2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Distributionally Robust Optimization with Unscented Transform for Learning-Based Motion Control in Dynamic Environments

Astghik Hakobyan Insoon Yang

***Abstract*— Safety is one of the main challenges when applying learning-based motion controllers to practical robotic systems, especially when the dynamics of the robots and their surrounding dynamic environments are unknown. This issue is further exacerbated when the learned information is unreliable and inaccurate. In this paper, we aim to enhance the safety of learning-enabled mobile robots in dynamic environments from the perspective of distributionally robust optimization (DRO) and the unscented transform (UT). Our method infers the unknown dynamics of both the robot and the environment by adopting Gaussian process regression with an uncertainty propagation scheme based on UT to improve prediction accuracy. This leads to a novel learning-based model predictive control (MPC) method in which state information about both the robot and the environment is propagated via UT. The proposed method uses DRO to proactively limit the risk of collisions or other unsafe events in the presence of learning errors. However, the distributionally robust risk constraint is intractable because it involves a separate infinite-dimensional optimization problem. To overcome this challenge, we exploit UT with modern DRO techniques to replace the risk constraint with its simple upper bound. The performance and the utility of our method are demonstrated through simulations in autonomous driving scenarios, showing its capability to enhance safety and computational efficiency.**

I. Introduction

Autonomous mobile robots have shown promise in many real-world applications ranging from indoor services to urban navigation. In general, information about the exact robot model and the environment dynamics is unavailable or highly limited. Learning-based control approaches are commonly used in such settings to infer unknown models and improve the overall control performance. However, the safety of learning-based controllers (e.g., collision-free navigation) remains a significant concern for the application of such methods, especially when the learned models are unreliable and inaccurate [1].

Existing learning-based control methods employ various machine learning techniques to infer the unknown dynamics of the robot and the environment. The learning models most commonly used for this purpose include deep neural networks [2]–[6], Bayesian linear regression [7]–[9], and Gaussian processes (GPs) [10]–[14], among others. One of the most popular approaches for learning-based motion control of robotic systems is model predictive control (MPC), where the unknown models are substituted with the learned ones. Most research efforts in this field have focused on improving the prediction model by learning the system dynamics or fine-tuning its parameters [10]–[13]. In contrast, a few works learn the dynamic environment model and apply the controller to a system with known dynamics [5], [8], [15]–[19]. Safety in such methods is often addressed via probabilistic constraints, such as chance constraints [20]–[24] or conditional value-at-risk (CVaR) constraints [25]–[29]. However, most existing methods do not address the learning inaccuracies or unreliability of the models, applying them directly to the controller. Such distributional uncertainties are handled in distributionally robust optimization (DRO) methods, where a given stochastic program is solved in the face of the worst-case distribution drawn from some ambiguity set [30]–[34]. Recently, the application of DRO has been extended to learning-based control problems to account for learning errors during the control stage [34]–[37]. However, these methods require nontrivial computational demand when solving DRO problems.

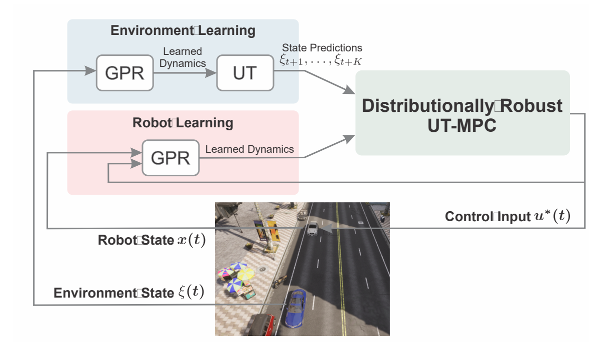

Fig. 1: The overview of our method.

Our paper is related to learning-based distributionally robust control in that we learn the unknown dynamics of both the robot and the environment, as well as address the learning errors in the motion control stage by adopting tools from DRO. As shown in Fig. 1, our framework consists of (i) separate learning modules for inferring the unknown models of both the robot and the dynamic environment via Gaussian process regression (GPR) [38] and the unscented transform (UT), and (ii) an MPC-based control module that uses the learned models with an accurate uncertainty propagation scheme and is robust against possible learning errors. Unlike typical uncertainty propagation schemes used in GP-based MPC methods [10]–[12], we propose exploiting UT to improve computational efficiency and prediction ac-

This work was supported in part by the National Research Foundation of Korea under MSIT2020R1C1C1009766, the Information and Communications Technology Planning and Evaluation under Grants MSIT2020-0-00857, MSIT2022-0-00124, MSIT2022-0-00480, and Samsung Electronics.

A. Hakobyan, and I. Yang are with the Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 08826, South Korea, {astghikhakobyan, insoonyang}@snu.ac.kr

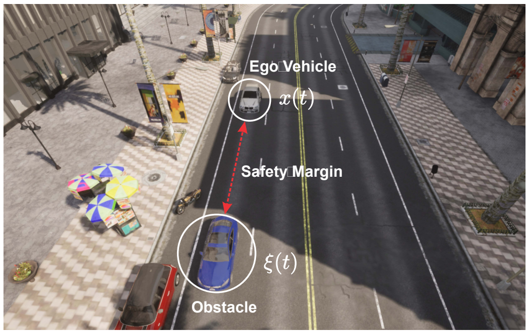

Fig. 2: An autonomous driving scenario.

curacy. Prior works utilize a similar uncertainty propagation approach in stochastic MPC settings with known system dynamics [39]–[41]. In contrast, we apply the UT method to the learned models to predict the states of both the robot and the environment. In addition, to immunize the system against learning errors, we adopt tools from Wasserstein DRO and design a risk constraint to limit the distributionally robust CVaR (DR-CVaR) of the safety loss. This leads to a novel distributionally robust UT-based MPC algorithm (UT-MPC), which combines the advantages of both UT and DRO within a single framework. Unfortunately, the DR-CVaR constraint is intractable as it involves an infinite-dimensional optimization problem over the space of probability distributions. To overcome this challenge, we devise a simple analytical upper bound of DR-CVaR that exploits UT to estimate the safety loss distribution. As a result, we obtain a tractable distributionally robust UT-MPC algorithm that guides the robot to take cautious actions despite learning inaccuracies. Finally, the performance and the utility of our method are demonstrated through simulations in an autonomous driving scenario. Our experiments show the capability of our algorithm to promote safe motion control in a dynamic environment, even in the presence of learning errors.

