On the Importance of Uncertainty Calibration in Perception-Based Motion Planning

Andrei Ivanovic[1] Masha Itkina[2] Rowan McAllister[2] Igor Gilitschenski[1] Florian Shkurti[1]

***Abstract*— Autonomous vehicles (AVs) are increasingly being deployed in urban environments. However, most AVs operate without accounting for uncertainty inherent to perceiving the world. To remedy this disregard, uncertainty-aware planners have recently been developed that account for upstream perception and prediction uncertainty, generating more efficient motion plans without sacrificing safety. However, such planners may be sensitive to prediction uncertainty miscalibration, the magnitude of which has not yet been characterized. Towards this end, we perform a detailed analysis of the impact that perceptual uncertainty propagation and uncertainty calibration has on perception-based motion planning. We do so with a comparison between two novel prediction-planning architectures with varying levels of uncertainty incorporation on a large-scale, real-world autonomous driving dataset. We find that, despite one model producing quantifiably better predictions, both methods produce similar motion plans with only minor differences.**

I. INTRODUCTION

Autonomous vehicles (AVs) are increasingly becoming adopted and continue to be deployed in highly-dynamic and uncertain urban environments. Uncertainty is present everywhere on our roads, whether it arises from multimodality in an agent's potential future motion or measurement uncertainty from onboard sensors. Despite this, most AVs operate without capturing this inherent uncertainty, even with some modules providing highly uncertain outputs [1], [2]. To improve their safety and robustness, AVs should adopt autonomy stacks that are uncertainty-aware and can use such information to better inform AV motion planning.

Currently, AV autonomy stacks are commonly architected with four main components: perception, prediction, planning, and control. While most trajectory predictors do not incorporate upstream sources of uncertainty, several recent works have augmented state-of-the-art (SOTA) methods to incorporate and implicitly propagate upstream perceptual uncertainty, such as state or class uncertainty [1], [3]. In doing so, these methods have shown improved results, both in traditional prediction metrics (e.g., displacement errors) and in model calibration; how closely a model's predicted uncertainty aligns with its empirical uncertainty [4], [5].

In parallel, uncertainty-aware planners have recently been developed that account for prediction uncertainty and generate motion plans which are more efficient without sacrificing safety [6], [7], [8]. However, such planners may be sensitive to prediction uncertainty miscalibration, manifested as, e.g., improperly-shaped covariances in a predictor's output distribution. The magnitude of this sensitivity is not known, however, as there have been no studies that measure the effect of prediction uncertainty calibration on uncertainty-aware motion planning. Towards this end, we seek to determine the importance of perception and prediction output uncertainty calibration on motion planning.

Contributions. Our key contribution is a detailed analysis of the impact that upstream uncertainty propagation, and, in turn, improved model calibration, has on perception-based motion planning. We construct a novel, uncertainty-aware, prediction-planning pipeline evaluated on the Perceptual Uncertainty in Prediction (PUP) dataset [1]. Our analysis compares the performance of two SOTA trajectory forecasting methods (with varying upstream uncertainty propagation [9], [1]) paired with an uncertainty-aware motion planner based on stochastic model predictive control (SMPC) [6].

II. RELATED WORK

Autonomous Vehicle Trajectory Forecasting. Recent trajectory forecasting methods employ Conditional Variational Autoencoders (CVAEs) [10] as in [11], [5], Graph Neural Networks (GNNs) [12] as in [13], and Transformers [14] as in [15] which have been used to explicitly model and capture all possible agent interactions and future trajectories in a scene [16], [17].

In this work, we employ Trajectron++ [9] and its follow-up work HAICU [1]. Both models use the exact same Trajectron++ backbone, producing Gaussian Mixture Model (GMM) predictions. HAICU additionally propagates upstream class uncertainty, enabling direct comparison of the isolated effect of uncertainty on downstream planning. Furthermore, PSU-TF [3] extends Trajectron++ [9] to account for state uncertainties stemming from detection and tracking. A detailed survey of uncertainty estimation and quantification methods can be found in [18]. Each of those works perform uncertainty estimation, but as we are dealing with a modular autonomous driving stack, each component is able to characterize its own uncertainty and provide it to following modules. Both Trajectron++ and HAICU are well established, achieving SOTA performance when released, with easy-to-use publicly-available codebases.

