Interventional Behavior Prediction: Avoiding Overly Confident Anticipation in Interactive Prediction

Chen Tang, Wei Zhan, Masayoshi Tomizuka

***Abstract*— Conditional behavior prediction (CBP) builds up the foundation for a coherent interactive prediction and planning framework that can enable more efficient and less conservative maneuvers in interactive scenarios. In CBP task, we train a prediction model approximating the posterior distribution of target agents' future trajectories conditioned on the future trajectory of an assigned ego agent. However, we argue that CBP may provide overly confident anticipation on how the autonomous agent may influence the target agents' behavior. Consequently, it is risky for the planner to query a CBP model. Instead, we should treat the planned trajectory as an intervention and let the model learn the trajectory distribution under intervention. We refer to it as the interventional behavior prediction (IBP) task. Moreover, to properly evaluate an IBP model with offline datasets, we propose a Shapley-value-based metric to testify if the prediction model satisfies the inherent temporal independence of an interventional distribution. We show that the proposed metric can effectively identify a CBP model violating the temporal independence, which plays an important role when establishing IBP benchmarks.**

I. Introduction

Behavior prediction is crucial for autonomous systems interacting with humans, such as autonomous vehicles. Most existing works focus on a *passive* prediction scheme [1], [2], [3], where the target agents' future trajectories are predicted given the historical trajectories of themselves and other surrounding agents. When using such a prediction model, downstream decision-making modules then determine the autonomous agent's action according to the predicted trajectories in a passive manner. To ensure safety under various predicted trajectories of others, the ego agent has to be overly conservative with inefficient maneuvers, especially in highly interactive scenarios. It is because passive prediction models ignore the fact that the autonomous agent's future actions can influence other agents' behavior. To this end, researchers started to investigate a more coherent interactive prediction and planning framework which relies on predicting the surrounding agents' future trajectories conditioned on the ego agent's future actions [4], [5], [6], [7], [8], [9], [10], [11]. Under such frameworks, the autonomous agents can reason over potential actions while considering their influence on surrounding agents. It can then induce more efficient and less conservative maneuvers in interactive scenes. Some of these prior works merely demonstrated that their models are able to support conditional prediction from the perspective of architecture [5], [7]. Another line of works focused on the closed-loop performance, which relies on a simulating environment [4], [6], [10].

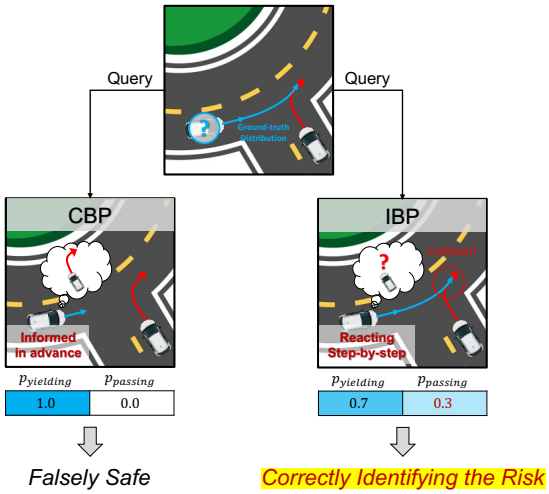


Fig. 1: An illustration of the difference between CBP and IBP. The robot car plans to enter the roundabout aggressively and force the human car in the roundabout to yield. It queries a prediction model for whether the human car will yield or pass. The CBP model predicts the posterior distribution of human car's behavior conditioned on the plan. Intuitively, it models the human behavior given that the driver is informed about the robot car's plan in advance. Therefore, the human car will always be predicted to yield to the robot car under CBP's prediction. In fact, the human car will only react to the robot car's action at each timestep. Consequently, the human car may attempt to pass first as it has the right-of-way, which may leads to collision. The IBP model is able to warn the robot car of the safety risk.

More interestingly, some existing works formulated an alternative prediction task to evaluate the prediction module in a self-contained way [9], [8], [11]. We follow [11] to refer to this task as *conditional behavior prediction* (CBP). In the CBP task, the future trajectories of the target agents are predicted conditioned on the ground-truth future trajectory of an assigned ego agent. Standard prediction metrics are adopted to quantify the performance. It allows us to leverage large-scale naturalistic traffic datasets to develop and validate a conditional prediction model before closed-loop testings. In those works, a model that can achieve the smallest prediction error after granted the additional future information of the ego agent is considered the best. However, we can only evaluate the prediction accuracy given the actual future

C. Tang, W. Zhan, and M. Tomizuka, are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA {chen_tang, wzhan, tomizuka}@berkeley.edu.

