Distributionally Safe Path Planning: Wasserstein Safe RRT

Paul Lathrop, *Graduate Student Member, IEEE*, Beth Boardman, and Sonia Martínez, *Fellow, IEEE*

***Abstract*—In this paper, we propose a Wasserstein metric-based random path planning algorithm. Wasserstein Safe RRT (W-Safe RRT) provides finite-sample probabilistic guarantees on the safety of a returned path in an uncertain obstacle environment. Vehicle and obstacle states are modeled as distributions based upon state and model observations. We define limits on distributional sampling error so the Wasserstein distance between a vehicle state distribution and obstacle distributions can be bounded. This enables the algorithm to return safe paths with a confidence bound through combining finite sampling error bounds with calculations of the Wasserstein distance between discrete distributions. W-Safe RRT is compared against a baseline minimum encompassing ball algorithm, which ensures balls that minimally encompass discrete state and obstacle distributions do not overlap. The improved performance is verified in a 3D environment using single, multi, and rotating non-convex obstacle cases, with and without forced obstacle error in adversarial directions, showing that W-Safe RRT can handle poorly modeled complex environments.**

***Index Terms*—Planning under uncertainty, robot safety, task and motion planning.**

I. Introduction

SAFETY is an essential requirement on the operation of autonomous systems in close proximity to humans, from self-driving cars to more complex human-robot teams. Our ability to guarantee safety is directly related to how we understand and manage uncertainty [1], from the epistemic kind —on environment, obstacle, and system modeling errors—to the aleatoric type —in the form of random noises.

In motion planning, the probabilistic modeling of uncertainty has enabled the integration of sensing, motion, and environmental uncertainty in a principled manner. This results in less conservative plans at the expense of higher computational and time complexity costs. However, distributional errors and lack of knowledge of the underlying probability distributions can nullify efforts to leverage this approach and guarantee safety. To address this, this work accounts for both state and obstacle distributional modeling errors through the Wasserstein metric and ambiguity sets. Using these, we approximate uncertain distributions with a quantification of the error for finite samples, and create a distributionally robust path planning algorithm for any finite sample set. The algorithm is applicable to vehicles navigating obstacles in a physical environment and extends to more abstract state spaces.

Various existing path planners have been designed to manage uncertainty. Chance-constrained optimization formulations in sampling-based planning [2]–[4] have taken the limelight because they can limit the probability of collision in a straightforward way provided that models are accurate. Two such algorithms, Chance Constrained Rapidly Exploring Random Trees (CC-RRT) [5], and Particle Chance Constrained RRT (PCC-RRT) [6], use continuous and discrete distributions, respectively, to model the vehicle state and limit the probability of collision with known obstacles. For tractability, CC-RRT uses continuous Gaussian distributions to represent the state, while PCC-RRT uses sample approximations that allow non-Gaussian models. Both algorithms return a probabilistic guarantee on safety for a given uncertainty model of the state (approximated in [6]) and fail to consider distributional modeling errors or unmodeled locational obstacle uncertainty. The work [7] characterizes path planning probabilistic completeness under uncertainty, but fails to consider distributional uncertainty and associated probabilistic guarantees.

Wasserstein-based ambiguity sets can handle these types of distributional errors [8], while retaining tractability in associated optimization problems. Thus, they have been applied in robust policy optimization against random disturbances [9], and distributed decision-making [10]. The work [11] characterizes the evolution of ambiguity sets under dynamic linear-systems transformations, which becomes useful in uncertainty quantification. Furthermore, chance constraints on moment-based ambiguity sets have been used in optimal motion planning [12], but require known, convex obstacles. Distributionally-robust chance constrained optimization for convex functions in the decision variables is studied in [13]. Here, for tractability, the distributional problem is inner-approximated by means of a conditional value-at-risk (CVaR)-like function. Returning trajectories with bounded-risk guarantees and probabilistic locations of obstacles is studied in [14], but these require an obstacle model and

Manuscript received June 14, 2021; accepted October 27, 2021. Date of publication November 17, 2021; date of current version November 29, 2021. This letter was recommended for publication by Associate Editor M. Morales and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported in part by Los Alamos National Laboratory and is approved for release under LA-UR-21-25559v4, with partial support from the Office of Naval Research under Grant N00014-19-1-2471. *(Corresponding author: Paul Lathrop.)*

Paul Lathrop is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92037 USA, and also with the Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: pdlathrop@gmail.com).

Beth Boardman is with the Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: bboardman@lanl.gov).

