ADA-Track: End-to-End Multi-Camera 3D Multi-Object Tracking with Alternating Detection and Association

Supplementary Material

A. Model Details

We provide the details of the model architectures as well as some design choices for the experiments with the DETR3D [6] and PETR [5] detector.

DETR3D For all DETR3D-based [6] experiments, we use ResNet-101 [2] as the backbone. A FPN [4] image neck is attached to the ResNet-101 which outputs multi-scale feature maps $\{C_2, C_3, C_4, C_5\}$ with downsampling rates $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ for the detection head. The detection head consists of 6 transformer decoder layers. Each decoder layer has a query-to-query self-attention, a DETR3D-based query-to-image cross-attention, and an edge-augmented cross-attention. The box regression and classification heads with two-layer MLPs are attached to the output of the DETR3D-based query-to-image cross-attention. All decoder layers have an embedding dimension of 256 and a feed-forward dimension of 512. Following DETR3D [6], the query embeddings and query position encodings of the detection queries are randomly initialized, and the initial reference points are estimated from their initial position encoding using a linear layer.

PETR For all PETR-based [5] experiments, we utilize VoVNetV2-99 [3] as backbone and a FPN [4] image neck. The FPN feature map C_5 is upsampled and fused with C_4, producing the final single-scale feature map with the downsampling rate $\frac{1}{16}$ for the detection head. The architecture of the detection head is the same as in the DETR3D-based experiments, except that the query-to-image cross-attention is based on PETR [5] with 3D position encodings. The embedding dimension is 256 and the feed-forward dimension is 2048. In contrast to [6], PETR [5] generates query position encodings using the uniformly initialized reference points while initializing detection queries. In the query propagation phase, we adhere to this setting to update the query position encodings using the updated reference point positions at each timestamp. We found this design choice to be essential for ensuring the effectiveness of the PETR-based ADA-Track.

B. Additional Experiments

We provide additional experiments of ADA-Track to validate our system design. All experiments are evaluated on the nuScenes validation set with the DETR3D detector.

	bicycle	bus	car	motorcycle	pedestrian	trailer	truck	average
TBA-Baseline	0.549	0.553	0.706	0.499	0.597	0.276	0.532	0.530
ADA-Track	**0.551**	**0.675**	**0.725**	**0.545**	**0.613**	**0.336**	**0.571**	**0.574**

Table A1. IDF1 for all categories on the nuScenes validation set.

	Params	FLOPS	Inference time
TBA-Baseline	59.73 M	1024 G	296 ms
TBD-Baseline	63.72 M	1025 G	297 ms
ADA-Track	63.73 M	1054 G	308 ms

Table A2. Complexity and runtime analysis based on DETR3D with ResNet-101 backbone and image resolution of 1600×900. Runtime is measured on an RTX 2080ti.

B.1. IDF1 Analysis

Approaches with an explicit association module typically exhibit a relatively higher IDS than TBA-based methods, as illustrated in our analysis in Tables 1 and 2. However, this discrepancy might stem from the evaluation protocol of nuScenes tracking benchmark [1] which computes IDS for each category at the recall where the highest MOTA is achieved. Nevertheless, different methods do not necessarily achieve the best IDS while achieving the highest MOTA due to trade-offs between FP, FN, and IDS. To verify the consistency of association of ADA-Track compared to TBA-Baseline, we show a supplementary comparison of IDF1 across all categories in Table A1. In comparison to TBA-Baseline, we observe an $8.3\%P$ higher average IDF1 and consistently higher IDF1 for all categories, especially for large objects such as buses and fast-moving objects such as motorcycles. Moreover, the IDF1 improvement of ADA-Track is particularly noteworthy for categories with lower occurrences (except for cars and pedestrians), showing its ability to handle class imbalance. The analysis based on IDF1 underscores the efficiency of ADA-Track in associating tracks consistently.

B.2. Complexity and runtime analysis

ADA-Track and TBD-Baseline require additional Edge-Augmented Cross-Attention modules compared to TBA-Baseline. We compare the number of parameters, FLOPs and runtime of ADA-Track with TBA-Baseline and TBD-Baseline in Table A2. As shown in Table A2, ADA-Track only adds about 6.7% parameters, 2.9% FLOPs, 4.1% inference time to TBA, and even less compared to TBD.

rel. pos. encoding	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
Center	0.366	1.392	**0.510**	0.325	**916**
None	0.341	1.442	0.474	0.300	1442
Appearance	0.358	1.417	**0.510**	0.318	1088
Box	**0.378**	**1.391**	0.507	**0.343**	981

Table A3. Ablation study on appearance and geometric features to build the relative positional encoding in the edge-augmented cross-attention. Our choice is *Box* (last row) which uses all the box parameters to build geometric-based relative position encoding. *Center* denotes that only box centers are used. *None* denotes no relative position encoding. *Appearance* uses the query feature differences as appearance-based relative position encoding.

