection Encoder (DE) to encode the detection outputs:

$$\text{emb}(\text{det}_i) = \text{DE}(g_i, a_i, m_i, \theta_{\text{DE}}) \tag{1}$$

Let det_i denote the ith detection, and let g_i, a_i, m_i be the corresponding geometry, appearance, and motion features for this detection respectively. θ_{DE} are the learned parameters of DE. DE is implemented as a multilayer perceptron (MLP) in our model.

After the DE comes a Temporal Fusion (TF) model that combines these detection embeddings over time to create a temporal embedding that describes each track's history. To better model the historical context of a track $\vec{\tau}_j^{t-1}$, we apply a self-attention model to the associated detection embeddings and obtain the track query $Q_{\vec{\tau}_j^{t-1}}$ at time $t-1$:

$$Q_{\vec{\tau}_j^{t-1}} = \text{TF}(\{\text{emb}(\text{det}_i)|i = 1, ..., t-1\}, \theta_{\text{TF}}) \tag{2}$$

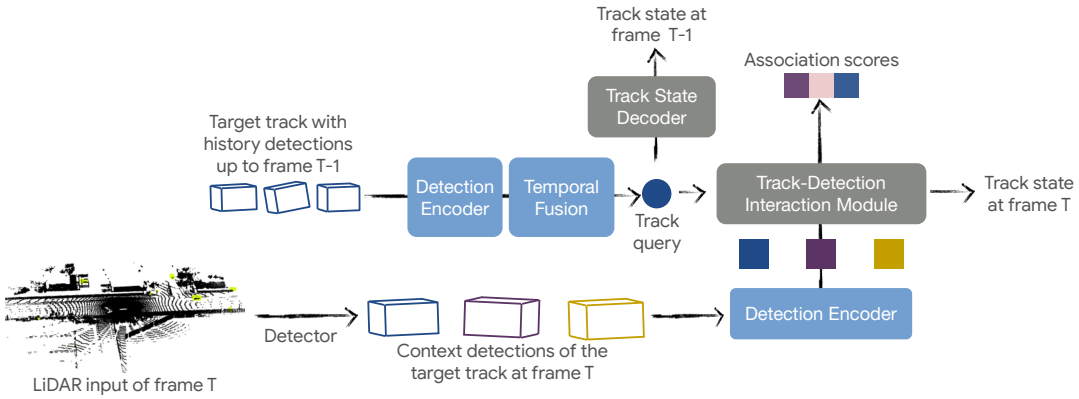

Fig. 2: **Overview of STT.** We first use the Detection Encoder to encode all of the 3D detections and extract temporal features for each track. The temporal features are fed into the Track-Detection Interaction module to aggregate information from surrounding detections and produce association scores and predicted states for each track. The Track State Decoder also takes the temporal features to produce track states in the previous frame $t-1$. All modules are jointly optimized.

where $\text{det}_i \in \mathbf{Det}(\vec{\tau}_j^{t-1})$, and $\mathbf{Det}(\vec{\tau}_j^{t-1})$ is the set of associated detections for track $\vec{\tau}_j^{t-1}$ until time $t-1$. After self-attention, TF aggregates the embeddings $\mathbb{R}^{1\times T\times D_q}$ across time and outputs the self-attended embedding in $\mathbb{R}^{1\times D_q}$ at time $t-1$. T is the track length, D_q is the feature size, and θ_{TF} are the learned parameters.

2) Track State Decoder: For a track $\vec{\tau}_j^{t-1}$ at time t, the track query $Q_{\vec{\tau}_j^{t-1}}$ encodes its history up to time $t-1$. Therefore, we can directly predict the state $\mathbf{S}_{t-1}$ for every track with a light-weight Track State Decoder (TSD) module:

$$\mathbf{S}_{t-1} = G(\mathbf{Q}_{t-1}, \theta_{\mathbf{g}}) \tag{3}$$

where $\mathbf{Q}_{t-1}$ is the list of all the track queries. G is a MLP and $\theta_{\mathbf{g}}$ are its learned parameters. TSD helps us supervise the track embedding, but it is also useful as a stand-alone state estimator for a given track embedding at any given timestamp. We will elaborate more on how this decoder is used during a typical tracker update loop in Section III-D.

