





OneTrack: Demystifying the Conflict Between Detection and Tracking in End-to-End 3D Trackers

Qitai Wang^{1,2}, Jiawei He², Yuntao Chen³, and Zhaoxiang Zhang^{1,2,3,4}

¹ School of Future Technology, University of Chinese Academy of Sciences (UCAS)

² NLPR, MAIS, Institute of Automation, Chinese Academy of Sciences (CASIA)
{wangqitai2020,zhaoxiang.zhang}@ia.ac.cn, jwhe2024@gmail.com

³ Center for Artificial Intelligence and Robotics, HKISI, CAS
chenyuntao08@gmail.com

⁴ Shanghai Artificial Intelligence Laboratory

Abstract. Existing end-to-end trackers for vision-based 3D perception suffer from performance degradation due to the conflict between detection and tracking tasks. In this work, we get to the bottom of this conflict, which was vaguely attributed to incompatible task-specific object features previously. We find the conflict between the two tasks lies in their partially conflicted classification gradients, which stems from their subtle difference in positive sample assignments. Based on this observation, we propose to coordinate those conflicted gradients from object queries with contradicted polarity in the two tasks. We also dynamically split all object queries into four groups based on their polarity in the two tasks. Attention between query sets with conflicted positive sample assignments is masked. The tracking classification loss is modified to suppress inaccurate predictions. To this end, we propose OneTrack, the first one-stage joint detection and tracking model that bridges the gap between detection and tracking under a unified object feature representation. On the nuScenes camera-based object tracking benchmark, OneTrack outperforms previous works by 6.9% AMOTA on the validation set and by 3.1% AMOTA on the test set.

Keywords: 3D Tracking · Camera-based Detection and Tracking · End-to-end Tracking · Gradient Coordination

1 Introduction

Reasoning about the location and trajectory of surrounding objects is a fundamental task for autonomous driving and robotic navigation systems. Due to the robustness and low cost of cameras, vision-based 3D detection and tracking have received widespread attention from the research community [14, 22, 23, 26, 36, 37]. Recently, several approaches [21, 31, 48] have been proposed to improve 3D multi-object tracking (MOT) by jointly optimizing the detection and tracking pipeline. Based on transformer architecture, those end-to-end 3D trackers use

Table 1: Comparison with previous works trying to solve the conflict between detection and tracking. *: DQTrack relies on frozen pre-trained detectors when compared with state-of-the-art trackers.

Method	setting	e2e	one-stage	detection supervision	train from scratch
MOTRv2 [50]	2D	×	×	×	×
MOTRv3 [45]	2D	✓	✓	×	✓
DQTrack [21]	3D	✓	×	✓	✓*
OneTrack	3D	✓	✓	✓	✓

track queries to track objects, being free from the dependence on offline detectors and post-processing operations.

However, most end-to-end 3D trackers suffer from the optimization conflict between detection and tracking, a problem also widely recognized in 2D MOT studies [45, 50]. This conflict results in the inferior precision and recall of end-to-end trackers compared to their baseline detectors. Previous works roughly attribute the conflict between two tasks to differences in required features for detection and tracking and attempted to alleviate this problem. Some of them train their model as detectors before the tracking training process [31, 33] or take an off-the-shelf detector as guidance [50]. Some separate the detection and association stages as well as their object feature representation to avoid conflict, such as DQTrack [21]. Although much progress has been made, training a one-stage end-to-end tracker from scratch remains challenging. It is also challenging to train trackers jointly under both detection and tracking supervision. We summarize recent works in Table 1. In this work, we aim to accomplish the true joint training of detection and tracking from scratch with a one-stage model and unified feature representation for both tasks.

We conduct a pilot study to investigate how and where the two tasks conflict during model optimization. As illustrated in Table 2, only jointly supervising the detector with an extra classification loss computed under the ground truth assignments for tracking will lead to severe performance degradation of detectors. Therefore, we conclude that the conflict between the two tasks originates from their conflicting classification gradients.

In this paper, we precisely identify and coordinate the conflicted classification gradient between the two tasks, which originates from their partially conflicted positive sample definition. We propose OneTrack, the first model capable of achieving the true joint training of detection and tracking in a unified, one-stage model. OneTrack follows the “tracking by track queries” paradigm. During the training process, we assign ground truth annotations to object queries twice based on positive sample definitions for detection or tracking. We dynamically categorize all object queries into four groups based on their polarity in two tasks. We have two lightweight classification heads in OneTrack which are respectively responsible for classifying detection or tracking positive samples. Queries that are

Table 2: Pilot study on supervising 3D detector under additional ground truth assignments for tracking. The stronger color means more degradation in performance. Experiments indicate the conflict between detection and tracking tasks lies in their classification supervision.

