Safe Planning in Dynamic Environments Using Conformal Prediction

Lars Lindemann, *Member, IEEE*, Matthew Cleaveland, *Member, IEEE*, Gihyun Shim, and George J. Pappas, *Fellow, IEEE*

***Abstract*—We propose a framework for planning in unknown dynamic environments with probabilistic safety guarantees using conformal prediction. Particularly, we design a model predictive controller (MPC) that uses i) trajectory predictions of the dynamic environment, and ii) prediction regions quantifying the uncertainty of the predictions. To obtain prediction regions, we use conformal prediction, a statistical tool for uncertainty quantification, that requires availability of offline trajectory data – a reasonable assumption in many applications such as autonomous driving. The prediction regions are valid, i.e., they hold with a user-defined probability, so that the MPC is provably safe. We illustrate the results in the self-driving car simulator CARLA at a pedestrian-filled intersection. The strength of our approach is compatibility with state of the art trajectory predictors, e.g., RNNs and LSTMs, while making no assumptions on the underlying trajectory-generating distribution. To the best of our knowledge, these are the first results that provide valid safety guarantees in such a setting.**

***Index Terms*—Planning under uncertainty, robot safety, motion planning, autonomous agents, AI-enabled robotics.**

I. Introduction

MOBILE robots and autonomous systems operate in dynamic and shared environments. Consider for instance a self-driving car navigating through urban traffic, or a service robot planning a path while avoiding other agents such as pedestrians, see Fig. 1. These applications are safety-critical and challenging as the agents' intentions and policies are unknown so that their a-priori unknown trajectories need to be estimated and integrated into the planning algorithm. We propose an uncertainty-informed planning algorithm that enjoys formal safety guarantees by using conformal prediction.

The problem of path planning in dynamic environments has found broad attention [1]. A large body of work focused on multi-agent navigation without incorporating predicted agent trajectories, e.g., the dynamic window approach [2], [3] or navigation functions [4], [5], [6]. However, predicted trajectories provide additional information and can significantly increase the quality of the robot's path in terms of safety and performance. Existing works that use trajectory predictions can be broadly classified into two categories: non-interactive and interactive. Non-interactive approaches predict agent trajectories and then integrate predictions into the planning algorithm [7], [8]. Interactive approaches simultaneously predict agent trajectories and design the path to take the coupling effect between a control action and the trajectories of other agents into account [9], [10]. While interactive approaches attempt to model interactions between actions and agents, this is generally a difficult task and existing works fail to provide quantifiable safety guarantees.

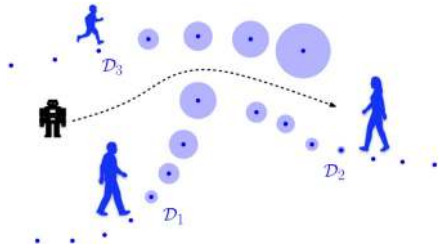


Fig. 1. We predict agent trajectories using state of the art prediction algorithms, such as LSTMs, and calculate valid prediction regions (blue circles) using conformal prediction.

In this letter, we focus on designing non-interactive planning algorithms with valid safety guarantees. Particularly, we use statistical tools from the conformal prediction literature [11], [12] to obtain valid prediction regions that quantify the uncertainty of trajectory predictions. We then formulate a model predictive controller (MPC) that incorporates trajectory predictions and valid prediction regions. While our framework is compatible with any trajectory prediction algorithm, we focus in the experiments on long short term memory (LSTM) networks [13], [14], [15], [16], [17], [18] which are special recurrent neural networks (RNN) that can capture nonlinear and long-term trends [19], [20]. Our contributions are:

- We propose a planning algorithm that incorporates trajectory predictions and valid prediction regions which are obtained using conformal prediction. The elegance in using conformal prediction is that prediction regions are easy to obtain and tight. Our algorithm is computationally tractable and, under reasonable assumptions, based on a convex optimization problem.

Manuscript received 22 October 2022; accepted 5 June 2023. Date of publication 4 July 2023; date of current version 12 July 2023. This letter was recommended for publication by Associate Editor J. Hu and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported in part by the NSF Award under Grant CPS-2038873 and in part by AFOSR under Grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments). *(Lars Lindemann and Matthew Cleaveland contributed equally to this work.) (Corresponding author: Lars Lindemann.)*

Lars Lindemann is with the Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: llindema@usc.edu).

Matthew Cleaveland, Gihyun Shim, and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: mcleav@seas.upenn.edu; gihyun@seas.upenn.edu; pappasg@seas.upenn.edu).

Digital Object Identifier 10.1109/LRA.2023.3292071

2377-3766 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

- We provide valid safety guarantees which guarantee that the system is safe with a user-defined probability. Larger user-defined probabilities naturally result in more conservative plans. The strength of our approach is compatibility with state of the art trajectory predictors, e.g., RNNs and LSTMs, while making no assumptions on the underlying trajectory-generating distribution.
- We provide numerical experiments of a mobile robot and a self-driving car using the TrajNet++ toolbox [17] and the autonomous driving simulator CARLA [21].

A. Related Work

The works in [22], [23], [24], [25], [26] present non-interactive sampling-based motion planners, while [8], [27], [28], [29] propose non-interactive receding horizon planning algorithms that minimize the risk of collision. A challenge in non-interactive methods is the robot freezing problem in which robots may come to a deadlock due to too large prediction uncertainty [7], e.g., for long time horizons. While this problem can be alleviated by receding horizon planning strategies, another direction to address this problem is to model social interaction as in interactive approaches where typically an interaction model is learned or constructed [9], [30], [31], [32], [33], [34], [35], [36], [37]. Reinforcement learning approaches that take social interaction into account were presented in [10], [38].

A particular challenge lies in selecting a good predictive model. Recent works have used intent-driven models for human agents where model uncertainty was estimated using Bayesian inference and then used for planning [39], [40], [41]. Other works considered Gaussian processes as a predictive model [42], [43], [44]. To the best of our knowledge, none of the aforementioned works provide valid safety guarantees unless strong assumptions are placed on the prediction algorithm and the agent model or its distribution, e.g., being Gaussian.

We focus instead on the predictive strength of neural networks. Particularly, RNNs and LSTMs have shown to be applicable to time-series forecasting [19], [20]. They were successfully applied in domains such as speech/handwriting recognition and image classification [45], [46], [47], [48], but also in trajectory prediction [13], [14], [15], [16], [17], [18]. We will specifically use the social LSTM presented in [16] that can jointly predict agent trajectories by taking social interaction into account.