3) Track-Detection Interaction Module: The Track-Detection Interaction (TDI) module calculates the relationship between tracks and their surrounding context detections at time t. For each track $\vec{\tau}_j^{t-1}$ from time $t-1$, we select k context detections $\mathbf{K}_n$ from all the detections $\mathbf{M}$ at time t in a small area around the track:

$$\mathbf{K}_n = \{b_i | D(\text{pred}(\vec{\tau}_j^{t-1}), b_i) < d, b_i \in p_i, p_i \in \mathbf{M}\} \tag{4}$$

where D computes the distance between detection b_i and the track's state estimation $\text{pred}(\vec{\tau}_j^{t-1})$ at time t. During training, we directly use the ground truth state at time t to represent $\text{pred}(\vec{\tau}_j^{t-1})$. During inference, we extrapolate the estimated track state at time $t-1$ to time t to search for the context detections effectively before running the model. In practice, we set threshold d to be small enough for efficiency, but large enough to ensure that all the detections of true positive association for track $\vec{\tau}_j^{t-1}$ are included in the context set $\mathbf{K}_n$.

We use the same Detection Encoder to create the detection embeddings $\mathbf{C_i}$ in $\mathbf{K}_n$. The TDI module then takes the list of queries $\mathbf{Q}_t$ and $\mathbf{C_i}$ as input to predict the association scores for all the tracks and detections:

$$\mathbf{AS} = \text{TDI}(\mathbf{Q}_t, \mathbf{C_i}, \theta_{\text{TDI}}) \tag{5}$$

where θ_{TDI} are learned parameters. $\mathbf{AS} = \{AS\}$, where $AS \in \mathbb{R}^{1\times k}$ are the association scores between a track query $Q_{\vec{\tau}_j^{t-1}}$ and the k context detections. TDI is a transformer-based model [50] with an added MLP to predict the track state at time t after cross-attending to the context detections.

C. Training

Our model is jointly trained using a data association loss L_d^t and state estimation losses L_s^t, L_s^{t-1}:

$$L_{\text{total}} = \gamma L_d^t + \lambda L_s^t + \alpha L_s^{t-1} \tag{6}$$

where γ, λ, and α are the weight of each loss term. We optimize the per-track query with per box association loss. Let AS_i be the association score between the track query $Q_{\vec{\tau}_j^{t-1}}$ and one of its context detections det_i. And let y be the ground-truth association with 0 as "not associated" or 1 as "association" Then the loss of this pair is:

$$L(Q_{\vec{\tau}_j^{t-1}}, \text{det}_i) = -(y\log(AS_i) + (1-y)\log(1-AS_i)) \tag{7}$$

For each track query, the total association loss is computed against all of its context detections as:

$$L_d^t = \sum_{i=1}^{k} L(Q_{\vec{\tau}_j^{t-1}}, \text{det}_i) \tag{8}$$

where k is the number of context detections.

The state estimation losses are the L1 loss between the predicted states and the ground truth states for each track at time t (via the output of TDI module) and $t-1$ (via the output of the TSD module):

$$L_s^t = \left|\mathbf{S}_j^t - \mathbf{S}_j^{*t}\right|, L_s^{t-1} = \left|\mathbf{S}_j^{t-1} - \mathbf{S}_j^{*t-1}\right| \tag{9}$$

where S_j^{*t} and S_j^{*t-1} is the ground truth state for the track $\vec{\tau}_j^t$ and $\vec{\tau}_j^{t-1}$ respectively.

D. Online Tracker Inference

During tracking inference, we apply STT over the laser stream frame by frame. For each frame at time t, a 3D object detection model is first applied over the laser spin

to get all N detection boxes. For each detection box, its geometry features, appearance features, and confidence score are collected as p_n^t, while p^t is the list of all the detections' feature vectors. For all tracks produced from the previous frame at time $t-1$, we cache their learned track query $\mathbf{Q}_{t-1}$. Then, the TDI module is applied over the queries $\mathbf{Q}_{t-1}$ and all detection embeddings $\mathbf{emb}(p^t)$ to produce the association likelihood 2D matrix $\mathbf{AS}$ between all the tracks and boxes.