Detection Classification	Detection Regression	Tracking Classification	Tracking Regression	AMOTA↑	IDS↓	NDS↑	mAP↑
✓	✓	✗	✗	-	-	0.570	0.479
✗	✗	✓	✓	0.305	121	-	-
✓	✓	✓	✗	0.404	363	0.512	0.432
✓	✓	✗	✓	0.417	8512	0.566	0.473
✓	✓	✓	✓	0.405	255	0.528	0.428

positive samples only for one task will only backpropagate classification gradients in the respective head to reduce conflicted gradients. We dynamically mask all attention between conflicted positive queries in each decoder layer to prevent unexpected query competition. We also adjust the tracking loss to suppress inaccurate predictions.

To summarize, our contributions are as follows:

- We reveal that the conflict between detection and tracking tasks lies in their diverged definition of positive samples and their partially conflicted classification gradients.
- We propose OneTrack, the first model capable of completely addressing the conflict between detection and tracking under a unified object feature representation. OneTrack can be trained as both a detector and a tracker from scratch in a single training stage and can perform tracking and detection in a one-stage fashion. The detection performance of OneTrack is comparable to detectors, while its tracking performance surpasses all previous 3D trackers.
- Our method establishes a new state-of-the-art on nuScenes MOT benchmark, surpassing previous methods by over 3.1% AMOTA.

2 Related Works

2.1 Query Propagation in MOT

Early works in 2D or 3D MOT most follow the “tracking by detection” paradigm [4] and associate detection results across frames through post-processing [1, 2, 8, 9, 12, 13, 16, 19, 25, 35, 40–42, 44, 46, 49]. Based on DETR [6] which introduced object queries to detect objects, MOTR [47] and Trackformer [30] extended the concept of object query to track query by propagating object queries across frames. By tracking objects through track queries, MOTR and Trackformer can perform end-to-end joint 2D detection and tracking without post-processing. However, they seriously suffer from the conflict between detection and tracking. MOTRv2 [50] and MOTRv3 [45] concentrate on this conflict. MOTRv2 proposes

using an off-the-shelf detector to guide the tracker, which is effective but damages the end-to-end fashion of the tracker. MOTRv3 proposes to balance the label assignment between detection queries and track queries to achieve a better trade-off between the detection and tracking performance of the model. MUTR3D [48] and PF-Track [31] applied the “tracking by track queries” paradigm to 3D MOT. MUTR3D [48] performs camera-based tracking in an end-to-end fashion by introducing 3D track queries to model spatial and appearance coherent tracks. PF-Track [31] further refine the tracks and predict the trajectories of objects with cross-object and cross-frame attention between object queries. Recently, DQTrack [21] proposes to separate the detection process and association process of trackers to avoid their conflicts in object feature representation. None of the previous joint detection and tracking approaches can truly bridge the gap between detection and tracking with a one-stage model and a unified object feature representation. Neither can they train the model from scratch and get rid of the reliance on pre-trained detectors while detecting and tracking objects with high precision and recall performance.

2.2 Camera-based 3D MOT

Camera-based 3D MOT has gained remarkable progress recently thanks to the developments in camera-based 3D detection and depth estimation [11, 14, 15, 22, 23, 26, 27, 36–38, 43, 51]. Early methods in camera-based 3D MOT perform tracking in 2D first and then lift the tracks to 3D space [34]. CC-3DT [10] fuses multi-view object features to enhance associating objects across different views. QD-3DT [13] perform 2D association first and enhance the instance association with the depth order and motions of 3D objects. TripletTrack [29] extracts local object feature embeddings and motion descriptors with CNN or LSTM to measure the affinity between objects. MUTR3D [48], PF-Track [31], and DQTrack [21] recently established new state-of-the-art tracking performance by applying the transformer-based tracking architecture in 3D.

3 Method

The model design of OneTrack is conceptually concise. We add an additional lightweight classification branch to a basic transformer-based tracker. Both ground truth assignments for the detection and tracking tasks are employed to supervise the model. We begin by reviewing the design of transformer-based trackers in Sec. 3.1. Then in Sec. 3.2, we propose to identify the conflicted positive sample assignments between detection and tracking through query grouping and prevent the contradicted gradient from back-propagating. In Sec. 3.3, we introduce the dynamic attention mask to further prevent gradient conflict between the two tasks. Additionally, the tracking classification loss is modified to suppress inaccurate boxes predicted by track queries, as described in Sec. 3.4.

3.1 Reviewing the ‘‘Tracking by Track Queries’’ Paradigm

As previous transformer-based trackers [21,30,31,45,47,48,50], OneTrack detect and track 3D objects with object queries. At each timestamp t , given multi-view images \mathbf{I} , the 3D detection task aim to predict a set of 3D bounding boxes $\mathbf{B} = \{\mathbf{b}^1, \mathbf{b}^2, \dots, \mathbf{b}^n\}$ and their confidence scores $\mathbf{S} = \{s^1, s^2, \dots, s^n\}$ to capture 3D objects. Meanwhile, 3D MOT requires a consistent ID for each object across frames. We represent the outputs for the 3D MOT task by $\{\mathbf{B}, \mathbf{S}, \mathbf{ID}\}$, including the predicted boxes, confidence scores, and ID of predictions.