II. Preliminaries

A. The Setup

Consider a mobile robot modeled by the following discrete-time dynamics:

$$x(t+1) = f(x(t), u(t)) + g(x(t), u(t)), \tag{1}$$

where $x(t) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $u(t) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ are the robot state and control input at time t, respectively. The dynamic model consists of a known part $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ that can be derived from the physics of the system and an unknown mismatch term $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, often occurring due to oversimplifying complex dynamics, unexpected interactions with the environment, etc.

Example 1: Consider the autonomous driving scenario in Fig. 2. The evolution of the ego vehicle can be described by the kinematic bicycle model, which disregards essential features, such as the slip angles, tire type, as well as driving ground. Therefore, it is reasonable to use the kinematic model with an additional mismatch term to compensate for the limited fidelity of the simple model.

The robot operates in a dynamic environment, whose state $\xi(t) \in \mathbb{R}^{n_\xi}$ evolves according to

$$\xi(t+1) = f_{\text{env}}(\xi(t)).$$

Such a model is reasonable as the environment evolves independently of the robot. For instance, the safety of the ego vehicle in Example 1 depends on the behavior of the blue car with state $\xi(t)$ in Fig. 2.

To promote the safe operation of our robot, we introduce a *safety loss function* $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\xi} \to \mathbb{R}$ and impose the following constraint:

$$\ell(x(t), \xi(t)) \leq 0. \tag{2}$$

In Fig. 2, the loss can be chosen to avoid collisions, e.g., $\ell(x, \xi) = r_{\text{safe}}^2 - \|C(x-\xi)\|_2^2$, where r_{safe} is a safety radius, and C maps the states to the position vector.

In this work, *assuming that the dynamics of the robot and the environment are unknown*, we aim to design a learning-based motion controller that guides the robot to perform a specified task in a cautious manner despite learning errors.

B. Uncertainty Propagation via UT

When the dynamics (1) is learned as a stochastic approximator, the uncertainty in the states is propagated over time. Unfortunately, it is challenging to compute the resulting state distribution for non-Gaussian uncertainties passing through nonlinear dynamics. Linearization techniques from extended Kalman filter (EKF) [42] suffer from large estimation errors and require the computationally expensive Jacobian matrix. An alternative approach is UT, which can be applied to an arbitrary nonlinear function. The intuition behind UT is that with fixed parameters, it is easier to approximate the given distribution than it is to approximate a nonlinear transformation [43]. Therefore, UT aims to find a parameterization that completely encodes the statistics of the inputs, allowing its accurate propagation through a nonlinear function.

Consider a random variable $x \in \mathbb{R}^n$ with a mean vector μ^x and a covariance matrix Σ^x that undergoes a nonlinear transformation $d : \mathbb{R}^n \to \mathbb{R}^m$. The goal of UT is to accurately calculate the statistics of the output $y = d(x)$. For that, first, a set of vectors called sigma points are generated in a way to capture the moments of the input distribution. It has been shown that choosing $2n+1$ points is sufficient for encoding the mean and covariance of the inputs [44]. The sigma points are selected according to the following rule:

$$\begin{aligned} \mathcal{X}^{(0)} &= \mu^x, \\ \mathcal{X}^{(i)} &= \mu^x + \left(\sqrt{(n+\lambda)\Sigma^x}\right)_i, \ i = 1, \ldots, n, \\ \mathcal{X}^{(n+i)} &= \mu^x - \left(\sqrt{(n+\lambda)\Sigma^x}\right)_i, \ i = 1, \ldots, n, \end{aligned} \tag{3}$$

where λ is a scaling parameter and $(\sqrt{\cdot})_i$ is the ith column of the matrix square root. Next, the sigma points are propagated through the nonlinear function to obtain the transformed points $\mathcal{Y}^{(i)} = d(\mathcal{X}^{(i)})$. The mean and covariance of the output y can then be computed as

$$\mu^y = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Y}^{(i)},\ \Sigma^y = \sum_{i=0}^{2n} W_c^{(i)} \big(\mathcal{Y}^{(i)} - \mu^y\big)\big(\mathcal{Y}^{(i)} - \mu^y\big)^\top,$$

where $W_m^{(i)}$ and $W_c^{(i)}$ are the weights chosen according to [44].

One of the main advantages of UT is the accuracy of uncertainty propagation. For any nonlinearity, UT captures the output mean and covariance accurately to the third order of the Taylor series expansion for Gaussian inputs and to at least the second order for non-Gaussian inputs [45]. In contrast, the EKF-based method provides only first-order accuracy. Another feature of UT is the implementation simplicity, as it involves only algebraic operations without the need to evaluate the Jacobian matrix needed in EKF.

