Supplementary Material for 3DMOTFormer: Graph Transformer for Online 3D Multi-Object Tracking

Shuxiao Ding[1,2], Eike Rehder[3], Lukas Schneider[1], Marius Cordts[1], Juergen Gall[2,4]
[1]Mercedes-Benz AG, Sindelfingen, Germany,
[2]University of Bonn, Bonn, Germany,
[3]Robert Bosch GmbH, Stuttgart, Germany
[4]Lamarr Institute for Machine Learning and Artificial Intelligence, Germany
{shuxiao.ding,lukas.schneider,marius.cordts}@mercedes-benz.com,
e.rehder@gmx.de, gall@iai.uni-bonn.de

A. Framework Details

A.1. Association Graph Representation

We show an illustration of our association graph in Figure A. At a time stamp t, the association graph is built between detection and track nodes, connected by dashed lines. The track nodes include inactive nodes that were unassociated at previous frame, *e.g.* yellow and orange nodes at $t-2$ and $t-3$. If a track node is unassociated for more than $T_d = 3$ time stamps, it is permanently deleted from the graph, *e.g.* red nodes at $t-4$. The association of past frames is shown in solid lines, where associated track nodes (dark green) are removed from the association graph. As track and detection nodes are two disjoint sets, the association graph is bipartite. This bipartite representation does not require a complex heuristic algorithm that decodes multi-frame network outputs into hard association. In contrast, in other approaches that use the spatiotemporal graph with a fixed time window [16, 10, 7], bipartite matching is carried out for every pair of timestamps and hence a conflict resolution step is needed.

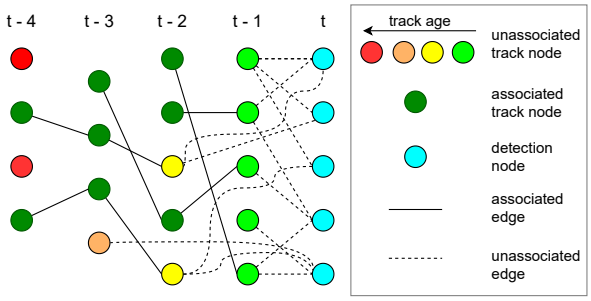

Figure A. An illustration of the association graph representation in our framework where each node represents a detection. The graph at time stamp t consists of two sets of nodes that are connected by dashed lines: detection nodes at t (cyan) and track nodes from past frames. To represent the age of a track node, we use a color encoding from light green to red. The associated nodes (dark green) are not processed by the graph.

A.2. Track Update Module

Figure B illustrates the details of our track update module. Given detections V_D and tracks V_T at time stamp t as well as their association score from the network, the greedy matching generates matched track-detection pairs as well as unmatched tracks and unmatched detections. For matched detection-track pairs, we replace the track features using the feature vectors $h_D^{(L_d)}$ of the matched detection instance. For example, track d is matched with detection 1, therefore the feature vector of detection 1 becomes the feature vector of track d in the next frame. Unmatched detections are initialized as new tracks with their features $h_D^{(L_d)}$, *e.g.* detection 5 is initialized with track ID h. We keep unmatched tracks for $T_d = 3$ frames and pass their features $h_T^{(L_e)}$ to the next frame, *e.g.* track a and c. More concisely, this procedure selects the encoder outputs $h_T^{(L_e)}$ and decoder outputs $h_D^{(L_d)}$ to build track features $h_T^{(0)}$ for the next frame, based on the matching results and the rules for spawning and termination. Besides node features, every track and detection node has additional fields, *e.g.* bounding box parameters, category and velocity, which are used to build the graph in the next frames. The fields of new track nodes are updated in the same way as the node features.