In each frame, OneTrack first receives a set of object queries $\mathbf{Q} = \{\mathbf{q}^1, \mathbf{q}^2, \dots\}$. Following previous works [21,31,47,48], \mathbf{Q} consists of track queries propagated from the previous frame and a fixed number of new detection queries \mathbf{Q}^d . The query propagation process can be formulated as follows:

$$\mathbf{Q}_t = \hat{\mathbf{Q}}_{t-1} \cup \mathbf{Q}^d, \quad (1)$$

where $\hat{\mathbf{Q}}_{t-1}$ represents the propagated queries from the last frame. We denote \mathbf{Q}_t as \mathbf{Q} in the subsequent descriptions for simplicity.

In each frame, given the multi-view image features \mathbf{F} extracted with image encoder, OneTrack utilizes a transformer decoder to extract object features. The decoder comprises six transformer decoder layers. In the i^{th} layer \mathbf{Q} is updated as follows:

$$\mathbf{Q} = \text{DecoderLayer}_i(\mathbf{F}, \mathbf{Q}, \mathbf{M}). \quad (2)$$

Here, \mathbf{M} represents the attention mask for the self-attention process in decoder layers. In end-to-end trackers, attention between all object queries is usually not masked. Then the object queries are fed into the classification and regression heads to generate prediction boxes. Ground truths are then assigned to the predictions for the calculation of classification loss and regression loss. Recent works [21,31] usually preserve historical track queries from a few past frames as tracking memory. These stored queries serve as references to the tracking process in the current frame.

Instead of associating boxes across frames with post-processing techniques, the ‘‘tracking by track queries’’ paradigm tracks objects with track queries propagated across frames. Therefore during the training of trackers, a ground truth instance is assigned to a specific track query in future frames once they are associated in the current frame, regardless of the accuracy of the assigned query’s predictions in future frames. We denote these ground truth-query pairs as ‘‘locked’’ [45] pairs, and the remaining ground truths and queries are denoted as ‘‘free’’ ground truths or queries. In contrast, for detectors, positive samples are simply defined as the most precise prediction in each frame. Hence, a positive sample for detection may be classified as a negative sample in tracking and be suppressed, and vice versa, as illustrated in Fig.1. This originates from ground truth objects being assigned to different object queries in the two tasks. This conflict will lead to conflicted classification gradients between the two tasks, hindering the convergence of the model. We found that those conflicted gradients are primarily responsible for the precision and recall degradation of end-to-end trackers.

from the last frame. Classification losses and regression losses are then computed for gradient back-propagation in each decoder layer.

However, we found that merely separating the classification heads for two tasks and supervising the model under both detection and tracking ground truth assignments can neither obtain a satisfying detector nor tracker, as shown in Table 4. Clearly, simply separating the classification heads cannot solve the conflict between the sample classification in detection and tracking. This is because as previously mentioned, a part of positive samples for training detectors will be assigned as negative samples for training trackers, and vice versa. Therefore even when the classification heads of two tasks are separated, classification gradients on those samples will still be conflicted when back-propagated to the decoder layers. To address this, we propose to decompose the classification gradients in both tasks to identify and avoid their partial conflict on those conflicted samples. We refer to this design as **classification gradient coordination** between the two tasks.

We dynamically categorize object queries into four query groups in each decode layer based on their polarity in both tasks: positive for both tasks; positive for detection and negative for tracking; negative for detection and positive for tracking; and negative for both tasks. We denote those four groups of queries as \mathbf{Q}^{pp} , \mathbf{Q}^{pn} , \mathbf{Q}^{np} , and \mathbf{Q}^{nn} . This proposed query grouping can be illustrated as follows:

$$\mathbf{Q}^{\text{pp}} = \{q^k | k \in \hat{\sigma}_{\text{det}}, k \in \hat{\sigma}_{\text{trk}}\}, \quad (6)$$

$$\mathbf{Q}^{\text{pn}} = \{q^k | k \in \hat{\sigma}_{\text{det}}, k \notin \hat{\sigma}_{\text{trk}}\}, \quad (7)$$

$$\mathbf{Q}^{\text{np}} = \{q^k | k \notin \hat{\sigma}_{\text{det}}, k \in \hat{\sigma}_{\text{trk}}\}, \quad (8)$$

$$\mathbf{Q}^{\text{nn}} = \{q^k | k \notin \hat{\sigma}_{\text{det}}, k \notin \hat{\sigma}_{\text{trk}}\}. \quad (9)$$

Through this, the classification gradients of the two tasks can also be decomposed as the sum of detection or tracking classification gradients on each group of queries. The classification gradient conflict only lies in the second and third groups of queries, as illustrated in Fig. 2(a). Therefore we propose to remove the gradient of \mathbf{Q}^{pn} contributed to the tracking classification loss and the gradient of \mathbf{Q}^{np} contributed to the detection classification loss, as illustrated in Fig. 2(b). Through this, the partial classification gradient conflict of the two tasks on \mathbf{Q}^{pn} and \mathbf{Q}^{np} is avoided. \mathbf{Q}^{pp} and \mathbf{Q}^{nn} still contribute gradient in both heads since the partial classification gradients upon them are not contradictory.