Neural network predictors, however, provide no information about the uncertainty of a prediction so that wrong predictions can lead to unsafe decisions. Therefore, monitors were constructed in [49], [50] to detect prediction failures – particularly [50] used conformal prediction to obtain guarantees on the predictor's false negative rate. Conformal prediction was also used to estimate reachable sets via neural network predictors [51], [52], [53]. Conceptually closest to our work is [54] where a valid predictor is constructed using conformal prediction, and then utilized to design a model predictive controller. However, no safety guarantees for the planner can be provided as the predictor uses a finite collection of training trajectories to represent all possible trajectories, implicitly requiring training and test trajectories to be similar. Our approach directly predicts trajectories of the dynamic environment (e.g., using RNNs or LSTMs) along with valid prediction regions so that we can provide end-to-end safety guarantees for our planner.

II. Problem Formulation

We first define the safe planning problem in dynamic environments that we consider, and then briefly discuss methods for trajectory prediction of dynamic agents.

A. Safe Planning in Dynamic Environments

Consider the discrete-time dynamical system

$$x_{t+1} = f(x_t, u_t), \quad x_0 := \zeta \tag{1}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ denote the state and the control input at time $t \in \mathbb{N} \cup \{0\}$, respectively. The sets $\mathcal{U}$ and $\mathcal{X}$ denote the set of permissible control inputs and the workspace of the system, respectively. The measurable function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ describes the system dynamics and $\zeta \in \mathbb{R}^n$ is the initial condition of the system.

The system operates in an environment with N dynamic agents whose trajectories are a priori unknown. Let $\mathcal{D}$ be an unknown distribution over agent trajectories and let

$$(Y_0, Y_1, \ldots) \sim \mathcal{D}$$

describe a random trajectory where the joint agent state $Y_t := (Y_{t,1}, \ldots, Y_{t,N})$ at time t is drawn from $\mathbb{R}^{Nn}$, i.e., $Y_{t,j}$ is the state of agent j at time t.[1] We assume at time t to have knowledge of $(Y_0, \ldots, Y_t)$. Modeling dynamic agents by a distribution $\mathcal{D}$ provides great flexibility, e.g., the pedestrians in Fig. 1 can be described by distributions $\mathcal{D}_1, \mathcal{D}_2$, and $\mathcal{D}_3$ with joint distribution $\mathcal{D}$, and $\mathcal{D}$ can generally describe the motion of Markov decision processes. We make no assumptions on the form of the distribution $\mathcal{D}$, but assume that $\mathcal{D}$ is independent of the system in (1) as formalized next.

Assumption 1: For any time $t \geq 0$, the control inputs $(u_0, \ldots, u_{t-1})$ and the resulting trajectory $(x_0, \ldots, x_t)$, following (1), do not change the distribution of $(Y_0, Y_1, \ldots) \sim \mathcal{D}$.

Assumption 1 approximately holds in many applications, e.g., a self-driving car taking conservative control actions that result in socially acceptable trajectories which do not change the behavior of pedestrians. We later comment on ways to deal with distribution shifts in practice, and reserve a thorough treatment of this issue for future papers. We further assume availability of training and calibration data drawn from $\mathcal{D}$.

Assumption 2: We have a dataset $D := \{Y^{(1)}, \ldots, Y^{(K)}\}$ in which each of the K trajectories $Y^{(i)} := (Y_0^{(i)}, Y_1^{(i)}, \ldots)$ is independently drawn from $\mathcal{D}$, i.e., $Y^{(i)} \sim \mathcal{D}$.

Assumption 2 is not restrictive in practice, e.g., pedestrian data is available in autonomous driving. Let us now define the problem that we aim to solve in this letter.

Problem 1: Given the system in (1), the random trajectory $(Y_0, Y_1 \ldots) \sim \mathcal{D}$, a mission time T, and a failure probability $\delta \in (0, 1)$, design the control inputs u_t such that the Lipschitz continuous constraint function $c : \mathbb{R}^n \times \mathbb{R}^{nN} \to \mathbb{R}$ is satisfied[2]

[1] For simplicity, we assume that the state of each dynamic agent is n-dimensional. This assumption can easily be generalized.

[2] We assume that c is initially satisfied, i.e., that $c(x_0, Y_0, 0) \geq 0$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

with a probability of at least $1-\delta$, i.e., that

$$P\left(c(x_t, Y_t) \geq 0, \forall t \in \{0, \dots, T\}\right) \geq 1-\delta.$$

We note that the function c can encode collision avoidance, but also objectives such as tracking another agent. In our solution to Problem 1, we additionally achieve cost optimality in terms of a cost function J (details below).

B. Trajectory Predictors for Dynamic Environments

Our goal is to predict future agent states $(Y_{t+1}, \dots, Y_T)$ from observations $(Y_0, \dots, Y_t)$. Our proposed planning algorithm is compatible with any trajectory prediction algorithm. Assume that PREDICT is a measureable function that maps observations $(Y_0, \dots, Y_t)$ to predictions $(\hat{Y}_{t+1|t}, \dots, \hat{Y}_{T|t})$ of $(Y_{t+1}, \dots, Y_T)$. We now split the dataset D into training and calibration datasets D_{train} and D_{cal}, respectively, and assume that PREDICT is learned from D_{train}.

A specific example of PREDICT are recurrent neural networks (RNNs) that have shown good performance [20]. For $\tau \leq t$, the recurrent structure of an RNN is given as

$$\begin{aligned} h^1_\tau &:= \mathcal{H}(Y_\tau, h^1_{\tau-1}), \\ h^i_\tau &:= \mathcal{H}(Y_\tau, h^i_{\tau-1}, h^{i-1}_\tau), \quad \forall i \in \{2, \dots, d\} \\ \hat{Y}_{\tau+1|\tau} &:= \mathcal{Y}(h^d_\tau), \end{aligned}$$

where Y_τ is the input that is sequentially applied and $\mathcal{H}$ is a function that can parameterize different types of RNNs, e.g., LSTMs [14]. Furthermore, d is its depth and $h^1_\tau, \dots, h^d_\tau$ are the hidden states. The output $\hat{Y}_{t+1|t}$ provides an estimate of Y_{t+1}, see e.g., [18], [55] where the function $\mathcal{Y}$ parameterizes a predictive conditional distribution. To obtain the remaining predictions $\hat{Y}_{t+2|t}, \dots, \hat{Y}_{T|t}$ of $Y_{t+2}, \dots, Y_T$, we use the RNN in a recursive way by sequentially applying $\hat{Y}_{t+1|t}, \dots, \hat{Y}_{T-1|t}$ instead of the unknown $Y_{t+1}, \dots, Y_{T-1}$.