Stochastic MPC for Autonomous Driving. For autonomous driving scenarios with a high density of overlapping, non-Markovian, multi-modal predictions, many SMPC works exploit the structure of GMMs [19], [20], [21], [6], [22] due to their memory-efficiency in representing multi-modal uncertainty [23]. To address the conservatism of emerging non-linear, non-convex SMPC methods [19], [20], a recent work [6] proposes a convex formulation optimizing over a novel class of policies, enhancing the feasibility of

[1]Andrei Ivanovic, Florian Shkurti, and Igor Gilitschenski are with the University of Toronto. a.ivanovic@mail.utoronto.ca, {florian,gilitschenski}@cs.toronto.edu.

[2]Masha Itkina and Rowan McAllister are with the Toyota Research Institute. {masha.itkina, rowan.mcallister}@tri.global.

the optimization problem. In this work we extend their formulation to directly incorporate uncertainty, multiple agents, and more sophisticated ego dynamics.

III. Input Uncertainty-Aware Stochastic MPC

We present two prediction-planning frameworks with varying amounts of uncertainty incorporation. The frameworks are modular, allowing for the integration of different prediction architectures with our novel uncertainty-aware SMPC formulation.

To investigate the effects of perception and prediction output uncertainty calibration on motion planning, we combine Trajectron++ and HAICU with a novel uncertainty-aware stochastic MPC (UA-SMPC) formulation. For architectural and implementation details regarding Trajectron++ [9] and HAICU [1], please refer to their original papers. Our formulation extends the SMPC framework presented in [6], modifying the objective function to more directly include prediction uncertainty. We also expand upon their method to account for multiple neighbouring agents, a more sophisticated dynamics model, and incorporate recent trajectory forecasting approaches to provide agent predictions. For consistency, the same notation convention as [6] is used in this report.

A. Notation

Let x_t represent the state vector of the ego-vehicle (EV) at time t. The state vector consists of the vehicle's 2D-position (X_t, Y_t) and velocity v_t. The control inputs $u_t = [v_x^t, v_y^t]^T$ are the vehicle's longitudinal and lateral velocity.

For all non-ego vehicles (NVs), $o_t^i = [X_t, Y_t, \phi_t]$ denotes the 2D-position and heading of NV $i \in \{1, \dots, A\}$, where A is the total number of neighbouring agents, at time t. We denote the future T-step predictions of the i-th NV's position as random variables $\{o_{k|t}^i\}_{k=1}^T$. As per [6], $o_{k|t}^i$ is a shortened notation for $o_{t+k}^i | o_t^i$; the NV's state at o_{t+k}^i conditioned on its current state at time t.

B. Overview of SMPC Formulation

The SMPC formulation in this work is used by the EV to track a reference trajectory safely and comfortably, while using multi-modal predictions for collision avoidance and navigating interactions with other agents. The optimization problem of our SMPC framework, in which the optimal solution returns the EV control inputs across the planning horizon, is formulated as

$$
\begin{aligned}
\min_{\theta_t} \quad & J_t(\mathbf{x}_t, \mathbf{u}_t), && (1)\\
\text{s.t.} \quad & x_{k+1|t} = f_k^{\text{EV}}(x_{k|t}, u_{k|t}), && (2)\\
& \mathbb{P}(x_{k+1|t}, o_{k+1|t}^i \in \mathcal{C}_{k+1}) \geq 1 - \epsilon, && (3)\\
& \mathbb{P}(x_{k+1|t}, u_{k|t} \in \mathcal{X} \times \mathcal{U}) \geq 1 - \epsilon, && (4)\\
& \mathbf{u}_t \in \Pi_{\theta_t}(\mathbf{x}_t, \mathbf{o}_t^i), && (5)\\
& x_{0|t} = x_t, o_{0|t}^i = o_t^i, && (6)\\
& \forall\ k \in \{0, \dots, N-1\},\\
& \forall\ i \in \{1, \dots, A\},
\end{aligned}
$$

where $\mathbf{x}_t = [x_{0|t}, ..., x_{N-1|t}]$, $\mathbf{u}_t = [u_{0|t}, ..., u_{N-1|t}]$, $\mathbf{o}_t^i = [o_{0|t}^i, ..., o_{N-1|t}^i]$, and $o_{k+1|t}^i | o_{k|t}^i \sim f_k^{\text{NV}}(o_{k|t}^i)$.