This work is supported by Denso International America, Inc.

trajectory of the ego agent with a static offline dataset. It is impossible to quantify the model performance when it is queried by an arbitrary plan of the ego agent. Therefore, we argue that we should be more careful when interpreting the evaluation results.

In particular, we argue that it is risky to train and evaluate the model for *conditional inference*. In the current CBP task, the prediction model essentially learns the posterior distribution of future trajectories conditioned on the future trajectory of the ego agent. In this way, the ego agent's future trajectory is treated as an *observation*. Since the actual ego agents in the offline dataset make decisions according to the states of the surrounding agents, the surrounding agents under CBP are implicitly assumed to get additional hints on the future behavior of the ego agents. With such an unrealistic assumption, it is natural to consider the CBP model with the lowest prediction error as the best option for the CBP task. However, the surrounding agents in the real world are not informed of the planned trajectories of the ego agents. Consequently, as illustrated in Fig. 1, there will be a discrepancy between: 1) what an autonomous agent is informed by querying a CBP model with a potential plan; and 2) how the others will actually react if the agent executes the plan. As we will show later in this work, this discrepancy may lead to overly confident anticipation on the ego agent's influence on the surroundings, resulting in potential safety hazards during online usage.

This discrepancy is formally captured in the theory of causality [12] by the difference between *observation* and *intervention*. With an intervention to a set of random variables, we enforce the value of a random variable without treating it as the consequence of other random variables. The resulting distribution of the remaining random variables under the intervention is consistent with what will actually happen if we have the privilege to manipulate the target random variable as desired. Consequently, we argue that we should build the prediction model to approximate the future trajectory distribution under the intervention of enforcing the ego agent's future trajectory. We refer to this new task as the *interventional behavior prediction* (IBP) task. In IBP, we still want to train and evaluate the model with an offline dataset. The task setting is essentially the same as CBP, except for learning an interventional distribution instead of a conditional one. The remaining issue is then how to properly evaluate an IBP model with an offline dataset. Without knowing the ground-truth distribution under intervention, we can only compare the model's output against the ground-truth future trajectories for evaluation. However, such evaluation metrics are naturally biased toward a CBP model. The dataset is collected without intervention. The ego agent in the dataset follows an internal reactive policy. Therefore, the distribution of ground-truth labels given the same input essentially follows a conditional distribution. As a result, a CBP model will always outperform an IBP model if prediction accuracy is the only evaluation metric with an offline dataset.

To this end, we propose to testify the inherent temporal independence of a prediction model before comparing the prediction performance to ensure a proper evaluation for the IBP task. Under the interventional distribution, the predicted states of the target agents at earlier timesteps should be independent from the ego agent's states at latter timesteps. We propose a Shapley-value-based [13], [14], [15] metric to verify if the model obeys this temporal independence. We show that we can effectively identify a model violating the expected temporal independence with the proposed metric, and therefore ensure a valid evaluation method.

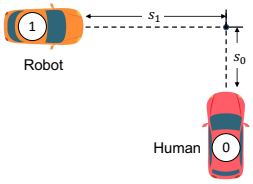


Fig. 2: A motivating toy example, where a human car and a robot car are driving towards a collision point.

The rest of the paper is organized as follows: 1) In Sec. II, we explain the difference between CBP and IBP and demonstrate the risk of using CBP with a motivating toy example; 2) In Sec. III, we formulate the Shapley-value-based metric we propose to testify the temporal independence and quantify the impact of doing CBP; 3) In Sec. IV, we study a CBP model with the proposed metric to demonstrate that such a CBP model indeed violates the temporal independence. Moreover, it results in misleading evaluation result without using the proposed metric. 4) In Sec. V, we wrap up the paper with a discussion on insights for future model design and IBP benchmarks.

II. A Motivating Example

We begin our discussion by studying a motivating toy example to demonstrate the issue of using conditional inference for interactive prediction. We consider the example depicted in Fig. 2, where two cars are driving towards a collision point. One of them is controlled by a human driver, while the other is an autonomous robot car. As an analogy of the CBP task, the robot car can query the posterior distribution of the human car's trajectory conditioned on a planned trajectory of the robot car. The robot car can then evaluate the risk of multiple planned trajectories and select the optimal one to execute.

