

Towards Motion Forecasting with Real-World Perception Inputs: Are End-to-End Approaches Competitive?

Yihong Xu, Loïck Chambon, Éloi Zablocki, Mickaël Chen, Alexandre Alahi, Matthieu Cord, Patrick Pérez

▶ **To cite this version:**

Yihong Xu, Loïck Chambon, Éloi Zablocki, Mickaël Chen, Alexandre Alahi, et al.. Towards Motion Forecasting with Real-World Perception Inputs: Are End-to-End Approaches Competitive?. 2024 IEEE International Conference on Robotics and Automation (ICRA), May 2024, Yokohama, Japan. pp.18428-18435, ⟨10.1109/ICRA57147.2024.10610201⟩. ⟨hal-05076997⟩

HAL Id: hal-05076997

https://hal.science/hal-05076997v1

Submitted on 21 May 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

arXiv:2306.09281v4 [cs.RO] 5 Mar 2024

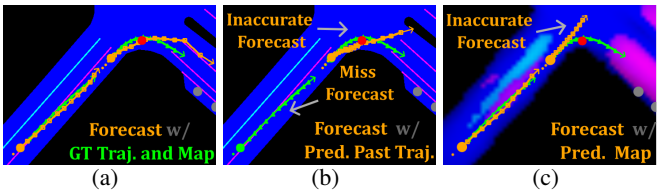
Towards Motion Forecasting with Real-World Perception Inputs: Are End-to-End Approaches Competitive?

Yihong Xu[1] Loïck Chambon[1,2] Éloi Zablocki[1] Mickaël Chen[1]
Alexandre Alahi[3] Matthieu Cord[1,2] Patrick Pérez[1]

***Abstract*— Motion forecasting is crucial in enabling autonomous vehicles to anticipate the future trajectories of surrounding agents. To do so, it requires solving mapping, detection, tracking, and then forecasting problems, in a multi-step pipeline. In this complex system, advances in conventional forecasting methods have been made using curated data, i.e., with the assumption of perfect maps, detection, and tracking. This paradigm, however, ignores any errors from upstream modules. Meanwhile, an emerging end-to-end paradigm, that tightly integrates the perception and forecasting architectures into joint training, promises to solve this issue. However, the evaluation protocols between the two methods were so far incompatible and their comparison was not possible. In fact, conventional forecasting methods are usually not trained nor tested in real-world pipelines (e.g., with upstream detection, tracking, and mapping modules). In this work, we aim to bring forecasting models closer to the real-world deployment. First, we propose a unified evaluation pipeline for forecasting methods with real-world perception inputs, allowing us to compare conventional and end-to-end methods for the first time. Second, our in-depth study uncovers a substantial performance gap when transitioning from curated to perception-based data. In particular, we show that this gap (1) stems not only from differences in precision but also from the nature of imperfect inputs provided by perception modules, and that (2) is not trivially reduced by simply finetuning on perception outputs. Based on extensive experiments, we provide recommendations for critical areas that require improvement and guidance towards more robust motion forecasting in the real world. The evaluation library for benchmarking models under standardized and practical conditions is provided: `https://github.com/valeoai/MFEval`.**

I. Introduction

Motion forecasting plays an important role for autonomous vehicles (i.e., ego vehicles), enabling them to anticipate future trajectories of agents in their surroundings (i.e., vehicles of interest) and, accordingly, to plan safely [1], [2], [3]. This complex task is usually shared between upstream modules for mapping, detecting, and tracking agents, and the forecasting module proper. In this system, most *conventional* forecasting works [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] set themselves in a setting with perfectly solved upstream tasks, and are trained primarily using inputs from curated offline annotations, including clean agent past trajectories and detailed road information [16]. A forecast with such curated inputs is shown in Figure 1a. However, when deployed in real-world settings, motion forecasting modules rely on data provided by upstream detection, tracking and mapping modules [17], [18], [19], [20], [21], often resulting in lower-quality predictions compared to the curated datasets used in research publications [22]. As an example, when the inputs (the past trajectories or the map) are degraded (in Figure 1b or Figure 1c respectively), the predictions become worse compared to Figure 1a, e.g., failing to anticipate well the turn (b and c in the figure) and to simply forecast an agent that is not detected (b in the figure).



(a) (b) (c)

Fig. 1: **Issues of deploying forecasting models to the real world**. We show in a nuScenes example [16] the forecasts (in orange) inferred by a motion forecasting model [6] compared to ground-truth annotations (in green), the ego car location (in red) and the static vehicles (in gray) on predicted or curated maps. (a) Satisfying forecasting performance in a curated setting; (b) When past trajectories are inferred from tracking models [23], [21], an agent is not detected and the forecasting model yields poor predictions; (c) When the map is inferred online [24], the forecasting model does not anticipate the future turn of one agent.

As an alternative to this conventional pipeline of a perception model followed by a forecasting model, *end-to-end* methods [25], [26] have recently received some attention. They advocate for joint perception-forecast training and inference with a more tightly integrated architecture, typically only using perception outputs as an intermediary representation or a multi-task training objective. Yet, both paradigms have not been compared. In fact, conventional forecasting models are not usually designed nor evaluated jointly with upstream perception models, and it is not known how they perform when integrated into the deployed pipeline.