A.3. Soft Association

A drawback of the bipartite graph representation is the limited temporal receptive field. As shown in Figure A, the associated nodes (dark green nodes) as well as the associated edges (solid lines) are not included in the graph that

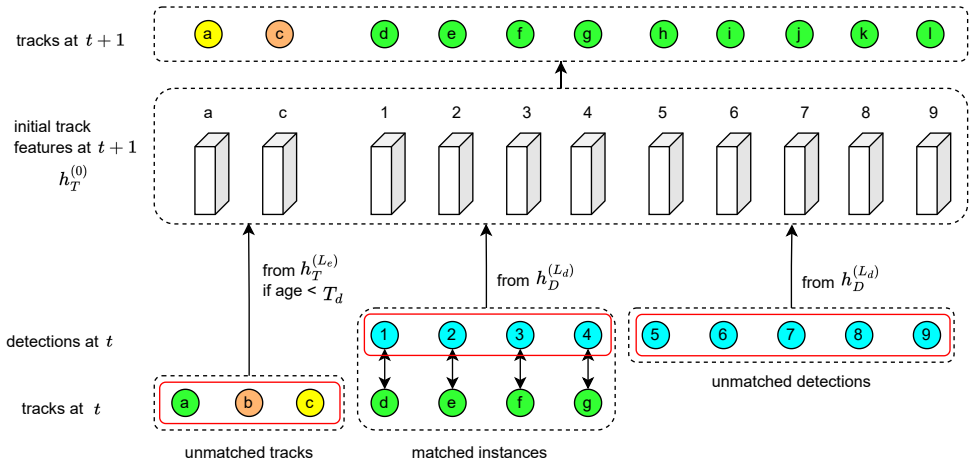

Figure B. An illustration of the track update module. Track nodes are indexed by letters in small case and detection nodes in arabic numbers. Color encoding of both track and detection nodes follows Figure A. The greedy matching generates matched instances, unmatched detections and unmatched tracks. For the three sets, different rules are used to initialize, terminate or update the track. The track node features $h_T^{(0)}$ at $t+1$ are initialized from detection $h_D^{(L_d)}$ and track features $h_T^{(L_e)}$ at time stamp t.

will be processed by the Graph Transformer model.

However, this problem is mitigated by the soft association characteristics of transformers. After being processed by the transformer model, the updated node feature $h_{D,i}^{(L_d)}$ is formed by features from all neighboring track nodes $h_{T,j}^{(L_e)}$ with $j \in \mathcal{N}(i)$ by a weighted average $h_{D,i}^{(L_d)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij} h_{T,j}^{(L_e)}$, where α_{ij} with $\sum_j \alpha_{ij} = 1$ represents the attention learnt by the transformer. This updated feature represents an implicit soft association to all combined track nodes, regardless to the hard association decided by greedy matching. We directly use this feature as initial node features when evolving the graph to the next time stamp $t+1$. Therefore, the network is aware of the historical information and soft association from last frames. Combined with our sequential batch optimization during training and back-propagation through time (BPTT) [11], the network is trained to be able to correct errors from past frames.

A.4. Model Details

Our model employs a feature dimension $d = 128$ in all fully connected layers, attention layers, FFNs, *etc*. We stack $L_e = 1$ encoder and $L_d = 3$ decoder layers. For all graph self-attention and edge-augmented graph cross-attention layers, we use $C = 8$ attention heads and a dropout rate [12] of 0.1. Following [14], we use LayerNorm [1] in front of all attention and FFN blocks, followed by residual connections [4].

Detector	Tracker	AMOTA↑	AMOTP↓	MOTA↑	IDS↓	FRAG↓
MEGVII	CenterPoint [15]	0.598	0.682	0.504	462	462
	OGR3MOT [16]	0.631	0.762	**0.541**	**263**	**305**
	GNN-PMB [8]	0.619	0.716	–	508	372
	3DMOTFormer	**0.641**	**0.639**	0.535	328	497
BEVFusion	CenterPoint [15]	0.712	**0.542**	0.616	696	**346**
	3DMOTFormer	**0.749**	0.550	**0.652**	**447**	443
	CAMO-MOT† [13]	0.760	0.561	–	243	–

Table A. Comparison with other methods using detections from different detectors on nuScenes validation set. † denotes method using additional appearance cue for data association.

B. Experiments

B.1. Comparisons using other Detectors

To further investigate the generalization on different detectors, we compare the results with other MOT approaches using MEGVII [17] and BEVFusion [9] detections in Table A. 3DMOTFormer outperforms OGR3MOT by 1.0%P AMOTA and achieves the highest AMOTA among all approaches with MEGVII detections. BEVFusion is published later than our geometry-based baselines and only CAMO-MOT [13] evaluated their method using BEVFusion detections. We first run the CenterPoint [15] tracking algorithm on BEVFusion detections which achieves a higher AMOTA compared to existing works using CenterPoint detections. However, our approach again surpasses it by 3.7%P AMOTA and improves the maximally achievable MOTA by 3.6%P. With additional image features for data association, CAMO-MOT achieves an AMOTA improvement of 1.1%P compared to our approach.