Sonia Martínez is with the Dept. of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, California 92037 USA (e-mail: soniamd@ucsd.edu).

Digital Object Identifier 10.1109/LRA.2021.3128696

obstacle movement model. Furthermore, this work considers bounding risk contours through an obstacle environment and not dynamic and distributional modeling errors.

Beyond this, risk-aware motion planning algorithms have been developed [15], [16] to limit the risk of collision as modeled by the CVaR metric. While [15] does not consider distributional uncertainty, [16] employs Wasserstein ambiguity sets around random obstacle drift vectors to constrain a distributionally robust model predictive control problem. However, the robot state is assumed to be known with certainty, and obstacles are modeled as non-rotating, convex polytopes, with random shifts that also belong to a known convex polytope. This results in large, poorly scaling optimization problems that are solved approximately via McCormick relaxations. Additionally, no results are shown employing the claimed sample guarantees.

In this work, we consider a motion planning problem where we account for uncertainty in not-necessarily-convex obstacles and in vehicle motion through a novel sampling based planner. W-Safe RRT is based upon checking possible paths and states in pre-planning rather than solving directly constrained optimal control problems online. Because our algorithm is a pre-planner, we create and maintain environmental and state models. Unlike previous Gaussian and particle-based planners, our algorithm makes use of the Wasserstein metric measured between individual vehicle-obstacle states to create a distributionally safe and probabilistically complete random path planner. By exploiting recent results on the number of samples required to obtain probabilistic guarantees over compact spaces, we can probabilistically guarantee with a precise bound that a resulting vehicle path is greater than a certain Wasserstein distance (W-distance) away from a moving obstacle model. We create a minimum encompassing ball algorithm inspired by PCC-RRT [6] that receives the same information with the same assumptions as W-Safe RRT for comparison purposes. We show that W-Safe RRT outperforms the comparison algorithm in simple convex obstacle and rotating non-convex multi-obstacle environments at the expense of computation time.

II. Problem Formulation

We introduce here the general notation, concepts, and problem formulation that are used throughout this work. Let $d, p \in \mathbb{N}$, and let $\mathbb{N}_N$ be the set of natural numbers from one to N. Let $\mathbb{R}^d$ be the d-dimensional real space with $x \in \mathbb{R}^d$ denoting a vector in it. We denote the p-norm on $\mathbb{R}^d$ as $\|x\|_p = \sqrt[p]{\|{}^1x\|^p + \cdots + \|{}^dx\|^p}$, $x \in \mathbb{R}^d$, with the Euclidean norm as $\|\cdot\| \equiv \|\cdot\|_2$ and ${}^1x, \ldots, {}^dx$ as the components of x. The diameter of a set S in the p-norm is denoted as $\mathrm{diam}_p(S) := \sup_{x,y\in S} \|x - y\|_p$. Let $\bar{S} = \mathbb{R}^d \setminus S$ be the complement of set $S \subseteq \mathbb{R}^d$.

Let the Borel σ-algebra on $\mathbb{R}^d$ be denoted as $\mathcal{B}(\mathbb{R}^d)$, and the set of probability distributions on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ as $\mathcal{P}(\mathbb{R}^d) \equiv \mathcal{P}$. In what follows, we identify probability distributions $\mathbb{P} \in \mathcal{P}$ with the measures μ used to generate them. Throughout this paper, the empirical distribution built on a set of samples $\hat{\xi}_i, i \in \{1, \ldots, N\}$ is $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i}$, where δ is the Dirac delta function. Let $X \in \mathbb{R}^d$ be a random vector variable. We denote the probability that $X \in S$, for $S \subseteq \mathbb{R}^d$, as $\mathbb{P}(X \in S) \in [0, 1]$, for a given $\mathbb{P} \in \mathcal{P}$.

The p-Wasserstein distance (W_p-distance) between two distributions $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ is defined as

$$W_p(\mu, \nu) := \inf_{\mathbb{P}\in\mathcal{P}(\mu,\nu)} \mathbb{E}_{\mathbb{P}}[\|\hat{\xi}_1 - \hat{\xi}_2\|_p],$$

where $\mathcal{P}(\mu, \nu)$ denotes the set of all probability distributions on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν, and ξ_1 and ξ_2 are the integration variables w.r.t. μ and ν, respectively.

