Valeo4Cast: A Modular Approach to End-to-End Forecasting

Yihong Xu[*1] Éloi Zablocki[*1] Alexandre Boulch[*1] Gilles Puy[1] Mickael Chen[1]
Florent Bartoccioni[1] Nermin Samet[1] Oriane Siméoni[1] Spyros Gidaris[1]
Tuan-Hung Vu[1] Andrei Bursuc[1] Eduardo Valle[1] Renaud Marlet[1,2]
Matthieu Cord[1,3]

[1]Valeo.ai, Paris, France
[2]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France
[3]Sorbonne Université, Paris, France

Abstract

Motion forecasting is crucial in autonomous driving systems to anticipate the future trajectories of surrounding agents such as pedestrians, vehicles, and traffic signals. In end-to-end forecasting, the model must jointly detect from sensor data (cameras or LiDARs) the position and past trajectories of the different elements of the scene and predict their future location. We depart from the current trend of tackling this task via end-to-end training from perception to forecasting and we use a modular approach instead. Following a recent study [27], we individually build and train detection, tracking, and forecasting modules. We then only use consecutive finetuning steps to integrate the modules better and alleviate compounding errors. Our study reveals that this simple yet effective approach significantly improves performance on the end-to-end forecasting benchmark. Consequently, our solution ranks first in the Argoverse 2 end-to-end Forecasting Challenge held at CVPR 2024 Workshop on Autonomous Driving (WAD), with 63.82 mAP$_f$. We surpass forecasting results by +17.1 points over last year's winner and by +13.3 points over this year's runner-up. This remarkable performance in forecasting can be explained by our modular paradigm, which integrates finetuning strategies and significantly outperforms the end-to-end-trained counterparts.

1. Introduction

Autonomous and assisted driving requires accurate understanding of the scene surrounding the vehicle. In particular, detecting [4, 9, 13, 16, 17], tracking [23, 26, 28] and forecasting [1, 7, 15, 20, 21] the behavior of the agents in the scene, agents which might be static or dynamic, is needed to plan the trajectory of the ego vehicle.

In the recent years, these tasks have been tackled conjointly in pipelines that performs detection, tracking, and forecasting, as part of the same integrated network trained end-to-end, with great success [18, 22]. We name such methods *end-to-end-trained*. Notably, VIP3D [8] introduced an end-to-end training pipeline from detection, tracking and mapping to forecasting, and UniAD [10] enhanced the forecasting performance and extended the pipeline to planning.

In spite of these achievements, a recent study [27] reveals that current state-of-the-art end-to-end-trained approaches [8, 10] are not without issues. Crucially, it shows that a simple baseline putting together independently trained detection, tracking and forecasting modules outperforms end-to-end training in the final forecasting task. However, because the modules of this simple pipeline are trained in isolation using curated data, the errors of the early modules are not compensated downstream, which can lead to dramatic compounding errors at the end of the pipeline.

Following the findings of [27], we focus on advancing the forecasting performance and build in this work a modular approach (illustrated in Fig. 1). In particular, we use BEVFusion [13] for detection, AB3DMOT [23] for tracking, and MTR [21] for forecasting, and work on integrating all three into an end-to-end forecasting pipeline. We start by pretraining the detection and forecasting modules individually with data curated for their respective tasks, the tracker having no trainable parameters. To mitigate the compounding errors, we then finetune the forecasting module, using as input the outputs of the previous blocks. We observe in this challenge the importance of this adaptation step which drastically boost performance. Overall, this modular approach has the benefit to (1) require limited resources as each functional block is trained separately — which is not the case

