Technical Report for Argoverse Challenges on Unified Sensor-based Detection, Tracking, and Forecasting

Zhepeng Wang[1], Feng Chen[1], Kanokphan Lertniphonphan[1], Siwei Chen[2*],
Jinyao Bao[3*], Pengfei Zheng[4*], Jinbao Zhang[5*], Kaer Huang[1], Tao Zhang[2]
[1]Lenovo Research,
[2]Tsinghua University, [3]Liaoning Petrochemical University,
[4]University of Science and Technology Beijing, [5]University of Science and Technology of China
{wangzpb, chenfeng13, klertniphonp}@lenovo.com

Abstract

This report presents our Le3DE2E solution for unified sensor-based detection, tracking, and forecasting in Argoverse Challenges at CVPR 2023 Workshop on Autonomous Driving (WAD). We propose a unified network that incorporates three tasks including detection, tracking, and forecasting. This solution adopts a strong Bird's Eye View (BEV) encoder with spatial and temporal fusion and generates unified representations for multi-tasks. The solution was tested on the Argoverse 2 sensor dataset [8] to evaluate the detection, tracking, and forecasting of 26 object categories. We achieve 1st place in Detection, Tracking, and Forecasting on the E2E Forecasting track in Argoverse Challenges at CVPR 2023 WAD.

1. Introduction

The challenge focuses on evaluating end-to-end perception tasks on detection, tracking, and multi-agent forecasting on Argoverse 2 sensor dataset. The dataset provides track annotations for 26 object categories. For testing, our algorithm needs to be able to detect objects in the current frame and forecast trajectories for the next 3 seconds. The end-to-end task is different from the motion forecasting task since the tracking ground truths are not provided.

2. Method

Motivated by UniAD [2], we propose an end-to-end framework for detection, tracking, and forecasting. We fuse the BEV features from LiDAR and multi-view cameras as a unified representation for all three downstream tasks. HD map is encoded as vectors to help with motion forecasting. The system overview is shown in figure 1.

*Work done as an intern at Lenovo Research.

2.1. BEV Feature

For LiDAR point cloud, we employ a LIDAR BEV encoder based on SECOND [10] to generate LIDAR BEV features B_l. For multi-view images, we adopt a spatio-temporal transformer based on BEVFormer [3] to generate BEV features from multi-view cameras B_c. The camera BEV branch has two modules: the backbone network and the BEV encoder.

The BEV features from LiDAR B_l and multi-view cameras B_c are fused into one BEV feature by a spatial encoder following BEVFusion [4]. The spatial encoder concatenates B_l and B_c and then reduces the feature dimensions through a convolution layer. After the spatial fusion, historical BEV features are fused with the current frame by the spatial-temporal transformer in BEVformer [3]. The spatial-temporal fused BEV feature is used as a 3D representation and input to downstream heads.

2.2. Detector

The detector is based on Deformable DETR [9]. The temporal fused BEV features are fed into the decoder as object queries. The Deformable DETR head is used to predict 3D bounding boxes and velocity without Non-Maximum Suppression (NMS). 3D box regression is supervised by using L1 loss. The detection queries capture the agent characteristic by attending to the BEV features.

2.3. Tracker

The tracking is initialized by object queries from the detector as the tracking candidate at each frame. While track queries, which are based on MOTR [11], are used to associate track queries in the current frame and the previous frame. The track queries which are matched with the history frame aggregate temporal information in a self-attention module until the agent disappears in a certain time period.

