Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 4865–4870

Robustness Benchmark of Road User Trajectory Prediction Models for Automated Driving ⋆

Manuel Muñoz Sánchez * **Emilia Silvas** *,** **Jos Elfring** *,*** **René van de Molengraft** *

* *Department of Mechanical Engineering, TU/e, Eindhoven, NL.*
** *Department of Integrated Vehicle Safety, TNO, Helmond, NL.*
*** *VDL CropTeq Robotics, Eindhoven, NL.*

Abstract: Accurate and robust trajectory predictions of road users are needed to enable safe automated driving. To do this, machine learning models are often used, which can show erratic behavior when presented with previously unseen inputs. In this work, two environment-aware models (MotionCNN and MultiPath++) and two common baselines (Constant Velocity and an LSTM) are benchmarked for robustness against various perturbations that simulate functional insufficiencies observed during model deployment in a vehicle: unavailability of road information, late detections, and noise. Results show significant performance degradation under the presence of these perturbations, with errors increasing up to +1444.8% in commonly used trajectory prediction evaluation metrics. Training the models with similar perturbations effectively reduces performance degradation, with error increases of up to +87.5%. We argue that despite being an effective mitigation strategy, data augmentation through perturbations during training does not guarantee robustness towards unforeseen perturbations, since identification of all possible on-road complications is unfeasible. Furthermore, degrading the inputs sometimes leads to more accurate predictions, suggesting that the models are unable to learn the true relationships between the different elements in the data.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Automated driving, automated vehicles, trajectory prediction, robustness.

1. INTRODUCTION

Trajectory prediction of road users (RUs) for automated vehicles (AVs) has received much attention in recent years, since it allows an AV to anticipate how the environment will evolve and react better, achieving safer, more comfortable, and more efficient reactions. Given the success of machine learning in various fields, a data-driven approach has also been adopted in trajectory prediction for AVs.

The main focus of early works on trajectory prediction was to increase predictive accuracy leveraging as much information as possible about the environment, such as surrounding RUs (Alahi et al., 2016; Kosaraju et al., 2019) and road infrastructure (Salzmann et al., 2020; Yoon et al., 2020). To develop these methods, a heavily processed dataset is often used, where detailed information about the RUs and road geometry is available. To deal with more realistic data and previously unseen situations, recently more emphasis is being placed on robustness of these methods (Roelofs et al., 2022; Cao et al., 2022; Bahari et al., 2022; Saadatnejad et al., 2022; Zhang et al., 2022), since they must achieve reliable predictions.

An obvious approach to evaluate and increase model robustness towards unseen situations is to collect more data containing those missing situations (e.g. different locations, weather conditions, faulty sensors, etc.), which is costly, time consuming, and not scalable. To address this lack of data, a common practice is to introduce *perturbations* (i.e. synthetic variations) to an already existing dataset to verify that the model still produces sensible predictions. Perturbations can be either *complementing*, which aim to enhance the recorded data with previously unseen situations (e.g.craft fake historical trajectories (Cao et al., 2022; Zhang et al., 2022)), or *disruptive*, which simulate complications such as sensor noise (Zamboni et al., 2022).

One of the limitations of current robustness evaluations through disruptive perturbations is that the choice of perturbations does not resemble harsh realistic conditions (e.g. longer data losses or high noise levels). These can occur and must be considered if AVs should function in various challenging environments (e.g. adverse weather, poor lighting, or infrastructural changes). When simulating disruptive perturbations, it is common to assume a *bias* in the observed trajectories, or sample artificial observation noise from some distribution, often Gaussian (Zamboni et al., 2022). However, the motivation behind the choice of parameters for sampling this noise is often lacking. If the analyzed perturbations are not representative of what one might encounter in reality, the identified performance degradation will not be representative of what one can expect in the vehicle. Consequently, the mitigation approaches will not be effective in reality.

⋆ This work was supported by SAFE-UP under EU's Horizon 2020 research and innovation programme, grant agreement 861570.