In this work, our objective is to bring forecasting models closer to real-world deployment. Accordingly, we first design a unified evaluation protocol for forecasting by integrating the upstream perception modules into the conventional forecasting evaluation. Second, this benchmark enables us to assess and compare end-to-end methods with conventional pipelines, which was previously not feasible. Third, we also uncover a substantial drop in performance when transitioning

[1] Valeo.ai, Paris, France; Email: `firstname.lastname@valeo.com`
[2] Sorbonne Université, Paris, France.
[3] EPFL, Lausanne, Switzerland; Email: `alexandre.alahi@epfl.ch`
Corresponding author: Y. Xu, `yihong.xu@valeo.com`

from the curated setting to a real-world scenario. While it may seem intuitive that a gap exists, the issue has mostly been overlooked in driving contexts, hence, it has never been properly measured and thoroughly characterized. We then conduct a complete study to identify a wide range of obstacles that cause this gap, and we demonstrate as not easily solved. In particular, our study covers the impacts of using state-of-the-art perception modules for detection and tracking [23], [21], [18], [20], [19], [25], [26], and online rasterized or vectorized mapping [24], [28], [29], on the performance of different motion forecasting models [6], [27].

As summarized in Figure 2: (1) This work provides an evaluation protocol to benchmark forecasting models with real-world inputs (section III); (2) This benchmark allows us to compare, for the first time, the end-to-end and conventional models under standardized and practical conditions (section III and section IV); (3) Being a missing brick in current literature, our extensive experiments on the impact of various perception errors (detection, tracking, and mapping) on forecasting shed light on the critical areas that need improvement (section IV and section V); (4) Based on the findings, we provide recommendations towards robust motion forecasting in the real world (section VI).

II. Related Work

Conventional motion forecasting methods [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] focus on better leveraging and modeling the interactions between their diverse inputs. For example, AgentFormer [6] formulates the forecasting problem as the modeling of all surrounding agents conditioned on their past trajectories and contextual road elements. LaPred [27] predicts per-agent future trajectories leveraging the closest vectorized lane information and past trajectories of neighboring agents. The lane and agent information is combined to enforce the use of vectorized map information and the behavior of neighbor agents, building a strong baseline even with a simple yet efficient MLP-based trajectory decoder. We assess the performance of the conventional methods by substituting the curated data by real-world inputs and highlight the open challenges.

Object tracking and online mapping. Motion forecasting takes as inputs agents' past trajectories and map information. In the real-world setting, they are inferred respectively by motion trackers and online mappers. Motion tracking jointly performs object detection and association with different modalities. For example, MUTR3D [18] utilizes a transformer architecture with 3D track queries to model spatial and appearance coherence across multiple cameras and frames. CenterPoint [20] leverages LiDAR information to first detect the centers of objects and then regress other attributes (size, orientation, etc.). Similarly, VoxelNext [19] directly detects and tracks objects in point-cloud based on sparse voxel features. For mapping, curated maps are in general costly and hard to maintain. As an alternative, some works have proposed to estimate them online, first from a single front camera [30], [31]. Then, the development of bird's-eye view (BEV) representations led to the use of surround RGB cameras [32], [29], [28], [24] optionally coupled with LiDAR point-clouds acquisitions [33], [34], [29], [35]. Our work studies the impact of predicted trajectories and online mapping compared to the curated settings most motion forecasting models rely on.

Motion forecasting in end-to-end driving pipelines. Recent works propose to learn forecasting models directly from raw data [36], [37], [32], [38], [39], [40], [11], [25], [26]. To address the issue of error propagation in the downstream forecasting module, MTP [11] and FutureDet [41] replace the one-to-one assignment in tracking with a one-to-many one, as motion forecasting performances can be deteriorated by identity switches and detection errors [11]. AffiniPred [39] and Zhang et al. [40] perform implicit data association by using detections and their affinity matrices as inputs instead of working on past trajectories. These studies have a special focus on the tracking error while [42] and [43] tackle a subset of imperfections with an adversarial scene or object generation. Differently, we focus on understanding the impacts of real-world inputs from various state-of-the-art perception methods on the different motion forecasting paradigms. Yet, joint perception-to-forecasting models (ViP3D [25], UniAD [26]) have not been directly compared to the established pure motion forecasting models, primarily due to differences in their approaches and evaluation criteria. An adapted evaluation protocol is needed with metrics considering the upstream errors [44], [45], [41], [46]. Our work is the first to pull both approaches into a single evaluation framework.

III. A Unified Evaluation Benchmark

In our study, we analyze conventional forecasting models when they are confronted, instead of the curated data, to outputs of various state-of-the-art perception modules for detection and tracking [23], [21], [18], [20], [19], [25], [26], and to online rasterized or vectorized mapping [24], [28], [29] on nuScenes dataset [16]. As illustrated in Figure 2, we replace the curated annotations with real-world predictions as inputs to conventional motion forecasting models [6], [27] and examine the forecasting and perception performance. By doing so, we can incorporate the same predicted perception inputs as in end-to-end approaches [25], [26]. However, the comparison is confronted with challenges that we first detail and then address here. As a direct application, we are able to compare, for the first time, recent methods in the end-to-end forecasting paradigm to conventional methods.