[*]Core contributors.
Correspondence to yihong.xu@valeo.com

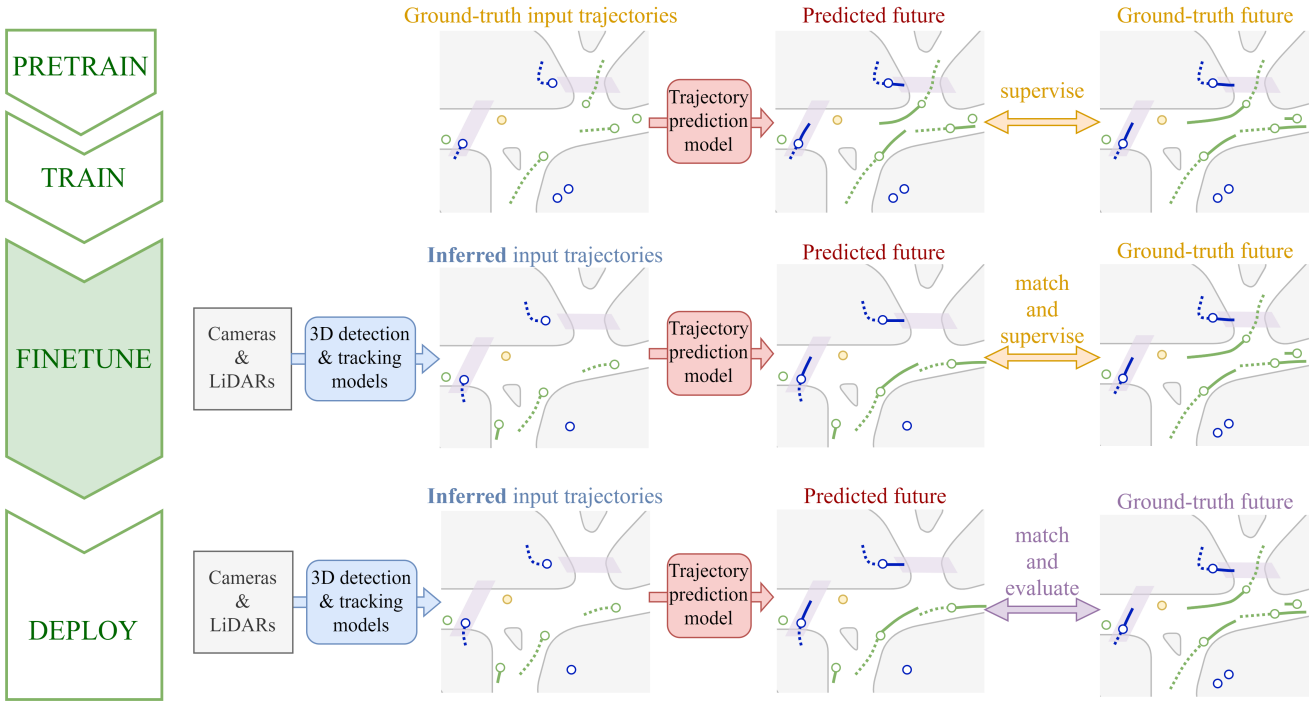

Figure 1. **Overview of the modular approach of *Valeo4Cast***. Conventional motion forecasting benchmarks provide curated annotations of the past trajectories. Differently in this 'end-to-end forecasting' challenge, we opt for a modular approach where the past trajectories are predicted by the detection and tracking modules. The predicted results contain imperfections such as FPs, FNs, IDS and localization errors, which hinder forecasting. For this reason, training only on curated data is not sufficient (top). We thus propose a finetuning strategy where we match the predicted results and ground-truth annotations. We finetune the model on the matched pairs (middle) and it shows significant improvements once the model is deployed in real-world end-to-end forecasting (bottom). The ego car, vehicles, and pedestrians are expressed in different colors. The past trajectories are shown with dotted lines and the future ones with plain lines. 'Pretrain' refers to the pretraining on the UniTraj [6] framework, and 'Train' to the step where we keep training on the curated Argoverse2-Sensor dataset.

for end-to-end training pipelines. It also (2) greatly improves the performances of the downstream blocks and (3) opens the possibility of updating/upgrading a block without retraining all the upstream components. The proposed pipeline is evaluated on the Argoverse Sensor forecasting benchmark [24] in the end-to-end forecasting paradigm.

We summarize here the main findings of the study which are later discussed:

- **Pretraining** it on a large dataset helps better initialize the model;
- **Finetuning** the forecasting module on predicted detection and tracking inputs helps to take into account the errors of the previous detection and tracking blocks;
- **Post-processing** is then needed to ensure a valid trajectory for static objects.

