Learning-enabled multi-modal motion prediction in urban environments

Vinicius Trentin
Center for Automation and Robotics
CSIC - Universidad Politécnica de Madrid
Arganda del Rey, Spain
vinicius.trentin@csic.es

Chenxu Ma
Computer Engineering Lab
TU Delft
Delft, The Netherlands
c.ma-6@student.tudelft.nl

Jorge Villagra
Center for Automation and Robotics
CSIC - Universidad Politécnica de Madrid
Arganda del Rey, Spain
jorge.villagra@csic.es

Zaid Al-Ars
Computer Engineering Lab
TU Delft
Delft, The Netherlands
Z.Al-Ars@tudelft.nl

***Abstract*—Motion prediction is a key factor towards the full deployment of autonomous vehicles. It is fundamental in order to assure safety while navigating through highly interactive complex scenarios. In this work, the framework IAMP (Interaction-Aware Motion Prediction), producing multi-modal probabilistic outputs from the integration of a Dynamic Bayesian Network and Markov Chains, is extended with a learning-based approach. The integration of a machine learning model tackles the limitations of the ruled-based mechanism since it can better adapt to different driving styles and driving situations. The method here introduced generates context-dependent acceleration distributions used in a Markov-chain-based motion prediction. This hybrid approach results in better evaluation metrics when compared with the baseline in the four highly-interactive scenarios obtained from publicly available datasets.**

***Index Terms*—motion-prediction, interaction-aware, learning-based**

I. Introduction

The anticipation of possible dangerous driving situations is fundamental to take preventive actions and to minimize potential risks accordingly. Indeed, in order to perform safe and efficient motion planning, autonomous vehicles need to predict the evolution of other traffic participants. However, the behavior of the surrounding agents is full of uncertainties in the real world and depends on the layout of the driving scene and on their interactions with others [1]. As a result, a reliable and robust mechanism for intention estimation and motion prediction is critical for autonomous vehicles.

The motion prediction problem has a multi-modal nature since a vehicle has many possible trajectories while navigating through the layout. To solve the multi-modal motion prediction problem, two categories of solutions are usually applied: model-based and learning-based. Model-based methods try to mathematically formalize the dynamics of the problem, whereas in learning-based approaches, the interactions and dynamics of the system are learned from data.

Model-based approaches include stochastic filters. In [2], the author predicts the probabilistic future states of the surrounding vehicles and the uncertainty of these predictions via a self-adaptive motion predictor. He does so by considering a kinematic model of the vehicles' motion and a Kalman filter to estimate the uncertainty. [3] proposed a motion prediction scheme based on an Interactive Multiple Model Kalman Filter that is able to infer the intentions and generate interaction-aware non-colliding predictions of multiple vehicles considering a priority list. Due to the uncertainty produced by the Kalman Filter, these methods should only be used to predict short prediction horizons. To analyze the interaction between vehicles and predict their route and maneuver intentions, the authors in [4] employed a Dynamic Bayesian Network with a particle filter. An action, indicated by acceleration and yaw rate values, is derived from these intentions, and the motion prediction is computed. This technique evaluates just the most likely action for the whole prediction horizon, which may have a detrimental impact on the motion planning search space in complicated scenarios.

In previous works from the authors [5], [6], they introduced a framework based on the combination of a Dynamic Bayesian Network to infer intentions and a Markov chain-based motion prediction to estimate the motion of the surrounding vehicles. Although the multi-modal model-based framework previously implemented can be successfully applied to any driving situation or scenario, it cannot adapt to different driving styles.