A. The need for matching

Standard forecasting datasets, such as the used nuScenes dataset [16], provide past information about the ground-truth agents, including their number, identities, and trajectories. Conventional forecasting models rely on these identities to train their models and compute their scores. However, in the real world, detected agents are not inherently linked to ground-truth agents because GT identities are not provided during inference, and arbitrary identities are assigned during tracking. To address this, a one-to-one matching is needed

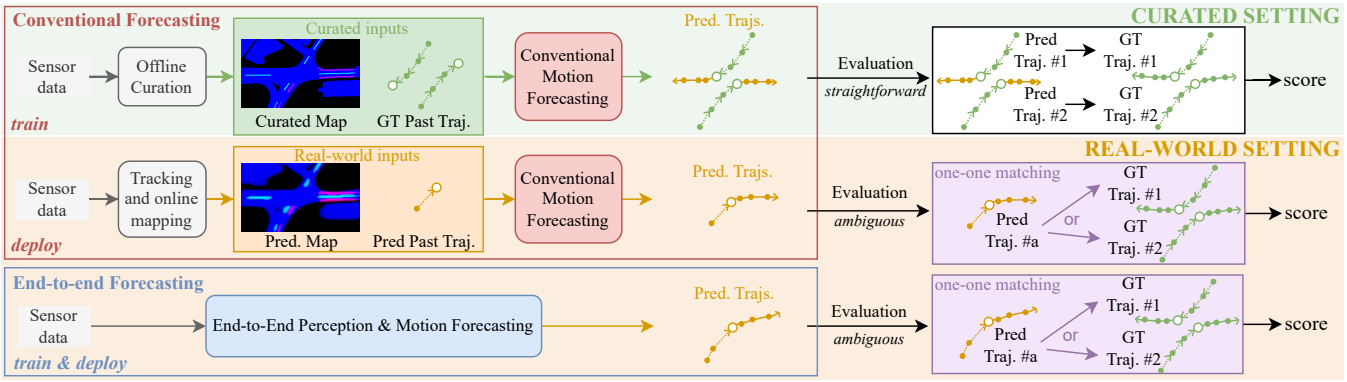


Fig. 2: **Study overview**. We study the challenges of deploying motion forecasting models into the real world when only predicted perception inputs are available. We compare (section III): (1) (*top*) 'conventional methods' [6], [27] (i.e., methods trained on curated input data) where (middle) we directly replace the curated inputs with real-world data, and (2) (*bottom*) 'end-to-end methods' [25], [26] that are trained and used with perception modules. In the real-world setting, evaluation is challenging as the past tracks are estimated with arbitrary identities, making it difficult to establish a direct correspondence to GT identities. Therefore, we propose a matching process (purple) to assign predictions to GT and thus evaluate forecasting performances (section III). Moreover, we study in depth the impact changing from curated data (green) to real-world (orange) mapping (section IV), or detection and tracking (section V) errors to motion forecasting.

to assign predictions to ground-truth agents. Similar to the multi-object tracking problem [47], we use Hungarian matching with a matching threshold of 2 meters between object center L2 distances at the starting frame for forecasting, i.e., $t = 0$.

B. The need for suitable metrics

For similar reasons, standard forecasting metrics — **minADE**$_k$, the minimum over k predictions of Average Distance Error (the average of point-wise L2 distances between the prediction and ground-truth forecasts), **minFDE**$_k$, the minimum over k predictions of Final Distance Error (the L2 distance at the final future time step), and **MR**$_{k@x}$, MissRate, the ratio of forecasts having **minFDE**$_k$ > x = 4 meters — are built on the assumption that the models have matching identities from past and future tracks. These metrics solely consider the forecasting quality of *matched* prediction-GT pairs, without penalizing missed or falsely predicted agents.

As this does not provide the full picture of real-world forecasting performance, we propose to consider the Mean Forecasting Average Precision (mAP$_f$) [41] that shares the same formulation as detection AP [48]. However, AP$_f$ considers as false positives not only trajectories with incorrect first-frame detections (center L2 distance at $t = 0$ bigger than 2m), but also those having correct first-frame detections but inaccurate forecasts (**minFDE**$_k$ > 4m) . The APs are then averaged over the classes 'car', 'truck', and 'bus' – mAP$_f$. Unlike [25], [26], to reflect the forecasting quality, we only evaluate ground-truth (GT) *vehicle* agents having full *moving* future trajectories as in nuScenes Prediction challenge, resulting in different and more realistic performance than the one shown in [25], [26] that consider mostly static objects including pedestrians.

TABLE I: **Comparison of end-to-end and conventional forecasting methods** given the same detection and tracking ('Det&Track') inputs; k, the number of possible forecasts, i.e., *modes*.

Det&Track Input	Forecast Method	mAP$_f$↑	minADE↓	minFDE↓	MR↓
Ground truth	AgentFormer (k = 5)	0.388	1.851	3.875	0.315
	LaPred (k = 5)	**0.588**	**1.547**	**3.176**	**0.208**
ViP3D (CVPR'23) [25]	AgentFormer (k = 5)	0.056	2.416	4.404	0.353
	LaPred (k = 5)	0.092	2.612	4.520	0.282
	ViP3D (k = 5)	0.021	4.018	7.040	0.505
	LaPred (k = 6)	**0.113**	**2.365**	**3.900**	**0.224**
	ViP3D (k = 6)	0.034	3.540	5.943	0.432
UniAD (CVPR'23) [26]	AgentFormer (k = 5)	0.069	2.530	4.613	0.384
	LaPred (k = 5)	0.123	2.684	4.678	0.278
	UniAD (k = 5)	0.094	2.071	3.810	0.283
	LaPred (k = 6)	**0.143**	2.499	4.212	0.237
	UniAD (k = 6)	0.117	**1.842**	**3.258**	**0.228**

C. Conventional vs. End-to-End forecasting

Recent end-to-end methods [25], [26], though providing a promising direction, are poorly evaluated (e.g., including static objects), which hinders the understanding of their underlying issues. Since both paradigms have never been fairly compared, we make the first step to compare them with the same perception inputs from end-to-end pipelines.