2405-8963 Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2023.10.1256

To bridge the gap between heavily processed rich learning datasets and in-vehicle data, in this work, we perform a robustness evaluation of several trajectory prediction models introducing extreme possible perturbations. Such perturbations are complete unavailability of road information, late detections with only one observation, and highly noisy heading angle measurements, which could be caused by faulty sensors or adverse weather conditions (Zang et al., 2019). To show the impact these perturbations can have in data-driven models that did not consider similar perturbations during training, we first assess the performance degradation using only perturbed data. Next, we train the models considering these perturbations and show the robustness increase towards perturbed data and potential performance decrease when using original data.

Thus, our work's main contribution is a benchmark that assesses the robustness of various machine learning trajectory prediction models against severe and highly detrimental perturbations. The choice of specific perturbations is motivated by real functional insufficiencies in world modeling that we observed when deploying our models in a vehicle. Additionally, we encourage the scientific community to perform similar types of robustness studies by sharing the framework we created for our study as reference [1] .

The remainder of this article is structured as follows. Section 2 presents the trajectory prediction problem and summarizes related work on robustness of trajectory prediction models. Section 3 outlines the benchmark procedure and introduces the perturbations considered. Section 4 summarizes the results, and Section 5 concludes the work and highlights future improvements.

2. PRELIMINARIES

2.1 Trajectory Prediction

Trajectory prediction refers to predicting the future positions of an RU. Given a trajectory prediction model $\mathcal{M}$ and its inputs $\mathcal{I}$, it will generate future predictions $\mathcal{P} = \mathcal{M}(\mathcal{I})$. The types of inputs and predictions are dependent on the model, although to generate predictions for a target RU $n \in N$, at least $\mathcal{I} = (\mathbf{s}^n_{t_0}, T, *)$ and $\mathcal{P} = (\hat{\mathbf{x}}^n_T, *)$, where $\mathbf{s}^n_{t_0}$ denotes the current state of n at time $t = 0$, T denotes the desired prediction horizons, and $\hat{\mathbf{x}}^n_T$ denotes one or more predictions of future positions of n at those horizons. Additionally, $*$ denotes other optional inputs and outputs.

Typically, additional inputs are a map $\mathbf{m}$ containing geometric and semantic information of the static environment (e.g. a road model); past states of the target $\mathbf{s}^n_{t:0}$ from time $t < 0$ up to but not including $t = 0$; and other RUs' states, $\mathbf{s}^{N\setminus n}_{t:0}$ from time $t < 0$ up to and including $t = 0$. Typical additional outputs are some measure of uncertainty $\mathcal{U}$ associated with the predictions.

To assess the accuracy of a trajectory prediction model, its predictions at times T for a set of RUs N, $\mathcal{P} = (\hat{\mathbf{x}}^N_T, *)$, are compared with the real (ground truth) future positions, $\mathbf{x}^N_T$. To quantify this accuracy, one of the most common metrics is minADE (Rasouli, 2020), defined as

$$\text{minADE}(\hat{\mathbf{x}}^N_T, \mathbf{x}^N_T) = \sum_{n\in N} \min_{\mathbf{y}\in\hat{\mathbf{x}}^n_T} \sum_{t\in T} \frac{\|\mathbf{y}_t - \mathbf{x}^n_t\|_2}{|N|\cdot|T|}, \quad (1)$$

where $|.|$ denotes the size of a set, and $\|.\|_2$ denotes the L2-norm of a vector. Note that this metric only considers $\hat{\mathbf{x}}$ and not $\mathcal{U}$. Despite this limitation, it has become widely popular since it allows direct comparison between models that provide $\mathcal{U}$ and those that do not.

2.2 Trajectory Prediction Robustness

To enable proactive automated driving behaviour that is safe and comfortable, predicting the behavior of RUs is crucial, even when facing challenging driving conditions or faulty data. To evaluate and increase robustness of prediction models, *perturbations* are often used (Roelofs et al., 2022).

Perturbations introduce synthetic variations to testing data, $\mathcal{I}' \sim \mathcal{I}_{test}$, to simulate unrecorded data. Perturbations can be *complementing*, or *disruptive*. Complementing perturbations aim to simulate unrecorded data which can be expected in reality. For instance Roelofs et al. (2022) show how the removal of irrelevant agents from the scene (e.g. parked vehicles) has a detrimental effect on predictive accuracy, and Zhang et al. (2022) show that subjecting the models to adversarial attacks (i.e. crafting new realistic historical trajectories) leads to lower accuracy. Disruptive perturbations aim to lower the reliability of the input data, most typically by simulating noise (Zamboni et al., 2022). Increasing robustness towards perturbations is often addressed introducing similar perturbations to the training data, $\mathcal{I}_{\text{train}}$ (Zhang et al., 2022; Zamboni et al., 2022).