We model the human drivers' behavior with the intelligent driver model (IDM) [16], [17]. Each car has two states at each timestep, $s_{i,t}$ and $v_{i,t}$, where $s_{i,t}$ is the displacement relative to the collision point and $v_{i,t}$ is the absolute velocity. We denote the state vector by $\mathbf{x}_{i,t} = [s_{i,t}\ v_{i,t}]^\intercal$. Each car is assigned a target position to follow at each step, depending on which car has the right-of-way. If a car has the right-of-way, it is asked to follow a target substantially far away. Otherwise, if the other car has the right-of-way and has not passed the collision point, the target is set as the collision point. We determine which car has the right-of-way based on which car has smaller time headway at the current timestep.

The time headway is defined as follows:

$$T_{head,i,t} = \max\left(\frac{s_{i,t}}{v_{i,t}}, 0\right).$$

Given a target position, the car dynamics is governed by the intelligent driver model as follows:

$$s_{i,t+1} = s_{i,t} - \Delta t \cdot v_{i,t},$$

$$v_{i,t+1} = v_{i,t} + \Delta t \cdot \left\{ a \left[1 - \left(\frac{v_{i,t}}{v_0}\right)^{\delta} - \left(\frac{s^*(v_{i,t}, \Delta v_{i,t})}{s_{i,t} - d_{i,t}}\right)^2 \right] + \omega_{i,t} \right\},$$

where

$$\Delta v_{i,t} = v_{i,t} - v_0,$$

$$s^*(v_{i,t}, \Delta v_{i,t}) = s_0 + \max\left(0, v_{i,t}T + \frac{v_{i,t}\Delta v_{i,t}}{2\sqrt{ab}}\right),$$

$$\omega_{i,t} \sim \mathcal{N}\left(0, \sigma^2\right).$$

The term $d_{i,t}$ denotes the target position. It equals to zero if the target point collides with the collision point. Otherwise, a large negative value is assigned to $d_{i,t}$. The Gaussian noise $\omega_{i,t}$ is added to inject randomness. The remaining parameters are defined as in the standard IDM model. Readers may refer to [16] for detailed definitions. In our experiments, we set $v_0 = 10\text{m/s}$, $T = 2\text{s}$, $s_0 = 4\text{m}$, $\delta = 4$, $a = 1\text{m/s}^2$, $b = 1.5\text{m/s}^2$, $\Delta t = 0.2\text{s}$, and $\sigma = 4\text{m/s}^2$.

In the CBP task, we aim to approximate the distribution of the human car's future trajectory conditioned on the initial states of the two cars and the future trajectory of the robot car, i.e., $p(\mathbf{x}_{0,1:T_H}|\mathbf{x}_{0,0}, \mathbf{x}_{1,0}, \mathbf{x}_{1,1:T_H})$, with T_H denotes the number of timesteps. The robot car can query the conditional distribution with a planned trajectory $\hat{\mathbf{x}}_{1,1:T_H}$. In our experiment, we use likelihood weighting [18] to estimate the conditional distribution given an evidence set $\{\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, \hat{\mathbf{x}}_{1,1:T_H}\}$. Meanwhile, we can approximate the actual distribution of $\mathbf{x}_{0,1:T_H}$ after executing $\hat{\mathbf{x}}_{1,1:T_H}$ via multiple simulation trials. In Fig. 3(a), we compare the two distributions under the same initial conditions and query trajectory. We set $s_{0,0} = s_{1,0} = 15\text{m}$, $v_{0,0} = 8\text{m/s}$, $v_{1,0} = 5\text{m/s}$, and $T_H = 10$. Since the robot car has smaller initial speed, it is more likely to yield to the human car. However, we let the robot car execute an aggressive maneuver, where the robot car accelerates with an acceleration of 5m/s^2 until reaching the speed of 10m/s.

The conditional distribution implies that the human car always yields to the robot car. However, the human car may actually not yield to the robot car when the robot car executes the planned trajectory. Even if the human car eventually yields to the robot car, it starts decelerating much later than the conditional distribution suggests. If we evaluate the risk based on the conditional distribution, we may falsely conclude that the human car will always yield to the robot car, so that the robot car can safely pass the intersection at high speed, which leads to an overly aggressive and unsafe maneuver. It can be further verified by estimating the histograms of the minimum distance between the two cars under these two distributions (Fig. 3(b)). The minimum