We consider AgentFormer [6], using rasterized maps, and LaPred [27], using vectorized maps, as strong representatives of conventional methods. For the end-to-end paradigm, we choose two state-of-the-art methods, namely, ViP3D [25], an end-to-end motion forecasting model, and UniAD [26], an end-to-end forecasting and planning model. The choice is made considering the availability of code and their distinct structures. Please refer to the related works for more details.

As shown in Table I, the first observation is that *recent state-of-the-art end-to-end models do not exhibit superior performance in forecasting, compared to conventional meth-*

TABLE II: **Performance of conventional and end-to-end methods with various types of input maps.** Map is a ground-truth curated map, an empty map or an online (rasterized or vectorized) map. In the two latter cases, the model is evaluated on the new map type either directly ('*transfer*') or after finetuning ('*finetune*').

Method	Map	Setting	$\text{mAP}_f\uparrow$	minADE↓	minFDE↓	MR↓
AgentFormer [6] (map: Raster)	ground truth	default	**0.388**	**1.851**	**3.875**	**0.315**
	empty map	*transfer*	0.057	2.747	6.165	0.668
	empty map	*finetune*	0.289	1.966	4.221	0.398
	LaRa [28]	*transfer*	0.028	3.246	7.358	0.750
	LaRa [28]	*finetune*	0.341	1.916	4.085	0.350
	SimpleBeV [24]	*transfer*	0.034	3.128	7.067	0.727
	SimpleBeV [24]	*finetune*	0.361	1.856	3.935	0.333
LaPred [27] (map: Vector)	ground truth	default	**0.760**	**1.237**	**2.344**	**0.118**
	empty map	*transfer*	0.239	2.385	5.152	0.481
	empty map	*finetune*	0.460	1.654	3.419	0.291
	MapTR [29]	*transfer*	0.302	2.269	4.863	0.433
	MapTR [29]	*finetune*	0.499	1.670	3.472	0.273
ViP3D [25] (map: Vector)	ground truth	default	0.034	3.540	5.943	0.432
	empty map	*transfer*	0.033	3.540	5.943	0.432
	empty map	*finetune*	**0.040**	**3.277**	**5.589**	**0.404**
UniAD [26] (map: Raster)	online map [26]	default	0.117	**1.842**	3.258	**0.228**
	empty map	*transfer*	0.112	1.908	3.441	0.247
	empty map	*finetune*	**0.118**	1.844	**3.250**	**0.228**

ods that have been trained only with curated inputs. Precisely, with the detection and tracking inputs from ViP3D, AgentFormer outperforms ViP3D significantly (more than 2 times higher mAP_f) without being jointly trained with such perception inputs. Since AgentFormer combines agent past history and inter-agent interactions in an attention module, without explicitly separating different trajectories, this joint interaction may help to resist the detection and tracking errors. Similarly, with no finetuning, LaPred leads the performance in mAP_f with a simpler trajectory predictor, compared to UniAD with its end-point attention-based refinement and physically-based kinematic model. Lastly, we observe that end-to-end pipelines (UniAD and ViP3D), despite operating in a much more realistic setting, are still *very far* from conventional forecasting methods with curated data (i.e., AgentFormer and LaPred with ground-truth inputs). The observations imply that it might not be trivial to train the end-to-end model jointly with perception modules without tackling their errors in downstream forecasting.

IV. Impacts of the Map Quality

We now study in more detail how conventional forecasting methods use their inputs and if we can replace them with real-world perceptions. In this section, we intervene on the map input of AgentFormer [6], LaPred [27], ViP3D [25] and UniAD [26]. Originally, AgentFormer uses *rasterized* curated maps; LaPred and ViP3D both leverage *vectorized* curated maps, while UniAD uses online mapping inferred from camera data.

A. Removing map information entirely

Our first goal is to assess the dependence of forecasting performances to map information. To do so, we replace the maps at the input of the forecasting modules with empty ones. In practice, we consider two distinct scenarios: 1) a direct '*transfer*' without finetuning on empty maps, and 2) a '*finetune*' one where each model uses empty maps throughout both finetuning and inference stages.

From Table II, our experiments reveal that *the presence or absence of maps has minimal impact on the performance of the end-to-end ViP3D and UniAD models.* For ViP3D, training the model with curated maps and replacing the map with an empty map during inference shows negligible change in results (*transfer* setting). Moreover, finetuning ViP3D without map information leads to slightly improved performance on the validation set (*finetune* setting), indicating that the map is not utilized at all in ViP3D. For UniAD, the model exhibits a slight performance drop when using an empty map during inference, but finetuning without map information yields similar results to using the online map, suggesting that it is not well utilized in UniAD either. On the other hand, for AgentFormer and LaPred, finetuning without maps does not close the performance gap compared to having maps, indicating their better utilization of contextual information. Although the poor usage of maps has already been pointed out by previous work [49] on some conventional methods [5], [50], *we show for the first time that the issue persists in both recent end-to-end models [26] and [25].* In the following, we study the impact of different aspects of the map in more detail. As we have determined that ViP3D and UniAD do not effectively utilize the map, our following analysis concentrates on AgentFormer and LaPred.