Method	Detector	mAP↑	NDS↑	Tracker	AMOTA↑	AMOTP↓	MOTA↑	IDS↓	FRAG↓
CAMO-MOT	BEVFusion [9] & FocalsConv [3]	70.23 63.86	72.88 69.41	CAMO-MOT† [13]	0.753	0.472	0.635	324	511
BEVFusion	BEVFusion-e‡ [9]	74.99	76.09	CenterPoint Tracker [15]	0.741	0.403	0.603	506	422
MSMDFusion-base	MSMDFusion-base [5]	71.50	74.00	CenterPoint Tracker [15]	0.740	0.549	0.624	1088	743
3DMOTFormer-BEVFusion	BEVFusion [9]	70.23	72.88	3DMOTFormer (ours)	0.725	0.539	0.609	593	499
TransFusion	TransFusion [2]	68.90	71.68	CenterPoint Tracker [15]	0.718	0.551	0.607	944	673
UVTR-Multimodal	UVTR-Multimodal [6]	67.10	71.10	CenterPoint Tracker [15]	0.701	0.686	0.618	941	798
TransFusion-Lidar	TransFusion-Lidar [2]	65.52	70.23	CenterPoint Tracker [15]	0.686	0.529	0.571	893	626
3DMOTFormer-CenterPoint	CenterPoint [15]	58.00	65.50	3DMOTFormer (ours)	0.682	0.496	0.556	438	529

Table B. Results on the nuScenes test set. We compare 3DMOTFormer using BEVFusion and CenterPoint detections with other tracking-by-detection approaches using different detections in terms of both detection and tracking performance. † denotes method using additional appearance cue for data association. ‡ denotes using model ensemble.

T_d	AMOTA↑	AMOTP↓	MOTA↑	FP↓	FN↓	IDS↓	FRAG↓
1	0.6620	0.6495	0.5750	**10899**	20395	1280	1070
2	0.7024	0.5577	**0.6091**	11887	19413	524	528
3	**0.7121**	0.5149	0.6071	13010	19281	341	436
4	0.7121	0.4937	0.6036	14101	**19198**	278	388
5	0.7060	0.4913	0.5956	14295	19712	222	363
6	0.7046	**0.4830**	0.5974	13604	20713	**195**	**348**

Table C. Ablation study on the maximum track age T_d.

d	C	L_e	L_d	AMOTA↑	AMOTP↓	MOTA↑	IDS↓	FRAG↓
128	8	1	3	**0.7121**	**0.5149**	0.6071	341	436
128	8	1	1	0.7065	0.5256	0.6025	371	428
128	8	1	2	0.7067	0.5225	0.6023	375	421
128	8	1	4	0.7113	0.5234	**0.6115**	**327**	423
128	8	0	3	0.7074	0.5288	0.6041	497	434
128	8	2	3	0.7106	0.5095	0.6070	367	**420**
128	8	3	3	0.7098	0.5214	0.6046	371	448
64	8	1	3	0.7084	0.5242	0.6029	360	423
256	8	1	3	0.7092	0.5201	0.6046	368	422
128	4	1	3	0.7098	0.5304	0.6112	368	437
128	16	1	3	0.7087	0.5208	0.6081	372	437

Table D. Ablation study on the model hyperparameters.