Our work falls under the category of disruptive perturbations. In particular, we evaluate robustness towards extreme perturbations that we observed while deploying prediction models in a real vehicle (i.e. complete road unavailability, late detections with only one observation, and highly noisy heading measurements). Additionally, we investigate the effectivity of introducing similar perturbations during training.

2.3 Data Preparation for a Trajectory Prediction Model

To reduce model complexity and required training time during model development, it is common practice to modify its inputs $\mathcal{I}$ before prediction with a 2-step process:

Firstly, instead of operating on world or AV coordinates (Fig. 1a), the RU's reference frame is used, shifting the scene to be centered on the last observed target position (Fig 1b), and then rotating it by the current RU's heading angle (Fig 1c). Secondly, all features are scaled to a common range, or to achieve zero mean and unit variance.

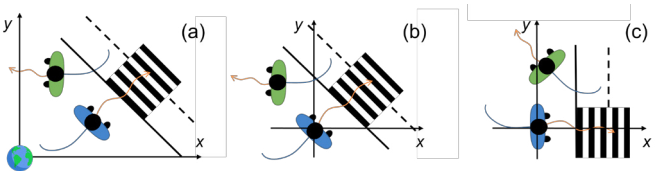

Fig. 1. Data preprocessing of trajectory prediction models.

[1] https://bit.ly/IFAC23-robustness-benchmark

3. METHODOLOGY & EXPERIMENTS

3.1 Benchmarked Models

Four models are evaluated: constant velocity (CV); an LSTM encoder-decoder similar to Muñoz Sánchez et al. (2022), MotionCNN (Konev et al., 2021), and MultiPath++ (Varadarajan et al., 2022; Konev, 2022). CV and LSTM are common baselines, and MotionCNN and MultiPath++ are state of the art models leveraging road geometry and surrounding RUs in their predictions.

Constant Velocity To make predictions for RU n with CV, the only required input is its current state consisting of positions and velocities, $\mathbf{s}_t = (\mathbf{x}_t, \mathbf{v}_t)$ for $t = 0$ (i.e. current time). Thus $\mathcal{I} = (\mathbf{s}_0^n)$. Then, for a given prediction horizon t', its future positions are computed as

$$\hat{\mathbf{x}}_{t+t'}^n = \mathbf{x}_t^n + t' \cdot \mathbf{v}_t^n.$$

LSTM A vanilla LSTM encoder-decoder with 3 layers of 128 neurons for the encoder and decoder that does not consider road information or surrounding RUs. With LSTM, the state of an RU at time t is given by

$$\mathbf{s}_t = (\mathbf{x}_t, \theta_t, \mathbf{v}_t, w, l, \nu_t, \boldsymbol{\tau}),$$

where θ denotes heading angle; w and l width and length; $\nu \in \{0, 1\}$ indicates validity (e.g. the target was temporarily occluded); and $\boldsymbol{\tau}$ is a one-hot encoding indicating the RU type (i.e. unset, vehicle, pedestrian, cyclist, or other). In the data we used, the length of historical and future observations are 1 and 8 seconds respectively, sampled at 10Hz. Thus, to make predictions of RU n at time t with LSTM, its inputs are $\mathcal{I} = (\mathbf{s}_{t-1:t}^n, T)$, where

$$T = \{\frac{t}{10} \mid 0 < t \leq 80 \wedge t \in \mathbb{Z}\}. \tag{2}$$

Its predictions $\mathcal{P} = (\hat{\mathbf{x}}_T^n)$ are produced recursively given the previous position and the network's hidden and cell states (Park et al., 2018).

MotionCNN A convolutional neural network-based architecture (Konev et al., 2021). Its inputs are raster images of 224x224 pixels with 25 channels, where the first three channels encode road geometry, the next eleven encode the past and current positions of the target, and the last 11 all other RUs. Thus, to make predictions of RU $n \in N$ at time t, MotionCNN uses $\mathcal{I} = (\mathbf{s}_{t-1:t}^N, T, \mathbf{m})$ with T as in (2), and makes predictions $\mathcal{P} = (\hat{\mathbf{x}}_T^n, \mathbf{p})$ with probabilities $\mathbf{p}$ of the predicted trajectories.