B. From curated to online mapping

We now assess the performance gap when going from curated maps to using the output of an online mapping method. We consider three different mapping methods: SimpleBeV [24] and LaRa [28] for rasterized BEV map prediction and MapTR [29], an online mapping method predicting vectorized map elements. LaRa obtains 0.361 mIoU and 0.458 for SimpleBeV. MapTR exhibits state-of-the-art performance with 62.8 mapping mAP [29].

In Table II, we observe that for direct *transfer*, SimpleBeV and LaRa maps perform worse than empty maps, indicating a significant domain gap as the forecasting model is unable to use online predictions directly. Finetuning improves performance for both empty and online maps, but the latter, although better, still underperforms curated maps. Thus, online maps are so far insufficient to replace curated maps. One issue could be their limited range around the ego car, much smaller than curated maps. Additionally, we find that high-level map elements such as 'lane', 'drivable area' ($0.388 \rightarrow 0.296$ mAP_f for AgentFormer, and $0.760 \rightarrow 0.611$ for LaPred) are more impactful than detailed map information ('road' and 'lane dividers') that could be better leveraged ($0.388 \rightarrow 0.337$ mAP_f for AgentFormer and $0.760 \rightarrow 0.668$ for LaPred).

V. Impacts of Detection and Tracking

In this section, we study the past trajectories inputs of conventional forecasting models [6], [27], by replacing GT

TABLE III: **Influence of the perception input type on tracking and forecasting metrics.** Forecasting methods are AgentFormer and LaPred. [1]The past is interpolated when it is incomplete hence the imperfect MOTA and FP values.

		Tracking metrics					*Forecasting metrics for AgentFormer*				*Forecasting metrics for LaPred*			
	Perception input	**MOTA↑**	**MOTP↓**	**FP↓**	**FN↓**	**IDS↓**	**mAP$_f$↑**	**minADE↓**	**minFDE↓**	**MR↓**	**mAP$_f$↑**	**minADE↓**	**minFDE↓**	**MR↓**
	GT position and identity	0.985[1]	0.000	155[1]	0	0	0.343	1.885	3.979	0.359	0.757	1.213	2.316	0.115
	GT position + Tracking model [20]	0.967	0.001	180	0	165	0.317	1.934	4.020	0.363	0.730	1.353	2.513	0.125
Camera-based	MUTR3D R50 [18] (CVPRW'22)	0.170	0.607	1828	6696	27	0.042	3.993	7.151	0.463	0.152	2.237	3.554	0.178
Camera-based	MUTR3D R101 [18] (CVPRW'22)	0.213	0.550	1373	6727	11	0.055	3.480	6.286	0.449	0.198	1.892	3.043	**0.150**
Camera-based	ViP3D Det&Track [25] (CVPR'23)	0.145	0.636	1855	6947	3	0.056	2.416	4.404	**0.353**	0.142	2.044	3.173	0.155
Camera-based	UniAD Det&Track [26] (CVPR'23)	0.195	0.471	**1199**	7076	20	0.069	2.530	4.613	0.384	0.180	2.142	3.424	0.169
LiDAR-based	MegVii [23]+AB3DMOT [21] (IROS'20)	0.226	0.320	1657	6232	79	0.089	2.356	4.412	0.382	0.227	2.143	3.561	0.168
LiDAR-based	CenterPoint [20] (CVPR'21)	**0.348**	**0.244**	1622	**5090**	5	**0.112**	**2.102**	**4.354**	0.413	0.285	**1.596**	**2.815**	0.151
LiDAR-based	VoxelNext [19] (CVPR'23)	0.328	0.263	1639	5283	**2**	0.096	2.134	4.409	0.426	**0.317**	1.669	2.914	0.166

past trajectories with outputs of real-world tracking models (subsection V-A) or by artificially intervening on them (subsection V-B). This allows us to study the importance of precise agents' positioning and identification. We also study in subsection V-C how much can simple finetuning can improve the forecasting models. To quantify the perception input quality, we count the number of false positives (FP), i.e., predicted objects not associated with any GT object, of false negatives (FN), i.e., GT objects not associated with any prediction, and of identity switches (IDS), i.e., assigning wrongly the detection to an agent of a different identity. For ease of interpretation, FP, FN and IDS are combined into Multiple Object Tracking Accuracy (MOTA) [47]. We also compute Multiple Object Tracking Precision (MOTP), quantifying the average positional accuracy over matched objects. Details can be found in [47].

A. From curated to predicted agents

We investigate how forecasting models would react in a realistic setting where the agents' positions and identities are not curated but predicted by perception models. Our selection of such models includes recent state-of-the-art methods that can be LiDAR-based, such as MegVii [23] + AB3DMOT [21], CenterPoint [20] and VoxelNext [19], or camera-based such as MUTR3D [18] with either ResNet-50 (R50) or ResNet-101 (R101) backbones, ViP3D [25], and UniAD [26]. In Table III, we also show results obtained with the curated GT as inputs for comparison. First, we observe that using real-world inputs leads to a very significant forecasting performance drop for both AgentFormer ($0.343 \rightarrow 0.112$ in mAP$_f$) and LaPred ($0.757 \rightarrow 0.317$). Second, while LiDAR-based methods are much better at detection and tracking than camera-based methods, we remark that this only translates into marginally better forecasting scores. For instance, ViP3D Det&Track even manages to achieve better MR and comparable minFDE to LiDAR-based methods, while UniAD Det&Track is not so far behind. Then, we can observe that replacing ground-truth identities of agents with outputs of a tracking model from [20] does not significantly degrade the performance (e.g., $0.343 \rightarrow 0.317$ in mAP$_f$ for AgentFormer) despite its worse IDS. This indicates that IDS in near agents is not as impactful as detection errors.