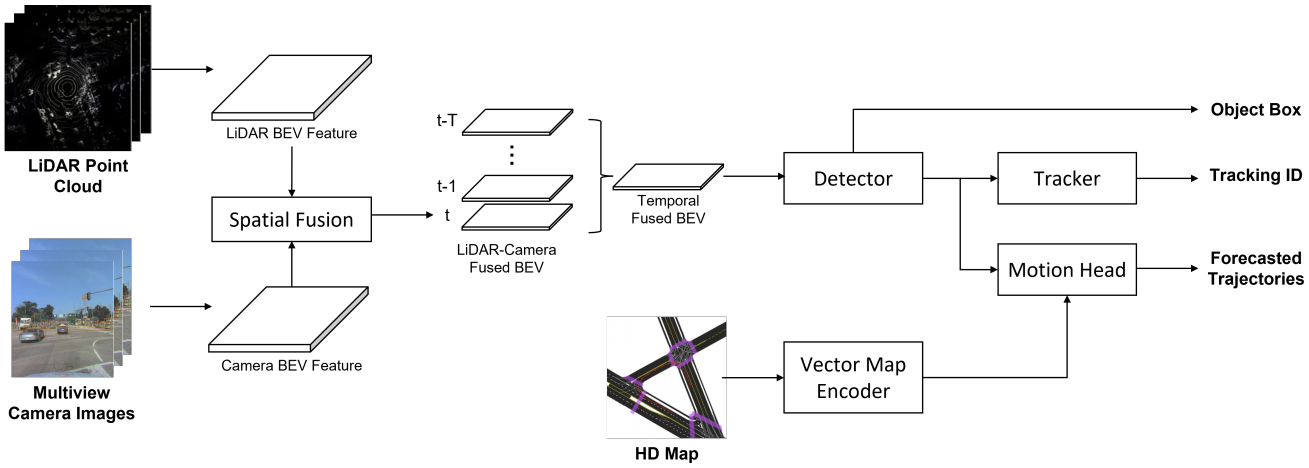

Figure 1. **System overview.** First, we extract BEV features from LiDAR point cloud and camera images separately. The LiDAR point clouds of the current frame are voxelized and encoded to the BEV feature map by **LiDAR backbone**. Image features are extracted from synchronized multi-view cameras by an **image backbone** and are encoded to a camera BEV feature by a transformer-based **BEV encoder**. Second, **spatial-fusion** module fuses LiDAR and Camera BEV into a unified BEV representation. The historical frame BEV feature maps are fused with the current frame by using a **temporal encoder**. Third, spatial-temporal fused BEV is fed into **Detector** which generates detection bounding boxes. **Tracker** utilizes object queries from the detector to associate track queries between frames. Also, **Motion Head** forecasts the future trajectories for each agent from Detector. In addition, **HD Map** is encoded to vectors and interacts with agents to help with motion forecasting.

Team	mAP_F(↑)	ADE(↓)	FDE(↓)
dgist−cvlab	45.83	4.09	4.53
Host_4626_Team	14.51	5.10	7.32
Le3DE2E (Ours)	**46.70**	**3.22**	**3.76**

Table 1. Forecasting Leaderboard on End-to-End Forecasting Challenge

2.4. VecterMap Encoder

HD maps are typically represented by vectorized spatial coordinates. To encode the information of lanes and pedestrian crossings, we adopt a vectorized encoding method called VectorNet [1], which operates on the vectorized HD maps to avoid lossy rendering and computationally intensive ConvNet encoding steps. The map elements are encoded by cross-attention layers and represented as map queries. We generate the position encoding with the center of each vector. The map queries and the position encoding are forwarded to Motion Head to help with motion forecasting.

2.5. Motion Head

The motion head takes in the agent's information from the Detection and map information from Vector Map Encoder. It then predicts the future trajectories for agents. The transformer structure has been proven to be effective in motion forecasting tasks in recent years. Thus we choose MotionFormer from UniAD [2] as a motion baseline. The motion head is a 3-layer transformer decoder and has BEV queries generated by the BEV encoder, agent queries generated by Detection, and map queries encoded by VectorMap as input. They interact with motion queries and help motion forecasting.

2.6. Test Time Augmentation and Ensemble

During inference, we apply Test Time Augmentation (TTA) to further improve the performance. Also, we use NMS to merge the results of augmented input.

We use the Weighted Box Fusion (WBF) [7] to ensemble multiple models with different training settings to improve detection and forecasting prediction accuracy. For E2E forecasting, we use a two-step ensemble procedure to ensemble not only the detection bounding boxes but also future trajectories. In step 1, we cluster the detection bounding boxes according to the intersection-over-union (IoU). In step 2, we cluster forecasting trajectories with L2 distances and adaptively adjust the threshold based on the speed of instances.

Team	HOTA(↑)	AMOTA(↑)	MOTA(↑)
AIDrive (v0)	44.36	17.47	32.61
dgist−cvlab	41.49	7.88	17.97
Host_4626_Team	39.98	7.10	16.21
Le3DE2E (Ours)	**56.19**	**19.53**	**39.34**

Table 2. Tracking Leaderboard on End-to-End Forecasting Challenge

Team	mCDS(↑)	mAP(↑)	mATE(↓)	mASE(↓)	mAOE(↓)
BEV (BEVFusion)	0.37	0.46	0.40	**0.30**	0.50
Detectors	0.34	0.42	**0.39**	**0.30**	0.50
AIDrive (Lv0)	0.27	0.35	0.45	0.33	0.84
Match (lt3d)	0.21	0.26	0.43	0.33	0.50
Host_75088_Team (CenterPoint)	0.14	0.18	0.49	0.34	0.72
zgzxy001	0.12	0.15	0.45	0.34	0.65
Le3DE2E (Ours)	**0.39**	**0.48**	0.41	0.31	**0.47**

Table 3. 3D Object Detection Leaderboard

3. Experiments

3.1. Dataset

The competition used the Argoverse 2 Sensor Dataset, which consisted of 1000 scenes (750 for training, 150 for validation, and 150 for testing) with a total of 4.2 hours of driving data. The total dataset is extracted in the form of 1 TB of data. Each vehicle log has a duration of approximately 15 seconds and includes an average of approximately 150 LiDAR scans with 10 FPS LiDAR frames. The dataset has 7 surrounding cameras with 20 FPS. For the E2E Forecasting track, 1 keyframe is sampled in 2Hz from the training, validation, and testing sets.