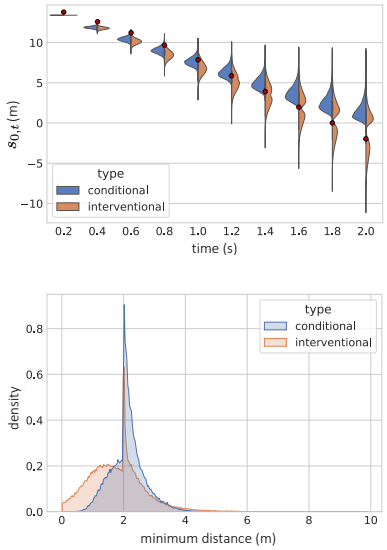


Fig. 3: **Top**: Histograms of $s_{0,t}$ for $t = 1, \cdots, T_H$ under the conditional distribution and the distribution given the interventional action. The red dots denote the planned trajectory of the robot car; **Bottom**: Normalized histograms of minimum distance between cars under these two distributions. The histograms are drawn with 10000 simulation trials.

distance is biased under conditional inference. In particular, the conditional distribution falsely implies that the two cars never collide.

The toy example demonstrates the discrepancy between the reality and the anticipation from conditional inference. Formally, the conditional distribution is governed by the Bayesian network in Fig. 4(a), where the initial states and the query trajectory are treated as an *observation*. However, the system is actually governed by the Bayesian network in Fig. 4(b) when the robot executes $\hat{\mathbf{x}}_{1,1:T_H}$. The incoming edges of $\mathbf{x}_{1,i}$ are removed because the robot car follows a fixed trajectory regardless of the other car's reaction. If fact, if we treat the Bayesian network governing the system as a causal Bayesian network [12], then Fig. 4(b) represents the distribution resulting from the interventional action $do(\mathbf{x}_{1,1:T_H} = \hat{\mathbf{x}}_{1,1:T_H})$, denoted by $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, do(\hat{\mathbf{x}}_{1,1:T_H}))$. The difference between the two distributions, $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, \hat{\mathbf{x}}_{1,1:T_H})$ and $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, do(\hat{\mathbf{x}}_{1,1:T_H}))$, mirrors the difference between seeing and doing [12]. By conditional inference, we aim to infer the distribution of $\mathbf{x}_{0,1:T_H}$ after *observing* $\hat{\mathbf{x}}_{0,1:T_H}$, intuitively speaking, how the human driver behaves if knowing the robot car will execute $\hat{\mathbf{x}}_{0,1:T_H}$ in advance. However, we should not inform the human driver of the robot car's future motion when evaluating the consequence of the action $do(\mathbf{x}_{1,1:T_H} = \hat{\mathbf{x}}_{1,1:T_H})$. It leads to overly confident anticipation on human's reaction on aggressive maneuvers, as demonstrated in our toy example. Instead, we should evaluate $\hat{\mathbf{x}}_{0,1:T_H}$ with a model approximating the distribution $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, do(\hat{\mathbf{x}}_{1,1:T_H}))$, in other words, a model designed for the IBP task.

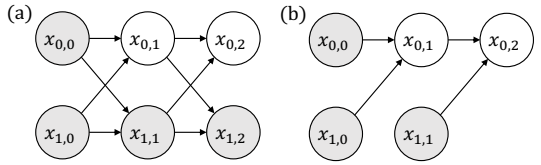


Fig. 4: (a) The Bayesian network representing the conditional distribution $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, \hat{\mathbf{x}}_{1,1:T_H})$; (b) The Bayesian network representing the distribution resulting from the intervention $do(\mathbf{x}_{1,1:T_H} = \hat{\mathbf{x}}_{1,1:T_H})$, denoted by $p(\mathbf{x}_{0,1:T_H}|\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{1,0}, do(\hat{\mathbf{x}}_{1,1:T_H}))$.

III. Quantifying the Impact of Conditional Inference on Real-World Datasets

From the toy example, we have made it clear that conditional inference leads to biased prediction and potential safety hazard. We are then curious about: 1) how conditional inference may impact interactive prediction in real-world scenarios; and 2) how we can identify a CBP model with potential safety risk. Unlike the toy example, we do not have access to the ground-truth dynamics governing the interacting agents. Meanwhile, it is expensive and dangerous to estimate and compare the conditional and interventional distributions via real-world experiments. Instead, we are interested in a method purely based on offline datasets.

Intuitively, the direct consequence of treating the query trajectory as observation is that the robot may anticipate the interacting agents to react to its future actions in advance. Therefore, we propose to detect and quantify the impact of conditional inference by looking into how much the future segment of the query trajectory contributes to the prediction at prior timesteps for a given CBP model. In particular, we adopt Shapley values as the evaluation tool.