B. Detection and tracking errors breakdown

A significant performance drop is observed using real-world inputs that contain a complex combination of highly correlated errors. To give insights on which type of errors dominates, we break down the errors that occur in the *past trajectories* into four types: FP by duplicating and perturbing ground-truth agents' locations within 5 meters; FN (except at t=0, i.e., the starting point of forecasting) by randomly removing ground-truth agents; localization error (Loc. Error correctly detected but misplaced) by perturbations of ground truth within 2 meters and, IDS with nearby neighbors within 5 meters. The choice of distance is based on real-world conditions (e.g., a vehicle tends to switch identities with nearby vehicles). We simulate errors with synthetic perturbations on the curated annotations, in varying proportions, and plot the performance of AgentFormer and LaPred in these settings in Figure 3. We note that for FN, we do not consider missing detections at t=0 since it will obviously cause catastrophic miss forecasts. Besides, we observe that localization errors are the most impactful on forecasting metrics while IDS errors have less impact, in complement with [11], [39].

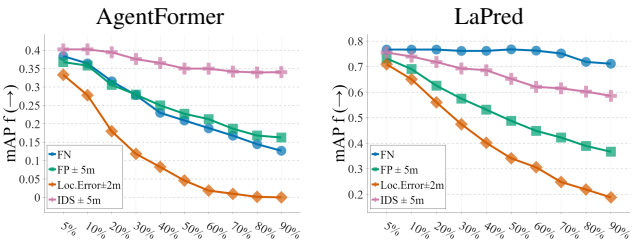


Fig. 3: **Impact of controlled input errors** . Forecasting performance (mAP$_f$) under different proportions of detection and tracking errors (x-axis); We simulate misdetections (FN, in blue), false detections (FP@5meters, in green), localization errors (Loc. Error@2meters in orange) and tracking errors (IDS@5meters, in pink) in the past trajectories.

C. Finetuning with imperfect inputs

Intuitively, one can think that the performance gap can be easily closed by finetuning. This experiment has already been conducted for map imperfections in Table II (*'finetune'*) where we show that the gap persists. We further conduct finetuning experiments on the stronger model (LaPred) with (1) data augmentation by randomly adding 30% GT past position perturbation (i.e., simulating Loc. Error). We also tried with 10% and 50%, yielding worse results. (2) real-world tracking results of different modalities (LiDAR-based VoxelNext [19], end-to-end UniAD [26] and camera-based

TABLE IV: **Impact of finetuning**. LaPred [27] pretrained on GT with data augmentation (DA) is finetuned by either randomly shifting GT past trajectories or using results from real-world tracking methods, for 30 epochs with a learning rate 2x slower than the default one (i.e., 5e-5). The finetuning is conducted on the train set and tested on the validation set. Results are relative to training with GT.

Finetuning input	Testing input	$\text{mAP}_f \uparrow$	minADE↓	minFDE↓	MR↓
No finetuning	GT	0.760	1.237	2.344	0.118
	VoxelNext	0.317	1.669	2.914	0.166
	UniAD	0.180	2.142	3.424	0.169
	MUTR3D R101	0.198	1.892	3.043	0.150
70% GT + 30% DA	VoxelNext	-0.007	+0.186	+0.056	+0.005
	UniAD	+0.007	-0.166	-0.324	-0.015
	MUTR3D R101	+0.009	-0.034	-0.191	-0.004
VoxelNext	VoxelNext	+0.022	-0.142	-0.283	-0.023
UniAD	UniAD	+0.018	-0.534	-0.782	-0.020
MUTR3D R101	MUTR3D R101	+0.010	-0.135	-0.294	-0.002

MUTR3D R101 [18]). We show in Table IV that finetuning with data augmentation has very limited help in improving forecasting performance with real-world inputs. A systematic but slight improvement is observed by finetuning with the actual real-world inputs (on the trainset) but the improved performance is not comparable with GT annotations.

The reasons are: (1) The distribution of perturbation in the trainset is different from the one in the validation set. (e.g., MapTR obtained 94.7 mapping mAP in the trainset vs. 62.8 in validation); (2) The forecasting models are not designed to handle such errors; (3) Unlike weather or road domain gap [51], [52], the map elements or the detections are often missing and simply finetuning with real-world inputs brings no benefits. This indicates that poor forecasting performance due to upstream perception errors cannot be easily fixed with basic domain adaptation methods.

D. Impact of the distance to ego vehicle

To better characterize the differences between different detection tracking methods, we group the agents per distance to the ego vehicle and report both tracking and forecasting performances in Figure 4. First, we observe that LiDAR-based trackers fare better in general. This advantage can be explained by the fact that LiDAR sensors keep good precision regardless of range. While the superiority is especially revealed with MOTP, the relatively similar MOTA scores indicate that LiDAR-based methods struggle nearly as much as camera-based trackers for detecting far-away agents. The superiority in MOTP however reflects that once detected, they are much better at precisely locating them. LiDAR-based methods are better in mAP_f despite similar MOTA is also compatible with our previous observation that the localization precision (in MOTP) is crucial for forecasting. This observation is significant because the ego vehicle position is often ignored in motion forecasting as the prediction and evaluation are usually done in an agent-centric view. In light of these findings, motion forecasting models and evaluation should consider agent-ego distance.