III. CONFORMAL PREDICTION REGIONS FOR TRAJECTORY PREDICTORS

The challenges in solving Problem 1 are twofold. First, the unknown distribution $\mathcal{D}$ over trajectories $(Y_0, Y_1, \dots)$ can be complex and may not follow standard assumptions, e.g., being Gaussian. Second, the function PREDICT can be highly nonlinear and predictions $\hat{Y}_{\tau|t}$ may not be accurate. To be able to deal with these challenges, we use conformal prediction to obtain *prediction regions* for $\hat{Y}_{\tau|t}$.

A. Introduction to Conformal Prediction

Conformal prediction was introduced in [11], [12] to obtain valid prediction regions for complex predictive models, i.e., neural networks, without making assumptions on the underlying distribution or the predictive model [56], [57], [58], [59], [60]. We first provide a brief introduction to conformal prediction.

Let $R^{(0)}, \dots, R^{(k)}$ be $k+1$ independent and identically distributed random variables.[3] The variable $R^{(i)}$ is usually referred to as the *nonconformity score*. In supervised learning, it may be defined as $R^{(i)} := \|Z^{(i)} - \mu(X^{(i)})\|$ where the predictor μ attempts to predict the output $Z^{(i)}$ based on the input $X^{(i)}$. A large nonconformity score indicates a poor predictive model. Our goal is to obtain a prediction region for $R^{(0)}$ based on $R^{(1)}, \dots, R^{(k)}$, i.e., the random variable $R^{(0)}$ should be contained within the prediction region with high probability. Formally, given a failure probability $\delta \in (0,1)$, we want to construct a valid prediction region C so that[4]

$$P(R^{(0)} \leq C) \geq 1-\delta.$$

By a surprisingly simple quantile argument, see [59, Lemma 1], one can obtain C to be the $(1-\delta)$th quantile of the empirical distribution of the values $R^{(1)}, \dots, R^{(k)}$ and ∞. By assuming that $R^{(1)}, \dots, R^{(k)}$ are sorted in non-decreasing order, and by adding $R^{(k+1)} := \infty$, we can equivalently obtain $C := R^{(p)}$ where $p := \lceil (k+1)(1-\delta) \rceil$ with $\lceil \cdot \rceil$ being the ceiling function, i.e., C is the pth smallest nonconformity score. In the next section, we discuss how to obtain conformal prediction regions for trajectory predictions. We are inspired by [61] where prediction regions for RNNs were derived, but present results that are more taylored for our problem setup.

B. Prediction Regions Using Conformal Prediction

Given observations $(Y_0, \dots, Y_t)$ at time t, recall that we can obtain predictions $\hat{Y}_{\tau|t}$ of Y_τ for all future times $\tau \in \{t+1, \dots, T\}$ via the PREDICT function. For a failure probability of $\bar{\delta} \in (0,1)$, our first goal is now to construct prediction regions defined by a value $C_{\tau|t}$ so that

$$P\left(\|Y_\tau - \hat{Y}_{\tau|t}\| \leq C_{\tau|t}\right) \geq 1-\bar{\delta}.$$

Following Section III-A, we define the nonconformity score $R_{\tau|t} := \|Y_\tau - \hat{Y}_{\tau|t}\|$ so that a small (large) nonconformity score indicates that the predictions $\hat{Y}_{\tau|t}$ are accurate (not accurate). Indeed, let us compute the nonconformity score for each trajectory $Y^{(i)}$ of the calibration dataset D_{cal} as

$$R^{(i)}_{\tau|t} := \|Y^{(i)}_\tau - \hat{Y}^{(i)}_{\tau|t}\|$$

where $\hat{Y}^{(i)}_{\tau|t}$ is the prediction obtained from $(Y^{(i)}_0, \dots, Y^{(i)}_t)$.

We can now obtain prediction regions by a direct application of [59, Lemma 1]. Assume hence that the values of $R^{(i)}_{\tau|t}$ are sorted in non-decreasing order, and let us add $R^{(|D_{\text{cal}}|+1)}_{\tau|t} := \infty$ as the $(|D_{\text{cal}}|+1)$th value.

Lemma 1 (Following [11]): Given the random trajectory $(Y_0, Y_1, \dots) \sim \mathcal{D}$, the calibration dataset D_{cal}, and predictions $\hat{Y}_{\tau|t}$ obtained from observations $(Y_0, \dots, Y_t)$. Let $\bar{\delta} \in (0,1)$ be a failure probability, then for $\tau > t$ we have

$$P\left(\|Y_\tau - \hat{Y}_{\tau|t}\| \leq C_{\tau|t}\right) \geq 1-\bar{\delta}$$

[3] In general only exchangeability is needed, which means that the joint distribution of $R^{(0)}, \dots, R^{(k)}$ is the same as the joint distribution of $R^{(\sigma(0))}, \dots, R^{(\sigma(k))}$ for any permutation σ on $\{0, \dots k\}$. Exchangeability is a strictly weaker requirement than independence and identical distribution.

[4] More formally, we would have to write $C(R^{(1)}, \dots, R^{(k)})$ as the prediction region C is a function of $R^{(1)}, \dots, R^{(k)}$. For this reason, the probability measure P is defined over the product measure of $R^{(0)}, \dots, R^{(k)}$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

where the prediction regions $C_{\tau|t}$ are defined by

$$C_{\tau|t} := R^{(p)}_{\tau|t} \text{ with } p := \lceil (|D_{\text{cal}}|+1)(1-\bar{\delta}) \rceil. \tag{2}$$

Proof: By construction of the nonconformity scores $R^{(i)}_{\tau|t}$, they are exchangeable and we can directly apply [59, Lemma 1] so that $P(\|Y_\tau - \hat{Y}_{\tau|t}\| \leq C_{\tau|t}) \geq 1-\bar{\delta}$. ■

We can now construct prediction regions over multiple future predictions by an application of Boole's inequality.