The objective function (1) (formally shown in Eq. (10)) serves three purposes: penalizing the EV's deviation from a kinematically-feasible reference trajectory $(x_n^{\text{ref}}, u_n^{\text{ref}})_{n=0}^N$, encouraging path efficiency, and rewarding conservative behaviour. EV predictions are obtained from a dynamics model f_k^{EV} (2) and NV predictions are obtained from the output of a prediction model f_k^{NV}. The framework uses these predictions to optimize over the parameterized policy class introduced in [6]. Collision avoidance and polytopic state and input constraints are imposed as chance constraints. EV and NV state feedback are incorporated in (6).

C. Ego-Vehicle and Non-Ego Prediction Models

EV predictions are obtained with the following dynamics model $f_k^{EV}(x_{k|t}, u_{k|t})$,

$$
\underbrace{\begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ v_x^{t+1} \end{bmatrix}}_{S^{t+1}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} X_t \\ Y_t \\ v_x^t \end{bmatrix}}_{S^t} + \underbrace{\begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 1 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} v_x^t \\ v_y^t \end{bmatrix}}_{u^t} + \underbrace{\begin{bmatrix} \sigma_X & 0 & 0 \\ 0 & \sigma_Y & 0 \\ 0 & 0 & \sigma_v \end{bmatrix}}_{D} w_t, \qquad (7)
$$

where $w_t \sim \mathcal{N}(0, I)$ captures process noise. This formulation, including the choice of velocity-based control inputs, incorporates steering into [6] while remaining compatible with the rest of their method.

Unlike the formulation in [6], we use Trajectron++ and HAICU as NV prediction models, $f_k^{NV}(o_{k|t})$. The output predictions for a single agent at time t are encoded as a bivariate GMM, $\mathcal{G}_t = \{p_{t,\xi}, \{\mathcal{N}(\mu_{k|t,\xi}, \Sigma_{k|t,\xi})\}_{k=1}^T\}_{\xi=1}^{\Xi}$, where $\Xi \leq |Z|$ is the number of *considered* modes (e.g., the Ξ most likely modes) and $p_{t,j} = \mathbb{P}(\sigma_t = \xi \mid o_t)$ denotes the probability of mode ξ at time t.

D. Collision Avoidance

For collision avoidance, the EV and NV vehicle geometries are approximated as ellipses. The collision avoidance condition ensures that these shape-approximating ellipses are non-overlapping with an affine inner-approximation denoted as $\mathcal{C}_t$. This condition is enforced as a chance constraint with risk level $1 - \epsilon$ along the prediction horizon due to both the process noise in the EV's dynamics model and uncertainty in the NV's future motion, written as $\mathbb{P}(x_{k|t}, o_{k|t}^i \in \mathcal{C}_k) \geq 1 - \epsilon$. Furthermore, due to the EV's process noise, the polytopic state and input constraints are also written as chance constraints with risk level $1 - \epsilon$, as in Eq. (4), where $\mathcal{X} \times \mathcal{U} = \{(x_t, u_t) \mid v_t \in [v_{min}, v_{max}], a_t \in [a_{min}, a_{max}]\}$.

E. Parameterized Policies Incorporating State Feedback

In this work, we use the same parameterized policies $\Pi_{\theta_t}(\mathbf{x}_t, \mathbf{o}_t)$ proposed in [6], where the EV's controls $\mathbf{u}_t$ are functions of the EV and NV trajectories. The feedback policy

is formulated as

$$\Delta u_{k|t} = \pi_{k|t}(x_{k|t}, o_{k|t}) = \begin{cases} h_{k|t} + \sum_{\ell=0}^{k-1} M_{\ell,k|t} w_{\ell|t} + K_{k|t} o_{k|t}, & \text{if } k < \overline{k}, \\ h^1_{k|t} + \sum_{\ell=0}^{k-1} M^1_{\ell,k|t} w_{\ell|t} + K^1_{k|t} o_{k|t}, & \text{if } k \geq \overline{k}, \sigma_t = 1, \\ \vdots & \\ h^\Xi_{k|t} + \sum_{\ell=0}^{k-1} M^\Xi_{\ell,k|t} w_{\ell|t} + K^\Xi_{k|t} o_{k|t}, & \text{if } k \geq \overline{k}, \sigma_t = \Xi, \end{cases} \tag{8}$$