The Hungarian matching algorithm [51] is then applied over $\mathbf{AS}$ to produce the assignment result. If the association score is lower than a pre-defined threshold, a new track will be created. Otherwise, the detection will be assigned to an existing track query and appended to its history. For the first frame of a track, all the detected boxes are treated as new tracks and their initial states (e.g. velocity and acceleration) will be set to 0. For all the subsequent frames, we use TSD to predict state for the track at time t as we find that it is slightly better than the output of TDI.

E. Stateful Evaluation Metrics

1) S-MOTA: MOTA [11] is one of the most commonly used metrics for multiple object tracking. Computing MOTA involves a matching step similar to the evaluation of object detection. A given prediction-label pair (p, g) is only considered for matching if their IoU is larger than a given threshold:

$$C(p,g) = \begin{cases} 1 - U(p,g), & \text{if } U(p,g) > t_u. \\ +\infty, & \text{otherwise.} \end{cases} \quad (10)$$

$U(\cdot)$ is the IoU function and t_u is a class-specific threshold. $C(\cdot)$ denotes the cost function of the matching algorithm. Consequently, MOTA primarily evaluates the quality of the detections as well as the predicted associations. The only component of the states defined in Section III-A evaluated here is the location (i.e., the detection box center), and the prediction accuracies of other states are only indirectly evaluated through the improvements they may bring to association.

To better evaluate data association and state estimation, we extend the MOTA to *Stateful Multiple Object Tracking Accuracy* (S-MOTA). This is computed using the same procedure as standard MOTA, but with additional requirements in the state estimation for a given prediction-label pair to be matched. Accurate state estimation such as a vehicle's velocity is critical for autonomous driving. In S-MOTA, the state estimation error of each pair must be below a class- and state-dependent threshold to allow matching:

$$C(p,g) = \begin{cases} 1 - U(p,g), & \text{if } U(p,g) > t_u \text{ and} \\ & \cap_{s \in \mathcal{S}} \|p_s - g_s\| < t_{u,s} \\ +\infty, & \text{otherwise.} \end{cases} \quad (11)$$

Let p_s and g_s denote predicted/ground-truth state vectors of type s. $\mathcal{S}$ is the set of states considered for the evaluation, and $t_{u,s}$ is the threshold for state type s and class u. Hence, maximizing S-MOTA requires track predictions to both have proper associations across time as well as reasonably close state predictions. For this work, $\mathcal{S}$ consists of velocity and acceleration. In principle, however, any combination of state types from a tracker can be used to derive a S-MOTA metric.

2) MOTP$_S$: The extended S-MOTA metric is designed to provide a comprehensive evaluation of tracking performance, including state estimation. As a complement, we extend the MOTP to Multiple Object Tracking Precision for General States (MOTP$_S$) to provide more fine-grained evaluation on the state estimation accuracy. Given the set $\mathcal{M}$ containing pairs of predictions p and label g which are matched during MOTA computation, MOTP$_S$ computes the average L2 error for each state type to measure the magnitude of the state error, for each state type $s \in \mathcal{S}^*$:

$$\text{MOTP}_s(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{(p,g) \in \mathcal{M}} \|p_s - g_s\| \quad (12)$$

We can further measure the count of objects with large state estimation errors, i.e.

$$|\text{MOTP}_s(\mathcal{M})| = |\{(p,g) \in \mathcal{M} \mid \|p_s - g_s\| > \alpha_s\}| \quad (13)$$

where α_s is a threshold for state s. Note that MOTP$_S$ is consistent with the definition of MOTP. In fact, the latter is a specific version of the former in the localization state. Rather than defining a single metric that aggregates across states, we use separate MOTP$_S$ metrics for each state type to highlight the performance of each type of state individually.

The evaluation dataset has a disproportionate amount of stationary objects. To ensure that the metrics properly evaluate performance on objects with different types of motion, we report the L2 state error in three different speed breakdowns: static, slow moving objects, and fast moving objects. We also count the number of predictions with L2 error larger than the threshold α_s to focus on challenging cases where the predictions are off significantly.

IV. Experiments

Datasets. We evaluate our STT model on the Waymo Open Dataset [12], which contains 798 sequences for training, 202 sequences for validation, and 150 sequences for testing. Each sequence lasts 20 seconds at 10 Hz. Following other popular methods, we evaluate our method on vehicles and pedestrians for the LEVEL 2 difficulty setting [12], which is more diffcult than LEVEL 1 because it includes objects with fewer than five laser points in their boxes. LEVEL 2 also includes all the objects in LEVEL 1.