This report is organized as follows. We summarize in Sec. 2.1 the used perception models that generate detection and tracking results. We detail in Sec. 2.2 the forecasting model and our pretraining, finetuning and post-processing strategies. In Sec. 3, we present our results and ablations on the Argoverse 2 Sensor forecasting benchmark.

2. Our approach

We use in this work the modular pipeline represented in Fig. 1. It consists of three independent modules for detection, tracking and forecasting. We describe the perception modules and the forecasting method in the following subsections. We also provide implementation details for each blocks.

2.1. Perception

We first discuss our detection module (Sec. 2.1.1), followed by our tracking block (Sec. 2.1.2).

2.1.1 Detection

Detection backbone. We use the LiDAR-only detector of BEVFusion [13]. It is composed of a sparse convolutional encoder which produces BEV features followed by a convolutional BEV backbone and multiple task-specific heads. These heads predict the box center, dimension and orientation of all objects.

Implementation details. We use voxel size of 0.075 m and a BEV cell size of 0.6 m. As an addition to the current frame, we load the 5 previous lidar sweeps to densify the point cloud. We train three different models: one detector working on a range of up to 54 m, another working a range of up to 75 m range, and a third one working on a range of up to 54 m but where the input features are enriched with ScaLR [19] point features. These detectors are trained using up to 8 NVIDIA RTX 2080 Ti.

Ensembling We then use a very simple heuristic to combine the detection of these three models. For each timestep and each predicted class category, given a reference detection, we combine all detections with centers at a distance less than r from the reference center. We proceed greedily and consider the boxes by decreasing confidence order, removing the merged detection from the pool of detection to be processed. The merging then consists in a simple weighted average, with the weights based on the boxes' center confidence, dimensions and orientations.

Going further. As our focus here was on forecasting, we used a simple LiDAR-based detector and trained it on only 10% of the train set, with a the limited range of 75m. Perspectives for this work include training the model on all annotations, and also leveraging the cameras and the map information available in the dataset, which could help produce stronger perception results and therefore improve the downstream forecasting.

2.1.2 Tracking

Tracking algorithm. We adopt the simple and effective training-free tracking algorithm of AB3DMOT [23] in order to associate the detection results obtained in different frames. Precisely, we perform per-class tracking by running a one-to-one matching algorithm based on the track (object being tracked)-detection distances. The distance is determined by the 3D intersection-over-union (IoU) between tracks and detection at each time step, and the matching threshold is set to 0.1 for all classes. Moreover, a track may be temporarily lost due to occlusions. In this case, this track is put into 'inactive' mode and its position is updated with a Kalman filter until it is matched to a detection. The 'inactive' mode can only last for 3 frames, beyond which the track is terminated.

Linear interpolation of tracks. When using the tracking algorithm presented above, we observe that the trajectories can be fragmented into sub-trajectories. In order to mitigate the problem, inspired by ByteTrack [29], we linearly interpolate between the fragment trajectories using a constant velocity calculated using the object locations at current and past timesteps. This interpolation improves HOTA by 0.45 points. [14]. The overall tracking approach forgoes any training.

2.2. Forecasting

To forecast the different agents, we use the MTR [21] forecasting model, which won the 2022 Waymo forecasting challenge. The architecture is transformer-based and have 60M trainable parameters. It jointly learns the agent's global intentions as well as their local movements.

Pretraining. We pretrain MTR on 1300+ hours of vehicle trajectories with the UniTraj framework that gathers nuScenes [3], Argoverse2-Motion [24], and Waymo Open Motion Dataset (WOMD) [5]. We then further train MTR on the curated Argoverse2-Sensor dataset.

Finetuning. At inference time, the forecasting model is applied to the outputs of the perceptions modules (detection and tracking), which are imperfect, with misdetections, hallucinations, tracking issues and localization errors. To deal with such mistakes, we finetune MTR on such imperfect data. In practice, given a track predicted by our upstream perception modules, we match it with the ground-truth trajectory provided in Argoverse 2 training annotations in order to get the future ground-truth to predict.