III. Unscented Transform and Distributionally Robust Optimization for Learning-Based Control

The overall structure of our learning-based control scheme is illustrated in Fig. 1. It consists of two main parts: (i) separate modules for learning the robot and environment dynamics, and (ii) a distributionally robust UT-MPC module for controlling the robot and addressing learning errors. First, the unknown dynamics are inferred via GPR using real-time observations and then used as prediction models in UT-MPC. However, due to the stochastic nature of the learned dynamics, state propagation through the GP models is not straightforward. Our algorithm mitigates this issue by exploiting UT for uncertainty propagation, achieving superior prediction accuracy and computational efficiency. Moreover, using DRO in UT-MPC immunizes the system against learning inaccuracies and promotes the robot's safety despite erroneous models.

A. Learning the Robot and Environment Dynamics

In this study, we use GPR, a non-parametric Bayesian regression method, to infer the dynamics of both the robot and the environment. A major challenge in GPR is the uncertainty propagation through the learned model, which is generally intractable. The most typical approach is linearizing the GP model around the current state mean [10]–[12]. However, as mentioned in Section II-B, such an approach is not only computationally demanding but also degrades the prediction accuracy. Motivated by the state update equations in GP-UKF [46], we propose an uncertainty propagation scheme for GP dynamics based on the concept of UT. This approach not only improves the prediction accuracy but also involves only simple algebraic operations, relieving the computational burden. Therefore, we apply the proposed scheme to learn the dynamics of both the robot and the environment.

At stage t, GPR for the robot is performed using the training input data $\mathbf{X}_t^{\text{rob}} = \{(x(t-1), u(t-1)), \ldots, (x(t-M), u(t-M))\}$ with the corresponding training output data $\mathbf{y}_t^{\text{rob}} = \{\Delta x(t), \ldots, \Delta x(t-M+1)\}$, where $\Delta x(t) = x(t+1) - f(x(t), u(t))$ is the residual between the observed system state and the nominal model. Following the ordinary GPR procedure, the unknown dynamics of the robot is approximated by $\mathcal{GP}(\mu^g, \Sigma^g)$. The dynamics of the robot is then inferred as

$$x(t+1) = f(x(t), u(t)) + \mu^g\left(x(t), u(t)\right) + w_t^{\text{rob}}, \quad (4)$$

where w_t^{rob} is a zero-mean noise with covariance $\Sigma^g(x(t), u(t))$.

For state prediction, we recursively apply the UT presented in Section II-B and propagate the states along the horizon. In particular, the distribution of the state vector can be predicted starting from the current observation $\mu_0^x = x(t)$ with $\Sigma_0^x = \mathbf{0}$ according to the following rule:

$$\mathcal{X}_k = \Big[\mu_k^x,\ \mu_k^x \pm \sqrt{(n_x + \lambda_x)\Sigma_k^x}\Big] \quad (5)$$

$$\mathcal{Y}_k^{(i)} = f(\mathcal{X}_k^{(i)}, u_k) + \mu^g(\mathcal{X}_k^{(i)}, u_k),\ i = 0, \ldots, 2n_x \quad (6)$$

$$\mu_{k+1}^x = \sum_{i=0}^{2n_x} W_{m_{\text{rob}}}^{(i)} \mathcal{Y}_k^{(i)} \quad (7)$$

$$\Sigma_{k+1}^x = \sum_{i=0}^{2n_x} W_{c_{\text{rob}}}^{(i)} \big(\mathcal{Y}_k^{(i)} - \mu_{k+1}^x\big)\big(\mathcal{Y}_k^{(i)} - \mu_{k+1}^x\big)^\top + \Sigma^g(\mu_k^x, u_k). \quad (8)$$

Similarly, using datasets $\mathbf{X}_t^{\text{env}} = \{\xi(t-1), \ldots, \xi(t-M)\}$ and $\mathbf{y}_t^{\text{env}} = \{\xi(t), \ldots, \xi(t-M+1)\}$, the dynamics of the environment is approximated by $\mathcal{GP}(\mu^{\text{env}}, \Sigma^{\text{env}})$. Then, the environment states evolve according to

$$\xi(t+1) = \mu^{\text{env}}(\xi(t)) + w_t^{\text{env}},$$

where w_t^{env} is a zero-mean noise with covariance $\Sigma^{\text{env}}(\xi(t))$. By applying a UT scheme similar to (5)–(8) with weights $W_{m_{\text{env}}}$ and $W_{c_{\text{env}}}$, the environment states can be predicted over the horizon to obtain μ_k^ξ and Σ_k^ξ starting from $\mu_0^\xi = \xi(t)$ and $\Sigma_0^\xi = \mathbf{0}$. Unlike the robot, the environment states are independent of the control inputs. Therefore, as illustrated in Fig. 1, the environment state prediction can be performed outside the control loop, saving computational resources.

For our further analysis, it is convenient to denote the joint state of the robot and the environment by $\mathbf{z}_k = [x_k^\top, \xi_k^\top]^\top$. Then, assuming the independence of the states, the estimated joint distribution $\mathbb{P}_k$ of z_k at any time step $t+k$ can be represented by its mean vector $\mu_k^{\mathbf{z}} = [(\mu_k^x)^\top, (\mu_k^\xi)^\top]^\top$ and covariance matrix $\Sigma_k^{\mathbf{z}} = \text{diag}(\Sigma_k^x, \Sigma_k^\xi)$.

B. Distributionally Robust UT-MPC

Since the state information is no longer deterministic due to the use of GPR, the MPC problem attains a stochastic formulation, where the deterministic constraint (2) is not valid anymore. Instead, a risk measure can be used to assess the risk of unsafe events using the learned joint state distribution. Among several risk measures, we use the CVaR, which is a coherent measure in the sense of Artzner et al. [47] and has been advocated as a rational risk measure in robotics [48]. CVaR of a random loss $X \sim \mathbb{P}$ is defined as

$$\text{CVaR}_\epsilon^{\mathbb{P}}[X] := \min_{z \in \mathbb{R}} \mathbb{E}^{\mathbb{P}}\left[z + \frac{(X-z)^+}{1-\epsilon}\right],$$

where $\epsilon \in (0, 1]$ is some confidence level. It quantifies the average loss beyond ϵ, accounting for rare but crucial events.