In recent years, many neural network architectures have been employed to model the surrounding context and interactions among traffic agents in motion prediction. [7] use a sparse context encoder to extract agent and map attributes, a dense goal encoder to retrieve the goal probability distribution, and a goal set predictor to provide predicted trajectories based on the goal probability distribution. [8] describes a cooperative multi-agent trajectory prediction system that uses a graph

This work has been partially funded by the Spanish Ministry of Science and Innovation with the National Project NEWCONTROL (PCI2019-103791), the Community of Madrid through SEGVAUTO 4.0-CM Programme (S2018-EMT-4362), and by the European Commission and ECSEL Joint Undertaking through the Project NEWCONTROL (826653).

encoder and a grid decoder to make goal-based predictions. It takes the agents' history and a lanelet map as inputs and generates a heatmap of potential ending locations, from which K trajectories are created by decoding K end points and generating the path with a fully connected neural network. [9] presents a network architecture based on a stacked transformer for modeling multi-modal predictions using predetermined trajectories collected from traffic data as input. [10] predicts the route of the target vehicles using a hierarchical graph neural network, which receives vector representations of agents and maps as input and retrieves interactions between agents via a fully-connected graph. [11] employs trajectory anchors from the training dataset to model intent uncertainty using a Gaussian Mixture Model whose parameters are generated using a deep neural network. Notice that despite the very promising prediction results of most of these methods, data-driven approaches rely on a vast amount of data, and the results obtained with them can degrade when applied to different scenarios from those selected for their training. Furthermore, these methods are usually based on complex machine learning systems that have limited interpretability.

In this paper, the previous work [5], [6] used to infer intentions and compute the motion prediction of the surrounding vehicles is extended to consider a learning-enabled motion prediction. The neural network model implemented provides personalized acceleration profiles from the most relevant information about the vehicle and its interactions with the layout and with other vehicles. These profiles serve as input to the Markov chain-based motion prediction, resulting in a hybrid approach that combines both model- and learning-based techniques. It can be applied to urban scenarios with high interactions where the autonomous vehicle needs accurate predictions in order to safely navigate, such as roundabouts and intersections. The extended framework is compared with the baseline in four situations obtained from publicly available datasets.

The outline of this paper is organized as follows: Section II presents a review of IAMP framework. Section III introduces the model implemented to generate the acceleration profiles. Section IV presents the scenarios being used and shows the results for the 4 driving environments. Finally, Section V provides some concluding remarks.

II. Interaction-aware motion prediction

The interaction-aware motion prediction uses a combination of Dynamic Bayesian Network (DBN) and Markov chains to infer the intentions and predict the motion of the surrounding vehicles. It explicitly considers interactions, road layout, and traffic rules.

At each time step, the vehicle-to-vehicle and vehicle-to-layout interactions are taken into account to infer the probability to stop or to cross the intersections, the probability to change lanes, as well as the probability of being in each of the possible navigable corridors. These intentions are fused with the motion predictions computed with a kinematic model to result in a motion grid used by the ego vehicle to navigate through the scene.

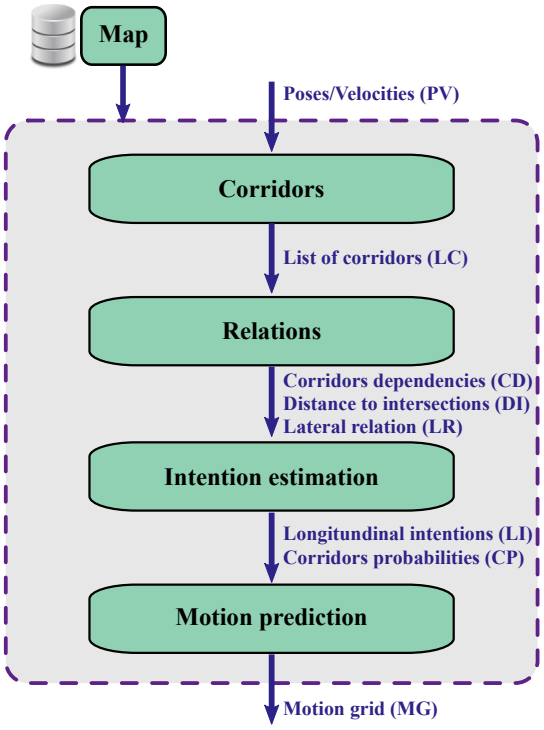


Fig. 1. Interaction-aware motion prediction flowchart.