A. Shapley Value in Explainable Deep Learning

Originated in cooperative game theory, Shapley values have been widely used in deep learning to quantify feature attribution of black-box models [14]. Shapley values quantify the attribution of each dimension of an input $x = (x_1, \cdots, x_n)$ to a function describing the model behavior $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$. The output of f could be the direct output of the model. Alternatively, f could also output an numerical value quantifying the performance or uncertainty of the model. Formally, one defines a set function $\nu : \mathbb{S} \to \mathbb{R}$ where $\mathbb{S}$ is the power set of $N := \{1, 2, \cdots, n\}$, i.e., $\mathbb{S} = P(N)$. For a subset $S \in \mathbb{S}$, the output $\nu(S)$ corresponds to running the model on a modified version of the input x for which features not in S are dropped or replaced. For instance, we may replace the dropped features $x_{N\setminus S}$ with samples drawn from their marginal distribution in the dataset [14], and then define:

$$\nu(S) = \mathbb{E}\left[f(x_S, X_{N\setminus S})\right]. \tag{1}$$

For each feature x_i, its Shapley value $\phi(x_i)$ is defined as:

$$\phi(x_i) = \sum_{S \subseteq N\setminus\{i\}} \frac{1}{n\binom{n-1}{|S|}} \left(\nu\left(S \cup \{i\}\right) - \nu\left(S\right)\right), \tag{2}$$

i.e., the difference in ν between including and not including the feature x_i averaged over all subsets S. In the context of trajectory prediction, Shapley values have been adapted to quantify the usage of social cues of prediction models [15].

B. Shapley Value for Interventional Behavior Prediction

In our case, we define the Shapley values of features based on their attribution to prediction performance. We adopt three evaluation metrics:

- *Average Displacement Error (ADE)*: Mean L_2 distance between the ground-truth and predicted trajectories averaged over K samples.
- *Final Displacement Error (FDE)*: L_2 distance between the ground-truth and predicted final position averaged over K samples.
- *Kernel Density Estimate-based Negative Log Likelihood (KDE NLL)*: Mean NLL of the ground-truth trajectory under a distribution created by fitting a kernel density estimate on K trajectory samples [8].

It is worth noting that we do not follow the common practice to evaluate the minimum of distance metrics over sampled trajectories. As argued in [15], computing the minimum leads to biased estimation. To minimize unnecessary bias in the computation of $\nu\left(S \cup \{i\}\right) - \nu\left(S\right)$, we choose a sufficiently large K and only consider the average values of the distance metrics.

The features of interest are essentially the states of the planned trajectory. The model is evaluated with the ground-truth future trajectory of the robot car, denoted by $\mathbf{x}_{r,1:T_H}$. Since additional future information is granted, we expect more accurate prediction in average after conditioning on the ground-truth robot future, which leads to non-negative Shapley values in general. The question remained is how to segment the robot future trajectory into features. Since the entire trajectory is typically treated as a sequence when encoded [8], perturbing the state at a single timestep may merely have minimal effect on the model output, as the encoder may manage to smooth out the perturbation. Also, treating the state at each timestep as a single feature leads to large n, which makes the computation of Shapley values expensive as $|P(N)|$ grows exponentially with n. Instead, we split $\hat{\mathbf{x}}_{r,1:T_H}$ into m segments, with each segment consists of states from multiple neighbouring timesteps:

$$\mathbf{x}_{r,1:T_H} = \left[\mathbf{x}_{r,1:t_1},\ \mathbf{x}_{r,t_1:t_2},\ \cdots,\ \mathbf{x}_{r,t_{m-1}:t_m}\right]. \tag{3}$$

We are then interested in evaluating the attribution of future segments to the prediction at earlier timesteps. To this end, we compute the Shapley values $\phi(\mathbf{x}_{r,t_j:t_{j+1}})$ regarding the prediction over the first t_1 timesteps. If the prediction model approximates the interventional distribution, we expect a large value for $\phi(\mathbf{x}_{r,1:t_1})$ but nearly zero for the latter segments. If any of the Shapley values for $\mathbf{x}_{r,t_j:t_{j+1}}$ with $j \geqslant 1$ is instead significant, it indicates the model learns a distribution with notable discrepancy to the interventional distribution, which may cause safety issue if deployed on on-road autonomous vehicles.

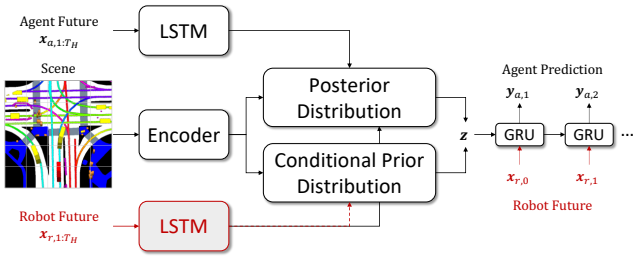


Fig. 5: The scheme of conditional behavior prediction with Trajectron++.