Training details. Our model is jointly trained on 16 TPUs with a batch size of 512. The AdamW [54] optimizer is used with 0.03 weight decay. The initial learning rate is 0.0001 with linear learning rate decay of 0.5. The model is trained for $125,000$ steps, including $1,000$ warm-up steps. We set association loss weight $\gamma = 10$ and we have different loss weights for different states: 1 for both position and velocity and 10 for acceleration. Unless explicitly specified, we set the maximum track length $T = 10$ for encoding track history and select a maximum of 20 context detections for training the model. We use SWFormer [53] as our detection backbone.

A. Overall Results

To demonstrate the effectiveness of our STT model, we compare it with published state-of-the-art methods on the

TABLE I: Comparison with state-of-the-art tracking methods on the validation set of Waymo Open Dataset.

Method	Vehicle					Pedestrian				
	S-MOTA↑	MOTA↑	FP↓	Miss↓	Missmatch↓	S-MOTA↑	MOTA↑	FP↓	Miss↓	Missmatch↓
CenterPoint [8]	-	55.1	10.8	33.9	0.26	-	54.9	10.0	34.0	1.13
SimpleTrack [1]	-	56.1	10.4	33.4	0.08	-	57.8	10.9	30.9	0.42
CenterPoint++ [8]	-	56.1	**10.2**	33.5	0.25	-	57.4	11.1	30.6	0.94
Immortal Tracker [3]	-	56.4	**10.2**	33.4	**0.01**	-	58.2	11.3	30.5	**0.26**
Kalman Filter (Ours)	34.6	56.5	10.6	32.8	0.1	41.8	59.7	10.1	**29.6**	0.5
STT (Ours)	**48.0**	58.2	10.4	31.3	0.1	**55.2**	59.9	10.2	**29.6**	0.3
TrajectoryFormer [52]	-	**59.7**	11.7	**28.4**	0.19	-	**61.0**	**8.8**	29.8	0.37

TABLE II: Comparisons for MOTP_S on the validation set of Waymo Open Dataset.

Method	Class	$\text{MOTP}_\text{velocity}$↓				$\lvert\text{MOTP}_\text{velocity}\rvert$↓	$\text{MOTP}_\text{acceleration}$↓				$\lvert\text{MOTP}_\text{acceleration}\rvert$↓
		Static	Slow	Fast	All		Static	Slow	Fast	All	
SWFormer[53]+SH	Vehicle	**0.016**	0.258	0.372	0.098	3063	**0.013**	0.864	0.758	0.179	11089
Kalman Filter	Vehicle	0.117	0.271	0.260	0.176	1890	0.217	0.683	0.665	0.418	25050
STT	Vehicle	0.049	**0.214**	**0.235**	**0.095**	**794**	0.026	**0.425**	**0.412**	**0.116**	**1528**
SWFormer[53]+SH	Pedestrian	**0.061**	0.179	0.307	0.162	147	**0.066**	**0.155**	0.340	**0.135**	121
Kalman Filter	Pedestrian	0.116	0.15	**0.183**	0.149	**25**	0.212	0.345	0.422	0.336	6930
STT	Pedestrian	0.066	**0.112**	0.205	**0.100**	39	0.082	**0.155**	**0.324**	0.141	**27**

Waymo Open Dataset. The majority of the 3D MOT algorithms adopt the tracking-by-detection paradigm, and each of them uses different detection backbones for their tracking algorithms [1], [3], [8], [52], [55], [56]. As STT is a stateful tracker that can be used with arbitrary detection models, we need to compare it with a tracking method that uses the same detection model as STT. Following [12], [2], [1], we develop a Kalman Filter baseline that uses the same detection backbone as STT.

We first compare our model with these state-of-the-art methods as well as our KF baseline on the official 3D tracking metrics of the Waymo Open Dataset. These metrics includes MOTA, MOTP, False Positives (FP), False Negatives (FN), and mismatches (Identity Switches). The results are shown in Table I. Our KF baseline, which uses a strong detection backbone [53], already achieves competitive performance compared with other existing methods. STT achieves a MOTA score that is +1.7 higher than our KF baseline on the vehicle type and on-par results on other metrics, demonstrating the benefit of including state estimation into the learning process of our tracking model. Note that the miss rate of the KF and STT models are slightly different due to the different cut-off scores used by the two methods. The strong performance of the KF baseline also indicates that these official metrics heavily rely on the quality of the detections. A simple tracker can achieve better performance than other highly-tuned approaches by using a stronger object detector (e.g. our KF baseline vs. CenterPoint [8]).