The grid-based representation of the predictions takes into account the uncertainties both in the motion model and in the input data, resulting in a more reliable and robust prediction when compared with a point-based trajectory prediction.

The framework can be divided in four main blocks: *Corridors*, *Relations*, *Intentions* and *Motion Prediction*. An overview of the framework is presented in Figure 1 where the data entering and leaving each block is shown. Each block will be briefly described below. For more details, the reader is referred to [5], [6].

A. Maps

The maps are loaded at the beginning of the simulation. They are made up of lanelets [12], which are interconnected driveable segments geometrically represented by a left and a right bound. The regulatory elements, which are linked to the lanelets, can indicate which lanelets have right of way, which must yield, and the exact placement of the stop lines in the case of an unsignalized intersection.

B. Corridors

A corridor is a lanelet sequence that represents a possible route a vehicle might take. They are obtained from the physical and relational layers extracted from the lanelets. The extension of these corridors is limited to the distance that the vehicle can travel in a given time interval at its current velocity, assuming constant acceleration.

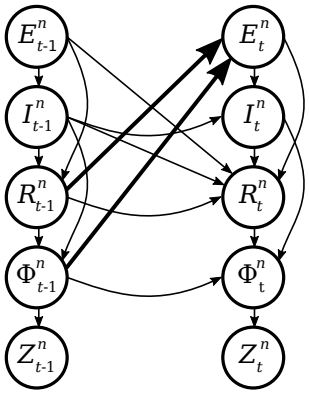


Fig. 2. Bayesian network for two consecutive time steps.

C. Relations

The interactions obtained from the list of corridors and the map are threefold: lateral relation, corridor-to-intersection (distance to intersections) and corridor-to-corridor (corridors dependencies);

- lateral relation: for each target vehicle, a search of the surrounding vehicles is performed, and the bumper-to-bumper distance and velocities are stored.
- corridor-to-intersection: the intersections each corridor goes through are determined by intersecting the lanelet's identifiers. For all intersections a corridor goes through, the distance to the intersection is computed in the Frenet frame. The entrance through which the corridor passes is also determined.
- corridor-to-corridor: determine which corridor will influence the predictions of the other acting as an obstacle ahead. The centerlines of all corridors are pairwise intersected, generating a list of possible collisions. Based on a set of rules, one (if any) corridor is selected as the corridor influencing the motion of a vehicle in a given corridor.

D. Intentions

The Dynamic Bayesian Network (DBN) described in [13] and implemented in [5], [6] is used to estimate the intention of traffic participants. The network represented in Figure 2 is instantiated for all vehicles present in the scene, where bold arrows represent the influences of the other vehicles on vehicle n through some key variables (E^n_t, I^n_t, R^n_t, Φ^n_t, Z^n_t).

- Expected maneuver E^n_t: reflects the expected behavior of the vehicle n at moment t in accordance with traffic laws.
- Intended maneuver I^n_t: reflects the intention of the vehicle.
- Route R^n_t: estimated route the vehicle is following.
- Physical vehicle state Φ^n_t: represents the pose, curvature, and speed of the vehicle. They are determined at each instant based on the vehicles' intentions.
- Measurements Z^n_t: represents the real measurements of the physical state of the vehicle, obtained directly from exteroceptive sensors of the EV or via V2X communications [14].

The relationships between the variables in Fig. 2 allow the following generalized distribution to be used to model the driving scene:

$$P(\boldsymbol{E_{0:T}}, \boldsymbol{I_{0:T}}, \boldsymbol{R_{0:T}}, \boldsymbol{\Phi_{0:T}}, \boldsymbol{Z_{0:T}}) = P(\boldsymbol{E_0}, \boldsymbol{I_0}, \boldsymbol{R_0}, \boldsymbol{\Phi_0}, \boldsymbol{Z_0}) \times \prod_{t=1}^{T} \times \prod_{n=1}^{N} [P(E^n_t | \boldsymbol{R_{t-1}}, \boldsymbol{\Phi_{t-1}}) \times P(I^n_t | I^n_{t-1}, E^n_t) \times P(\Phi^n_t | \Phi^n_{t-1}, R^n_t, I^n_t) \times P(Z^n_t | \Phi^n_t)] \tag{1}$$

Since an exact inference of (1) is generally impractical, a particle filter is employed to estimate the hidden states $\mathbf{E_t}$, $\mathbf{I_t}$, $\mathbf{R_t}$ and $\mathbf{\Phi_t}$, given the observed variables $\mathbf{Z_t}$.