Theorem 1: Given the random trajectory $(Y_0, Y_1, \ldots) \sim \mathcal{D}$, the calibration dataset D_{cal}, and predictions $\hat{Y}_{\tau|t}$ obtained from observations $(Y_0, \ldots, Y_t)$. Let $\bar{\delta} := \delta/T$ where $\delta \in (0,1)$ is a failure probability, then the following two statements hold where $C_{\tau|t}$ is constructed as in (2):

$$P\left(\|Y_\tau - \hat{Y}_{\tau|0}\| \leq C_{\tau|0},\ \forall \tau \in \{1,\ldots,T\}\right) \geq 1-\delta, \tag{3}$$

$$P\left(\|Y_{t+1} - \hat{Y}_{t+1|t}\| \leq C_{t+1|t},\ \forall t \in \{0,\ldots,T-1\}\right) \geq 1-\delta. \tag{4}$$

Proof: Let us first show that the statement in (3) holds. According to Lemma 1, it holds that $P(\|Y_\tau - \hat{Y}_{\tau|0}\| \leq C_{\tau|0}) \geq 1-\bar{\delta}$ for each $\tau \in \{1,\ldots,T\}$ individually. We consequently know that $P(\|Y_\tau - \hat{Y}_{\tau|0}\| > C_{\tau|0}) \leq \bar{\delta}$. Applying Boole's inequality gives us

$$P\left(\exists \tau > 0 \text{ s.t. } \|Y_\tau - \hat{Y}_{\tau|0}\| > C_{\tau|0}\right) \leq \sum_{i=t}^{T} \bar{\delta} = \sum_{i=t}^{T} \frac{\delta}{T} = \delta$$

so that we can finally conclude that

$$P\left(\|Y_\tau - \hat{Y}_{\tau|0}\| \leq C_{\tau|0},\ \forall \tau \in \{1,\ldots,T\}\right) \geq 1-\delta$$

which proves (3). Statement (4) follows analogously. ■

Equation (3) guarantees that all τ-step ahead prediction regions at time zero are valid, while (4) guarantees that all one-step ahead prediction regions are valid. We note that we set $\bar{\delta} := \delta/T$ to compute $C_{\tau|t}$ according to (2) to obtain prediction regions over multiple time steps. As a consequence, the value of $C_{\tau|t}$ increases with increasing T or decreasing δ as smaller $\bar{\delta}$ result in larger quantiles p.

Remark 1: The result in (3) was similarly shown in [61], but without permitting recursive RNNs, i.e., RNNs where predictions $\hat{Y}_{\tau|t}$ are recursively used to predict $\hat{Y}_{\tau+1|t}$. We permit any measurable predictor PREDICT. Importantly, we also show that (4) holds, which will be important for proving correctness guarantees of our proposed MPC.

Remark 2: For the applications that we plan to address, it can be useful to instead consider a nonconformity score

$$R_{\tau|t} := \max_{j \in \{1,\ldots,N\}} \|Y_{\tau,j} - \hat{Y}_{\tau|t,j}\|$$

where we recall that $Y_{\tau,j}$ is the state of agent j at time τ, and where $\hat{Y}_{\tau|t,j}$ is the corresponding prediction for agent j. We can obtain prediction regions $P(R_{\tau|t} \leq C_{\tau|t}) \geq 1-\bar{\delta}$ in the same way as described above. Using this nonconformity score, we get prediction regions individually for each agent.

IV. Model Predictive Control With Conformal Prediction Regions

We next propose an MPC that uses the predictions $\hat{Y}_{\tau|t}$ and the prediction regions defined by $C_{\tau|t}$ to solve Problem 1. Let us first present the optimization problem that will be iteratively solved within the MPC. Therefore, denote the Lipschitz constant of c by L. For instance, the collision avoidance constraint $c(x,y) := \|x-y\| - 0.5$ has Lipschitz constant one, i.e., $|c(x,y') - c(x,y'')| \leq \|y' - y''\|$. At time t, we solve the following optimization problem to obtain an open-loop control sequence $u_t, \ldots, u_{T-1}$:

$$\min_{(u_t,\ldots,u_{T-1})} J(x,u) \tag{5a}$$

$$\text{s.t. } x_{\tau+1} = f(x_\tau, u_\tau), \qquad \tau \in \{t,\ldots,T-1\} \tag{5b}$$

$$c(x_\tau, \hat{Y}_{\tau|t}) \geq LC_{\tau|t}, \qquad \tau \in \{t+1,\ldots,t+H\} \tag{5c}$$

$$u_\tau \in \mathcal{U}, x_{\tau+1} \in \mathcal{X}, \qquad \tau \in \{t,\ldots,T-1\} \tag{5d}$$

where H is a prediction horizon and J is a cost function over states $x := (x_1, \ldots, x_T)$ and control inputs $u := (u_0, \ldots, u_{T-1})$. The optimization problem (5) is convex if the functions J and f are convex, the function c is convex in its first argument, and the sets $\mathcal{U}$ and $\mathcal{X}$ are convex.

If one can solve the optimization problem (5) at time $t=0$, one obtains a control sequence u that solves Problem 1.

Theorem 2 (Open-loop control): Let the system (1) be given, and let the conditions from Theorem 1 and Assumption 1 hold. If the optimization problem (5) is feasible at time $t=0$ with prediction horizon $H=T$, then the open-loop control sequence u from (5) is s.t.

$$P\left(c(x_\tau, Y_\tau) \geq 0, \forall \tau \in \{1,\ldots,T\}\right) \geq 1-\delta.$$

Proof: Due to constraint (5c), it holds that

$$\begin{aligned} 0 &\leq c(x_\tau, \hat{Y}_{\tau|0}) - LC_{\tau|0} \\ &\leq c(x_\tau, Y_\tau) + L\|Y_\tau - \hat{Y}_{\tau|0}\| - LC_{\tau|0} \end{aligned}$$

where the latter inequality follows by Lipschitz continuity. Now, by (3) in Theorem 1, it directly follows that $P(c(x_\tau, Y_\tau) \geq 0, \forall \tau \in \{1,\ldots,T\}) \geq 1-\delta$. ■

While the open-loop controller in Theorem 2 provides a solution to Problem 1, the controller will admittedly be conservative, or the optimization problem in (5) may even be infeasible. The main reason for this is that the prediction regions defined by $C_{\tau|0}$ will be large for large τ as the predictions $\hat{Y}_{\tau|0}$ will lose accuracy. Another drawback of an open-loop controller is the missing robustness due to the lack of feedback. We propose a receding horizon control strategy in Algorithm 1 that reduces conservatism and is robust.