where $\overline{k} \leq T$ denotes the minimum time step within the prediction horizon such that the β-confidence ellipsoids $\varepsilon(\mu, \Sigma, \beta) = \{x \mid \|x - \mu\|^2_{\Sigma^{-1}} \leq \beta\}$ are non-overlapping for all future timesteps and $o_{k|t}$ refers to all A NVs. The feedback policy uses state feedback for NV states and affine disturbance feedback for EV states [6]. This formulation causes the two chance constraints to become affine, reducing their computational cost. Given x_t and o_t for a mode $\xi \in \{1, \ldots, \Xi\}$, the set of parameters satisfying both chance constraints for all $k \in \{0, \ldots, N-1\}$ is

$$\Theta_t(x_t, o_t) = \left\{ \{\mathbf{h}^\xi_t, \mathbf{M}^\xi_t, \mathbf{K}^\xi_t\}_{\xi=1}^{\Xi} \;\middle|\; (3), (4), \Delta x_{0|t} = x_t - x^{\text{ref}}_t, o_{0|t} = o_t \right\}. \tag{9}$$

F. Cost Function and Optimization Problem

We propose a novel cost function that expands upon that of [6] which only penalizes deviations of the EV state and input trajectories from the reference trajectory. In addition to penalizing deviations, we introduce two additional terms: The first promotes comfort by discouraging changes in control inputs across subsequent timesteps. Intuitively, this mitigates occurrences of jerky controls and results in a smoother EV trajectory. The second term encourages conservative behaviour by minimizing the probability that the EV's trajectory lies in a neighbour's potential future path. This probability is defined as the log-likelihood of the EV's motion under the predicted distribution from a distribution-producing predictor (e.g., Trajectron++, HAICU). Formally, our novel objective function is

$$\begin{aligned} J_t(\mathbf{x}_t, \mathbf{u}_t) = \mathbb{E}_{\xi \sim p_t} \Bigg[& \sum_{k=0}^{N-1} \left(\Delta x^\xi_{k+1|t}\right)^T Q \Delta x^\xi_{k+1|t} \\ & + \left(u^\xi_{k+1|t} - u^\xi_{k|t}\right)^T R \left(u^\xi_{k+1|t} - u^\xi_{k|t}\right) \\ & + \left(\Delta u^\xi_{k|t}\right)^T R \Delta u^\xi_{k|t} + T \log p\left(x^\xi_{k+1|t}\right) \Bigg], \end{aligned} \tag{10}$$

where $\Delta x_{k+1|t} = x_{k+1|t} - x^{\text{ref}}_{k+1|t}$, $\Delta u_{k|t} = u_{k|t} - u^{\text{ref}}_{k|t}$, and $Q \succ 0, R \succ 0$, $T > 0$ represent weights for the different cost terms. In this work, Q is selected to be larger than R and T. Once dynamical constraints are substituted in, the EV control sequence can be obtained by minimizing cost.

IV. EXPERIMENTS

We compare two different prediction-planning methods on the PUP Dataset [1]: Trajectron++ [9] and HAICU [1], both paired with our proposed UA-SMPC. The two prediction architectures are highly similar, with the difference being HAICU's incorporation of input class uncertainty, allowing us to quantify the impact of increased uncertainty incorporation and improved model calibration on uncertainty-aware motion planning.

A. Methodology

For each scenario, consisting of 25 timesteps, both prediction-planning methods are run with the final position of the ground truth trajectory provided as the goal point. Reference waypoints, comprising states and control actions, for each timestep of the prediction/planning horizon are provided using a constant velocity model between the EV's current position and goal position. Evaluation is conducted in an open-loop manner, where each approach is compared on a per-timestep basis, with the initial states for all agents coming from the dataset.

B. Metrics

The two methods are compared on various common deterministic and probabilistic prediction and planning metrics. For prediction metrics, we are interested in comparing the fidelity of the output distributions, thus we measure:

1) *minADE*$_5$: ADE/FDE between the ground truth and best of 5 predicted output samples [24],
2) *Average Negative Log-Likelihood (NLL)*: The average NLL of the ground truth trajectory under the predicted future distribution, and
3) *Average Entropy*: The entropy of each GMM component distribution averaged across all agents and time.

To evaluate open-loop planning, we use three separate metrics that quantify for accuracy, path efficiency, and safety:

1) *Rollout ADE/FDE (Accuracy)*: Average/final ℓ_2 distance between the trajectory rollout using UA-SMPC's control actions and the ground truth trajectories,
2) *Path Length Difference to GT (Efficiency)*: The total distance between waypoints for the UA-SMPC's rollout relative to the total path length of the ground truth trajectory, and
3) *Closest Distance to Neighbour (Safety)*: The closest ℓ_2 distance reached between the ego-vehicle and nearest neighbouring vehicle for each scenario.