E. Motion Prediction

The computation of the predictions of the surrounding vehicles is inspired by the method proposed by [15]. The system dynamics are abstracted into Markov chains, where the state space X and input space U are discretized into intervals. The state space consists of longitudinal position s and velocity v, and the input space represents the potential acceleration.

The transition probability matrices of the Markov chains for a time step $\Upsilon(\tau)$, and for a time interval $\Upsilon([0,\tau])$, where τ is the time increment, are computed offline with reachability analysis [15], using the following differential equation as the vehicle's longitudinal dynamics:

$$\begin{aligned} \dot{s} &= v \\ \dot{v} &= \begin{cases} a^{max} u, & 0 < v < v^{max} \\ 0, & v \leq 0 \vee v \geq v^{max} \end{cases} \end{aligned} \tag{2}$$

where a^{max} and v^{max} are the maximum allowed acceleration and velocity, respectively, and u is sampled for the discretized input space U.

The states probability distributions for future time steps $p(t_{k+1})$ and time intervals $p(t_k, t_{k+1})$ are computed as follows:

$$\begin{aligned} p(t_{k+1}) &= \Gamma(t_k)\Upsilon(\tau)p(t_k) \\ p(t_k, t_{k+1}) &= \Upsilon([0,\tau])p(t_k) \end{aligned} \tag{3}$$

where t_k is the discrete time step and $\Gamma(t_k)$ is the input transition matrix that represents the transition probabilities between the input states. This matrix is computed for each prediction step, considering two parts: a matrix Ψ that depicts the vehicle's inherent behavior when there are no priorities for certain input values (all inputs will eventually have the same probability) and a priority vector λ that takes into account the dependencies between corridors, the road layout, and the velocity limitations. The dependencies between corridors are computed considering an interaction matrix that contains the probability of collision between two states given two acceleration inputs, one for each state. When this matrix is multiplied with the predictions of the corridor causing the

dependency, it results in a probability vector that represents the percentage that an acceleration input is allowed at a given state. It is a costly operation since the interaction matrix is dense and its size depends on the number of states and inputs.

The predictions of each corridor are fused together with their probabilities, inferred with the DBN, to generate a motion grid. An example of the generated grid for the last instant of the time horizon is shown in Figure 5a.

III. Learning model

In this work, the aforementioned architecture is enhanced with the integration of a learning approach. This implementation tackles the limitation of the current motion prediction module, since it cannot adapt to different driving styles and driving scenarios. To do so, the priority vector λ for the acceleration input is replaced by an acceleration profile generated as the output from a neural network model. These acceleration profiles take into account vehicle-to-vehicle and vehicle-to-layout interactions, as well as the history data from the target vehicle.

The replacement of the acceleration input greatly reduces the computational time of the system. Besides the time component, machine learning approaches can increase the accuracy of the predictions since the acceleration profile generated can self-adapt to different driving contexts and driver styles.

For the aforementioned task, an Auto-Regressive (AR) neural network [16] was selected due to its ability to describe time-varying processes and to predict future behavior based on past behavior. The AR model has been used in monitoring systems [17], financial time series [18] and chemistry [19].

The model takes as input the history of 4 s containing the acceleration a_{in}, the curvature of the path, the distance to the next intersection d_{int}^{target}, the distance to the leading vehicle d_{lead} and its velocity v_{lead}, and the two vehicles that the most influence the path of the corridor in the next intersection (if exist), represented by their distance to the intersection d_{int}^{n}, their velocity v_{int}^{n} and their priority p_{int}^{n} with respect to the target vehicle. These inputs are represented in Figure 3 and the variables are described in Table I. To obtain the curvature of the path, the centerline is equally divided into 6 segments and the positive and negative values of the curvature of each segment are integrated resulting in 12 discrete values (κ_p^i, κ_n^i), with $i \in \{1, \ldots, 6\}$ [20].