However, the quality of risk assessment highly depends on the accuracy of the learned safety loss distribution. Unfortunately, in our case, the learned information might be unreliable for measuring the robot's safety due to inaccuracies in GP models. To immunize the system against such distributional uncertainties, we propose evaluating the following distributionally robust version of CVaR:

$$\text{DR-CVaR}_{\epsilon}^{\mathbb{P}_k^{\ell}}\left[\ell(\mathbf{z}_k)\right] := \sup_{\mathbb{Q}_k \in \mathbb{D}_{\theta}(\mathbb{P}_k^{\ell})} \text{CVaR}_{\epsilon}^{\mathbb{Q}_k}\left[\ell(\mathbf{z}_k)\right], \quad (9)$$

which evaluates the worst-case CVaR over an ambiguity set $\mathbb{D}_{\theta}(\mathbb{P}_k^{\ell})$ constructed using the learned safety loss distribution $\mathbb{P}_k^{\ell}$. In this work, we define the ambiguity set as a Wasserstein ball of radius $\theta > 0$ centered at $\mathbb{P}_k^{\ell}$:

$$\mathbb{D}_{\theta}(\mathbb{P}_k^{\ell}) = \left\{\mathbb{Q}_k \in \mathcal{P}_2(\mathbb{R}) \mid W_2(\mathbb{P}_k^{\ell}, \mathbb{Q}_k) \leq \theta\right\}, \quad (10)$$

where $\mathcal{P}_2(\mathcal{W})$ is the space of Borel probability measures on $\mathcal{W}$ with a finite second moment. Here, $W_2(\mathbb{P}, \mathbb{Q})$ is the 2-Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$, which is defined as $W_2(\mathbb{P}, \mathbb{Q}) := \inf_{\kappa \in \mathcal{P}(\mathcal{W}^2)} \left\{ \left(\int_{\mathcal{W}^2} \|x - y\|^2 \, \mathrm{d}\kappa(x, y) \right)^{1/2} \middle| \, \Pi^1\kappa = \mathbb{P}, \Pi^2\kappa = \mathbb{Q} \right\}$, where κ is the *transport plan*, with $\Pi^i\kappa$ denoting its ith marginal, and $\|\cdot\|$ is the Euclidean norm quantifying the transportation cost. Wasserstein distance represents the minimum cost of transporting mass from one distribution to another using nonuniform perturbations. It has received great interest in DRO for its superior features, such as providing a finite-sample performance guarantee and addressing the closeness between two points in the support [31], [49], [50].

Combining the UT-based GP dynamics (5)–(8) and the DR-CVaR risk (9), we formulate the following distributionally robust UT-MPC problem:

$$\min_{\mathbf{u}} \sum_{k=0}^{K-1} \mathbb{E}^{\mathbb{P}_k}\left[c(x_k, u_k)\right] + \mathbb{E}^{\mathbb{P}_K}\left[q(x_K)\right] \quad (11a)$$

$$\text{s.t. } (5) - (8) \quad (11b)$$

$$\text{DR-CVaR}_{\epsilon}^{\mathbb{P}_k^{\ell}}\left[\ell(\mathbf{z}_k)\right] \leq 0 \quad (11c)$$

$$\mu_k^x \in \mathbb{X} \quad (11d)$$

$$u_k \in \mathbb{U} \quad (11e)$$

$$\mu_0^x = x(t), \Sigma_0^x = \mathbf{0}, \quad (11f)$$

where $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the stage-wise cost function, and $q : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost function. Here, the constraints (11b) and (11e) hold for $k = 0, \ldots, K-1$, while the constraints (11c) and (11d) hold for $k = 0, \ldots, K$.

The constraint on UT-based GP dynamics (11b) plays an important role in our distributionally robust UT-MPC problem. First, the UT-based state propagation scheme provides better state prediction accuracy than the linearization technique often met in prior GP-based MPC approaches [10]–[12]. Second, the nonconvexities in the equality constraints involve simple algebraic operations and, thus, are relatively easy to handle than the derivatives in linearization. Another key component of our MPC problem is the DR-CVaR constraint (11c). It limits the safety risk under the worst-case distribution within the ambiguity set of the learned loss distribution for all time stages. Notably, adjusting the radius θ changes the conservativeness of the constraint, as it determines the range of distributions in the neighborhood of $\mathbb{P}_k^{\ell}$ to be included in the ambiguity set. In summary, the combination of UT-based GP dynamics and DR-CVaR risk constraint reinforces our distributionally robust UT-MPC problem with superior prediction accuracy and computational efficiency, as well as the capability of limiting the safety risk despite learning inaccuracies.

IV. Tractable Reformulation and Algorithm

Despite the advantages of the distributionally robust UT-MPC problem (11), it is intractable due to the objective function (11a) and the safety constraint (11c). The objective function can be handled relatively easily by approximating it around the predicted state mean. Our primary concern is the DR-CVaR in constraint (11c), which is challenging to evaluate as it involves an infinite-dimensional optimization problem over the ambiguity set of probability distributions. In our method, we overcome the intractability of the MPC problem using a novel UT-based approximation scheme. Specifically, we take advantage of UT to estimate the statistics of the safety loss distribution. Then, we use the approximate distribution and modern tools from DRO to derive an upper bound of DR-CVaR. As a consequence, we arrive at a tractable distributionally robust UT-MPC algorithm.