We consider a pre-planning mobile robot problem where we assume obstacle and state dynamics to allow dynamic path planning. Before the algorithm is run, in an observational period, we observe the environment and robot details to create $\mathcal{M}_r$ and $\mathcal{M}_O$, the assumed dynamic robot and obstacle models, and the initial states. To cope with incorrect modeling and unknown disturbances, we will make use of Wasserstein ambiguity sets, see Section III. The robot has state $x \in \mathbb{R}^d$ which is constrained within a compact configuration space, $Q \subseteq \mathbb{R}^d$, and control $u \in \mathbb{R}^n$, with,

$$\mathcal{M}_r: \quad x(t+1) = Ax(t) + Bu(t) + w(t), \quad \forall\, t \in \mathbb{N},$$

where (A, B) is controllable. $\mathcal{M}_r$ is a linear time invariant dynamic model chosen to approximate unknown, possibly nonlinear dynamics. If bounds on the control $u(t)$ are known, they can be directly applied in the calculation of $x(t+1)$ above. The noise model is chosen as $w(t) \sim \mathcal{N}(0, P_w)$. The robot must avoid N_O rigidly rotating and translating obstacles, $\mathcal{O}_1, \ldots, \mathcal{O}_{N_O} \subset Q$, to stay safe. Obstacle movement is modeled as a rigid body transformation,

$$\mathcal{M}_O: \quad \mathcal{O}_k(t+1) = \hat{R}_k\mathcal{O}_k(t) + \hat{\gamma}_k, \text{ for } k \in [1, N_O],$$

where $\hat{R}_k$ is a time-invariant rotation matrix and $\hat{\gamma}_k \in \mathbb{R}^d$ is a time-invariant vector translation. $\hat{R}_k$ and $\hat{\gamma}_k$ are estimates of the unknown true dynamics R_k and γ_k based upon the observation period. The goal is for the robot to navigate a path within the free space $\bar{\mathcal{O}}(t) := Q \setminus (\mathcal{O}_1(t) \cup \cdots \cup \mathcal{O}_{N_O}(t))$, from the initial state $x_I \in \mathbb{R}^d$ to the goal state $x_G \in \mathbb{R}^d$. The path is denoted as an ordered set of states $Z : x_I, x_2, \ldots, x_G$. For the path to be considered safe, $x_i \in \bar{\mathcal{O}}(t), \forall\, t \geq 0$.

To account for state modeling errors, the state is represented by a distribution,

$$\mathbb{P}_1(t) := \mathcal{N}(x(t), P_x(t)), P_x(t+1) = AP_x(t)A^T + P_w. \quad (1)$$

To account for obstacle model errors, each obstacle location is represented by a distribution evolving according to,

$$\mathbb{P}_{2,k}(t) := \mathcal{N}(\mathcal{O}_k^C(t), P_k(t)), P_k(t+1) = P_k(t) + \kappa_k, \quad (2)$$

for $k \in [1, N_O]$, where $\mathcal{O}_k^C(t)$ represents the center of mass of $\mathcal{O}_k(t)$, and κ_k represents the distribution spread since the observation period.

III. Algorithm: Wasserstein-Safe RRT

In this section, we define a novel path planning algorithm for uncertain robotic states and environments based on Rapidly Exploring Random Trees [17]. In Proposition 1, we leverage

the recent probabilistic guarantees on the discrete W-distance from [11], [18] to create a safety search criterion.

To determine whether states at time t are safe, Algorithm 1 uses empirical distributions sampled from $\mathbb{P}_1(t)$ and $\mathbb{P}_{2,1}(t), \ldots, \mathbb{P}_{2,N_O}(t)$, which are available for sampling for all $t \geq 0$ because the algorithm creates and maintains them. To sample $\mathbb{P}_1(t)$ and $\mathbb{P}_{2,1}(t), \ldots, \mathbb{P}_{2,N_O}(t)$ at time t, a temporarily truncated and finitely supported version of each distribution is created. Wasserstein truncation errors can be found and added to ε_1 and ε_2 at each time step to maintain probabilistic guarantees. Alternatively, any finitely supported distributions can be used.