B.2. Test Results using BEVFusion Detections

As shown in Table A, using BEVFusion detections leads to better MOT performance which we can attribute to its higher detection performance. To explore the potential of 3DMOTFormer, we show the tracking results using BEVFusion detections on nuScenes test split and compare with the highest-ranking tracking-by-detection methods in Table D. We also show the mAP (mean Average Precision) and NDS (nuScenes detection score) of the object detectors for all methods because they strongly affect the tracking performance. Many works [9, 5, 2, 6] focus on improving the object detector and use the CenterPoint Tracker [15] for test submission. As can be observed, the tracking performance of these approaches, especially AMOTA highly depends on the detection performance. The submission of BEVFusion [9] uses model ensemble which yields 4.76 %P mAP and 3.21 %P NDS improvements compared to a model without ensemble. However, only a checkpoint without an ensemble of models of BEVFusion is made publicly available for generating detections for tracking. In addition, this available checkpoint is trained solely on the training set, whereas the reported 70.23 mAP and 72.88 NDS on the test set are from a model trained on both training and validation set. This can result in a slightly lower real detection performance on the test set of our used detections than the reported 70.23 mAP and 72.88 NDS. Despite considerable lower detection performance than BEVFusion-e, 3DMOTFormer achieves an AMOTA of 0.725 on the nuScenes test split. Considering the improvements of 3DMOTFormer against the CenterPoint Tracker in Table A, we believe that 3DMOTFormer can achieve a much higher performance if the unavailable detections of BEVFusion-e are used. CAMO-MOT [13] uses the same BEVFusion checkpoint as ours to generate detections, but augments it with detections from FocalsConv [3]. Combined with a tracker based on both geometry and appearance cues, CAMO-MOT achieves the highest 0.753 AMOTA among all methods.

B.3. Ablation Studies

We provide more supplementary ablation studies of 3DMOTFormer to verify our design choices. All experiments are evaluated on the NuScenes validation set using CenterPoint detections.

Maxixum track age The maximum track age T_d is one of the few hyperparameters in the track update module. In Table C, we compare the performance of using different values for T_d, ranging from 1 to 6. With $T_d = 1$, unassociated tracks are immediately removed from the graph, which introduces considerable ID switches and fragmentation and finally leads to a significantly lower AMOTA. The AMOTA peaks at $T_d = \{3, 4\}$ and we use $T_d = 3$ as the default setting due to inference efficiency with fewer tracks. Higher T_d increases the robustness against occlusions or missed detections, but it increases the size of the graph, which makes the training more difficult. Therefore, when increasing T_d, we see a tendency of fewer ID switches and fragmentation, but the AMOTA decreases gradually due to more FPs and FNs.

Model architecture We ablate the model-related hyperparameters in Table D, which includes model dimension d, number of attention heads C, number of encoder layers L_e and decoder layers L_d. Compared to the default setting in the first row, fewer decoder layers causes a drop of AMOTA between 0.5%P and 0.6%P. With $L_e = 0$, the tracks from the last frames are directly associated with new detections without updating the track features in advance. This leads to an AMOTA decrease of 0.47%P. More encoder or decoder layers do not lead to further improvements. Similarly, varying model dimension and head numbers lead to an AMOTA decrease ranging between 0.2%P and 0.4%P.

Loss weights We evaluate the impact of the weighting of the loss terms $\mathcal{L}_\text{a}$ and $\mathcal{L}_\text{v}$ by varying the velocity loss weight λ_v in Tab. E. Using lower or higher weights than $\lambda_\text{v} = 1.0$ does not improve AMOTA or AMOTP.

Label assignment As shown in Section 3.3, we use Hungarian Matching (HM) with 3D IoU as matching cost to assign object IDs for detections. However, different ways of label assignment are possible, *e.g*. Greedy Matching (GM), or using other matching costs, *e.g*. center distance. We show a comparison in Table F. Compared to Hungarian Matching, using Greedy Matching decreases the performance slightly. This observation is different than the track-detection data association that we evaluated in Table 5, where greedy matching performed better. We argue that the label assignment requires a geometric matching cost like 3D IoU or center distance whereas the data association uses the estimated affinity, thus different matching methods work best. Results of 3D IoU is better than center distance since it measures the box similarity and not only the location, which results in a preciser lable assignment.

Other factors We evaluate four different variants of our approach: (1) *data augmentation:* the model is trained with random detection dropout and box jitter; (2) *enc. fully-connected:* we use a standard transformer encoder which utilizes a fully-connected graph; (3) *w/o velo. estimation:* we remove the velocity estimation of our model, whereas the association graph is built using the velocity from the detector; (4) *cross entropy:* we replace the focal loss with a binary cross entropy loss.