IV. Experiments

With the proposed toolkit, we now study the impact of conditional inference for a state-of-the-art model on a real-world prediction dataset.

A. Model and Dataset

We conduct our experiments with a state-of-the art trajectory prediction model, Trajectron++ [8], on the nuScenes dataset [19]. We choose Trajectron++ because it supports conditional trajectory prediction. More importantly, the authors showed that conditioning Trajectron++ on the future trajectory of the ego agent—referred to as *robot future* in [8]—indeed improved the model's prediction performance on the nuScenes dataset.

Trajectron++ leverages the Conditional Variational Autoencoder (CVAE) [20] framework to explicitly model the multimodality in trajectory distribution. As shown in Fig. 5, the robot future trajectory is fed into Trajectron++ through three channels: 1) feeding step-by-step into the corresponding Gated Recurrent Unit (GRU) cell [21] of the trajectory decoder; 2) feeding into the encoder modeling the posterior distribution of latent variables after encoded by a Long Short-Term Memory (LSTM) network [22]; 3) feeding into the encoder modeling the conditional prior distribution of latent variables after encoded by the same LSTM network.

The first two channels are not problematic. The computational graph of the first channel is consistent with the causal Bayesian network after intervention as in Fig. 4. For the second channel, the posterior distribution is only used during training. However, the last channel is potentially defective. During the inference stage, the latent variables are sampled from the conditional prior distribution. It takes the embedding generated by the LSTM network as input. The embedding fuses information along the entire planning horizon. Consequently, the model has access to the robot future states at latter timesteps when predicting the target agent's states at former timesteps.

We are then curious about how much performance gain is attributed to this faulty shortcut. To this end, we use the Shapley values proposed in Sec. III-B to analyze the model behavior on the nuScenes dataset. The nuScenes benchmark sets a prediction horizon of 6 seconds. We then equally split the robot future trajectory into three segments. Afterwards, we compute the Shapley values to quantify their attribution to the prediction performance within the first 2 seconds in the future. In addition, we compare Trajectron++ with a variant of it, created by masking out the third channel (i.e., the dash line in Fig. 5). By comparing the prediction performance of these two models, we can also have an idea on the effect of this shortcut on the model behavior.

To compute the Shapley values, we need to define the set function in Eqn. (1), which requires a marginal distribution of dropped features to define the expectation. Since we do not have access to the ground-truth distribution of the dataset, we train an unconditioned Trajectron++ model as an approximation. When computing the Shapley values for a given data sample, we sample trajectories from the trained Trajectron++ model's prediction of the robot car. The resulting Shapley values unravel the characteristics of the prediction model when it is queried by a motion planner imitating human behavior. Alternatively, we may estimate the expectation with the exact motion planner that will be deployed for a customized analysis.

B. Results

We computed the Shapley values over the test set for the models with and without masking the channel feeding the robot future embedding into the conditional prior encoder. The results are summarized in Table I and Fig. 6, where we summarize the statistics of ϕ_j^{ADE}, ϕ_j^{FDE}, and ϕ_j^{KDE} for $j = 1, 2, 3$. The superscript denotes the corresponding performance metric. The subscript denotes the segment of the robot future trajectory. By masking the input channel, the model satisfies the temporal independence inherited in the causal Bayesian network after intervention. Therefore, it is as expected that the Shapley values are minimal for $j > 1$. However, we can see from Fig. 6 that the values

TABLE I: Shapley Values Comparison

Mask	ϕ_1^{ADE}	ϕ_2^{ADE}	ϕ_3^{ADE}
-	0.0148 ± 0.0839	0.0049 ± 0.0444	0.0044 ± 0.0376
✓	0.0053 ± 0.0197	0.0000 ± 0.0024	0.0000 ± 0.0024
Mask	ϕ_1^{FDE}	ϕ_2^{FDE}	ϕ_3^{FDE}
-	0.0332 ± 0.1569	0.0117 ± 0.0829	0.0109 ± 0.0716
✓	0.0156 ± 0.0568	0.0000 ± 0.0045	0.0000 ± 0.0045
Mask	ϕ_1^{KDE}	ϕ_2^{KDE}	ϕ_3^{KDE}
-	0.0636 ± 0.3365	0.0192 ± 0.1725	0.0179 ± 0.1676
✓	0.0119 ± 0.0857	0.0000 ± 0.0527	0.0001 ± 0.0524

The results are presented in the format of $\text{mean} \pm \text{std}$.