Since the detections are not filtered by any detection score, they are typically redundant and cannot be match one-to-one with the ground truth. To provide rich supervision for the forecasting, we perform a many (predictions)-to-one (ground truth) matching based on the Euclidean distance between the past trajectories of the tracks and the ground-truth annotations at each inference time step. In the distance calculation, we only consider the past timestamps where the prediction and ground truth are both available. For the matched track and ground truth, we train MTR to predict the corresponding ground-truth future trajectory. We finetune the model only on the trainset for 15 epochs and choose the checkpoint with the lowest brier-FDE on the validation set. Given our results, discussed in Tab. 2, this finetuning strategy appears crucial for the forecasting performance.

Post-processing. Because the pretraining was performed on standard (non end-to-end) forecasting data that do not contain static trajectories, our model tends to avoid predicting static motion. This can be easily solved using a post-processing step. During inference, we conduct the following steps:

- Static trajectories are the most prevalent in the dataset, however the forecasting module is trained mainly on *moving objects*. We therefore insert a static trajectory in the predictions: we replace the least probable mode with a stationary future, and assign it a score of 1.
- For object classes that are always static[1], we predict a single static trajectory with a probability of 1. This only marginally impacts the scores.

[1]These include 'bollard', 'construction cone', 'construction barrel', 'sign', 'mobile pedestrian crossing sign', and 'message board trailer'

		Forecasting			Detection			Tracking
		mAP_f (↑)	ADE (↓)	FDE (↓)	Inputs	Training range	mCDS (↑)	HOTA (↑)
2023	CenterPoint [28]	-	-	-	L	150	14	-
	BEVFusion [13]	-	-	-	L+C	150	37	-
	Anony_3D (v0) [11]	-	-	-	L	150	31	44.36
	Host_4626_Team [17]	14.51	5.10	7.32	-	-	-	39.98
	Dgist-cvlab [25]	42.91	4.11	4.59	L+C	150	34	41.49
	Le3DE2E [22]	46.70	3.22	3.76	L+C	150	39	56.19
2024	Dgist-cvlab	45.83	4.09	4.53	L+C	150	34	41.49
	Le3DE2E	50.53	4.07	4.60	L+C	150	**43**	**64.60**
	Valeo4Cast	**63.82**	**2.14**	**2.43**	L	75	31	61.28

Table 1. **Leaderboard results.** Test set results of the Argoverse 2 end-to-end forecasting leaderboard, with three sub-challenges: forecasting, detection and tracking. We distinguish the detectors by their input modality with 'L' for LiDAR and 'C' for Camera. For all methods, the *evaluation* range is fixed to 150 m for detection, and 50 m for tracking and forecasting. Our modular strategy significantly outperforms others in the forecasting challenge (by more than 13 mAP_f pts).

As the mAP_f computation is done at the level of trajectory type, and then averaged, we find that both steps significantly boosts the score, as seen in Tab. 2 ('w/o post-processing' line). The post-processing currently requires no training, but could be improved by predicting if a future trajectory is likely to be static or not.

Implementation details We use the implementation of UniTraj [6]. We use the past 2 seconds (or until the beginning of the sequence), even though the challenge allows to use all past frames. The finetuning takes around 12 hrs on a single node with 8×A100 GPUs.

3. Experiments and Results

3.1. Dataset

We train and evaluate our method on the Argoverse 2 Sensor Dataset [24]. It contains 4.2 hours of driving data, split into 1000 scenes (750/150/150 for train/val/test). Each scene lasts about 15 seconds with sensor data and annotations being provided at a 10Hz sampling rate. The input data include images captured from a 360°-rig of 7 cameras, LiDAR scans, and HD-maps of the environment that include information about lines, drivable area and cross-walks. Annotations are provided for 26 different semantic classes, including common ones like vehicle and bus and less frequent ones like wheelchair, construction cone, dog, message board trailer.

3.2. Evaluation metrics

Detection. The detection performance is measured with the mCDS metric. The Composite Detection Score (CDS) gathers in a single score the detection precision, recall, and the quality of the estimation of the object extent, positioning and orientation. This metric is averaged over all object classes to form mCDS. The evaluation range is 150m around the ego-vehicle.