C. Quantitative Results

Table I summarizes the performance of the Trajectron++ and HAICU-backed UA-SMPC on the PUP dataset. The prediction metrics demonstrate HAICU clearly outperforming Trajectron++ in both the accuracy and calibration of its output distribution. On average, HAICU's predictions are more informative, achieving a lower NLL, and less entropic.

When compared on open-loop planning metrics, the HAICU-backed approach incrementally outperforms the Trajectron++-backed one on *Rollout ADE/FDE* and *Closest Distance to Neighbour*, with HAICU UA-SMPC achieving better path efficiency.

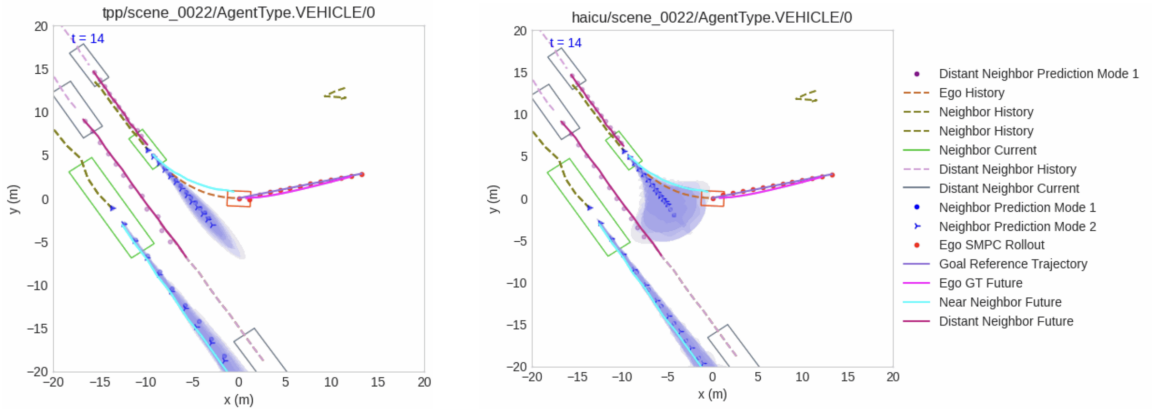

Fig. 1: Comparison between Trajectron++ (left) and HAICU (right) and the resulting rollout of UA-SMPC on a left turn surrounded by several agents. HAICU produces more uncertain predictions for the nearest neighbour than Trajectron++, covering both primary potential modes of continuing straight or turning left. Trajectron++ confidently predicts the nearby agents' trajectory incorrectly for all of its top output modes. Despite this difference in output distributions, both motion plans and control actions are very similar.

TABLE I: HAICU UA-SMPC notably outperforms Trajectron++ UA-SMPC on prediction metrics, while both methods perform similarly on planning metrics. Bold indicates best.

Metric	GT	Methods	
		T++ [9]	HAICU [1]
Prediction Metrics			
minADE_5 (m) [↓]	-	1.11	**0.96**
NLL [↓]	-	-2.30	**-2.96**
Average Entropy [↓]	-	-1.01	**-1.32**
Planning Metrics (Combining Prediction with UA-SMPC)			
Rollout ADE (m) [↓]	-	0.454	**0.449**
Rollout FDE (m) [↓]	-	0.495	**0.492**
Δ Path Length (m) [↓]	-	0.667	**0.537**
Closest Neighbour (m) [↑]	8.494	8.455	**8.476**

D. Qualitative Results

Visually, HAICU produces more calibrated outputs than Trajectron++, capturing the ground truth future with more likelihood than Trajectron++. In Fig. 1, Trajectron++ confidently overshoots the ground truth future of the neighbour turning left, predicting it to continue straight.

Irrespective of the calibration of the distributions, both models produce highly accurate predictions in initial timesteps, with most uncertainty towards the end of the prediction horizon. Despite such uncertainty and prediction errors, UA-SMPC was able to converge to nearly identical control actions with either prediction model.

E. Discussion

Our results demonstrate that, despite HAICU producing more accurate predictions and well-informed prediction distributions, both predictor-planner combinations yielded similarly-performing motion plans.