3.3 Dynamic Attention Mask

Following previous end-to-end trackers [21, 30, 31, 45, 47, 48, 50], OneTrack generates predictions in a set-prediction fashion. Hence the model is encouraged to suppress redundant predictions regarding the same ground truth object through self-attention between object queries in each decoder layer. As previously mentioned, \mathbf{Q}^{pn} and \mathbf{Q}^{np} originates from the same ground truth object being assigned to different object queries in the two tasks. To avoid unexpected competition

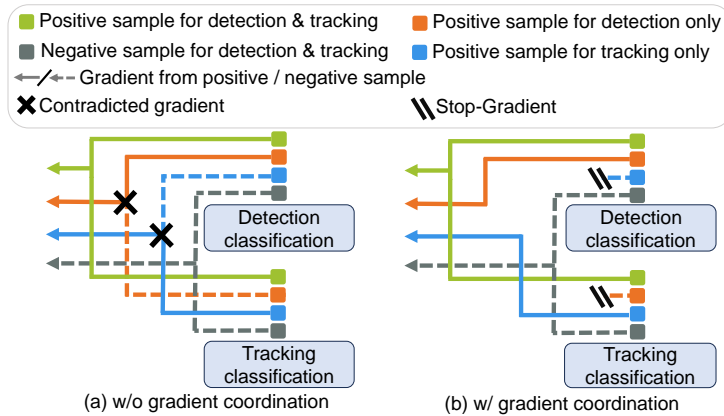


Fig. 2: Backpropagation of Classification Gradients. Gradient coordination prevents the classification gradient conflict between detection and tracking. Best viewed in color.

between task-specific queries regarding the same object, in each layer of the decoder, we dynamically mask the attention between Q^{pn} and Q^{np} . Attention between all other query groups is not masked. Through this, Q^{pn} and Q^{np} are not directly exposed to each other during the forward process.

We utilize the ground truth assignment results in each decoder layer to generate the self-attention mask for the next layer. Before the first decoder layer where detection queries are initialized as empty queries, we use distances between their initial reference points and centers of ground truths to perform bipartite matching. After each decoder layer, we perform bipartite matching based on matching costs between ground truths and predicted 3D boxes defined in [37]. We employ the Hungarian algorithm [17] for bipartite matching. As for outputs, the detection results of OneTrack include predictions from Q^{pp} , Q^{pn} and Q^{nn} . The tracking results includes predictions from Q^{pp} , Q^{pn} and Q^{nn} . We illustrate the pipeline of OneTrack in Fig. 3.

During inference, we categorize object queries into four groups based on the detection and tracking confidence scores of their predictions. Queries with high detection or tracking confidence scores over T_p are classified as positive samples for the respective tasks. We set $T_p = 0.2$ in our experiments.

At last, in both the training or inference process, we propagate the Top- K object queries with high confidence scores from both detection and tracking results to the next frame as \hat{Q}_t . In particular, we will still propagate the “locked” track queries to the future frames even if they are not in the Top- K object queries. This is for addressing challenging tracking scenarios, such as an object being occluded for several frames. We propagate “locked” track queries through this for up to $A_{age} = 3$ frames into future frames.

Following [31, 36], we also preserve the propagated object queries as the memory queue for the detection or tracking process for up to 4 frames. The mem-

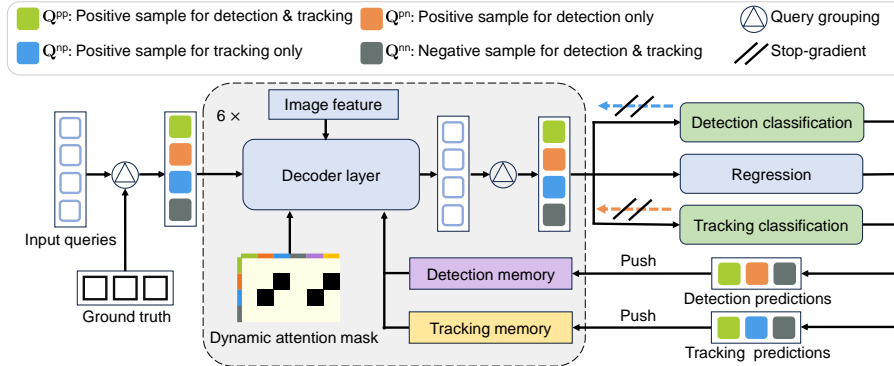


Fig. 3: Training pipeline of OneTrack. For each decoder layer in OneTrack, We identify object queries into four groups based on their positivity as training samples in detection and tracking tasks. Task-specific queries that are positive samples only for one task will only backpropagate gradients in the respective task-specific classification head to prevent conflicted classification gradients. We dynamically mask all attention across positive sample queries only for detection or tracking. Best viewed in color.

ory queues for the detection and tracking processes are updated independently. Those memory queries are provided to the decoder for performing hybrid attention proposed by [36]. As keys and values for the hybrid attention layer, detection and tracking memory queries are masked for Q^{pp} and Q^{pn} , respectively. We show the design of decoder layers in OneTrack in Fig. 4.