To demonstrate STT's advantage on state estimation over the KF baseline, we further compare them using the stateful metric S-MOTA, as shown in Table I. This metric requires prediction/ground-truth matches to have sufficiently high predicted velocity and acceleration quality. The velocity and acceleration thresholds are set to 1.0 m/s and 1.0 m/s^2 for vehicles and 0.5 m/s and 0.5 m/s^2 for pedestrians. The S-MOTA score of STT is 13.4 higher than the KF baseline for both vehicles and pedestrians. This shows that while STT performance is close to the KF baseline on the data association metrics, it actually outperforms the KF model significantly on state estimation. This result also indicates that the S-MOTA metric is useful to distinguish between methods having similar association quality in MOTA results.

To evaluate inference time, we compile the STT model with XLA [57] and run inference on the same scenario as reported in [53]. We use a Nvidia PG189 GPU which shares the same hardware architecture as Nvidia T4 GPU but with less memory to meet the power constraints of autonomous vehicles. The inference time for STT alone is 2.9 ms. Combined with the fastest version of SWFormer as reported in their paper, we can achieve real-time performance for the end-to-end tracking.

We also compare our method to TrajectoryFormer [52], which is the current state-of-the-art 3D MOT method on the WOD. We report their CenterPoint [8] configuration. It has higher MOTA score than STT due to improved FN (vehicle) and FP (pedestrian) achieved by taking the trajectory hypothesis from track history as model input. We highlight it in a separate row for that a direct comparison with ours is unfair, as TrajectoryFormer uses extra detection boxes. This improvement is orthogonal to our approach. STT still performs better in other two sub-metrics of MOTA. Moreover, TrajectoryFormer does not predict or evaluate on full state estimates, nor does it run in real-time.

B. $MOTP_S$ Results

To further understand the improvements of STT on state estimation, we report the MOTP_S metric results for STT and two baselines: i) Kalman Filter, and ii) SWFormer+State Head (SH), for which we add a state head to the original SWFormer detector to predict velocity and acceleration for each detected box. The three methods all use the same detection model, which removes the impact of detection quality and allows us to concentrate on the performance of state estimation itself.

TABLE III: Ablation studies with the proposed STT model on the validation set of Waymo Open Dataset.

Tracker	Detector	Track Length	Joint Optimization w/ State Estimation	Vehicle		Pedestrian	
				MOTA↑	S-MOTA↑	MOTA↑	S-MOTA↑
Joint Optimization of Association and State Estimation							
STT	SWFormer[53]	10	N	56.4	30.9	55.9	13.1
STT	SWFormer[53]	10	Y	**58.2**	**48.0**	**59.9**	**55.2**
Long-term Temporal Modeling							
STT	SWFormer[53]	3	Y	58.1	37.7	59.9	52.9
STT	SWFormer[53]	5	Y	**58.2**	40.4	**60.0**	54.1
STT	SWFormer[53]	10	Y	**58.2**	48.0	59.9	55.2
STT	SWFormer[53]	20	Y	**58.2**	**49.2**	**60.0**	**55.4**
Tracking Performance with Different Detectors							
Kalman Filter	UPillar[58]	N/A	N/A	55.7	34.0	57.1	39.8
STT	UPillar[58]	10	Y	**57.1**	**46.3**	**57.4**	**52.1**
Kalman Filter	SWFormer[53]	N/A	N/A	56.5	34.6	59.7	41.8
STT	SWFormer[53]	10	Y	**58.2**	**48.0**	**59.9**	**55.2**

As shown in Table II, our STT model achieves the best overall state estimation results compared with the two baselines. In terms of velocity estimation, SWFormer+SH is surprisingly the best state estimator for static objects, but STT performs better for moving objects. SWFormer+SH also produces the highest value of $|\text{MOTP}_{\text{velocity}}|$ whereas STT has the lowest, indicating that the superior performance of SWFormer+SH on static objects may due to overfitting. On the other hand, the KF baseline struggles to predict accurate states for static objects but can achieve decent performance on moving ones. This may be because small jittering from static objects can create large noise in KF state estimation while learning-based methods are more robust to this.