Tracking. HOTA [14] is the main metric used for evaluating the tracking performance. It breaks down the detection and association evaluation by calculating separately the False Positives (FP), False Negatives (FN) and True Positives (TP). It alleviates the issue of overly emphasizing detection performance in the multi-object tracking accuracy (MOTA) [2] metric, which calculates the sum of FP, FN and IDentity Switch (IDS) over the total number of ground-truth objects. MOTA reflects the overall performance of a multi-object tracker with a focus on the detection. Since MOTA only considers the tracking result after thresholding, a variant of MOTA – AMOTA averaging all recall thresholds [23]. These metrics are averaged over all object classes. The evaluation range is 50m around the ego-vehicle.

Forecasting. The challenge's primary metric is the mean Forecasting Average Precision (mAP_f) [18]. This metric shares the same formulation as detection AP [12]. However, in the case of mAP_f, a true positive is defined for trajectories that match at the current time step (i.e., successfully detected agents) and the final time step (i.e., successfully forecasted agents). For agents that are successfully detected, the other considered metrics are Average Displacement Error (ADE) and Final Displacement Error (FDE), where ADE measures the average Euclidean distance between the predicted and ground-truth positions over the future time horizon, while FDE measures the distance at the final time step. We stress that ADE and FDE can only compute a distance when a ground truth and a predicted detection have been matched and therefore do not account for miss-detected or hallucinated agents. In fact, not detecting the more difficult agents can improve ADE and FDE errors. Therefore, these metrics should be used carefully in the context of end-to-

Figure 2. **Qualitative visualizations** of randomly sampled frames of the validation set. The ego car, vehicles, wheeled devices, pedestrians and ground-truth annotations are expressed in different colors. The numbers represent the detection scores.

end forecasting benchmarks to avoid erroneous interpretations. The evaluation range is 50m around the ego-vehicle.

3.3. Results and discussion

Before discussing the quantitative results, we visualize examples of forecasting obtained on randomly selected frames in Fig. 2.

Leaderboard results. We provide the learderboard results from Argoverse 2 end-to-end forecasting challenge in Tab. 1. The proposed modular solution *Valeo4Cast* achieves strong performance on forecasting, outperforming the second-best solution by more than 13 points on mAP_f. This demonstrate the usefulness of our modular approach, which allows to easily transfer the strong pretrained forecasting model such as MTR to the end-to-end task via a suitable finetuning strategy. Interestingly, even though we depart from lower detection and tracking results compared to other current methods, our finetuned trajectory prediction model can overcome the difference and significantly outperform them.

Ablation study. From Tab. 2, we observe that finetuning MTR on the trainset predictions of the used detector and tracker on the Argoverse2-Sensor data is crucial for the forecasting performance. It improves drastically the mAP_f from 43.3 to 63.0. This is mainly because the vanilla MTR model has never been trained on predicted inputs. The finetuning can adapt the model not only to new trajectory types but also to the inaccuracies in predicted inputs.

	mAP_f (↑)	mAP_f (↑) non-linear	ADE (↓)	FDE (↓)
Valeo4Cast	**63.0**	67.5	**0.60**	0.96
w/o finetuning with perception inputs	43.3	33.6	1.97	3.24
w/o post-processing	54.6	33.2	0.61	**0.94**
w/o UniTraj pretraining	62.9	**68.3**	0.62	0.98

Table 2. **Ablation study of the forecasting module.** Scores are reported on the validation set. The evaluation is conducted in a 50m-range around the ego-car. mAP_f is the main metric of the challenge.

Surprisingly, after finetuning, we find that using the model pretrained on 1300+ hours of vehicles trajectories does not bring significant benefit compared to training from scratch (62.9 mAP_f). We believe that this is due to the difference in object classes and the data distribution between ground truth and predicted past trajectories.

And finally, compensating for the lack of static trajectories in pretraining, the post-processing effectively helps to improve the forecasting performance.

4. Conclusion

The modular pipeline *Valeo4Cast* ranks first in the AV2 E2E Forecasting challenge 2024 and outperforms other solutions by +13 mAP_f pts. This design allowed us to start from

a state-of-the-art motion forecasting model, that we integrate into an end-to-end pipeline by finetuning the forecasting module. We include a post-processing step to account for the absence of static objects in the conventional motion forecasting pretraining but which are important in the end-to-end benchmark. In this work, we confirm the findings of [27], verifying the superiority of modular approaches in the end-to-end forecasting task, and their capacity to handle detection and tracking imperfections.