As demonstrated in Table 4, with the proposed dynamic hybrid-attention mask, the true joint training of detection and tracking can be achieved. This allows for training detection and tracking in a single training stage and inferring them in a single forward pass with a unified model. This feature of OneTrack significantly reduces its training cost and inference latency.

3.4 Modified Tracking Classification Loss

In addition to introducing the true joint training of detection and tracking to reduce the detection capability degradation of trackers, we also adjusted the tracking classification loss to suppress the in fact inaccurate predictions. As previously mentioned, previous end-to-end trackers regard track queries whose predictions are associated with ground truths as positive samples equally, regardless of the precision of their current predictions. This encourages the tracker to assign high confidence to the in fact inaccurate predictions from track queries. Therefore we propose to restrain the positive assignment of those in fact inaccurate predictions.

In previous end-to-end trackers, the assignment result of a ground truth object g can be denoted as a one-hot vector: $A_g^{n \times 1} = [0, \dots, 1, \dots, 0]$, where n is the number of object queries and the ground truth object is exclusively

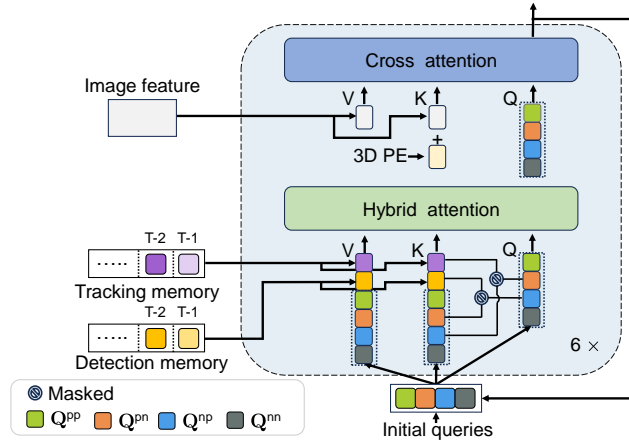


Fig. 4: Design of decoder layers in OneTrack. In addition to current object queries, queries in detection memory and tracking memory are fed to the hybrid attention layer as keys and values. As keys and values, Q^{pn} and detection memory are masked to Q^{np} . As keys and values, Q^{pp} and tracking memory are masked to Q^{pn} .

assigned to the i^{th} object query. This assignment is unaffected by the matching cost between the i^{th} object query and g if they have been associated in recent frames.

In OneTrack, we propose to suppress the assignment weight between ground truths and queries that are locked but have high matching costs. We adjust the assignment result of locked ground truth g as follows:

$$A_g^{n \times 1} = [0, \dots, W_{gi}, \dots, 0], \quad (10)$$

$$W_{gi} = -\frac{1}{\gamma} * \text{Max}((\text{dist}(g, q^i) - 0.5), 0) + 1, \quad (11)$$

where $\text{dist}(g, q^i)$ represents the distance between the centers of ground truth box g and the predicted box of the locked query q^i . γ is a hyper-parameter greater than zero. The assigned weight W_{gi} between g and q^i decreases linearly from 1.0 as $\text{dist}(g, q^i)$ increases from 0.5m. The assignment weight serves as the weighting factor for calculating box regression loss and the targets for predicting tracking confidence score. Instead of Focal Loss [24], we employ the Quality Focal Loss proposed by [20] to supervise the tracking classification head, given the continuous 0~1 confidence label.

4 Experiments

In this section, we first introduce our detailed experimental setup. Then we provide comparisons with recent works on nuScenes camera-based 3D tracking benchmark. Ablation studies on each component are presented in the last part.

Table 3: Compare with recent works on nuScenes dataset. †: trained from scratch. OneTrack-F / OneTrack-S : OneTrack trained under full-resolution / half-resolution as described in Sec. 4.2.

	Backbone	e2e	Resolution	AMOTA↑	AMOTP↓	MOTA↑	RECALL↑	IDS↓
<i>Validation Split</i>								
QD3DT [13]	R101	✓	1600×900	0.242	1.518	0.218	39.9%	5646
MUTR3D [48]	R101	✓	1600×900	0.294	1.498	0.267	42.7%	3822
CC-3DT [10]	R101	×	1600×640	0.429	1.257	0.385	53.4%	2219
PF-Track-S [31]	V2-99	✓	800×320	0.408	1.343	0.376	50.7%	166
PF-Track-F [31]	V2-99	✓	1600×640	0.479	1.227	0.435	59.0%	181
DQTrack [21]	V2-99	✓	800×320	0.446	1.251	-	-	1193
OneTrack-S†	V2-99	✓	800×320	0.492	1.122	0.409	58.8%	315
OneTrack-F†	V2-99	✓	1600×640	0.548	1.088	0.479	61.8%	389
<i>Test Split</i>								
QD3DT [13]	R101	✓	1600×640	0.217	1.550	0.198	37.5%	6856
MUTR3D [48]	R101	✓	1600×640	0.270	1.494	0.235	41.1%	6018
SRCN3D [32]	V2-99	×	1600×640	0.398	1.317	0.359	53.8%	3334
CC-3DT [10]	R101	×	1600×640	0.410	1.274	0.357	53.8%	3334
PF-Track [31]	V2-99	✓	1600×640	0.434	1.252	0.378	53.8%	249
DQTrack [21]	V2-99	✓	1600×640	0.523	1.096	0.444	62.2%	1204
OneTrack-F†	V2-99	✓	1600×640	0.554	1.021	0.461	60.8%	481