MultiPath++ The state of an RU at time t is given by

$$\mathbf{s}_t = (\mathbf{x}_t, \theta_t, u_t, w, l, \nu_t, \tau, \Delta\mathbf{x}_t, \Delta\theta_t, \Delta u_t, \Delta\nu_t),$$

where u denotes speed, and Δ denotes the change in the variable that follows with respect to the previous time. Thus, to make predictions of RU $n \in N$ at time t, MultiPath++ uses $\mathcal{I} = (\mathbf{s}_{t-1:t}^N, T, \mathbf{m})$ with T as in (2), and makes predictions $\mathcal{P} = (\hat{\mathbf{x}}_T^n, \mathbf{p}, \Sigma)$ with probabilities $\mathbf{p}$ and covariance matrix Σ of the predicted trajectories.

3.2 Baseline Evaluation

To establish a baseline performance, we evaluate the models under nominal conditions (i.e. training and evaluating them with the original data). To that end, we use the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021). To reduce computational load, we only used trajectories of RUs labeled *tracks to predict*, which feature more diverse behavior than the rest (Ettinger et al., 2021). Since the ground truth for the test data is kept hidden for motion prediction challenges [2], we reserved 1/3 of the original validation split for testing. Thus, our testing data consists of trajectories of 54903 vehicles, 6958 pedestrians and 1784 cyclists. Prediction accuracy is reported by means of the predictions' minADE, as defined in (1), over all prediction horizons T as defined in (2).

3.3 Robustness Evaluation with Perturbations

Based on our experience deploying trajectory prediction models in a vehicle, we focus on three highly detrimental perturbations that can render the models unusable. Such perturbations are unavailability of road information; late detections; and highly noisy RU's heading angle.

Missing road information Availability of road information is not guaranteed. If models are always trained with it, and it is temporarily missing, the results could be unpredictable. To evaluate this case, all road information is removed.

Late detections When perturbing the dataset to simulate perceptual loss, it is common to drop some observations of the historical trajectories with certain probability (Konev, 2022). This way, even if there are gaps in the observed trajectories, the models can exploit the remaining historical data to produce better predictions. With late detections, however, there is no historical data to exploit, thus prediction accuracy will naturally suffer. To investigate the potential performance degradation when lacking past observations, only the most recent observation is considered.

(Highly) noisy heading It is common practice to transform the inputs to a trajectory prediction model relative to the RU's coordinate frame, as introduced in Section 2.3. If the last observed RU state is highly noisy, the entire scene will be transformed erroneously, which can have a highly detrimental effect on the predictions. During model deployment in the vehicle, higher heading angle errors than anticipated were observed, resulting in invalid predictions because the models were never trained with similar noise levels. To evaluate robustness towards extreme cases, we introduce a 90 degree offset in the last observed target heading angle.

As illustrated in Fig. 2(top), for each perturbation introduced we report its minADE averaged over all trajectories using the perturbed dataset, its performance degradation Δ, and its relative increase $\%\Delta$ given by

$$\Delta = \text{minADE\_perturbed} - \text{minADE\_original},$$

$$\%\Delta = \frac{\Delta}{\text{minADE\_original}}.$$

Additionally, to investigate the impact of these perturbations for each prediction, we also analyze the change of minADE per trajectory.