Edge feat. iteration	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
	0.346	1.421	0.498	0.310	1275
✓	**0.378**	**1.391**	**0.507**	**0.343**	**981**

Table A4. Ablation study on the iterative refinement of edge features over decoder layers.

attention det → track	attention track → det	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
		0.367	1.404	0.504	0.320	936
✓		0.364	1.411	0.503	0.317	**899**
	✓	**0.378**	**1.388**	**0.513**	**0.344**	904
✓	✓	**0.378**	1.391	0.507	0.343	981

Table A5. Ablation study on computing attention across different query types in self-attention.

Therefore, the additional computational overhead of ADA-Track is minimal. Despite this slight increase in complexity, the performance gain of ADA-Track is much more significant, *e.g.* 17.6% higher AMOTA than TBA-Baseline with DETR3D (see Table 1).

B.3. Ablation Studies

Appearance and geometry cues for association We investigate the role of appearance and geometric cues in the learnable association module based on edge-augmented cross-attention. As shown in Table A3, using only the center position (first row) instead of the complete box parameters (fourth row) leads to 1.2%P decrease in AMOTA, which underscores the significance of leveraging the entire box information for robust data association. If geometric features are excluded (second row), the zero-initialized edge features are refined exclusively through appearance-based query features layer-by-layer, resulting in a substantial performance drop across all metrics when compared to the use of geometric-based edge features. This observation highlights the usage of geometric features in enhancing the model's ability to distinguish between object instances. Even without geometric features, using relative positional encodings derived from query feature differences (third row) yields a notable AMOTA increase of 1.7%P compared to scenarios without relative positional encoding (second row). This finding shows the importance of the exact edge feature encoding in the model architecture of edge-augmented cross-attention.

Edge feature refinement Table A4 illustrates the effectiveness of the iterative refinement of edge features over decoder layers. In the case where edge features remain independent within each decoder layer (first row), there is a notable decrease in AMOTA by 3.2%P when compared to scenarios where edge feature refinement occurs across layers (second row). This experiment shows the potential for iterative optimization of data association across decoder layers, aligning with the fundamental design of our architecture. In addition, since the edge features also participate in the query feature update in the edge-augmented cross-attention, the refinement of the edge features itself also contributes to the iterative refinement of query representations, which also improves the overall performance.

Masked self-attention The self-attention layer in ADA-Track facilitates temporal modeling between queries. As shown in Table A5, when we use only the attention from track to detection queries in the self-attention layer (third row), we observe no changes in AMOTA compared to our default setting (fourth row). The other metrics are even slightly better, indicating that this setting is slightly preferable compared to the default setting that we used in all other experiments. Conversely, using only the attention from detection to track queries (second row) results in a noticeable drop of 1.4%P in AMOTA. The results are similar when attention is not computed across different query types (first row). These results show that self-attention is important for enhancing detection queries, enabling detection queries to incorporate information from past tracks and frames.

Association module Table A6 compares Edge-Augmented Cross-Attention with alternative association modules. We replace the Edge-Augmented Cross-Attention with association networks utilizing the difference or concatenation of detection and track query features (node features). In both cases, we use an MLP and sigmoid to obtain the association scores S as before. Using only the difference or concatenation of node features results in a significant performance drop, highlighting the necessity of using explicit edge features and the effectiveness of Edge-Augmented Cross-Attention in data association.

Robustness against appearance change One of the biggest challenges of MOT in autonomous driving is ego-

Association module	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
Node Difference	0.355	1.421	0.483	0.314	1249
Node Concatenation	0.349	1.421	0.451	0.306	**977**
Edge-Aug. Cross Attn.	**0.378**	**1.391**	**0.507**	**0.343**	981

Table A6. Ablation study on different association modules.

ego speed	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
$\geq 5m/s$	0.377	**1.378**	**0.512**	**0.355**	**589**
$\geq 0m/s$	**0.378**	1.391	0.507	0.343	981

Table A7. Abaltion study on the impact of ego-motion.