In lines 1 and 2 of Algorithm 1, we set the variables $\bar{\delta}$ and p according to Lemma 1 and Theorem 1. Lines 3-9 present the computation of the conformal prediction regions by: 1) calculating the predictions $\hat{Y}^{(i)}_{\tau|t}$ on the calibration data D_{cal}, 2) calculating the nonconformity scores $R^{(i)}_{\tau|t}$, and 3) obtaining $C_{\tau|t} := R^{(p)}_{\tau|t}$ according to Theorem 1. Lines 10–14 are the

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: MPC With Conformal Prediction Regions.

Input: Failure probability δ, calibration dataset D_{cal} obtained from $\mathcal{D}$, prediction and task horizons H and T
Output: Feedback controller $u_0(x_0), \dots, u_{T-1}(x_{T-1})$
1: $\bar{\delta} \leftarrow \delta / T$
2: $p \leftarrow \lceil (|D_{\text{cal}}| + 1)(1 - \bar{\delta}) \rceil$
3: **for** t from 0 to $T-1$ **do** # offline computation
4: **for** τ from $t+1$ to $t+H$ **do** # conformal prediction
5: Obtain predictions $\hat{Y}^{(i)}_{\tau|t}$ for each $Y^{(i)} \in D_{\text{cal}}$
6: $R^{(i)}_{\tau|t} \leftarrow \|Y^{(i)}_{\tau} - \hat{Y}^{(i)}_{\tau|t}\|$ for each $Y^{(i)} \in D_{\text{cal}}$
7: $R^{(|D_{\text{cal}}|+1)}_{\tau|t} \leftarrow \infty$
8: Sort $R^{(i)}_{\tau|t}$ in non-decreasing order
9: $C_{\tau|t} \leftarrow R^{(p)}_{\tau|t}$
10: **for** t from 0 to $T-1$ **do** # real-time planning loop
11: Sense x_t and Y_t
12: Obtain predictions $\hat{Y}_{\tau|t}$ for $\tau \in \{t+1, \dots, t+H\}$
13: Calculate controls $u_t, \dots, u_{T-1}$ as the solution of (5)
14: Apply u_t to (1)

real-time planning loop in which we observe our states x_t and Y_t (line 11), obtain new predictions $\hat{Y}_{\tau|t}$ based on Y_t (line 12), and solve the optimization problem in (5) of which we apply only u_t (lines 13–14). The MPC presented in Algorithm 1 enjoys the following guarantees.

Theorem 3 (Closed-loop control): Let the system (1) be given, and let the conditions from Theorem 1 and Assumption 1 hold. If the optimization problem (5) is feasible at each time $t \in \{0, \dots, T-1\}$, then the closed-loop control $u_0(x_0), \dots, u_{T-1}(x_{T-1})$ obtained from Algorithm 1 is s.t.

$$P\left(c(x_t, Y_t) \geq 0, \forall t \in \{1, \dots, T\}\right) \geq 1 - \delta.$$

Proof: By assumption, the optimization problem (5) is feasible at each time $t \in \{0, \dots, T-1\}$. Due to constraint (5c) and Lipschitz continuity of c, it hence holds that

$$\begin{aligned} 0 &\leq c(x_{t+1}, \hat{Y}_{t+1|t}) - LC_{t+1|t} \\ &\leq c(x_{t+1}, Y_{t+1}) + L\|Y_{t+1} - \hat{Y}_{t+1|t}\| - LC_{t+1|t} \end{aligned}$$

at each time $t \in \{0, \dots, T-1\}$. By (4) in Theorem 1, it now follows that $P(c(x_{t+1}, Y_{t+1}) \geq 0, \forall t \in \{0, \dots, T-1\}) \geq 1 - \delta$ which proves the main result. ■

We conclude with final remarks on parameter choices and recursive feasibility of the MPC and distribution shifts in $\mathcal{D}$.

Remark 3: A smaller failure probability δ leads to larger prediction regions $C_{\tau|t}$, as remarked before, and hence to less optimal paths w.r.t. the cost function J. Larger prediction horizons H may also lead to less optimal paths as future predictions lose accuracy. For too small H, however, one may experience recursive feasibility issues of the optimization problem (5). While we assume recursive feasibility in Theorem 3, we note that this is a reasonable assumption when $\|Y_{t+H} - Y_{t+H+1}\|$ is not changing too much. Note that the time-varying nature of prediction regions is typically no problem for recursive feasibility as prediction regions shrink over time, e.g., $C_{\tau|t}$ will typically be smaller than $C_{\tau+1|t}$.

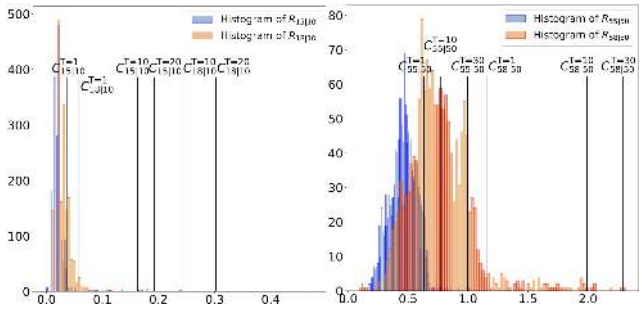


Fig. 2. Left: nonconformity scores $R^{(i)}_{15|10}$ and $R^{(i)}_{18|10}$ on D_{cal} along with $C_{15|10}$ and $C_{18|10}$ for different T for Scenario 1. Right: nonconformity scores $R^{(i)}_{55|50}$ and $R^{(i)}_{58|50}$ on D_{cal} along with $C_{55|50}$ and $C_{58|50}$ for different T for Scenario 2.

Remark 4: We assume that test and calibration trajectories follow the same distribution $\mathcal{D}$ per Assumptions 1 and 2. We hence ignore that the trajectory x may change $\mathcal{D}$ during test time, e.g., when a robot is too close to a pedestrian. While we do not address distribution shifts in full generality, e.g., when $\mathcal{D}(x)$ depends explicitly on x, we can use robust conformal prediction [60] to obtain valid prediction regions for all distributions that are "close" to $\mathcal{D}$ (in terms of the f-divergence), and integrate these in an MPC.

V. Case Studies

In the first case study, we consider navigating a mobile robot around pedestrians whose trajectories are generated in TrajNet++ [17] using the ORCA simulator [62]. In the second case study, we control a self-driving car at an intersection filled with pedestrians in CARLA [21], see Fig. 1.

For trajectory prediction, we use the social LSTM from [16]. Compared to vanilla LSTMs as introduced in Section II-B, the social LSTM uses one LSTM for each agent while sharing LSTM weights via a pooling layer to model social interactions. These pooled LSTM weights are then used as inputs of the individual LSTMs, see [16] for details. For both case studies, we trained a social LSTM with a depth of $d := 128$ for the individual agent LSTM.