The efficient nature of the end-to-end approaches is still appealing. In future work, we are interested in investigating how to better train the end-to-end approaches in order to achieve performances on-par with *Valeo4Cast*. Besides, future work may also consider more challenging settings in which the map information is not provided, at any stage, and has to be inferred in an online fashion, as the ego car drives and discovers its environment.

Acknowledgment

The project was partially funded by ANR grant MultiTrans (ANR-21-CE23-0032). This research received the support of EXA4MIND project, funded by a European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement N°101092944 views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. The project also receives compute resources from the petascale system Karolina, acquired as part of the EuroHPC. We thank the authors of UniTraj [6] for the release of the forecasting framework. We also thank the organizers of the challenge.

References

[1] Hedi Ben-Younes, Éloi Zablocki, Mickaël Chen, Patrick Pérez, and Matthieu Cord. Raising context awareness in motion forecasting. In *CVPRW*, 2022. 1

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008. 4

[3] 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 *CVPR*, 2020. 3

[4] Yilun Chen, Zhiding Yu, Yukang Chen, Shiyi Lan, Anima Anandkumar, Jiaya Jia, and José M. Álvarez. Focalformer3d : Focusing on hard instance for 3d object detection. In *ICCV*, 2023. 1

[5] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *ICCV*, 2021. 3

[6] Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud Ben Amor, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. *CoRR*, abs/2403.15098, 2024. 2, 4, 6

[7] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022. 1

[8] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*, 2023. 1

[9] Haotian Hu, Fanyi Wang, Jingwen Su, Yaonong Wang, Laifeng Hu, Weiye Fang, Jingwei Xu, and Zhiwang Zhang. Ea-lss: Edge-aware lift-splat-shot framework for 3d bev object detection. *arXiv preprint arXiv:2303.17895*, 2023. 1

[10] 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 *CVPR*, 2023. 1

[11] Jae-Keun Lee, Jin-Hee Lee, Joohyun Lee, Soon Kwon, and Heechul Jung. Re-VoxelDet: Rethinking neck and head architectures for high-performance voxel-based 3d detection. In *WACV*, 2024. 4

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4

[13] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 1, 2, 4

[14] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *IJCV*, 2021. 3, 4

[15] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *ICRA*, 2023. 1

[16] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, 2021. 1

[17] Neehar Peri, Achal Dave, Deva Ramanan, and Shu Kong. Towards long-tailed 3d detection. In *CoRL*, 2022. 1, 4

[18] Neehar Peri, Mengtian Li Jonathon Luiten, Aljoša Ošep, Laura Leal-Taixé, and Deva Ramanan. Forecasting from lidar via future object detection. In *CVPR*, 2022. 1, 4

[19] Gilles Puy, Spyros Gidaris, Alexandre Boulch, Oriane Siméoni, Corentin Sautier, Patrick Pérez, Andrei Bursuc, and Renaud Marlet. Three pillars improving vision foundation model distillation for lidar. In *CVPR*, 2024. 3

[20] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 1

[21] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *NeurIPS*, 2022. 1, 3

[22] Zhepeng Wang, Feng Chen, Kanokphan Lertniphonphan, Siwei Chen, Jinyao Bao, Pengfei Zheng, Jinbao Zhang, Kaer Huang, and Tao Zhang. Technical report for argoverse challenges on unified sensor-based detection, tracking, and forecasting. *CoRR*, abs/2311.15615, 2023. 1, 4

[23] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IEEE Int. Conf. Intell. Robots Syst.*, 2020. 1, 3, 4

[24] 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 *NeurIPS Track on Datasets and Benchmarks*, 2021. 2, 3, 4

[25] Jungwan Woo, Jaeyeul Kim, and Sunghoon Im. Motion forecasting via coordinate transformations and object trajectory modifications. 4

[26] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense representations for multiple-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 1

[27] Yihong Xu, Loïck Chambon, Éloi Zablocki, Mickaël Chen, Alexandre Alahi, Matthieu Cord, and Patrick Pérez. Towards motion forecasting with real-world perception inputs: Are end-to-end approaches competitive? In *ICRA*, 2024. 1, 6

[28] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 1, 4

[29] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022. 3