N_D	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
100	0.341	1.437	0.461	0.301	1213
200	0.370	1.397	0.471	0.327	847
300	**0.378**	**1.391**	0.507	**0.343**	981
400	0.377	1.395	0.486	0.329	**805**
500	0.377	1.397	**0.514**	0.335	919

Table A8. Ablation study on the number of detection queries N_D.

motion, where the camera mounting points move with the ego-vehicle, leading to appearance changes of observed objects due to varying observation angles. We evaluate the sequences where the average speed of the ego-vehicle is $\geq 5m/s$, indicating significant appearance changes in observed objects due to ego-motion. As shown in Table A7, ADA-Track achieves similar AMOTA for sequences with ego-motion ($\geq 5m/s$) compared to all scenes ($\geq 0m/s$). Some secondary metrics are even better for the sequences with ego-motion.

Number of queries Table A8 shows the comparison of varying the number of detection queries N_D, where AMOTA increases until $N_D = 300$. Using fewer detection queries typically causes missed detections and lower detection performance, inevitably affecting the tracking performance. However, continuing to increase the number of detection queries does not yield further improvements. This is attributed to the risk of introducing an imbalance in the classification for the data association task, potentially reducing the association performance. As a result, we opt for $N_D = 300$ as the default setting.

Association loss weight We evaluate the weight of the association loss λ_{asso} in Table A9. Using $\lambda_{\text{asso}} = 5$ and $\lambda_{\text{asso}} = 10$ yield the same AMOTA of 0.378. Lower or higher association loss weights result in performance drops with different ranges, which can be attributed to the imbalance of the multi-task training. We choose $\lambda_{\text{asso}} = 10$ as default.

λ_{asso}	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
2	0.371	1.408	0.507	0.332	1000
5	**0.378**	**1.386**	**0.518**	0.331	922
10	**0.378**	1.391	0.507	**0.343**	981
20	0.377	1.389	0.504	0.336	**920**
50	0.365	1.399	0.516	0.327	911

Table A9. Ablation study on the association loss weight λ_{asso}.

γ	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
0.0	0.358	1.404	0.496	0.323	1659
0.5	0.359	1.409	0.506	0.323	1118
1.0	**0.378**	1.391	0.507	**0.343**	981
1.5	0.363	**1.384**	**0.516**	0.326	984
2.0	0.366	1.406	0.465	0.322	**952**

Table A10. Ablation study on the focusing parameter γ of the association loss.

T_D	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
2	0.361	1.421	0.495	0.337	1246
3	0.369	1.404	0.496	0.338	1070
4	0.375	1.395	0.503	0.340	1072
5	**0.378**	**1.391**	**0.507**	**0.343**	981
6	0.373	1.392	0.501	0.337	**934**

Table A11. Ablation study on the duration of keeping unmatched tracks T_d.

τ_{new}	AMOTA↑	AMOTP↓	Recall↑	MOTA↑	IDS↓
0.0	0.203	1.421	0.347	0.213	1575
0.1	0.248	1.409	0.380	0.246	1827
0.2	0.315	1.386	0.462	0.280	1520
0.3	0.365	**1.379**	0.495	0.327	1217
0.4	**0.378**	1.391	**0.507**	**0.343**	981
0.5	0.364	1.424	0.489	0.337	**869**

Table A12. Ablation study on the score threshold for track spawning τ_{new}.

Focusing parameter in association loss We use the focal loss as the association loss $\mathcal{L}_{\text{asso}}$ with a focusing parameter of $\gamma = 1.0$. This choice is validated in Table A9, where lowering or raising the focusing parameter γ results in a notable decrease of AMOTA, ranging from 1.2%P to 2.0%P. Therefore $\gamma = 1.0$ is a reasonable choice for effectively controlling the class imbalance in the data association task.

Hyperparameters during inference During inference, we use two hyperparameters: the number of frames until

unmatched tracks are kept and the score threshold τ_{new} to spawn new tracks. We evaluate both hyperparameters in Table A11 and Table A12. As shown in Table A11, low values of T_d cause a significant performance drop caused by the insufficient handling of occluded objects. The AMOTA peaks at $T_d = 5$ and a higher value again leads to a decrease of AMOTA, which might keep too many tracks and cause a higher class imbalance in the data association. As for the score threshold τ_{new}, when setting $\tau_{\text{new}} \leq 0.3$, the tracker initializes excessive noisy detections, which results in a significant performance drop. We choose $\tau_{\text{new}} = 0.4$ as the default value.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1

[4] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[5] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022. 1

[6] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 1