To train the model, the Pytorch Lightning framework was used. For the optimization, the Adam algorithm [21] was selected, considering mean square error (MSE) and mean average error (MAE) as the losses. The learning rate was annealed in the form of a cosine curve every 10 epochs during the training procedure.

The basic principle of the model is expressed in the following equation:

$$y_t = W x_{t-1} + b + \epsilon_t \tag{4}$$

where ϵ_t is the white noise at timestamp t, x_{t-1} is the input state in timestamp $t-1$, W and b are the weight matrix and bias, respectively, that are needed to be updated. The output y_t from the model is an acceleration time series representing 4 s into the future discretized at 0.1 s.

TABLE I
Input features variables.

Parameter	Variable
target vehicle's acceleration	a_{in}
distance to leading vehicle	d_{lead}
leading vehicle's velocity	v_{lead}
target vehicle's distance to the intersection	d_{int}^{target}
curvature parameters	$(\kappa_p^i, \kappa_n^i), i \in \{1, \ldots, 6\}$
intersection parameters	$(d_{int}^j, v_{int}^j, p_{int}^j), j \in \{1, 2\}$

TABLE II
Information about the scenarios

Situation	Number of vehicles	Duration
A	8	15.1 s
B	6	19.8 s
C	18	34.2 s
D	15	28.4 s

The datasets openDD [22] and inD [23] were used to generate the input data to train the model. The whole dataset has been processed with the framework from Figure 1 to generate the input for the training.

IV. Results

A. Scenarios

From the datasets inD [23] and INTERACTION [24], 4 situations were extracted following the procedures described in [5]. Table II contains the information regarding the number of vehicles and duration of each situation followed by a brief description of each of them.

- Situation A: four-armed intersection with two center left-turn lanes. The crossroad contains a lane with the right of way.
- Situation B: four-armed intersection where no lane has the right of way over the other.
- Situation C: T-junction intersection where the main road has the right of way and there is a left turn lane into the side road.
- Situation D: single-lane roundabout containing 3 entrances and 3 exits.

B. Evaluation metrics

For the comparison with the physical-base model, two evaluation metrics have been implemented to evaluate the quality of the predictions: *ADE* (Average Displacement Error) and *FDE* (Final Displacement Error). Since the predictions are multi-modal, these metrics are used in the min form, which