A. UT-Based Upper Bound of DR-CVaR

The Wasserstein ambiguity set in (10) is built around the learned loss distribution $\mathbb{P}_k^{\ell}$. However, in each prediction step k, we are given only the mean and covariance of the joint state vector $\mathbf{z}_k$. Therefore, our first goal is to determine $\mathbb{P}_k^{\ell}$ by propagating $\mathbf{z}_k$ through the loss function. Fortunately, we can apply the UT-based uncertainty propagation scheme in Section II-B to directly estimate the statistics of the loss distribution. For that, we first generate sigma points $\mathcal{Z}_k$ for $\mu_k^{\mathbf{z}}$ and $\Sigma_k^{\mathbf{z}}$ according to (3), pass them through the loss function, and obtain transformed points $\mathcal{L}_k^{(i)} = \ell\big(\mathcal{Z}_k^{(i)}\big)$, $i = 0, \ldots, 2(n_x + n_{\xi})$. Then, the mean and the variance of the loss can be obtained as

$$\mu_k^{\ell} = \sum_{i=0}^{2(n_x+n_{\xi})} W_{m_{\text{loss}}}^{(i)} \mathcal{L}_k^{(i)} \quad (12)$$

$$\left(\sigma_k^{\ell}\right)^2 = \sum_{i=0}^{2(n_x+n_{\xi})} W_{c_{\text{loss}}}^{(i)} \left(\mathcal{L}_k^{(i)} - \mu_k^{\ell}\right)\left(\mathcal{L}_k^{(i)} - \mu_k^{\ell}\right)^{\top}. \quad (13)$$

Though there is still no full knowledge about the distribution $\mathbb{P}_k^{\ell}$, UT provides us with knowledge about its mean and variance. In the following proposition, we show how this statistical information can be used to obtain a tractable and simple upper bound on DR-CVaR.

Proposition 1: Let $\mathbb{P}_k^{\ell}$ be the distribution of the loss $\ell(\mathbf{z}_k)$ with mean and variance defined in (12) and (13), respectively.

Algorithm 1: Distributionally Robust UT-MPC

1 **Input:** UT parameters $W_{m_i}, W_{c_i}, i = \{\text{rob}, \text{env}, \text{loss}\}$ and risk parameters ϵ, θ
2 Collect M observations to $\mathbf{X}_0^{\text{rob}}, \mathbf{y}_0^{\text{rob}}, \mathbf{X}_0^{\text{env}}, \mathbf{y}_0^{\text{env}}$
3 Observe $x(0)$ and $\xi(0)$
4 **for** $t = 0, 1, \dots$ **do**
5 Train GPs for μ^g, Σ^g and $\mu^{\text{env}}, \Sigma^{\text{env}}$
6 Predict μ_k^ξ and Σ_k^ξ for $k = 0, \dots, K-1$ starting from $\mu_0^\xi = \xi(t), \Sigma_0^\xi = \mathbf{0}$
7 Solve problem (11) with (14)
8 Apply $u(t) = u_0^*$ and observe $x(t+1), \xi(t+1)$
9 Update $\mathbf{X}_{t+1}^{\text{rob}}, \mathbf{y}_{t+1}^{\text{rob}}$ and $\mathbf{X}_{t+1}^{\text{env}}, \mathbf{y}_{t+1}^{\text{env}}$

Then, the DR-CVaR (9) with a radius $\theta > 0$ has the following upper bound:

$$\text{DR-CVaR}_\epsilon^{\mathbb{P}_k^\ell}[\ell(\mathbf{z}_k)] \le \mu_k^\ell + \gamma\sigma_k^\ell + \theta\sqrt{1+\gamma^2}, \tag{14}$$

where $\gamma = \sqrt{\epsilon/(1-\epsilon)}$.

The proof of this proposition can be found in Appendix I. The upper bound (14) is attained for elliptical distributions and is tight for all other distributions. Despite relying solely on the mean and variance, this bound exhibits exceptional computational properties, as it requires simple algebraic operations. Moreover, unlike the existing methods (e.g., [35], [36]), we directly estimate the loss distribution $\mathbb{P}_k^\ell$, enabling the use of our approach for any safety loss function.

B. Tractable Algorithm

The UT-based upper bound of DR-CVaR can be directly incorporated into the distributionally robust UT-MPC problem (11) to alleviate the intractability without significantly affecting the computational complexity. For that, we replace the risk constraint (11c) with a constraint on the upper bound (14) and introduce additional equality constraints (12) and (13) for estimating the loss distribution. As a result, the reformulated MPC problem constitutes a tractable nonlinear optimization problem. Despite its nonconvexity due to the GP dynamics and the UT approximations, it can be efficiently solved using existing algorithms, such as interior-point and sequential quadratic programming methods [51].

The overall distributionally robust UT-MPC scheme is presented in Algorithm 1, given the UT weights $W_{m_i}, W_{c_i}, i = \{\text{rob}, \text{env}, \text{loss}\}$, as well as risk parameters ϵ and θ. First, GPR training datasets $\mathbf{X}_0^{\text{rob}}, \mathbf{y}_0^{\text{rob}}$ and $\mathbf{X}_0^{\text{env}}, \mathbf{y}_0^{\text{env}}$ are initialized by collecting M observations (line 2). Next, the states of the robot and the environment are observed to begin the main loop (line 3). In each time stage, GP models for the robot and environment are learned (line 5). Then, we predict the environment states for K time stages starting from the current state $\xi(t)$ (line 6). Using the learned models, the UT-MPC problem (11) is solved with the DR-CVaR upper bound (14) (line 7). The first element of the optimal control sequence $\mathbf{u}^*$ returned by the UT-MPC is then applied to the robot (line 8). Finally, we observe the new states and update the GPR datasets with the latest M observations (line 9).