The relative gain of STT is more prominent for the acceleration estimation. STT achieves the best acceleration for moving objects and comparable performance with the SWFormer+SH on static objects. STT has the lowest variance compared to the two baselines as reflected by $|\text{MOTP}_{\text{acceleration}}|$. Acceleration, as a second order statistic, is more challenging to estimate. Therefore, models must be able to robustly handle small noise and effectively reason about long-term motion. STT possesses both of these qualities, and its robustness and consistency are reflected in the metric results.

C. Ablation Studies

Joint optimization with state estimation is important. One of the key innovations of STT is its unified learning framework which jointly optimizes for both data association and state estimation tasks. To validate the claim that the joint optimization with state estimation can improve the data association performance, we create a STT baseline that is only trained with the data association loss. The results are reported in the first two rows of Table III. With the joint optimization of state estimation and data association, STT achieves MOTA improvement of +1.8 and +4 for the vehicle and pedestrian classes, respectively. Similarly, S-MOTA improvements of +17.1 and +42.1 are observed for these two classes from STT. These results suggest that data association and state estimation are highly complementary tasks that should be jointly optimized.

Longer-term temporal modeling improves data association quality with more accurate state estimation. To verify the impact of the temporal features on tracking performance, we evaluate STT with different track history lengths. The results, shown in rows 3 to 6 of Table III, demonstrate that longer track history can lead to improved tracking performance. The MOTA score increases as the track history length increases to 5, after which it saturates. However, the S-MOTA score continues to increase by a large margin, even for track history lengths of 20. This suggests that longer-term temporal modeling is critical for data association and state estimation tasks.

Improvements from STT are robust with different detectors. As our KF baseline experiment shows, the performance of a tracking system can be significantly affected by the quality of the upstream object detector. To understand the sensitivity of STT to different detectors, we compared STT and KF using two different detectors: SWFormer [53] and UPillar [58]. The results in Table III show that our STT model outperforms the Kalman Filter on all metrics with different object detectors, which indicates that our model is robust to the choice of detector.

V. Conclusion

In this paper, we propose STT, a transformer-based model that jointly conducts data association and state estimation in one model. We emphasize the importance of this joint estimation task for autonomous driving, which requires consistent tracking and accurate state estimation for objects in 3D real-world-space. To address the limitations of existing evaluation methods, we extend MOTA metrics to S-MOTA, which enforces the consideration of state estimation quality when evaluating association quality, and MOTP to MOTP_s, which captures broader motion state of objects. Evaluation has shown that STT achieves the competitive results on the Waymo Open Dataset with strong performance in state estimation. We hope that our proposed solutions and extended metrics will facilitate future work in this area.

Acknowledgements. We would like to thank Luming Tang, Andy Tsai, Shirley Chung, Yang Wang, Chao Jia, Zhaoqi Leng, Yu Zhu, Nichola Abdo, Henrik Kretzschmar, Marshall Tappen, and Dragomir Anguelov for their invaluable contributions to this paper.

REFERENCES

[1] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," *arXiv:2111.09621*, 2021.

[2] X. Weng and K. Kitani, "A baseline for 3D multi-object tracking," *arXiv:1907.03961*, 2019.

[3] Q. Wang, Y. Chen, Z. Pang, N. Wang, and Z. Zhang, "Immortal tracker: Tracklet never dies," *arXiv:2111.13672*, 2021.

[4] S. Lee and J. McBride, "Extended object tracking via positive and negative information fusion," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1812–1823, 2019.

[5] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.

[6] E. Cortina, D. Otero, and C. D'Attellis, "Maneuvering target tracking using extended kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, no. 1, pp. 155–158, 1991.

[7] S. Lee, J. Lee, and I. Hwang, "Maneuvering spacecraft tracking via state-dependent adaptive estimation," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 9, pp. 2034–2043, 2016.

[8] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *CVPR*, 2021.

[9] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *ICCV*, 2015.

[10] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *ECCV*, 2020.

[11] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, 2006.

[12] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *CVPR*, 2020.