As can be seen in Table G, using data augmentation even results in a performance decrease, which again verifies the generalization ability due to our online training strategy. A standard transformer encoder leads to an AMOTA decrease of 0.37%P, which shows the benefit of a sparse graph in the encoder. *w/o velo. estimation* achieves an AMOTA of 0.7084. This indicates an accurate enough velocity estimation of the detector for graph building, but our approach estimates velocity more accurately using temporal information of tracked objects. Cross entropy loss results in an AMOTA decrease of 0.58%P and this shows the effectiveness of the focal loss for the imbalanced data in our framework.

λ_v	0.2	0.6	**1.0**	2.0	5.0
AMOTA↑	0.7078	0.7100	**0.7121**	0.7111	0.7106
AMOTP↓	0.5199	0.5217	**0.5149**	0.5317	0.5243

Table E. Ablation study on velocity loss weight λ_{L1}.

Matching	Cost	AMOTA↑	AMOTP↓	MOTA↑	IDS↓	FRAG↓
HM	3D IoU	**0.7121**	**0.5149**	**0.6071**	341	436
GM	3D IoU	0.7103	0.5244	0.6065	363	429
HM	center dist.	0.7077	0.5155	0.6045	**331**	410
GM	center dist.	0.7059	0.5190	0.6000	354	**401**

Table F. Ablation study on matching variants for label generation.

Variant	AMOTA↑	AMOTP↓	MOTA↑	IDS↓	FRAG↓
data augmentation	0.7092	0.5266	**0.6076**	**334**	440
enc. fully-connected	0.7084	0.5209	0.6016	378	439
w/o velo. estimation	0.7084	0.5299	0.6121	389	**430**
cross entropy	0.7063	0.5345	0.6059	369	441
3DMOTFormer	**0.7121**	**0.5149**	0.6071	341	436

Table G. Ablation study on other factors.

B.4. Detailed Metrics of our Test Submissions

We show the diagrams of different metrics over recalls of our submitted tracking results on the test split. Figure C shows the diagrams of the submission with CenterPoint detections, Figure D with BEVFusion detections. We refer to the nuScenes tracking benchmark website[1] for a comparison with other test submissions and a detailed interpretation

[1]benchmark url: `https://www.nuscenes.org/tracking?externalData=all&mapData=all&modalities=Any`

of all metrics.

C. Visualization

We show visualizations of our tracking results (left side) on the nuScenes validation set and compare it with ground truth (right side) in Figure E and Figure F. We use a unique color to represent a track ID. We use arrows to show the velocities of moving objects, where objects with velocity < 0.2 m/s are considered as stationary and no arrows are shown. Figure E shows a scenario where the ego vehicle is waiting at a crossing. We can see a consistent tracking of both moving and static objects as well as an accurate velocity estimation of moving vehicles. However, an inaccurate orientation of a car in front (frame 12, 16 and 20) or false positives (frame 12) from the object detector can not be corrected by the tracker. Figure E shows a scenario where the ego vehicle is moving on a crowded street. Similarly, we see a consistent tracking, even though cars that are close to each other and extremely small objects such as pedestrians.

References

[1] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089, 2022.

[3] Yukang Chen, Yanwei Li, X. Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5418–5427, 2022.

[4] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Yang Jiao, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Msmdfusion: A gated multi-scale lidar-camera fusion framework with multi-depth seeds for 3d object detection. *ArXiv*, abs/2209.03102, 2022.

[6] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *ArXiv*, abs/2206.00630, 2022.

[7] Mingchao Liang and Florian Meyer. Neural enhanced belief propagation for data association in multiobject tracking. *25th International Conference on Information Fusion (FUSION)*, pages 1–7, 2022.

[8] Jianan Liu, Liping Bai, Yuxuan Xia, Tao Huang, and Bing Zhu. Gnn-pmb: A simple but effective online 3d multi-object tracker without bells and whistles. *ArXiv*, abs/2206.10255, 2022.

[9] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *ArXiv*, abs/2205.13542, 2022.

[10] Akshay Rangesh, Pranav Maheshwari, Mez Gebre, Siddhesh Mhatre, Vahid Reza Ramezani, and Mohan Manubhai Trivedi. Trackmpnn: A message passing graph neural architecture for multi-object tracking. *ArXiv*, abs/2101.04206, 2021.