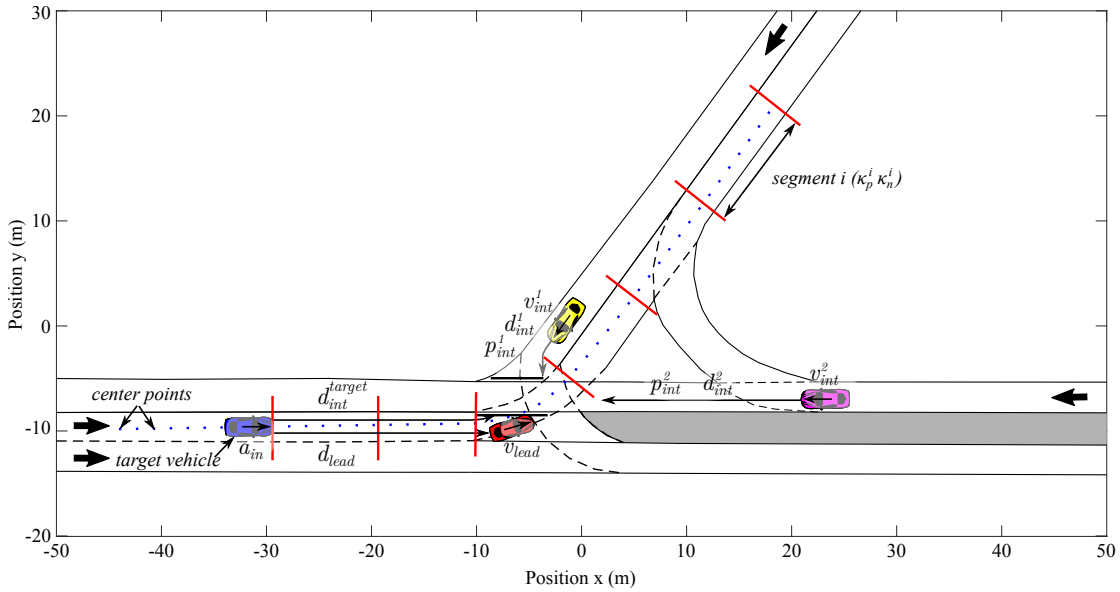


Fig. 3. Input features example.

means that for k given corridors for the same vehicle, the minimum value is selected.

- ADE: average L_2 distance between the ground truth positions and the weighted average position of the predictions.

$$\text{ADE}_k = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i^{GT}-\hat{x}_i)^2+(y_i^{GT}-\hat{y}_i)^2}$$
$$\text{mADE} = \text{argmin}_k\,\text{ADE}_k$$

- FDE: L_2 distance between the last ground truth position and the weighted average position of the last prediction.

$$\text{FDE}_k = \sqrt{(x_n^{GT}-\hat{x}_n)^2+(y_n^{GT}-\hat{y}_n)^2}$$
$$\text{mFDE} = \text{argmin}_k\,\text{FDE}_k$$

where (x_i^{GT}, y_i^{GT}) is the ground truth position obtained from the dataset, $(\hat{x}_i, \hat{y}_i)$ is the estimated position computed as the weighted average of the predictions and n is the number of predictions.

C. Comparison with baseline

The model was evaluated in the situations described in Section IV-A. It was executed three times to take into account the stochastic nature of the framework, and the results here presented are the average of the executions.

In order to use the outputs from the model in the Markov chain-based motion prediction, the acceleration time series is converted into n normal acceleration distributions. To represent the 4 s prediction horizon, 10 distributions are defined, each representing 0.4 s. These distributions are obtained by the evaluation of the input intervals (ranging from -3 m/s^2 to 2 m/s^2) using a mean and a standard deviation obtained from the equally divided profile, as illustrated in Figure 4. In this figure, the blue line is the ground truth acceleration of the target vehicle, the red line represents the output y_t from the AR model, and the black intensity corresponds to the probability of having this acceleration interval in a given prediction step inside the prediction horizon. Notice that this distribution is repeated in every state of the state space.

An example of the predictions generated with the AR model (hybrid-IAMP) is presented in Figure 5a. In this figure, each footprint is the prediction for the last time step of the prediction horizon and corresponds to a vehicle with the same color. Notice that the prediction for the vehicle arriving from the right entrance respects the priority of the vehicle inside