V. Experiment Results

In this section, we present the simulation results of our algorithm in an autonomous driving scenario performed in an open-source traffic simulation platform CARLA [52]. The goal is to control the ego vehicle to follow the given waypoints without colliding with the obstacle.[1]

A. Experiment Settings

In our experiments, the nominal model of the ego vehicle is chosen as the following kinematic bicycle model:

$$\begin{aligned}
\mathrm{x}(t+1) &= \mathrm{x}(t) + T_s v(t)\cos(\phi(t) + \beta_s(t)) \\
\mathrm{y}(t+1) &= \mathrm{y}(t) + T_s v(t)\sin(\phi(t) + \beta_s(t)) \\
\phi(t+1) &= \phi(t) + T_s v(t)\tan(\delta_f(t))\cos(\beta_s(t))/L \\
v(t+1) &= v(t) + T_s a(t),
\end{aligned}$$

where $x(t) = [\mathrm{x}(t), \mathrm{y}(t), \phi(t), v(t)]^\top$ is the ego vehicle's state vector, consisting of its position, heading angle and velocity, $u(t) = [a(t), \delta_f(t)]^\top$ is the control input vector, comprising acceleration and steering angle, $\beta_s(t) := \arctan\left(\frac{1}{2}\tan(\delta_f(t))\right)$ is the slipping angle, $T_s = 0.1\,\text{sec.}$ is the sampling time, and $L = 4.611\,\text{m.}$ is the car length. The control inputs are limited to $|a(t)| \le 3\,\text{m/sec.}^2, |\delta_f(t)| \le 1.22\,\text{rad.}$ with an additional limit on the change of front steering angle $|\Delta\delta_f(t)| \le 0.05\,\text{rad}$. The cost function is chosen to track the waypoints p_t and penalize control input changes, i.e., $c(x_k, u_k) = \|x_k - p_{t+k}\|_Q^2 + \|\Delta u_k\|_R^2, q(x_K) = \|x_K - p_{t+K}\|_Q^2$, where $Q = \text{diag}(1, 1, 0, 0.2)$ and $R = \text{diag}(1.5, 3)$. We consider an MPC with a horizon of $K = 30$ and a zero-mean GPR with a radial basis function kernel trained on $M = 50$ real-time observations. The parameters for DR-CVaR are tuned to $\epsilon = 0.95$ and $\theta = 0.1$.

Due to the simulation model, the obstacle's behavior is not deterministic and varies in each execution. Therefore, for reliability, we have performed 20 simulation runs under identical conditions. We compare our method to an MPC without any learning component (Vanilla-MPC), a learning-based MPC with the safety constraint (2) evaluated at the mean of the predicted state (Mean-MPC), and a non-robust version of UT-MPC with a CVaR constraint (CVaR-MPC).

B. Results

Snapshots of representative scenarios for distributionally robust UT-MPC and Mean-MPC are demonstrated in Fig. 3.[2] Initially, the obstacle navigates far from the ego vehicle and plans to continue in the same lane. Therefore, in the early stages, both controllers drive the car along the reference path. However, when the vehicles approach the intersection, the obstacle suddenly steers to the left. This situation causes errors in the GPR prediction, making the learned distribution unreliable. Nevertheless, the Mean-MPC trusts the learned information even in such a situation and decides to perform a cut-in maneuver, eventually leading to a collision between the ego vehicle and the obstacle. On the contrary, UT-MPC

[1]The source code of our implementation is available at https://github.com/CORE-SNU/DR-UT-MPC

[2]More scenarios are available in the supplementary video.

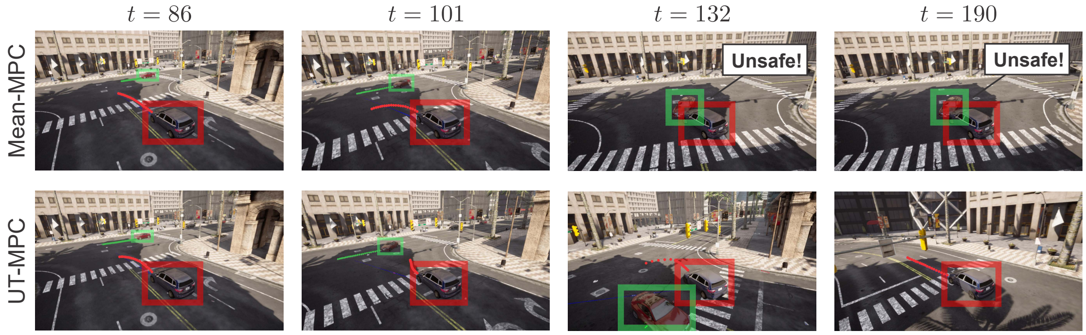

Fig. 3: Snapshots of simulations for Mean-MPC and UT-MPC. The MPC predictions for the ego vehicle are shown in red, while the GP predictions for the obstacle are drawn in green.

TABLE I: The total cost, average computation time per stage, and maximum safety loss value for for all algorithm computed over 20 simulations (mean ± std).

	Total Cost ($\times 10^3$)	Comp. Time (sec)	Safety Loss (m^2)
UT-MPC	28.056 ± 0.204	0.467 ± 0.013	1.864 ± 0.133
Vanilla-MPC	∞	0.031 ± 0.002	12.342 ± 0.742
Mean-MPC	∞	0.435 ± 0.001	9.314 ± 1.054
CVaR-MPC	29.356 ± 0.142	0.418 ± 0.023	2.032 ± 0.101

makes the car stop at the intersection and then slowly bypass the obstacle from the right. Due to its robustness to learning errors, UT-MPC takes cautious actions and overtakes the obstacle without any collisions. Consequently, it outperforms Mean-MPC in terms of navigation quality and safety.