[2] `https://waymo.com/open/challenges/`

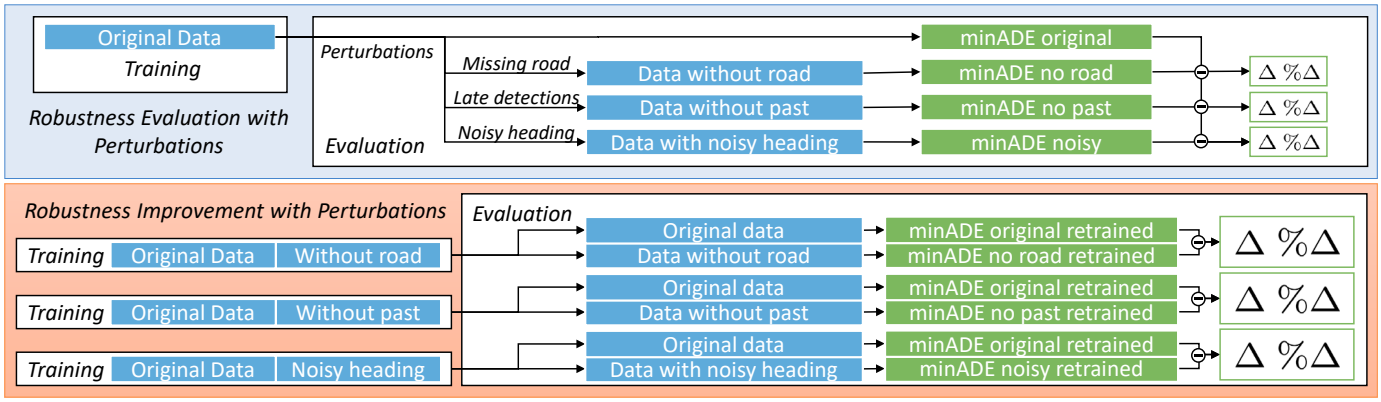

Fig. 2. Overview of the benchmarking procedure to assess the robustness of models towards different perturbations.

3.4 Robustness Improvement with Perturbations

For each perturbation, a new dataset is generated consisting of the original data and a perturbed copy. As shown in Fig. 2(bottom), a new version of the models is then trained with each of the new perturbed datasets, after which a similar evaluation is performed to assess the robustness improvement. Additionally, we report the minADE relative increase of each model evaluated on the original data after it is trained with perturbed data.

4. RESULTS

4.1 Robustness Towards Perturbations

Table 1 shows the performance of each model when provided with the original and perturbed data. As expected, MotionCNN and MultiPath++ clearly outperform CV and LSTM, since they leverage environmental information such as road geometry and surrounding RUs. Additionally, Fig. 3(top) shows a density plot of Δ for each individual prediction instead of averaged over all trajectories. Finally, Fig. 4(top) shows examples of predictions produced by MultiPath++ given the original and perturbed data.

Missing road information CV and LSTM do not use road information, and are therefore unaffected by its absence. On the other hand, MotionCNN and MultiPath++ suffer a severe performance drop (+110.79% and +71.64% respectively), although they still outperform the environment-unaware models. Despite this average performance drop, there are several cases where prediction accuracy improves after removing the road, as seen by the negative Δ in Fig. 3a. Fig. 4b shows an example of a prediction where the lack of road information has a negative impact, causing off-road predictions.

Late detections CV only uses the most recent observation, and as such is unaffected by lack of past observations. LSTM and MultiPath++ undergo an error increase of approximately +85%, resulting in higher errors than those of CV in the case of LSTM. Since MotionCNN's raster inputs do not contain information about the target's current velocities, it is unable to infer it from its past positions and its minADE almost triples (+184.89%). Despite having road information, the models sometimes give predictions going off-road (Fig. 4c) due to lack of historical data, which suggests the model was unable to infer behavior that is obvious to humans: vehicles drive on the road (at least in nominal conditions). Nevertheless, under this perturbation there are also a significant number of predictions for which minADE improves (Fig. 3b).

Noisy heading angle CV does not use heading angle, as the velocities in each direction are provided as input directly, therefore it remains unaffected by this perturbation. All machine learning models suffer an extreme performance degradation of up to +1444.78%, resulting in significantly worse performance than that of CV. This performance degradation is to be expected, since the models have been trained modifying their inputs as described in Section 3.2 to reduce model complexity and training times, and if the entire scene is rotated with a wrong angle, they are unable to produce sensible predictions.