TABLE II: Prediction Performance Comparison

Ablation		$minADE_{K=6}$	$minFDE_{K=6}$	KDE NLL
Robot	Mask			
-	-	1.73 ± 2.32	4.02 ± 5.62	1.86 ± 3.23
✓	-	$\mathbf{1.61 \pm 2.44}$ $(\mathbf{-6.94\%})$	3.76 ± 5.99 (-6.47%)	$\mathbf{1.61 \pm 3.73}$ $(\mathbf{-13.4\%})$
✓	✓	$\mathbf{1.61 \pm 2.43}$ $(\mathbf{-6.94\%})$	$\mathbf{3.72 \pm 5.92}$ $(\mathbf{-7.46\%})$	1.77 ± 3.43 (-4.84%)

The results are presented in the format of mean $\pm$ std. The number in the parentheses indicates the percentage of improvement regarding the unconditioned model.

are not strictly zero for all the data samples, because of the randomness in model output. In contrast, the model without masking, i.e., the original Trajectron++ model, has significantly larger Shapley values for the latter two segments. More importantly, their magnitude is not negligible compared to the Shapley value of the first segment. It means the future states of the robot can falsely affect the model's prediction at earlier timesteps.

In summary, the Shapley values suggest that the CBP model is indeed biased and could potentially cause safety hazard after deployment. It is difficult though to precisely measure its consequence without online testing. Even during online testing, the effect of the biased prediction could only be observed in those highly interactive scenarios which make up a small proportion of real-world traffic scenarios. Therefore, we argue that a cheaper solution is to prevent the bias at the design stage. Instead of developing models for the CBP task, we should turn to the IBP task. The model should be carefully designed and implemented to follow an interventional distribution. Meanwhile, the proposed Shapley values should be used to monitor the model behavior.

In particular, a prediction benchmark designed for the IBP task should include such a quantitative metric as a complement to the current prediction metrics. For instance, we may set constraints $\phi_2^{\text{FDE}} \leqslant \epsilon$ and $\phi_3^{\text{FDE}} \leqslant \epsilon$ for some small threshold value ϵ. Only those models satisfying the constraints are qualified for performance comparison. Such constraints are crucial for prediction benchmark, because the models are evaluated as black boxes. It is expensive and time-consuming in general to check that leakage does not occur in the model design and implementation. Without the constraints, the performance comparison could be misleading and unfair. For instance, we compare the performance of the models with and without masking against the unconditioned model in Table II. While the masking only slightly affects the values of $minADE$ and $minFDE$, the model without masking gains significant improvement in terms of KDE NLL. Without the Shapley values, one may consider this model with the defective input channel a better prediction model.

V. Discussion

A. Training Prediction Model for IBP

In our experiments, we demonstrated one practical way to design a prediction model for the IBP task. Same as a CBP model, the model takes the ego agent's future trajectory as input during training. However, we should ensure that the model architecture reserves the structure of the causal Bayesian network under intervention (e.g. Fig. 4(b)). Alternatively, we may train a prediction model for the joint behavior of all the agents, including the ego agent and its surroundings. For online usage, we can then conduct intervention on this joint prediction model when given a planned trajectory. In fact, some prior works follow this scheme implicitly for conditional prediction [4], [5], [7]. However, they mean to approximate the conditional distribution by enforcing the ego agent's action sequence, as it is intractable to conduct exact conditional inference on the joint prediction model. We are interested in formally comparing these two training schemes in the future.

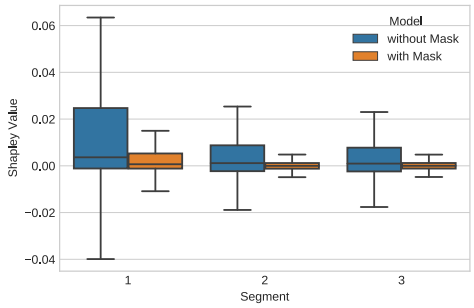
B. Establishing Prediction Benchmark for IBP

To compute the Shapley values in our experiments, we sample the ego agent's future trajectories from an unconditioned Trajectron++ model for our convenience. However, the sample distribution is sensitive to the training dataset. Also, if we want to establish a formal IBP benchmark, we cannot ensure a transparent and fair evaluation with a black-box sampling method for Shapley value computation. As a solution, we may evaluate the Shapley values with a set of plausible future trajectories generated by a model-based motion planner [23]. In our future work, we will investigate it and develop IBP benchmarks on public datasets.