There are several potential reasons for this: First, each of the output differences visualized occur with vehicles that would normally not have any safety implications for the EV (under the given goal point). Despite incorrectly predicting an agent's turn in long-horizons, Trajectron++ still produced equally accurate next-step predictions as HAICU did, and any far-horizon errors are corrected in later scene timesteps. Second, due to limitations of the PUP dataset and a lack of HD maps, both agents were given the same goal as the final ground truth state in a scenario and same initial state every timestep due to our open-loop evaluation scheme. With the added primary objective of UA-SMPC not deviating from the reference trajectory significantly, there were seldom cases where one method would act substantially riskier than another, given that the reference trajectories were equal for both. These results imply that, with our current uncertainty-awareness formulation, motion planners can be made robust to upstream output miscalibration, especially if the predictor's first timestep predictions are accurate.

V. CONCLUSION

In this work, we present a comparison between two prediction-planning architectures with varying levels of input uncertainty propagation. We evaluate the two architectures on the PUP dataset with several prediction and planning metrics. We demonstrate that, despite having quantifiably better predictions, HAICU UA-SMPC only produces incrementally better plans compared to Trajectron++ UA-SMPC.

Future Work. To address the limitations of open-loop evaluation, an immediate next step is to perform the same experiments in the recently-released nuPlan planning benchmark [25]. The NVs would be reactive and difficult scenarios could be generated using pre-specified options such as lane changing in dense traffic. Another future direction is the addition of other prediction-planning methods, one immediate example being PSU-TF [3] for trajectory prediction combined with UA-SMPC for planning.

Acknowledgements. The Toyota Research Institute partially supported this work. This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

REFERENCES

[1] B. Ivanovic, K.-H. Lee, P. Tokmakov, B. Wulfe, R. McAllister, A. Gaidon, and M. Pavone, "Heterogeneous-agent trajectory forecasting incorporating class uncertainty," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2022.

[2] M. Henne, A. Schwaiger, and G. Weiss, "Managing uncertainty of ai-based perception for autonomous systems," in *Int. Joint Conf. on Artificial Intelligence Workshop on AI Safety*, 2019.

[3] B. Ivanovic, Y. Lin, S. Shrivastava, P. Chakravarty, and M. Pavone, "Propagating state uncertainty through trajectory forecasting," in *Proc. IEEE Conf. on Robotics and Automation*, 2022.

[4] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Conf. on Neural Information Processing Systems*, 2019.

[5] B. Ivanovic, J. Harrison, and M. Pavone, "Expanding the deployment envelope of behavior prediction via adaptive meta-learning," in *Proc. IEEE Conf. on Robotics and Automation*, 2023.

[6] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with multi-modal predictions for traffic intersections," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2022.

[7] S. H. Nair, V. Govindarajan, T. Lin, Y. Wang, E. H. Tseng, and F. Borrelli, "Stochastic MPC with dual control for autonomous driving with multi-modal interaction-aware predictions," in *Int. Symp. on Advanced Vehicle Control*, 2022.

[8] S. H. Nair, E. H. Tseng, and F. Borrelli, "Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control," in *Proc. IEEE Conf. on Decision and Control*, 2022.

[9] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conf. on Computer Vision*, 2020.

[10] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Conf. on Neural Information Processing Systems*, 2015.

[11] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *IEEE Int. Conf. on Computer Vision*, 2019.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, 2009.

[13] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *European Conf. on Computer Vision*, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Conf. on Neural Information Processing Systems*, 2017.

[15] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021.

[16] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[17] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[18] J. Gawlikowski, C. Rovile Njieutcheu Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, *et al.*, "A survey of uncertainty in deep neural networks," *arXiv:2107.03342*, 2021.

[19] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving," in *Proc. IEEE Conf. on Robotics and Automation*, 2018.

[20] A. Wang, A. Jasour, and B. C. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, 2020.

[21] K. Ren, H. Ahn, and M. Kamgarpour, "Chance-constrained trajectory planning with multimodal environmental uncertainty," *IEEE Control Systems Letters*, vol. 7, pp. 13–18, 2022.

[22] J. Zhou, B. Olofsson, and E. Frisk, "Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions," *IEEE Transactions on Intelligent Vehicles*, 2023.

[23] S. H. Nair, H. Lee, E. Joa, Y. Wang, H. E. Tseng, and F. Borrelli, "Predictive control for autonomous driving with uncertain, multi-modal predictions," *arXiv:2310.20561*, 2023.

[24] A. Gupta, J. Johnson, F. Li, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[25] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv:2106.11810*, 2021.