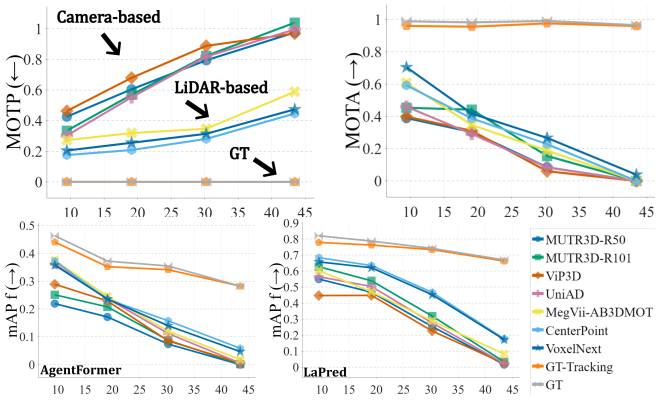


Fig. 4: **Impact of agent-ego distance.** Tracking and forecasting performances w.r.t. agent-ego distance (x-axis in meters) for tracking methods: (camera-based) MUTR3D-R50, MUTR3D-R101, ViP3D, UniAD; (LiDAR-based) MegVii-AB3DMOT, CenterPoint, VoxelNext; GT-Tracking and GT.

VI. Conclusion

This work brings conventional and end-to-end methods into a joint evaluation protocol representative of real-world constraints and enables the study of diverse forecasting methods (4 of them in the experiments) when facing outputs of trackers (7 considered), and online mapping methods (3 considered, vectorized or rasterized, using LiDAR or camera). This comparison sheds light on the poor performance of end-to-end methods and highlights the challenges that arise when interfacing perception and forecasting models. The main findings and corresponding recommendations are:

- (section III and section IV) *The emerging 'end-to-end forecasting' paradigm is so far not better than the conventional one, even in a real-world setting without finetuning. Besides, end-to-end models do not utilize map information.* A better multi-task learning strategy and map integration design is needed to advance the end-to-end paradigm.
- (section IV and section V) *There is a large and systematic performance gap going from curated annotations to perception predictions, which is not reduced by simple techniques,* requiring more effort than just joint training. Also, for perception tasks, precise localization should be considered along with detection mAP or tracking accuracy and the perception range should be enlarged to a physically achievable range.
- (section V) *We show that the perception and forecasting quality depends on the agent-ego distance.* While intuitive once formulated, this information is missing in current benchmarks, which should be included by stratifying the evaluation according to the distance to ego-vehicle.

Finally, we encourage the community to publicly release codes and models to benchmark them with a broader spectrum of driving scenes in the future. The advantages of end-to-end models, such as a single-loop training and easier deployment, motivate us to further investigate and improve this promising paradigm. We also note that our study can be further extended to the downstream task of motion planning.

ACKNOWLEDGMENT

This work was supported by the ANR MultiTrans (ANR-21-CE23-0032) and CINES (HORS DARI N°A0141014181). 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. Finally, we want to thank Florent Bartoccioni and Kaouther Messaoud for helpful discussions.

REFERENCES

[1] H. Nishimura, J. Mercat, B. Wulfe, R. T. McAllister, and A. Gaidon, "RAP: risk-aware prediction for robust planning," in *CoRL*, 2022, pp. 381–392. 1

[2] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone, "Robust trajectory prediction against adversarial attacks," in *CoRL*, 2022, pp. 128–137. 1

[3] X. Li, J. A. DeCastro, C. I. Vasile, S. Karaman, and D. Rus, "Learning A risk-aware trajectory planner from demonstrations using logic monitor," in *CoRL*, 2021, pp. 1326–1335. 1

[4] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *CVPR*, 2020, pp. 14 074–14 083. 1, 2

[5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *ECCV*, 2020, pp. 683–700. 1, 2, 4

[6] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *ICCV*, 2021, pp. 9813–9823. 1, 2, 3, 4

[7] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *ICCV*, 2021, pp. 15 303–15 312. 1, 2

[8] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *CoRL*, 2019, pp. 86–99. 1, 2

[9] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. Lam, D. Anguelov, and B. Sapp, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *ICRA*, 2022, pp. 7814–7821. 1, 2

[10] Y. C. Tang and R. Salakhutdinov, "Multiple futures prediction," in *NeurIPS*, 2019, pp. 15 424–15 434. 1, 2

[11] H. Cui, V. Radosavljevic, F. Chou, T. Lin, T. Nguyen, T. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *ICRA*, 2019, pp. 2090–2096. 1, 2, 5

[12] P. Kothari, D. Li, Y. Liu, and A. Alahi, "Motion style transfer: Modular low-rank adaptation for deep motion forecasting," in *CoRL*, 2022, pp. 774–784. 1, 2

[13] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun, "Gorela: Go relative for viewpoint-invariant motion forecasting," in *ICRA*, 2023, pp. 7801–7807. 1, 2

[14] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang, "Ganet: Goal area network for motion forecasting," in *ICRA*, 2023, pp. 1609–1615. 1, 2

[15] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *ICRA*, 2023, pp. 2980–2987. 1, 2

[16] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020, pp. 11 621–11 631. 1, 2

[17] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robotics and Automation Letters*, pp. 3054–3061, 2022. 1

[18] T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Mutr3d: A multi-camera tracking framework via 3d-to-2d queries," in *CVPRW*, 2022, pp. 4537–4546. 1, 2, 5, 6

[19] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking," in *CVPR*, 2023, pp. 21 674–21 683. 1, 2, 5

[20] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *CVPR*, 2021, pp. 11 784–11 793. 1, 2, 5