In both case studies, we consider a bicycle model [63]

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \Delta v_t \cos(\theta_t) \\ x_t + \Delta v_t \sin(\theta_t) \\ \theta_t + \Delta \frac{v_t}{l} \tan(\phi_t) \\ v_t + \Delta a_t \end{bmatrix}$$

where $p_\tau := (x_t, y_t)$ is the two-dimensional position, θ_t is the vehicles's orientation, v_t is the velocity, $l := 1$ is the vehicles's length, and Δ is the sampling time. The control inputs are the steering angle ϕ_t and the acceleration a_t.

The objective is to reach a goal region, while avoiding the pedestrians by means of the constraint function

$$c(p_\tau, Y_\tau) := \min_{j \in \{1, \dots, N\}} \|p_\tau - Y_{\tau,j}\| - \epsilon$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

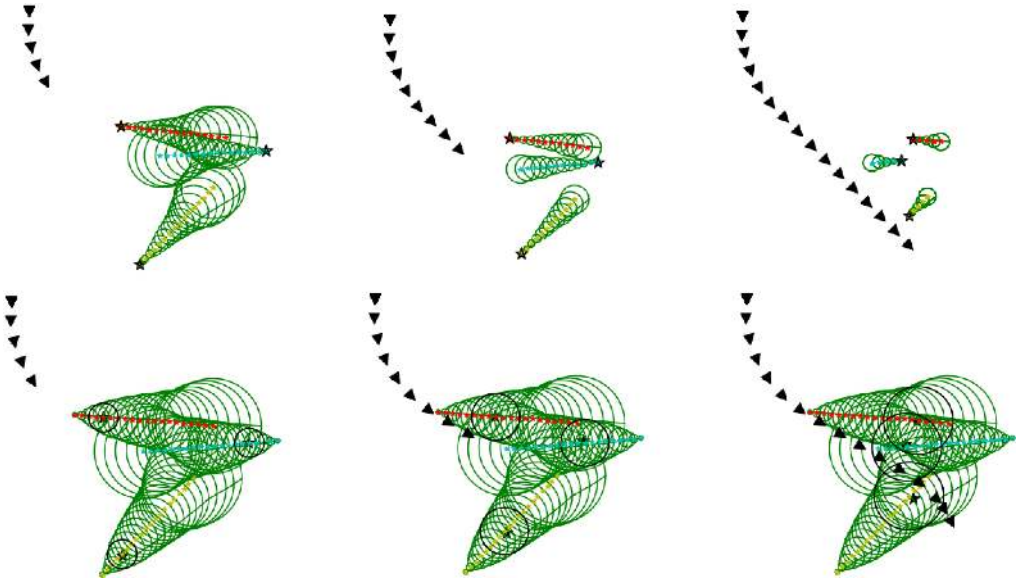


Fig. 3. Scenario 1. The top three plots show the robot trajectory (black triangles) produced by the proposed MPC at times 4, 8, and 15. There are three pedestrians, with their actual trajectories indicated by black stars, that the robot safely avoids. The LSTM predictions are indicated by red stars with the corresponding prediction regions shown in green. The prediction regions are updated at each time making the MPC less conservative than the open-loop controller, which is presented in the bottom three plots. For the open-loop controller, the prediction regions are fixed from time zero and given by $C_{\tau|0}$.

where ϵ is a user-defined safety distance. In the first case study, we encode reaching the goal region as a constraint $\|p_T - p_{\text{goal}}\| \leq 0.25$, while we minimize $\|p_T - p_{\text{goal}}\|$ in the cost function J in the second case study. To solve the optimization problem in (5), we use CasADi [64] with the Ipopt nonlinear programming solver. Animations for both case studies can be found at https://tinyurl.com/ecz2a9c4https://tinyurl.com/ecz2a9c4.

Scenario 1 (ORCA). We collected 4500 synthetic trajectories in a scene consisting of three pedestrians, and created training, calibration, and test datasets with sizes $|D_{\text{train}}| = 2000$, $|D_{\text{cal}}| = 2000$, and $|D_{\text{test}}| := 500$, respectively. The sampling time is $\Delta := 1/8$, and Fig. 2 (left) shows histograms of the nonconformity scores $R^{(i)}_{\tau|t}$ evaluated on D_{cal} for $\tau \in \{15, 18\}$ and $t := 10$ (with R defined as in Remark 2). Based on these nonconformity scores, we can calculate $C_{\tau|t}$ according to Theorem 1 by using $\delta := 0.05$ and $T := 20$. In Fig. 2 (left), we additionally indicate $C_{\tau|t}$ for different values of T, and we can observe that larger T naturally result in larger prediction regions. Next, we empirically evaluate the correctness of the prediction regions by checking whether or not (3) holds on D_{test}. Indeed, we find that 498 of the 500 test trajectories are such that $\|Y^{(i)}_\tau - \hat{Y}^{(i)}_{\tau|0}\| \leq C_{\tau|0}$ which empirically confirms (3) in Theorem 1. For one of these test trajectories, Fig. 3 shows the prediction regions defined by $\hat{Y}_{\tau|t}$ and $C_{\tau|t}$ for $\tau > t$ where $t = 0$ (bottom plots) $t = 4$ (top left plot), $t = 8$ (top middle plot), $t = 15$ (top right plot). We note that the LSTM at time $t = 0$ for simplicity uses information $Y_{-20}, \ldots, Y_{-1}$. We can observe that prediction regions become larger for larger τ.

Fig. 3 shows the result of the proposed MPC (here with $H := T$) in the top three plots for a trajectory $(Y_0, Y_1, \ldots)$ from D_{test}. For comparison, we present the results for the open-loop controller from Theorem 2, i.e., only applying the control sequence obtained at time zero, in the bottom three plots of Fig. 3. For illustration purposes, we have set $\epsilon := 0$. It is visible that both the open and closed loop MPC controllers are such that the pedestrians are avoided according to the constraint function c. This has to hold in at least 95 percent of the cases by Theorems 2 and 3 since we have selected $\delta = 0.05$. The cost function J in the open and closed loop MPC controllers additionally minimizes $\sum_{\tau=1}^{T} v_\tau^2$ to avoid large velocities, which also acts as a proxy to minimize the total distance traveled. The closed loop MPC is, as expected and motivated before, much less conservative than the open-loop controller and finds a more direct trajectory towards the goal region. To corroborate this observation, we calculated the averaged cost over 500 test trajectories which are 414 for the open-loop controller and 367.1 for the closed loop controller.