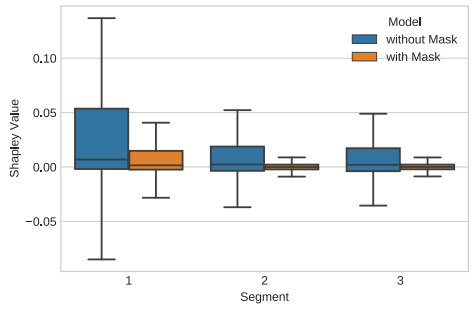
Besides, we would like to emphasize that ensuring the temporal dependence is only the basic requirement for a good IBP model. Since a planner may query an IBP model with an arbitrary planned trajectory, ideally the IBP model need to be accurate over the entire input space of planned trajectories. However, it is prohibitive in general to train such a perfect model with offline datasets. Instead, a practical solution is to equip the prediction model with a module detecting out-of-distribution (OOD) inputs of planned trajectories [24], [25], which can be utilized to prevent the planning module from exploiting the prediction model with those OOD inputs. Therefore, it is necessary to require such an OOD module for an IBP model and include the evaluation of OOD detection as a part of an IBP benchmark.

VI. Conclusion

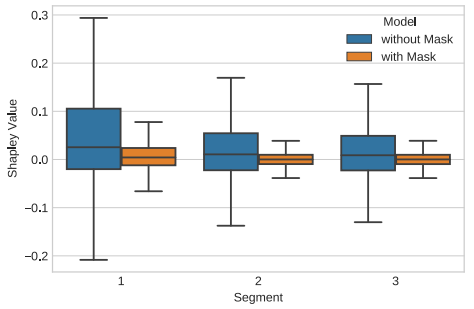
In this work, we study the problem of conditional behavior prediction, which builds up the foundation for an interactive prediction and planning framework. We argue that it is risky for the planner to query a prediction model trained for the CBP task. Instead, we should treat the planned trajectory



(a) ϕ_j^{ADE} for $j = 1, 2, 3$.



(b) ϕ_j^{FDE} for $j = 1, 2, 3$.



(c) ϕ_j^{KDE} for $j = 1, 2, 3$.

Fig. 6: Box plots of Shapley values regarding different performance metrics. We compare the Shapley values of different segments of the robot future for the two models.

as an intervention and let the model learn the trajectory distribution under intervention, which we refer to as the IBP task. Moreover, to properly evaluate an IBP model with offline datasets, we propose a Shapley-value-based metric to testify if the prediction model satisfies the inherent temporal independence of an interventional distribution. We show that the proposed metric can effectively identify a CBP model violating the temporal independence. When establishing IBP benchmarks, we can then set constraints based on the proposed metric to ensure a fair evaluation of an IBP model.

REFERENCES

[1] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 532–539, IEEE, 2021.

[2] J. Gu, Q. Sun, and H. Zhao, "Densetnt: Waymo open dataset motion prediction challenge 1st place solution," *arXiv preprint arXiv:2106.14160*, 2021.

[3] C. Tang, W. Zhan, and M. Tomizuka, "Exploring social posterior collapse in variational autoencoder for interaction modeling," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[4] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3399–3406, IEEE, 2018.

[5] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2821–2830, 2019.

[7] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.

[8] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*, pp. 683–700, Springer, 2020.

[9] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "Pip: Planning-informed trajectory prediction for autonomous driving," in *European Conference on Computer Vision*, pp. 598–614, Springer, 2020.

[10] J. Liu, W. Zeng, R. Urtasun, and E. Yumer, "Deep structured reactive planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4897–4904, IEEE, 2021.

[11] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3473–3479, IEEE, 2021.

[12] J. Pearl, *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd ed., 2009.

[13] L. S. Shapley, *Notes on the N-Person Game - II: The Value of an N-Person Game*. Santa Monica, CA: RAND Corporation, 1951.

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[15] O. Makansi, J. von Kügelgen, F. Locatello, P. Gehler, D. Janzing, T. Brox, and B. Schölkopf, "You mostly walk alone: Analyzing feature attribution in trajectory prediction," *arXiv preprint arXiv:2110.05304*, 2021.

[16] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[17] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 ed., 2010.

[19] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[20] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Conference on Robot Learning*, pp. 1035–1045, PMLR, 2022.

[24] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2020.

[25] L. Sun, X. Jia, and A. D. Dragan, "On complementing end-to-end human behavior predictors with planning," *arXiv preprint arXiv:2103.05661*, 2021.