3.2. Evaluation Metrics

Detection. Argoverse [8] proposes a new metric Composite Detection Score (CDS) which simultaneously measures precision, recall, object extent, translation error, and orientation. The mean metrics are computed as an average of 26 different object categories.

Tracking. HOTA [5] is the key metric for the challenge, while AMOTA and MOTA are also important metrics for reference. HOTA explicitly balances the effect of performing accurate detection, association, and localization into a single unified metric. MOTA combines false positives, missed targets, and identifies switches to compute the tracking accuracy. AMOTA, similar to MOTA, is averaged over all recall thresholds to consider the confidence of predicted tracks.

Forecasting. The main evaluation metric is Forecasting mAP (mAP_F) [6], ADE, and FDE which are averaged over static, and non-linearly moving cohorts. mAP_F is the key metric for the challenge, which defines a true positive when there is a positive match in both the current timestamp T and the future (final) at T + N time slot. ADE is an average L2 distance between the best-forecasted trajectory and the ground truth. FDE is an L2 distance between the endpoint of the best-forecasted trajectory and the ground truth.

3.3. Implementation Details

Architecture details. In the LiDAR branch, the voxel size of LiDAR encoder is (0.075m, 0.075m, 0.2m) and the point clouds range is limited to [-54m, 54m] x [-54m, 54m] x [-3m, 3m] to adapt the max range of E2E forecasting. In LiDAR backbone, we down-sampled voxels to 1/8. For the camera branch, we crop and resize camera images to 976x1440 to save GPU memory. we use the ResNet-101 as a backbone and a 4-layer FPN as a neck to extract features from multi-view cameras.

Training. We apply a 2-step training procedure. First, we train the detector for 6 epochs. Then, we train the whole end-to-end network to optimize the detector, tracker, and motion head simultaneously for 20 epochs. We freeze the LiDAR and image backbones in step 2 to save GPU memory.

The models are trained by AdamW optimizer, with a learning rate of 2e-4, a weight decay of 0.01, and a total batch size of 8 on 8 V100 GPUs. We use cosine annealing to decay the learning rate. We applied CBGS (Class-balanced Grouping and Sampling) [12] to get the expert model for balanced data distribution.

TTA and Ensemble. For every model, we employ global scaling with [0.95, 1, 1.05] and flipping with respect to the xz-plane and yz-plane for TTA. We trained multiple mod-

els with three voxel sizes of [0.05m, 0.075m, 0.1m], with or without CBGS augmentation and with or without camera input. Totally we ensemble 8 models to generate final results.

3.4. Final Results

We test our solution on 3 sub-challenges of Detection, Tracking, and Forecasting in the E2E Forecasting track of the Argoverse Challenge. Table 1 is the final leaderboard of Forecasting and shows that our solution achieves 46.70 mAP_F and ranks 1st place in Forecasting. Table 2 is the final leaderboard of Tracking and shows that our solution achieves 56.19 HOTA and ranks 1st place in Tracking. Table 3 is the final leaderboard of 3D Object Detection and shows that our solution achieves 0.34 CDS and ranks 1st place in Detection.

4. Conclusion

We devise a unified framework of detection, tracking, and forecasting for Autonomous Driving. Our solution ranks 1st place in Detection, Tracking, and Forecasting of the E2E Forecasting track in Argoverse Challenges at CVPR 2023 WAD.

References

[1] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11522–11530, 2020. 2

[2] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[3] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. 1

[4] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1

[5] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and B. Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129:548 – 578, 2020. 3

[6] Neehar Peri, Jonathon Luiten, Mengtian Li, Aljovsa Ovsep, Laura Leal-Taix'e, and Deva Ramanan. Forecasting from lidar via future object detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17181–17190, 2022. 3

[7] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, pages 1–6, 2021. 2

[8] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 1, 3

[9] Lewei Lu Bin Li Xiaogang Wang Jifeng Dai Xizhou Zhu, Weijie Su. Deformable detr: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations*, 2021. 1

[10] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1

[11] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[12] 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. 3