Fig. 4. Intersection with pedestrians in CARLA [21].

Scenario 2 (CARLA). Within the autonomous driving simulator CARLA we have created a scene consisting of an intersection that is populated with four pedestrians, see Fig. 4. These pedestrians have random initial and goal positions, and walk at random speeds to reach their goal. We collected 2600 trajectories, and then created training, calibration, and test datasets with sizes $|D_{\text{train}}| = 1000$, $|D_{\text{cal}}| = 1500$, and $|D_{\text{test}}| = 100$, respectively, and train an LSTM from D_{train} to predict pedestrian trajectories. We run CARLA at a frequency of 20 Hz for 30 seconds, but use the LSTM and solve the MPC at 2 Hz. We show the histograms of the nonconformity scores $R^{(i)}_{\tau|t}$ evaluated on D_{cal} for $\tau \in \{55, 58\}$ and $t := 50$ which are hence the 2.5

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

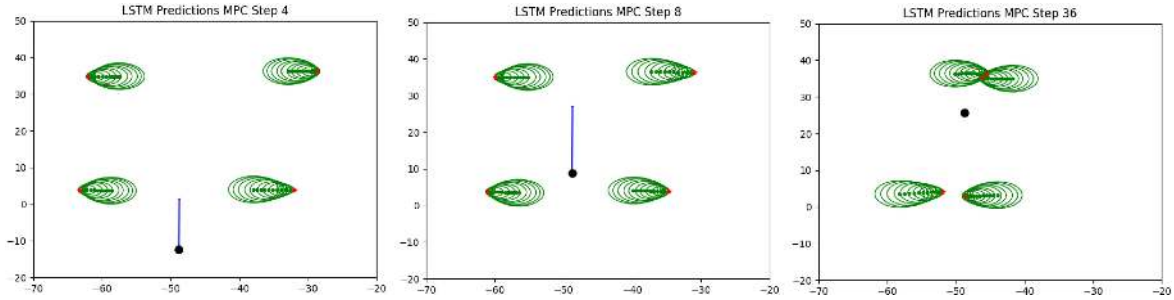


Fig. 5. Scenario 2. The car (black dot) navigating an intersection while avoiding four pedestrians (current position marked in red, predictions and prediction regions indicated by green dots and circles, respectively). Plots at times 4, 8, and 36.

and 4 seconds ahead prediction regions (with R defined as in Remark 2) in Fig. 2 (right). We again indicate $C_{\tau|t}$ for different values of T, and observe that the size of $C_{\tau|t}$ increases with larger T. As we run the car for 30 seconds, the mission time T is technically 60. However, as we have a limited amount of calibration trajectories (data collection in CARLA is real-time), we only use $T := 30$ to calculate prediction regions $C_{\tau|t}$ from D_{cal}, i.e., we use $\bar{\delta} := \delta/30$ with $\delta = 0.05$ in Theorem 1. This choice is practically motivated and larger calibration datasets D_{cal} will allow to select larger T. Nonetheless, the obtained prediction regions are valid for T steps.

We use the MPC again with the constraint function c and the bicycle model discretized with $\Delta = 1/2$, but we now set the prediction horizon to $H := 10$, i.e., to five seconds. Due to the large time horizon of T, we note that the open-loop controller with $H := T$ is infeasible. We made two practically motivated modifications to the optimization problem in (5). First, for the κ step-ahead predictions we observed that the values of $C_{t+\kappa|t}$ are similar at different times t, and we have hence used the smallest value of $C_{t+\kappa|t}$ among all t for the κ step-ahead prediction region. Second, due to a model mismatch between the bicycle model and the CARLA model, we also included slack variables on the constraint in (5c) to obtain recursive feasibility.

The MPC result for a single trajectory from the test set D_{test} is shown in Fig. 5. We also ran the MPC for all 100 test trajectories, and only in one case the safety constraint $c(x_\tau, Y_\tau) \geq 0$ was violated. This confirms Theorem 3 which states that in at most 5 of the 100 MPC runs the constraint is violated. We next checked the correctness of the one-step ahead prediction regions and found that in 99.985 % it holds that $\|Y_\tau - \hat{Y}_{\tau|t}\| \leq C_{\tau|t}$ are satisfied which is higher than the theoretically ensured value of $100(1 - \delta/T) = 99.83$ %.

VI. Conclusion

We presented a model predictive controller (MPC) that uses conformal prediction for probabilistic safe planning in dynamic environments. The MPC uses complex trajectory predictors, such as (but not limited to) long short term memory networks, to predict future states of the dynamic environment and incorporates valid prediction regions using conformal prediction to quantify the uncertainty. To the best of our knowledge, these are the first results that provide valid safety guarantees for planning in dynamic environments without making assumptions on the predictor or the environment. To corroborate our results, we presented two numerical experiments of a mobile robot and a self-driving vehicle safely navigating around other agents.

For future work, we plan to do a comparative study using different trajectory predictors and to analyze their interplay with the MPC in more detail. We will also extend the presented MPC to be able to handle an arbitrary number of agents in the environment. We further plan to investigate potential conservatism in the prediction regions (induced by union bounding over events over several time steps in Theorem 1) by using adaptive methods.

References

[1] C. Mavrogiannis et al., "Core challenges of social robot navigation: A survey," *ACM Trans. Hum.-Robot Interact.*, vol. 12, no. 3, pp. 1–39, 2023.
[2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
[3] S. Mitsch, K. Ghorbal, and A. Platzer, "On provably safe obstacle avoidance for autonomous robotic ground vehicles," in *Proc. Robot.: Sci. Syst. IX*, 2013, pp. 81–102.
[4] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
[5] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.
[6] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Automat.*, vol. 19, no. 1, pp. 53–64, Feb. 2003.
[7] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE Conf. Intell. Robots Syst.*, 2010, pp. 797–803.
[8] N. E. D. Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 101–115, Feb. 2012.
[9] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.
[10] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.
[11] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Berlin, Germany: Springer, 2005.
[12] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, no. 3, pp. 371–421, 2008.
[13] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6262–6271.
[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.

[15] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *Proc. Conf. Comput. Vis.*, 2019, pp. 2821–2830.

[16] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. Conf. Comp. Vis. Pattern Recog.*, 2016, pp. 961–971.

[17] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7386–7400, Jul. 2022.

[18] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Europ. Conf. Comp. Vis.*, 2020, pp. 683–700.

[19] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*.