4.1 Datasets and Metrics

We evaluate our proposed method on the nuScenes dataset. **nuScenes** [5] dataset is a large-scale autonomous driving benchmark containing 1000 multi-modal videos. Videos recorded by six cameras are divided into 700, 150, and 100 scenes for training, validation, and testing, respectively. 3D box annotations of 10 object classes are provided for keyframes at 2Hz. For nuScenes 3D tracking benchmark, we report AMOTA, AMOTP [39], MOTA [3], Recall and identity switches (IDS). For the 3D detection task, we report mean Average Precision (mAP) and nuScenes Detection Score (NDS).

4.2 Implementation Details

Our implementation is mainly based on StreamPETR [36]. We define two implementation settings: The full-resolution setting for state-of-the-art comparison and the half-resolution setting for ablation studies. We conduct experiments using V2-99 [18] as the image backbone. OneTrack is trained using AdamW [28] optimizer with a batch size of 8 and a base learning rate of 4e-4. The cosine annealing policy is employed to adjust the learning rate. We use 644 initialized detection queries in each frame and propagate the Top-256 object queries into the next frame. When training, we record all matched ground truth and object query pairs as “locked” pairs for label assignment. An object query matched with a ground truth will be recorded as a track query. As previously mentioned, a “locked” pair will be preserved in memory for up to $A_{\text{age}} = 3$ frames, even if

Table 4: Ablation study on components breakdown.

Training task	Initialized from pre-trained detector	Gradient coordination	Dynamic mask	High-cost assign suppress	AMOTA↑	IDS↓	NDS↑	mAP↑
detection	-	-	-	-	-	-	0.570	0.479
tracking	×	-	-	×	0.305	121	0.455	0.313
tracking	✓	-	-	×	0.375	466	0.503	0.409
joint	×	×	×	×	0.425	246	0.517	0.410
joint	×	✓	×	×	0.441	231	0.555	0.462
joint	×	✓	✓	×	0.469	261	0.575	0.484
joint	×	✓	✓	✓	0.492	315	0.566	0.478

the respective track query is not in the Top- K object queries in each frame. For track queries that failed to predict confident prediction over A_{age} frames, its “locked” ground truth will be assigned to other “free” object queries. During inference, we initialize an object query as a track query if it predicts a 3D box with confidence over $T_{\text{conf}} = 0.4$. Additionally, all track queries are preserved and propagated across frames for up to $A_{\text{age}} = 3$ frames. Predictions from the same track query propagated across frames will be assigned the same tracking ID. We output all detection predictions with confidence scores over 0.05 and all tracking predictions with confidence scores over 0.3. We introduce other setting-specific training configurations as follows.

OneTrack-F: We crop input images from the initial resolution of 1600×900 to 1600×640 . We then conduct end-to-end training for 48 epochs. **OneTrack-S:** We first crop the input images to the resolution of 1600×640 and downsample them to 800×320 . Models are trained for 24 epochs.

4.3 Comparisons with State-of-the-arts

As shown in Table 3, OneTrack outperforms all previous trackers in both the validation and test set of nuScenes dataset by a significant margin. On the validation set, OneTrack surpasses previous state-of-the-art PF-Track [31] by 10.2% AMOTA, On the test set, OneTrack surpasses DQTrack [21] by 3.3% AMOTA. Notably, OneTrack reports fewer than half of the identity switches compared to those reported by DQTrack. It is also worth mentioning that OneTrack is the only model in Table 3 that is trained from scratch.

4.4 Ablations

We conduct ablation studies on each proposed component with OneTrack-S on the nuScenes validation set.