[11] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[12] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

[13] Li Wang, Xinyu Newman Zhang, Wenyuan Qin, Xiaoyu Li, Lei Yang, Zhiwei Li, Lei Zhu, Hong Wang, Jun Li, and Hua Liu. Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-lidar fusion. *ArXiv*, abs/2209.02540, 2022.

[14] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, 2020.

[15] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11779–11788, 2021.

[16] Jan-Nico Zaech, Alexander Liniger, Dengxin Dai, Martin Danelljan, and Luc Van Gool. Learnable online graph representations for 3d multi-object tracking. *IEEE Robotics and Automation Letters*, 7(2):5103–5110, 2022.

[17] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *ArXiv*, abs/1908.09492, 2019.

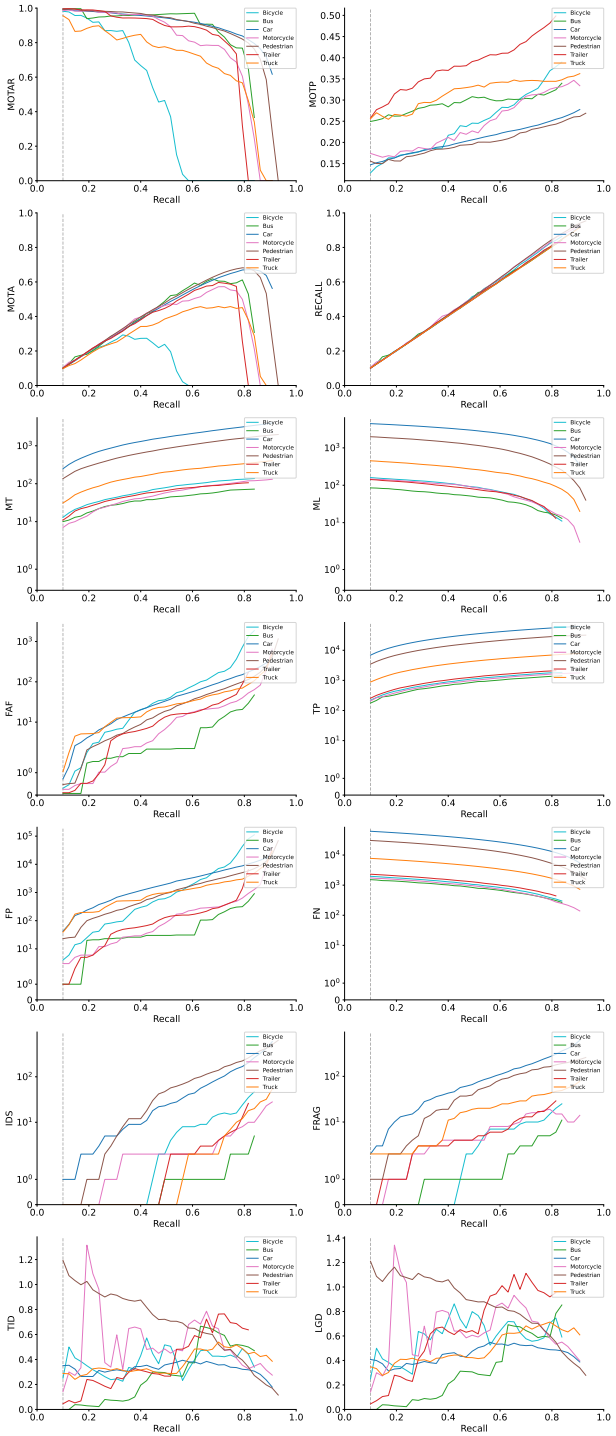

Figure C. Test results using CenterPoint detections.