[20] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, 2020.

[21] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[22] M. Phillips and M. Likhachev, "SIPP: Safe interval path planning for dynamic environments," in *Proc. Conf. Robot. Automat.*, 2011, pp. 5628–5635.

[23] V. Renganathan, S. Safaoui, A. Kothari, B. Gravell, I. Shames, and T. Summers, "Risk bounded nonlinear robot motion planning with integrated perception & control," *Artif. Intell.*, vol. 314, 2023, Art. no. 103812.

[24] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auton. Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[25] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, 2021, pp. 5794–5800.

[26] S. Kalluraya, G. J. Pappas, and Y. Kantaros, "Multi-robot mission planning in dynamic semantic environments," 2022, *arXiv:2209.06323*.

[27] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Control Syst. Lett.*, vol. 7, pp. 289–294, 2022.

[28] A. Wang, C. Mavrogiannis, and A. Steinfeld, "Group-based motion prediction for navigation in crowded environments," in *Proc. Conf. Robot Learn.*, 2022, pp. 871–882.

[29] A. Thomas, F. Mastrogiovanni, and M. Baglietto, "Probabilistic collision constraint for motion planning in dynamic environments," in *Proc. Conf. Intel. Auton. Syst.*, 2021, pp. 141–154.

[30] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: The case for cooperation," in *Proc. Conf. Robot. Autom.*, 2013, pp. 2153–2160.

[31] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Proc. Robot.: Sci. Syst.*, 2012, pp. 193–200.

[32] C. Mavrogiannis, J. A. DeCastro, and S.S. Srinivasa, "Implicit multiagent coordination at unsignalized intersections via multimodal inference enabled by topological braids," 2020, *arXiv:2004.05205*.

[33] H. Hu and J. F. Fisac, "Active uncertainty reduction for human-robot interaction: An implicit dual control approach," in *Proc. Workshop Algorithmic Found. Robot.*, 2022, pp. 385–401.

[34] K. Muvvala, P. Amorese, and M. Lahijanian, "Let's collaborate: Regret-based reactive synthesis for robotic manipulation," in *Proc. IEEE Conf. Robot. Automat.*, 2022, pp. 4340–4346.

[35] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora, "Learning interaction-aware trajectory predictions for decentralized multi-robot motion planning in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2256–2263, Apr. 2021.

[36] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2020, pp. 11785–11792.

[37] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020.

[38] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2017, pp. 1343–1350.

[39] J. F. Fisac et al., "Probabilistically safe robot planning with confidence-based human predictions," in *Proc. 14th Robot.: Sci. Syst.*, 2018.

[40] D. Fridovich-Keil et al., "Confidence-aware motion prediction for real-time collision avoidance[1]," *Int. J. Robot. Res.*, vol. 39, no. 2/3, pp. 250–265, 2020.

[41] S. Bansal, A. Bajcsy, E. Ratner, A. D. Dragan, and C. J. Tomlin, "A Hamilton-Jacobi reachability-based framework for predicting and analyzing human motion for safe planning," in *Proc. IEEE Conf. Robot. Automat.*, 2020, pp. 7149–7155.

[42] S. Choi, E. Kim, K. Lee, and S. Oh, "Real-time nonparametric reactive navigation of mobile robots in dynamic environments," *Robot. Auton. Syst.*, vol. 91, pp. 11–24, 2017.

[43] M. Omainska, J. Yamauchi, T. Beckers, T. Hatanaka, S. Hirche, and M. Fujita, "Gaussian process-based visual pursuit control with unknown target motion learning in three dimensions," *SICE J. Control, Meas., Syst. Integration*, vol. 14, no. 1, pp. 116–127, 2021.

[44] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes," in *Proc. IEEE Conf. Intell. Robots Syst.*, 2008, pp. 1056–1062.

[45] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.

[46] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 545–552.

[47] T. Du et al., "Cert-RNN: Towards certifying the robustness of recurrent neural networks," in *Proc. Conf. Comput. Commun. Secur.*, 2021, pp. 516–534.

[48] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *Proc. Conf. Mach. Learn.*, 2016, pp. 1120–1128.

[49] A. Farid, S. Veer, B. Ivanovic, K. Leung, and M. Pavone, "Task-relevant failure detection for trajectory predictors in autonomous vehicles," in *Proc. Conf. Robot Learn.*, 2023, pp. 1959–1969.

[50] R. Luo et al., "Sample-efficient safety assurances using conformal prediction," in *Proc. Workshop Algorithmic Found. Robot.*, 2022, pp. 149–169.

[51] T. G. Dietterich and J. Hostetler, "Conformal prediction intervals for Markov decision process trajectories," 2022, *arXiv:2206.04860*.

[52] L. Bortolussi, F. Cairoli, N. Paoletti, S. A. Smolka, and S. D. Stoller, "Neural predictive monitoring," in *Proc. Conf. Runtime Verification*, 2019, pp. 129–147.

[53] X. Qin, Y. Xian, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification of cyber-physical systems using surrogate models and conformal inference," in *Proc. IEEE 13th Int. Conf. Cyber- Phys. Syst.*, 2022, pp. 116–126.

[54] Y. Chen, U. Rosolia, C. Fan, A. Ames, and R. Murray, "Reactive motion planning with probabilisticsafety guarantees," in *Proc. Conf. Robot Learn.*, 2021, pp. 1958–1970.

[55] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[56] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," 2021, *arXiv:2107.07511*.

[57] M. Fontana, G. Zeni, and S. Vantini, "Conformal prediction: A unified review of theory and new challenges," *Bernoulli*, vol. 29, no. 1, pp. 1–23, 2023.

[58] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, 2018.

[59] R. J. Tibshirani, R. F. Barber, E. Candes, and A. Ramdas, "Conformal prediction under covariate shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2526–2536.

[60] M. Cauchois, S. Gupta, A. Ali, and J. C. Duchi, "Robust validation: Confident predictions even when distributions shift," 2020, *arXiv:2008.04267*.

[61] K. Stankeviciute, A. M. Alaa, and M. van der Schaar, "Conformal time-series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 6216–6228.

[62] J. V. d. Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Conf. Robot. Automat.*, 2008, pp. 1928–1935.

[63] R. Pepy, A. Lambert, and H. Mounier, "Path planning using a dynamic vehicle model," in *Proc. IEEE Conf. Inf. Commun. Technol.*, 2006, pp. 781–786.

[64] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on March 12,2024 at 13:59:26 UTC from IEEE Xplore. Restrictions apply.