Table 1. Model minADE with original data and perturbed variants

Model	Original	Missing Road			Late Detection			Noisy Heading		
			Δ	%Δ		Δ	%Δ		Δ	%Δ
CV	8.83	-	-	-	-	-	-	-	-	-
LSTM	5.16	-	-	-	9.44	4.28	82.95%	23.45	18.29	354.46%
MotionCNN	1.39	2.93	1.54	110.79%	3.96	2.57	184.89%	18.74	17.35	1248.20%
MultiPath++	1.34	2.3	0.96	71.64%	2.49	1.15	85.82%	20.7	19.36	1444.78%

Table 2. Model minADE after training with original and perturbed data

Model	Missing Road				Late Detection				Noisy Heading			
	Original	Perturbed	Δ	%Δ	Original	Perturbed	Δ	%Δ	Original	Perturbed	Δ	%Δ
LSTM	-	-	-	-	5.17 (+0.19%)	9.37	4.20	81.24%	5.39 (+4.46%)	6.3	0.91	16.88%
MotionCNN	1.35 (-2.88%)	1.59	0.24	17.78%	1.34 (-3.60%)	1.75	0.41	30.60%	1.45 (+4.32%)	1.46	0.01	0.69%
MultiPath++	1.56 (+16.42%)	1.81	0.25	16.03%	1.28 (-4.48%)	2.4	1.12	87.50%	1.33 (-0.75%)	1.31	-0.02	-1.50%

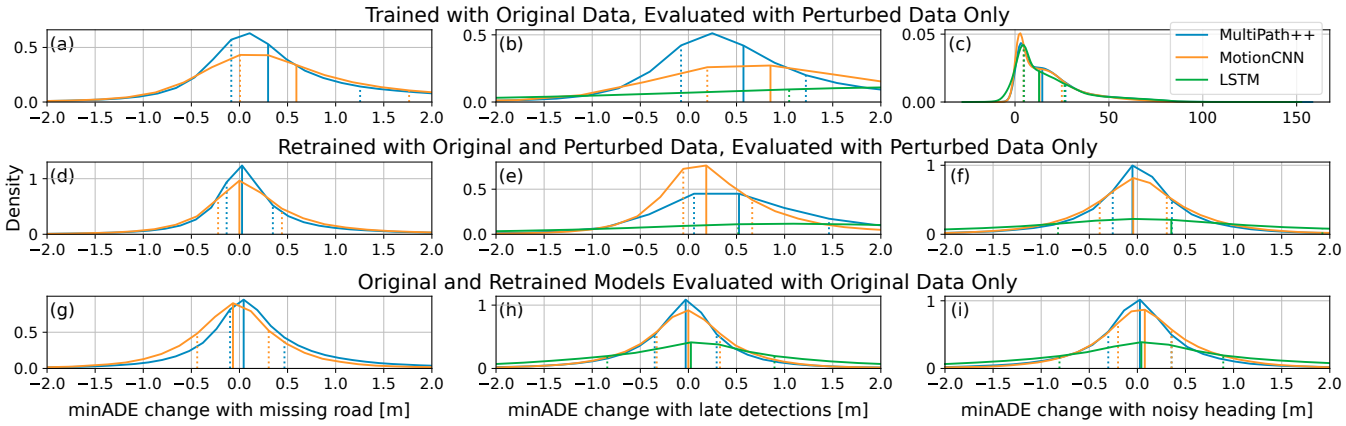

Fig. 3. Density plots of minADE change per trajectory. Top: Δ of original models evaluated on perturbed data. Middle: Δ of retrained models evaluated on perturbed data. Bottom: minADE difference of original and retrained models when evaluated on original data. Solid vertical lines denote the median, and dotted lines the 25th and 75th percentiles. Note: horizontal axis is fixed to [-2,2]m in most plots, and the long tails could extend further than shown.

4.2 Robustness Towards Perturbations after Retraining

Table 2 shows the performance of each machine learning model when provided with the original data and the three perturbations after being trained with original and perturbed data. Additionally, when evaluated using the original dataset, the performance increase with respect to the model that was trained without perturbations is reported. Fig. 3(middle) summarizes Δ of each individual prediction. Finally, Fig. 4(bottom) shows example predictions with a retrained model.

Missing road information Applying this perturbation during training leads to different results. MotionCNN not only achieves lower performance degradation on perturbed data (+17.78% vs. +110.79%), but it also achieves a lower minADE on the original data (-2.88%). Additionally, for half of the perturbed trajectories minADE improves (Fig. 3d), and for more than half on original data (Fig. 3g). On the other hand, MultiPath++ also achieves a lower average performance degradation (+16.03% vs. +71.64%), but increased minADE on the original data (+16.42%). Fig. 4f shows that despite missing road information, there is sufficient information to make significantly better predictions, suggesting that the models were unnecessarily reliant in road information, and learn to better exploit other information in the scene applying this perturbation.