Components breakdown. The breakdown analysis of proposed components is presented in Table 4. Here “det” or “trk” in the first row stands for only training the model for detection or tracking tasks. And “joint” stands for jointly training the model for both detection and tracking tasks, with an extra classification head attached to the model. As shown in Table 4, the model trained solely

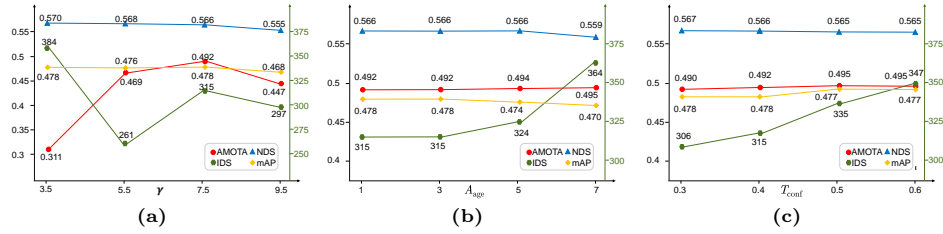


Fig. 5: Ablation on hyper-parameters. (a) On parameter γ in the modified tracking losses. (b) On the max preserved frames of track queries A_{age} . (c) On the score threshold T_{conf} for track query initialization.

for the tracking task suffers from severe degradation in detection performance, which results in their inferior detection performance compared to the detector. Initializing the tracker from a detector checkpoint or trivially supervising the model with both detection and tracking losses can only slightly alleviate this problem. Our proposed components including classification gradient coordination, dynamic attention mask, and high-cost tracking assignment suppress, are all validated as highly effective in achieving the true joint training of detection and tracking tasks. It is noteworthy that OneTrack without modifying its tracking loss achieves an even slightly better detection performance than the model trained solely as a detector. This observation suggests the potential for further enhancing the detection process through the integrated tracking process in a unified model.

Modified tracking loss. In Fig. 5(a), we conduct ablation studies on the hyper-parameter γ in the modified tracking loss for suppressing positive assignments of inaccurate predictions. OneTrack achieves optimal tracking performance when γ is set to 7.5. While the detection performance of OneTrack shows improvement as γ increases, this gain becomes limited when γ exceeds 7.5.

Track age. We conduct ablation studies on the maximum preservation age A_{age} of track queries. As illustrated in Fig. 5(b), a longer track age will lead to more identity switches and decreased detection performance. We speculate that this is because propagating too many outdated track queries will mislead the object feature extraction process in the current frame.

Confidence threshold for track query initialization. We assess the impact of T_{conf} in Fig. 5(c). The AMOTA performance will slightly increase as T_{conf} increases, which results from the more accurate tracking predictions on average. However, a higher T_{conf} reduces the initialization of track queries from detection queries with low confidence. This limitation hinders tracking objects under poor observation conditions, resulting in a higher number of identity switch cases.

Separated classification heads. As shown in Table 5, when its two classification heads share weights except for the final linear layer, the tracking and detection performance of OneTrack is slightly worse than when the classification heads are independent. We speculate that this is because the separation of classification heads leads to the easier convergence of both heads.

Table 5: Ablation on the separated classification heads for two tasks.

Cls heads	AMOTA↑	IDS↓	NDS↑	mAP↑
shared	49.0	376	0.559	0.473
separated	49.2	315	0.566	0.478

Table 6: Computational cost comparison with recent works. All training time costs and inference latencies are measured on a single A100 GPU. *: PF-Track is first trained as a detector and then as a tracker for 12 epochs each. We train DQTrack and OneTrack from scratch for 24 epochs.

Method	Backbone	Resolution	Training time cost	Inference latency
PF-Track	V2-99	800×320	34h+82h*	124.4ms
DQTrack	V2-99	800×320	176h	115.2ms
OneTrack	V2-99	800×320	87h	83.7ms

Computational cost. To ensure fair comparison, we apply gradient checkpointing [7] on the decoder layers of all trackers. We measure their training cost and inference latency on a single A100 GPU. As shown in Table 6, the training cost of OneTrack is only half that of previous state-of-the-art trackers. The inference latency of OneTrack is also significantly lower. We attribute the low computational cost of OneTrack to its elegant model design. We do not extend the tracker with various plug-in modules [31] or separate the detection and tracking stages of the tracker [21], which will introduce additional computational burdens. Notably, OneTrack is trained from scratch while PF-Track and DQTrack rely on extra training stages for achieving high tracking performance.

5 Conclusion

In this work, we delve into the optimization conflict between detection and tracking in end-to-end trackers, which arises from the partially conflicted classification gradients between two tasks. Based on this observation, we propose to identify the conflicted positive samples between two tasks through query grouping and solve the conflicted gradients with the proposed gradient coordination, dynamic attention mask, and suppression tracking positive samples of low quality. On the nuScenes dataset, OneTrack outperforms all previous end-to-end trackers.

Acknowledgement

This work was supported in part by the National Key R&D Program of China (No. 2022ZD0160102), the National Natural Science Foundation of China (No. U21B2042, No. 62320106010), and in part by the 2035 Innovation Program of CAS, and the InnoHK program.