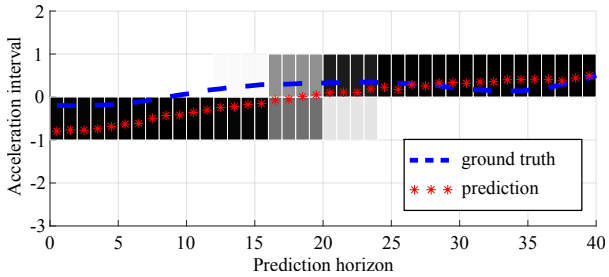
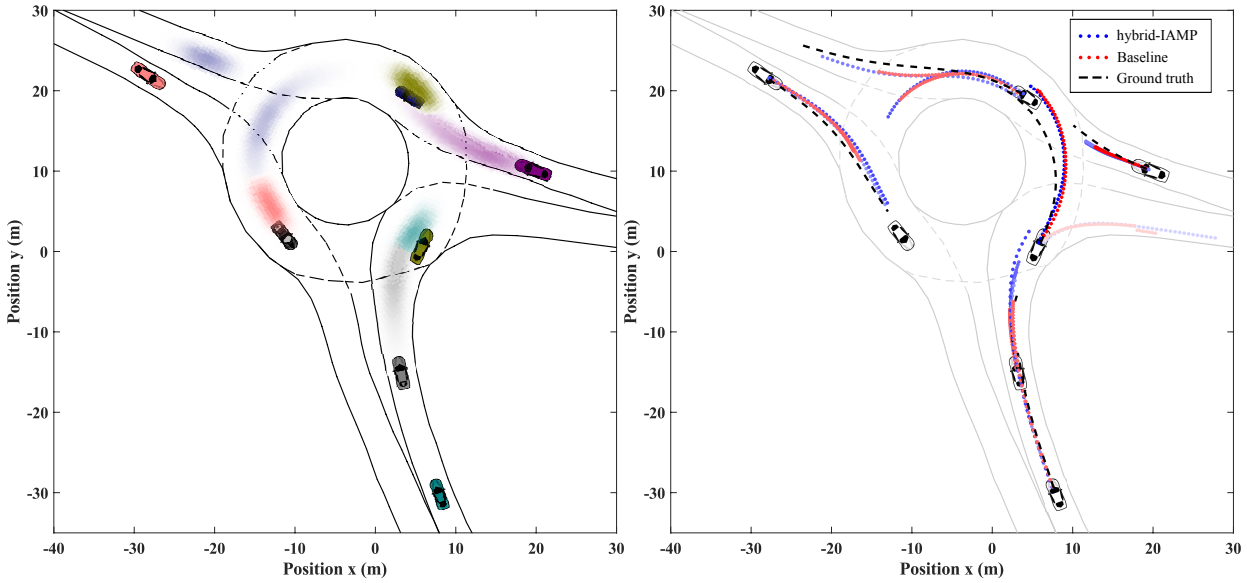


Fig. 4. Example of acceleration distribution obtained from the output of the model.

TABLE III
COMPARISON METRICS FOR THE INSTANT FROM FIGURE 5B.

Metric \ Model	hybrid-IAMP	Baseline
mADE	1.30 m	2.10 m
mFDE	3.73 m	5.74 m



(a) (b)

Fig. 5. (a) Motion grid representing the predictions at 4.0 s in a given time step of the situation with hybrid-IAMP. (b) Comparison of the predictions: hybrid-IAMP (blue), baseline (red) and ground truth (black). The intensity of the predictions represents the probability of the corridor.

TABLE IV
QUANTITATIVE RESULTS.

Situation \ Model	Hybrid-IAMP		Baseline	
A	time	**0.97 s**	time	3.83 s
	mADE	**1.18 m**	mADE	1.43 m
	mFDE	**2.75 m**	mFDE	3.22 m
B	time	**1.14 s**	time	5.96 s
	mADE	**1.62 m**	mADE	1.92 m
	mFDE	**4.49 m**	mFDE	4.83 m
C	time	**0.80 s**	time	5.60 s
	mADE	**1.49 m**	mADE	1.58 m
	mFDE	3.75 m	mFDE	**3.73 m**
D	time	**1.22 s**	time	3.18 s
	mADE	**1.43 m**	mADE	1.96 m
	mFDE	**3.75 m**	mFDE	5.23 m

the roundabout and stays behind its prediction. The black vehicle is the ego vehicle. It also correctly predicts that the vehicle coming from the top left entrance can safely enter the roundabout.

The whole prediction horizon for this same instant is compared with the baseline and with the ground truth in Figure 5b. As mentioned in Section IV-B, the predictions in each time step are converted to the most probable position by computing a weighted average of the probabilistic footprint. Only vehicles having the ground truth position contained in the dataset are considered. As can be seen, the hybrid-IAMP can better adapt to different driving styles, and in general, generate a prediction closer to the ground truth when compared with the baseline. The metrics for this instant are presented in Table III.

Table IV presents the prediction metrics for the hybrid-IAMP and for the baseline. These metrics are the average for the whole simulation time. It also includes the average time it took to compute the predictions. As can be seen, the hybrid implementation outperformed the baseline in both metrics, as well as in computational time.