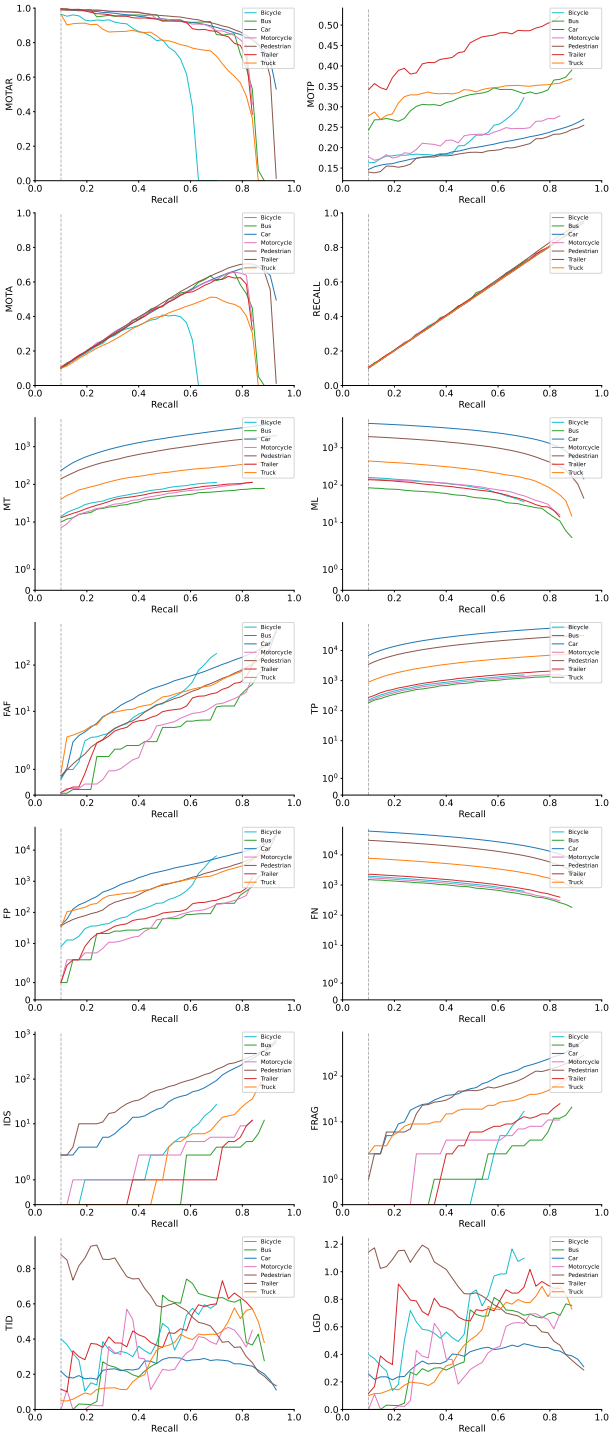

Figure D. Test results using BEVFusion detections.