The statistics of our quantitative analysis for 20 simulation runs are reported in Table I. In terms of safety, our algorithm outperforms all the baselines, followed by the CVaR-MPC. Such results are expected, as UT-MPC is the only method that accounts for learning errors. On the other hand, both Vanilla-MPC and Mean-MPC become infeasible after colliding with the obstacles, making the total cost infinitely large. In terms of computation time, Vanilla-MPC surpasses all the baselines due to its simplicity. Meanwhile, all the learning-based algorithms, including our UT-MPC, require similar computation time for solving the problem. As a result, we confirm the capabilities of our algorithm for promoting safety with a comparably short computation time.

VI. Conclusions and Future Work

We have proposed a novel learning-based MPC framework for robotic systems in unknown environments. Our method exploits the learned dynamics and UT-based uncertainty propagation scheme for accurate and efficient prediction of the robot and environment states. Furthermore, it uses a DR-CVaR constraint to proactively limit the risk of unsafety even under errors in the learned models. To tackle the computational intractability of the resulting UT-MPC problem, we have approximated the safety loss distribution using UT and derived a simple upper bound of DR-CVaR. The experiment results demonstrate the computational efficiency of our method and its capability to promote safety.

It remains a future work to extend the current approach to the partially observable settings and conduct experiments on physical robots. In addition, we plan to enhance the adaptivity of the risk constraint by adjusting the level of robustness online.

Appendix I
Proof of Proposition 1

Proof: We use the Gelbrich bound on Wasserstein distance, for which $W_2(\mathbb{P}_k^\ell, \mathbb{Q}_k) \geq \sqrt{(\mu_k - \mu_k^\ell)^2 + (\sigma_k - \sigma_k^\ell)^2}$, where $\mu_k \in \mathbb{R}$ and $\sigma_k^2 \in \mathbb{R}_+$ are the mean and variance of the loss under the distribution $\mathbb{Q}_k$ [53, Proposition 8]. The bound is exact if $\mathbb{P}_k^\ell$ and $\mathbb{Q}_k$ are elliptical distributions with the same density generator. It follows that the DR-CVaR is bounded as

$$\text{DR-CVaR}_\epsilon^{\mathbb{P}_k^\ell}[\ell(\mathbf{z}_k)] \leq \sup_{\mathbb{Q}_k \in \tilde{\mathbb{D}}_\theta(\mathbb{P}_k^\ell)} \text{CVaR}_\epsilon^{\mathbb{Q}_k}[\ell(\mathbf{z}_k)],$$

where $\tilde{\mathbb{D}}_\theta(\mathbb{P}_k^\ell)$ is an ambiguity set with respect to the Gelbrich bound defined as $\tilde{\mathbb{D}}_\theta(\mathbb{P}_k^\ell) := \{\mathbb{Q}_k \in \mathcal{P}_2(\mathbb{R}) \mid (\mu, \sigma^2) \in \mathcal{U}_\theta(\mu_k^\ell, (\sigma_k^\ell)^2), \mathbb{E}^{\mathbb{Q}_k}[\ell(\mathbf{z}_k)] = \mu, \mathbb{E}^{\mathbb{Q}_k}[(\ell(\mathbf{z}_k) - \mu)^2] = \sigma^2\}$, and $\mathcal{U}_\theta(\mu_k^\ell, (\sigma_k^\ell)^2) := \{(\mu, \sigma^2) \in \mathbb{R} \times \mathbb{R}_+ \mid (\mu - \mu_k^\ell)^2 + (\sigma - \sigma_k^\ell)^2 \leq \theta^2\}$ is the mean-covariance uncertainty set around the estimated mean μ_k^ℓ and variance $(\sigma_k^\ell)^2$.

In order to solve the right-hand side of the inequality, also known as the *Gelbrich risk*, we decompose it into

$$\begin{aligned} &\sup_{\mathbb{Q}_k \in \tilde{\mathbb{D}}_\theta(\mathbb{P}_k^\ell)} \text{CVaR}_\epsilon^{\mathbb{Q}_k}[\ell(\mathbf{z}_k)] \\ &= \sup_{(\mu_k, \sigma_k^2) \in \mathcal{U}_\theta(\mu_k^\ell, (\sigma_k^\ell)^2)} \sup_{\mathbb{Q}_k \in \mathcal{C}(\mu_k, \sigma_k^2)} \text{CVaR}_\epsilon^{\mathbb{Q}_k}[\ell(\mathbf{z}_k)], \end{aligned} \tag{15}$$

where $\mathcal{C}(\mu, \sigma^2)$ is the Chebyshev uncertainty set with mean μ and variance σ^2.

To solve the inner supremum, we apply [54, Proposition 2] according to which $\sup_{\mathbb{Q}_k \in \mathcal{C}(\mu_k, \sigma_k^2)} \text{CVaR}_\epsilon^{\mathbb{Q}_k}[\ell(\mathbf{z}_k)] = \mu_k + \gamma\sigma_k$. By substituting the above solution into (15), the problem reduces to the following convex optimization problem, which is a quadratically constrained quadratic program:

$$\max_{\mu_k, \sigma_k \geq 0} \{\mu_k + \gamma\sigma_k \mid (\mu_k - \mu_k^\ell)^2 + (\sigma_k - \sigma_k^\ell)^2 \leq \theta^2\}.$$

Using the standard duality, the solution of the above optimization problem corresponds to the right-hand side of (14). ∎

REFERENCES

[1] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[3] T. Wei and C. Liu, "Safe control with neural network dynamic models," in *Learning for Dynamics and Control Conference*, 2022, pp. 739–750.