References

1. Ali, A., Jalil, A., Niu, J., Zhao, X., Rathore, S., Ahmed, J., Aksam Iftikhar, M.: Visual object tracking—classical and contemporary approaches. *Frontiers of Computer Science* **10**, 167–188 (2016)
2. Benbarka, N., Schröder, J., Zell, A.: Score refinement for confidence-based 3d multi-object tracking. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8083–8090. IEEE (2021)
3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing* **2008**, 1–10 (2008)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
7. Chen, T., Xu, B., Zhang, C., Guestrin, C.: Training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174 (2016)
8. Chiu, H.k., Li, J., Ambruş, R., Bohg, J.: Probabilistic 3d multi-modal, multi-object tracking for autonomous driving. In: ICRA (2021)
9. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: Transmot: Spatial-temporal graph transformer for multiple object tracking. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4870–4880 (2023)
10. Fischer, T., Yang, Y., Kumar, S., Sun, M., Yu, F.: CC-3DT: panoramic 3d object tracking via cross-camera fusion. In: CoRL. Proceedings of Machine Learning Research, vol. 205, pp. 2294–2305. PMLR (2022)
11. Guan, H., Song, C., Zhang, Z.: Gramo: geometric resampling augmentation for monocular 3d object detection. *Frontiers of Computer Science* **18**(5), 185706 (2024)
12. He, J., Huang, Z., Wang, N., Zhang, Z.: Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5299–5309 (2021)
13. Hu, H.N., Yang, Y.H., Fischer, T., Darrell, T., Yu, F., Sun, M.: Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022)
14. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
15. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
16. Kim, A., Ošep, A., Leal-Taixé, L.: Eagermot: 3d multi-object tracking via sensor fusion. In: 2021 IEEE International conference on Robotics and Automation (ICRA). pp. 11315–11321 (2021)
17. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval research logistics quarterly* **2**(1-2), 83–97 (1955)

18. Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 0–0 (2019)
19. Li, J., Ding, Y., Wei, H.L., Zhang, Y., Lin, W.: Simpletrack: Rethinking and improving the jde approach for multi-object tracking. *Sensors* **22**(15), 5863 (2022)
20. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems* **33**, 21002–21012 (2020)
21. Li, Y., Yu, Z., Phillion, J., Anandkumar, A., Fidler, S., Jia, J., Alvarez, J.: End-to-end 3d tracking with decoupled queries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18302–18311 (2023)
22. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1477–1485 (2023)
23. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
24. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
25. Liu, C., Chen, X.F., Bo, C.J., Wang, D.: Long-term visual tracking: review and experimental comparison. *Machine Intelligence Research* **19**(6), 512–530 (2022)
26. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
27. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3262–3272 (2023)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
29. Marinello, N., Proesmans, M., Van Gool, L.: Triplettrack: 3d object tracking using triplet embeddings and lstm. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 4499–4509 (2022)
30. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8844–8854 (2022)
31. Pang, Z., Li, J., Tokmakov, P., Chen, D., Zagoruyko, S., Wang, Y.X.: Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17928–17938 (2023)
32. Shi, Y., Shen, J., Sun, Y., Wang, Y., Li, J., Sun, S., Jiang, K., Yang, D.: Srcn3d: Sparse r-cnn 3d surround-view camera object detection and tracking for autonomous driving. arXiv preprint arXiv:2206.14451 (2022)
33. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10860–10869 (2021)
34. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. In: ICCV. pp. 10840–10849. IEEE (2021)

35. Wang, Q., Chen, Y., Pang, Z., Wang, N., Zhang, Z.: Immortal tracker: Tracklet never dies. arXiv preprint arXiv:2111.13672 (2021)
36. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3621–3631 (2023)
37. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
38. Wang, Y., Chen, Y., Zhang, Z.: Frustumformer: Adaptive instance-aware resampling for multi-view 3d detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5096–5105 (2023)
39. Weng, X., Kitani, K.: A baseline for 3d multi-object tracking. arXiv preprint arXiv:1907.03961 **1**(2), 6 (2019)
40. Weng, X., Wang, J., Held, D., Kitani, K.: 3d multi-object tracking: A baseline and new evaluation metrics. IROS (2020)
41. Weng, X., Wang, Y., Man, Y., Kitani, K.: Gnn3dmot: Graph neural network for 3d multi-object tracking with multi-feature learning. arXiv preprint arXiv:2006.07327 (2020)
42. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
43. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird’s-eye-view recognition via perspective supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
44. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. CVPR (2021)
45. Yu, E., Wang, T., Li, Z., Zhang, Y., Zhang, X., Tao, W.: Motrv3: Release-fetch supervision for end-to-end multi-object tracking. arXiv preprint arXiv:2305.14298 (2023)
46. Zaech, J.N., Liniger, A., Dai, D., Danelljan, M., Van Gool, L.: Learnable online graph representations for 3d multi-object tracking. IEEE Robotics and Automation Letters **7**(2), 5103–5110 (2022)
47. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision. pp. 659–675. Springer (2022)
48. Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4537–4546 (2022)
49. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision **129**, 3069–3087 (2021)
50. Zhang, Y., Wang, T., Zhang, X.: Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22056–22065 (2023)
51. Zhao, H., Zhang, J., Chen, Z., Yuan, B., Tao, D.: On robust cross-view consistency in self-supervised monocular depth estimation. Machine Intelligence Research **21**(3), 495–513 (2024)