Late detections LSTM achieves a marginally improved performance degradation on perturbed data (+81.24% vs. +82.95%) and a marginal performance degradation on the original data (+0.19%). MotionCNN achieves a significant improvement on perturbed data (+30.60% vs. +184.89%), and lower minADE on the original data (-3.6%). MultiPath++ maintains a similar degradation on perturbed data (87.5% vs. 85.82%), but also lower minADE on the original data (-4.48%). Simulating late detections during training proves beneficial unless past observations are the only additional information the model can leverage.

Noisy heading angle Introducing this perturbation during training yields a significant improvement for all models, but in some cases at the expense of higher minADE with the original data. Performance degradation on perturbed data for LSTM is lowered from +354.56% to +16.88%, and for MotionCNN from +1248.2% to +0.69%. However, minADE increases approximately 4.4% for both models when using original data. On the contrary, MultiPath++ not only achieves a performance improvement on perturbed data (-1.5% vs. +1444.78%), but also slightly lower minADE on original data (-0.75%). For more than half of the trajectories, predicting with the retrained MotionCNN and MultiPath++ using perturbed trajectories yields higher accuracy than using the original trajectories (Fig. 3, bottom).

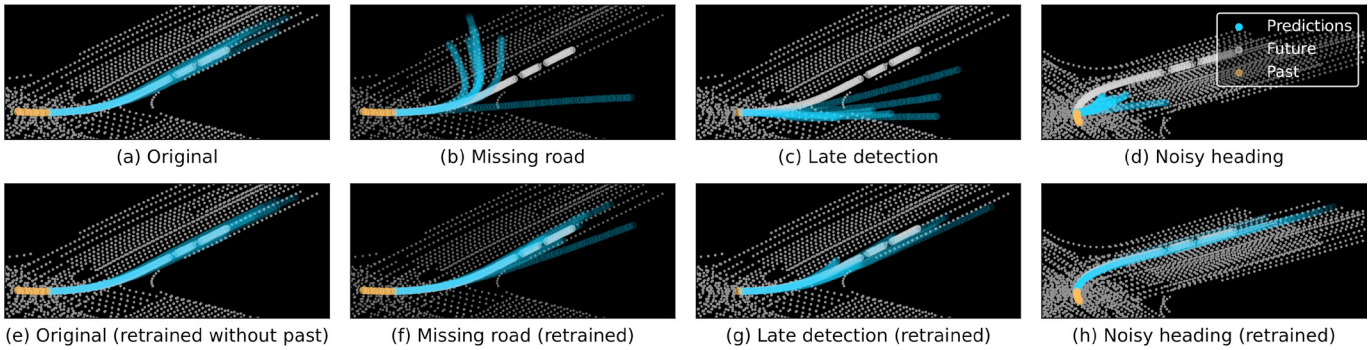

Fig. 4. Examples of MultiPath++ predictions on original and perturbed data (scene 106fb050cdf836af, agent 1117). Trained and tested on original data (a), trained on original data and tested with pertubed data (b-d), trained with both original and perturbed data (e-h), and tested with original (e) and perturbed data (f-h).

4.3 Are Perturbations the Solution to Achieve Robustness?

Perturbations are an effective strategy to improve model robustness. Examples from our analysis are shown in Fig. 4. For instance, the removal of road information initially causes off-road predictions (Fig. 4b), although there seems to be sufficient information in historical data to produce better predictions even without any road information (Fig. 4f). Another example is the removal of historical data leading to off-road predictions (Fig. 4c) despite road information being available, which is prevented after simulating the same perturbation during training (Fig. 4d).

Training with perturbed data for the identified perturbations effectively mitigates performance degradation, although it does not address the fact that if the model is presented with previously unseen data, its behavior can be erratic once again. Guaranteeing robustness with perturbations would therefore require identification of all possible complications to simulate the necessary perturbations, which is likely unfeasible.

Additionally, there are a significant number of cases for which the perturbed data leads to better predictions. For instance, when removing the road with models that were always trained with road (Fig. 3a). Retraining the models without road not only improves performance degradation but also increases the number of cases with better predictions without road (Fig. 3d), suggesting the models may be unable to learn the correct relationships in the data.