[4] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for Lidar 3D vehicle detection," in *IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 3266–3273.

[5] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 8552–8558.

[6] T. Salzmann, E. Kaufmann, M. Pavone, D. Scaramuzza, and M. Ryll, "Neural-MPC: Deep learning model predictive control for quadrotors and agile robotic platforms," *arXiv preprint arXiv:2203.07747*, 2022.

[7] L. Brunke, S. Zhou, and A. P. Schoellig, "Barrier Bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems," in *Learning for Dynamics and Control Conference*, 2022, pp. 881–892.

[8] C. D. McKinnon and A. P. Schoellig, "Learn fast, forget slow: Safe predictive learning control for systems with unknown and changing dynamics performing repetitive tasks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2180–2187, 2019.

[9] K. Seel, M. Haring, E. I. Grøtli, K. Y. Pettersen, and J. T. Gravdahl, "Learning-based robust model predictive control for sector-bounded lur'e systems," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 46–52, 2021.

[10] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[11] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[12] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.

[13] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[14] J. Fan and W. Li, "Safety-guided deep reinforcement learning via online Gaussian process estimation," *arXiv preprint arXiv:1903.02526*, 2019.

[15] A. Hakobyan and I. Yang, "Learning-based distributionally robust motion control with Gaussian processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 7667–7674.

[16] C. Richter, J. Ware, and N. Roy, "High-speed autonomous navigation of unknown environments using learned probabilities of collision," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 6114–6121.

[17] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: a data-driven risk-aware approach," *IEEE Control Systems Letters*, 2022.

[18] D. Li, D. Fooladivanda, and S. Martínez, "Online learning of parameterized uncertain dynamical environments with finite-sample guarantees," in *IEEE American Control Conference*, 2021, pp. 2005–2010.

[19] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 11 205–11 212.

[20] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, "Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 389–396, 2020.

[21] J. Ma, Z. Cheng, X. Zhang, A. A. Mamun, C. W. de Silva, and T. H. Lee, "Non-parametric behavior learning for multi-agent decision making with chance constraints: A data-driven predictive control framework," *arXiv preprint arXiv:2011.03213*, 2020.

[22] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

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

[24] A. Thorpe, T. Lew, M. Oishi, and M. Pavone, "Data-driven chance constrained control using kernel distribution embeddings," in *Learning for Dynamics and Control Conference*, 2022, pp. 790–802.

[25] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using CVaR-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019.

[26] V. D. Sharma, M. Toubeh, L. Zhou, and P. Tokekar, "Risk-aware planning and assignment for ground vehicles using uncertain perception from aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 11 763–11 769.

[27] S. Singh, J. Lacotte, A. Majumdar, and M. Pavone, "Risk-sensitive inverse reinforcement learning via semi- and non-parametric methods," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1713–1740, 2018.

[28] I. G. Jin, B. Schürmann, R. M. Murray, and M. Althoff, "Risk-aware motion planning for automated vehicle among human-driven cars," in *American Control Conference*, 2019, pp. 3987–3993.

[29] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-averse control via CVaR barrier functions: Application to bipedal robot locomotion," *IEEE Control Systems Letters*, vol. 6, pp. 878–883, 2021.

[30] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, 2018.

[31] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with Wasserstein distance," *arXiv preprint arXiv:1604.02199*, 2016.

[32] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[33] A. R. Hota, A. Cherukuri, and J. Lygeros, "Data-driven chance constrained optimization under Wasserstein ambiguity sets," in *2019 American Control Conference (ACC)*, 2019, pp. 1501–1506.

[34] D. Boskos, J. Cortés, and S. Martínez, "Data-driven ambiguity sets with probabilistic guarantees for dynamic processes," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 2991–3006, 2020.

[35] J. Coulson, J. Lygeros, and F. Dorfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2021.

[36] A. Hakobyan and I. Yang, "Distributionally robust risk map for learning-based motion planning and control: A semidefinite programming approach," *IEEE Transactsions on Robotics*, 2022.

[37] A. Zolanvari and A. Cherukuri, "Data-driven distributionally robust iterative risk-constrained model predictive control," in *IEEE European Control Conference*, 2022, pp. 1578–1583.

[38] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[39] C. Liu, A. Gray, C. Lee, J. K. Hedrick, and J. Pan, "Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles," in *IEEE American Control Conference*, 2014, pp. 5574–5579.

[40] E. Bradford and L. Imsland, "Stochastic nonlinear model predictive control with state estimation by incorporation of the unscented Kalman filter," *arXiv preprint arXiv:1709.01201*, 2017.

[41] T. Knudsen and J. Leth, "Stochastic MPC using the unscented transform," in *IEEE Annual American Control Conference*, 2018, pp. 4718–4724.

[42] M. I. Ribeiro, "Kalman and extended Kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, p. 46, 2004.

[43] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, 1997, pp. 182–193.

[44] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[45] S. J. Julier, "The scaled unscented transformation," in *IEEE American Control Conference*, vol. 6, 2002, pp. 4555–4559.

[46] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1901–1907.
[47] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.
[48] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Robotics Research*. Springer, 2020, pp. 75–84.
[49] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*. INFORMS, 2019, pp. 130–166.
[50] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3863–3870, 2020.
[51] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
[52] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017, pp. 1–16.
[53] V. A. Nguyen, S. S. Abadeh, D. Filipović, and D. Kuhn, "Mean-covariance robust risk measurement," *arXiv preprint arXiv:2112.09959*, 2021.
[54] Y.-L. Yu, Y. Li, D. Schuurmans, and C. Szepesvári, "A general projection property for distribution families," *Advances in Neural Information Processing Systems*, vol. 22, 2009.