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

[14] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942*, 2015.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[16] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Transmot: Spatial-temporal graph transformer for multiple object tracking," *arXiv:2104.00194*, 2021.

[17] J. Peng, T. Wang, W. Lin, J. Wang, J. See, S. Wen, and E. Ding, "Tpm: Multiple object tracking with tracklet-plane matching," *Pattern Recognition*, 2020.

[18] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *ECCV*, 2020.

[19] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *CVPR*, 2021.

[20] Q. Yu, G. Medioni, and I. Cohen, "Multiple target tracking using spatio-temporal markov chain monte carlo data association," in *CVPR*, 2007.

[21] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," in *ECCV*, 2020.

[22] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding, "Learning a proposal classifier for multiple object tracking," *CVPR*, 2021.

[23] F. Zeng, B. Dong, T. Wang, C. Chen, X. Zhang, and Y. Wei, "End-to-end multiple-object tracking with transformer," *ECCV*, 2022.

[24] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense queries for multiple-object tracking," *arXiv:2103.15145*, 2021.

[25] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *CVPR*, 2021.

[26] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple-object tracking with transformer," *arXiv:2012.15460*, 2020.

[27] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," *CVPR*, 2021.

[28] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, "Global tracking transformers," in *CVPR*, 2022.

[29] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *ICCV*, 2019.

[30] H. Xiang, R. Xu, and J. Ma, "Hm-vit: Hetero-modal vehicle-to-vehicle cooperative perception with vision transformer," *arXiv:2304.10628*, 2023.

[31] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *CVPR*, 2022.

[32] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016.

[33] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *ICIP*, 2017.

[34] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *ICCV*, 2019.

[35] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *CVPR*, 2017.

[36] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *arXiv:2004.01888*, 2020.

[37] Q. Zhou, S. Agostinho, A. Osep, and L. Leal-Taixe, "Is geometry enough for matching in visual localization?" *ECCV*, 2022.

[38] A. Kim, G. Brasó, A. Ošep, and L. Leal-Taixé, "Polarmot: How far can geometric relations take us in 3d multi-object tracking?" in *ECCV*, 2022.

[39] M. Gladkova, N. Korobov, N. Demmel, A. Ošep, L. Leal-Taixé, and D. Cremers, "Directtracker: 3d multi-object tracking using direct image alignment and photometric bundle adjustment," *IROS*, 2022.

[40] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *ICRA*, 2021.

[41] W.-C. Hung, H. Kretzschmar, T.-Y. Lin, Y. Chai, R. Yu, M.-H. Yang, and D. Anguelov, "Soda: Multi-object tracking with soft data association," *arXiv:2008.07725*, 2020.

[42] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *ICRA*, 2022.

[43] H. Kuang Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3d multi-object tracking for autonomous driving," *arXiv 2001.05673*, 2020.

[44] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[45] H.-N. Hu, Y.-H. Yang, T. Fischer, T. Darrell, F. Yu, and M. Sun, "Monocular quasi-dense 3d object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1992–2008, 2022.

[46] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking," *arXiv:2303.11301*, 2023.

[47] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[48] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The clear 2006 evaluation," in *International evaluation workshop on classification of events, activities and relationships*. Springer, 2006.

[49] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *IJCV*, 2021.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[51] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[52] X. Chen, S. Shi, C. Zhang, B. Zhu, Q. Wang, K. C. Cheung, S. See, and H. Li, "Trajectoryformer: 3d object tracking transformer with predictive trajectory hypotheses," in *ICCV*, 2023.

[53] P. Sun, M. Tan, W. Wang, C. Liu, F. Xia, Z. Leng, and D. Anguelov, "Swformer: Sparse window transformer for 3d object detection in point clouds," in *ECCV*, 2022.

[54] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv:1711.05101*, 2017.

[55] P. Li and J. Jin, "Time3d: End-to-end joint monocular 3d object detection and tracking for autonomous driving," in *CVPR*, 2022.

[56] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *IROS*, 2020.

[57] A. Sabne, "Xla: Compiling machine learning for peak performance," 2020.

[58] Z. Leng, G. Li, C. Liu, E. D. Cubuk, P. Sun, T. He, D. Anguelov, and M. Tan, "Lidaraugment: Searching for scalable 3d lidar data augmentations," *arXiv:2210.13488*, 2022.