5. CONCLUSIONS & FUTURE WORK

To enable better trajectory prediction of other road users in real-life experiments, in this work we have compared several models against three types of perturbations we observed when deploying our models in a vehicle, and found that their predictions are severely degraded with minADE increases of up to +110.8% when road information is unavailable, up to +184.9% with late detections, and up to +1444.8% when the last observed heading angle contains high errors.

We show that introducing similar perturbations during model training is effective to mitigate erratic predictions, leading to much lower minADE increases of up to +17.8% when road information is unavailable, up to +87.5% with late detections, and up to +16.9% with noisy heading angles. Despite its effectiveness, this approach requires identification of relevant perturbations before model deployment, and there will likely be new unforeseen circumstances that were not considered and lead to erratic predictions again. Thus, preemptively introducing these perturbations, while effective, does not ensure model robustness.

Future work will focus on three aspects. Firstly, a comprehensive analysis of how robustness improvement is affected by the severity of the perturbations and the proportion of perturbed data used during retraining. Secondly, an evaluation of the in-vehicle robustness enhancement after retraining with simulated perturbations. Lastly, further examination of cases where degraded inputs yield improved predictions, which suggest the models do not learn the right relationships between the elements in the scene.

REFERENCES

Alahi, A. et al. (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *CVPR*, 961–971.

Bahari, M. et al. (2022). Vehicle trajectory prediction works, but not everywhere. In *CVPR*, 17102–17112.

Cao, Y., Xiao, C., Anandkumar, A., Xu, D., and Pavone, M. (2022). AdvDO: Realistic Adversarial Attacks for Trajectory Prediction. volume 13692 of *Lecture Notes in Computer Science*, 36–52.

Ettinger, S. et al. (2021). Large Scale Interactive Motion Forecasting for Autonomous Driving : The Waymo Open Motion Dataset. In *ICCV*, 9690–9699.

Konev, S. (2022). MPA: MultiPath++ Based Architecture for Motion Prediction. 5–7.

Konev, S., Brodt, K., and Sanakoyeu, A. (2021). MotionCNN: A Strong Baseline for Motion Prediction in Autonomous Driving.

Kosaraju, V. et al. (2019). Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. In *Advances in Neural Information Processing Systems*.

Muñoz Sánchez, M., Elfring, J., Silvas, E., and van de Molengraft, R. (2022). Scenario-based Evaluation of Prediction Models for Automated Vehicles. In *ITSC*, 2227–2233.

Park, S.H., Kim, B., Kang, C.M., Chung, C.C., and Choi, J.W. (2018). Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture. In *Intelligent Vehicles Symposium*, 1672–1678.

Rasouli, A. (2020). Deep Learning for Vision-based Prediction: A Survey.

Roelofs, R. et al. (2022). CausalAgents: A Robustness Benchmark for Motion Forecasting using Causal Relationships. 1–36.

Saadatnejad, S., Bahari, M., Khorsandi, P., Saneian, M., Moosavi-Dezfooli, S.M., and Alahi, A. (2022). Are socially-aware trajectory prediction models really socially-aware? *Transportation Research Part C: Emerging Technologies*, 141, 1–17.

Salzmann, T., Ivanovic, B., Chakravarty, P., and Pavone, M. (2020). Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In *Lecture Notes in Computer Science*, volume 12363 LNCS, 683–700.

Varadarajan, B. et al. (2022). MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction. In *ICRA*, 7814–7821.

Yoon, Y., Kim, T., Lee, H., and Park, J. (2020). Road-aware trajectory prediction for autonomous driving on highways. *Sensors (Switzerland)*, 20(17), 1–20.

Zamboni, S., Kefato, Z.T., Girdzijauskas, S., Norén, C., and Dal Col, L. (2022). Pedestrian trajectory prediction with convolutional neural networks. *Pattern Recognition*, 121, 108252.

Zang, S., Ding, M., Smith, D., Tyler, P., Rakotoarivelo, T., and Kaafar, M.A. (2019). The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car. *IEEE Vehicular Technology Magazine*, 14(2), 103–111.

Zhang, Q., Hu, S., Sun, J., Chen, Q.A., and Mao, Z.M. (2022). On Adversarial Robustness of Trajectory Prediction for Autonomous Vehicles.