[21] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *IROS*, 2020, pp. 10 359–10 366. 1, 2, 5

[22] B. Okumura, M. R. James, Y. Kanzawa, M. Derry, K. Sakai, T. Nishi, and D. V. Prokhorov, "Challenges in perception and decision making for intelligent automotive vehicles: A case study," *IEEE Trans. Intell. Veh.*, pp. 20–32, 2016. 1

[23] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019. 1, 2, 5

[24] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-BEV: What really matters for multi-sensor bev perception?" in *ICRA*, 2023, pp. 2759–2765. 1, 2, 4

[25] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Vip3d: End-to-end visual trajectory prediction via 3d agent queries," in *CVPR*, 2023, pp. 5496–5506. 1, 2, 3, 4, 5

[26] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *CVPR*, 2023, pp. 17 853–17 862. 1, 2, 3, 4, 5

[27] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *CVPR*, 2021, pp. 14 636–14 645. 2, 3, 4, 6

[28] F. Bartoccioni, É. Zablocki, A. Bursuc, P. Pérez, M. Cord, and K. Alahari, "Lara: Latents and rays for multi-camera bird's-eye-view semantic segmentation," in *CoRL*, 2022, pp. 1663–1672. 2, 4

[29] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized HD map construction," in *ICLR*, 2023. 2, 4

[30] Y. B. Can, A. Liniger, D. P. Paudel, and L. V. Gool, "Structured bird's-eye-view traffic scene understanding from onboard images," in *ICCV*, 2021, pp. 15 661–15 670. 2

[31] Y. B. Can, A. Liniger, O. Unal, D. P. Paudel, and L. V. Gool, "Understanding bird's-eye view of road semantics using an onboard camera," *IEEE Robotics Autom. Lett.*, pp. 3302–3309, 2022. 2

[32] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *ECCV*, 2020, pp. 194–210. 2

[33] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *ICRA*, 2022, pp. 4628–4634. 2

[34] Y. Liu, Y. Wang, Y. Wang, and H. Zhao, "Vectormapnet: End-to-end vectorized HD map learning," in *ICML*, 2023, pp. 22 352–22 369. 2

[35] R. Sun, D. Lingrand, and F. Precioso, "Exploring the road graph in trajectory forecasting for autonomous driving," in *ICCVW*, 2023. 2

[36] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *CVPR*, 2018, pp. 3569–3577. 2

[37] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "Pnpnet: End-to-end perception and prediction with tracking in the loop," in *CVPR*, 2020, pp. 11 553–11 562. 2

[38] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "FIERY: future instance prediction in bird's-eye view from surround monocular cameras," in *ICCV*, 2021, pp. 15 273–15 282. 2

[39] X. Weng, B. Ivanovic, K. Kitani, and M. Pavone, "Whose track is it anyway? improving robustness to tracking errors with affinity-based trajectory prediction," in *CVPR*, 2022, pp. 6573–6582. 2, 5

[40] P. Zhang, L. Bai, J. Xue, J. Fang, N. Zheng, and W. Ouyang, "Trajectory forecasting from detection with uncertainty-aware motion encoding," *arXiv preprint arXiv:2202.01478*, 2022. 2

[41] N. Peri, J. Luiten, M. Li, A. Osep, L. Leal-Taixé, and D. Ramanan, "Forecasting from lidar via future object detection," in *CVPR*, 2022, pp. 17 202–17 211. 2, 3

[42] M. Bahari, S. Saadatnejad, A. Rahimi, M. Shaverdikondori, A. H. Shahidzadeh, S.-M. Moosavi-Dezfooli, and A. Alahi, "Vehicle trajectory prediction works, but not everywhere," in *CVPR*, 2022, pp. 17 123–17 133. 2

[43] J. Sarva, J. Wang, J. Tu, Y. Xiong, S. Manivasagam, and R. Urtasun, "Adv3d: Generating safety-critical 3d objects through closed-loop simulation," in *CoRL*, 2023. 2

[44] J. Philion, A. Kar, and S. Fidler, "Learning to evaluate perception models using planner-centric metrics," in *CVPR*, 2020. 2

[45] E. Weng, H. Hoshino, D. Ramanan, and K. Kitani, "Joint metrics matter: A better standard for trajectory forecasting," *arXiv preprint arXiv:2305.06292*, 2023. 2

[46] B. Ivanovic, K.-H. Lee, P. Tokmakov, B. Wulfe, R. Mcllister, A. Gaidon, and M. Pavone, "Heterogeneous-agent trajectory forecasting incorporating class uncertainty," in *IROS*, 2022, pp. 12 196–12 203. 2

[47] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Eurasip J.*, pp. 1–10, 2008. 3, 5

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755. 3

[49] H. Ben-Younes, É. Zablocki, M. Chen, P. Pérez, and M. Cord, "Raising context awareness in motion forecasting," in *CVPRW*, 2022, pp. 4409–4418. 4

[50] D. Zhu, M. Zahran, L. E. Li, and M. Elhoseiny, "Halentnet: Multimodal trajectory forecasting with hallucinative intents," in *ICLR*, 2021. 4

[51] Y. Lee, Y. Kim, J. Yu, and M. Jeon, "Learning to remove bad weather: towards robust visual perception for self-driving," *IEEE Robotics and Automation Letters*, 2022. 6

[52] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu, "Shift: a synthetic driving dataset for continuous multi-task domain adaptation," in *CVPR*, 2022, pp. 21 371–21 382. 6