V. CONCLUDING REMARKS

In this paper, the framework IAMP was extended to include a learning mechanism able to generate personalized acceleration profiles that are used to compute the motion prediction of the surrounding vehicles. The hybrid-IAMP provided better performance in the predictions metrics and reduced computational cost when applied to four use cases extracted from the publicly available datasets *inD* and *INTERACTION*, proving that a hybrid approach can adapt better to different driving styles and driving scenarios.

As future work, more complex models will be implemented at this and other stages of the framework, always trying to improve the adaptability to the context while guaranteeing its interpretability.

REFERENCES

[1] J. Villagra, M. Clavijo, A. Díaz-Álvarez, and V. Trentin, "Motion prediction and risk assessment," in *Decision-Making Techniques for Autonomous Vehicles*. Elsevier, 2023, pp. 61–101.
[2] Y. Jeong, "Self-adaptive motion prediction-based proactive motion planning for autonomous driving in urban environments," *IEEE Access*, vol. 9, pp. 105 612–105 626, 2021.
[3] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger, "Interaction-aware motion prediction for autonomous driving: A multiple model kalman filtering scheme," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 80–87, 2021.
[4] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3999–4006.
[5] V. Trentin, A. Artuñedo, J. Godoy, and J. Villagra, "Interaction-aware intention estimation at roundabouts," *IEEE Access*, vol. 9, pp. 123 088–123 102, 2021.

[6] V. Trentin, A. Artuñedo, J. Godoy, and J. Villagra, "Interaction-aware motion prediction at highways: A comparison of three lane changing models," in *Smart Cities, Green Technologies, and Intelligent Transport Systems*, C. Klein, M. Jarke, M. Helfert, K. Berns, and O. Gusikhin, Eds. Cham: Springer International Publishing, 2022, pp. 274–296.

[7] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 283–15 292.

[8] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "THOMAS: Trajectory Heatmap Output with learned Multi-Agent Sampling," in *International Conference on Learning Representations*, France, Apr. 2022.

[9] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. abs/2103.11624, 2021.

[10] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 522–11 530, 2020.

[11] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 86–99.

[12] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 420–425.

[13] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Intention-Aware Risk Estimation for General Traffic Situations, and Application to Intersection Safety," INRIA, France, Research Report RR-8379, Oct. 2013.

[14] J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra, "A grid-based framework for collective perception in autonomous vehicles," *Sensors*, vol. 21, no. 3, p. 744, 2021.

[15] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[16] O. Triebe, N. Laptev, and R. Rajagopal, "Ar-net: A simple auto-regressive neural network for time-series," 2019. [Online]. Available: https://arxiv.org/abs/1911.12436

[17] Y. Bai, X. Jin, X. Wang, T. Su, J. Kong, Y. Lu, and T. Craciunescu, "Compound autoregressive network for prediction of multivariate time series," *Complex.*, vol. 2019, jan 2019. [Online]. Available: https://doi.org/10.1155/2019/9107167

[18] L. Ranaldi, M. Gerardi, and F. Fallucchi, "Cryptonet: Using auto-regressive multi-layer artificial neural networks to predict financial time series," *Information*, vol. 13, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/11/524

[19] T. D. Barrett, A. Malyshev, and A. I. Lvovsky, "Autoregressive neural-network wavefunctions for ab initio quantum chemistry," 2022. [Online]. Available: https://arxiv.org/abs/2109.12606

[20] A. Artuñedo, G. Corrales, J. Villagra, and J. Godoy, "Machine learning based motion planning approach for intelligent vehicles," in *2020 IEEE Intelligent Vehicles Symposium*. IEEE, 2020.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] A. Breuer, J.-A. Termöhlen, S. Homoceanu, and T. Fingscheidt, "Opendd: A large-scale roundabout drone dataset," in *Proceedings of International Conference on Intelligent Transportation Systems*, September 2020.

[23] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.

[24] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.