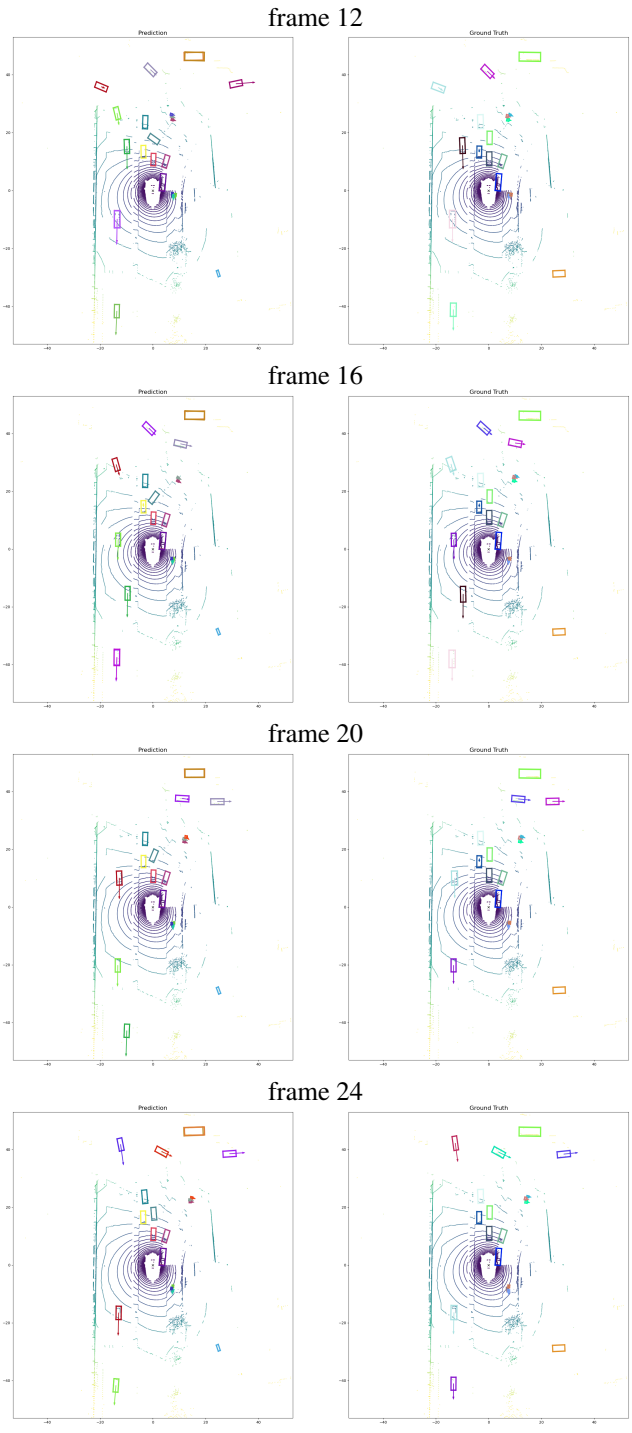

Figure E. Waiting at intersection.

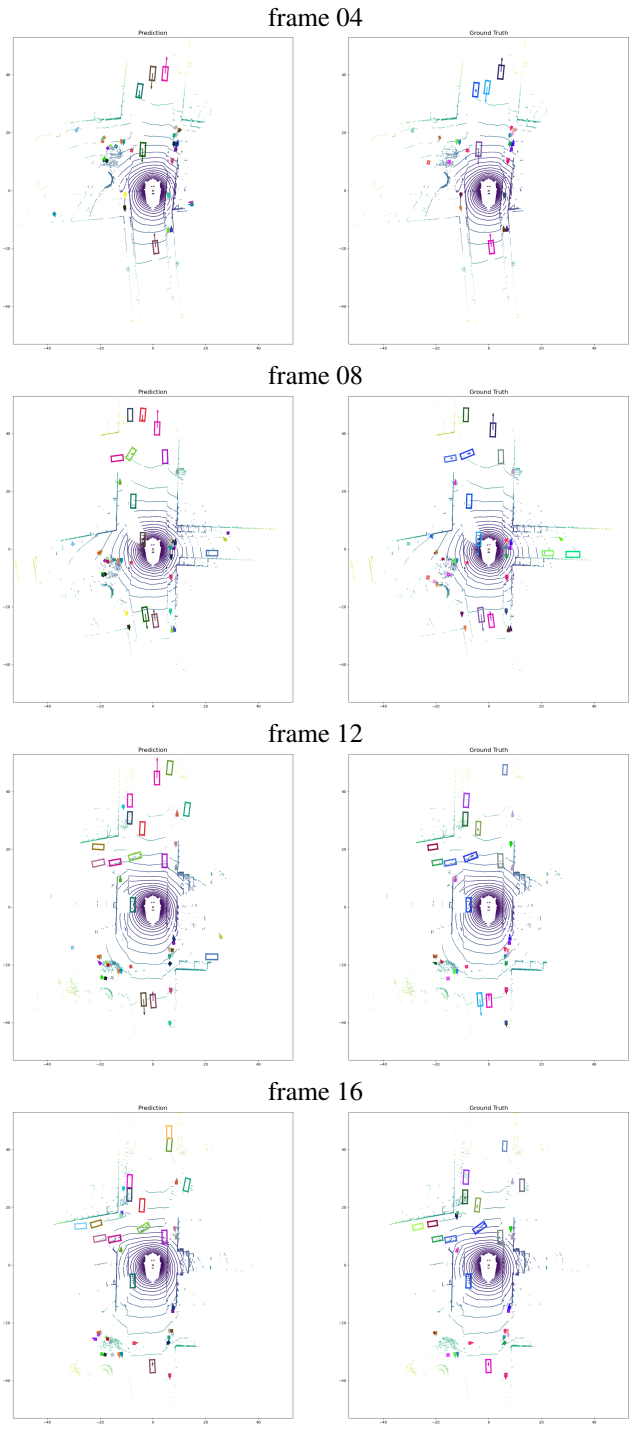

Figure F. Driving on a street with many traffic participants.