Proposition 1: Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be two probability distributions over the compact configuration space $Q \subseteq \mathbb{R}^d$. Let $\mathbb{P}_i$ be supported on $B_i \subseteq Q$ with $\rho_i = 1/2\text{diam}_\infty(B_i), \rho_i < \infty$. Let $\hat{\mathbb{P}}_1^N$ and $\hat{\mathbb{P}}_2^M$ be the empirical distributions defined from taking N and M samples of $\mathbb{P}_1$ and $\mathbb{P}_2$, respectively, at an arbitrary time t. Given a confidence $1-\beta \in [0,1]$, for any $p \geq 1, N \geq 1$ when $p < d/2$, it holds that

$$W_p(\mathbb{P}_1, \mathbb{P}_2) \geq W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_2^M) - \varepsilon_1^N - \varepsilon_2^M, \tag{3}$$

with ε_i^N given by,

$$\varepsilon_i^N(\beta, \rho_i) = \left(\frac{\ln(C\beta^{-1})}{c}\right)^{1/d} \frac{\rho_i}{N^{1/d}},$$

constants C and c given by,

$$C = \frac{(C_*)^d}{2\sqrt{d}^d}, \text{ and } c = \frac{1}{2^d\sqrt{d}^d}, \tag{4}$$

and constant C_* given by,

$$C_* = \sqrt{d}\, 2^{\frac{d-2}{2p}} \left(\frac{1}{1-2^{p-d/2}} + \frac{1}{1-2^{-p}}\right)^{\frac{1}{p}}. \tag{5}$$

Proof: Note first that by the triangular inequality,

$$W_p(\mathbb{P}_1, \mathbb{P}_2) \geq W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_2^M) - W_p(\mathbb{P}_1, \hat{\mathbb{P}}_1^N) - W_p(\hat{\mathbb{P}}_2^M, \mathbb{P}_2). \tag{6}$$

Let the W-distance between a distribution and an empirical one built with samples of said distribution be constrained within a bound ε,

$$W_p(\mathbb{P}_1, \hat{\mathbb{P}}_1^N) \leq \varepsilon_1^N, \quad W_p(\mathbb{P}_2, \hat{\mathbb{P}}_2^M) \leq \varepsilon_2^M. \tag{7}$$

Proposition 1 Equation (3) follows via substitution of (7) into (6). According to [11, Proposition 6 and 20], the nominal bound ε_i between the continuous and empirical distributions with a confidence $1-\beta$ is given below. Consider a sequence $(X_i)_{i\in\mathbb{N}}$ of i.i.d $\mathbb{R}^d$-valued random variables supported compactly on distribution μ. Then, for $p < d/2$ and $N \geq 1$, we have $P(W_p(\mu^N, \mu) \leq \varepsilon^N(\beta, \rho)) \geq 1-\beta$, where,

$$\rho = \frac{1}{2}\text{diam}_\infty(\text{supp}(\mu)). \tag{8}$$

Under the assumptions of Proposition 1, we can select ε^N,

$$\varepsilon^N(\beta, \rho) = \rho(C_* N^{-\frac{1}{d}} + \sqrt{d}(2\ln\beta^{-1})^{\frac{1}{2p}} N^{-\frac{1}{2p}}), \tag{9}$$

with C_* as given in (5). Proposition 1 Equation (4) follows as shown in [11, Corollary 21].

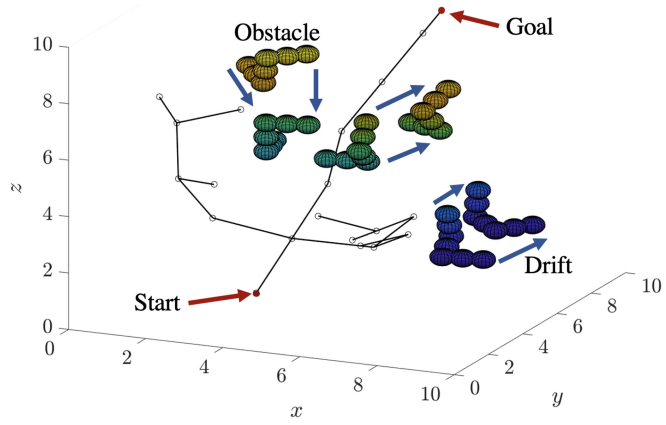

Fig. 1. Search tree created by W-Safe RRT to path plan in a multi-nonconvex-obstacle 3D configuration space with drift and rotational uncertainty present.

We design a sampling-based path planner similar to RRT, but which compares $\mathbb{P}_1(t)$ and $\mathbb{P}_{2,k}(t)$ for each obstacle k instead of using a collision checker. In particular, the algorithm bounds the W-distance between the probability distribution of the vehicle state and the obstacle state by means of the associated empirical distributions and the bounds of Proposition 1. The Wasserstein metric is used over other approaches, such as Kullback-Leibler divergence, due to its consistency with state-space Euclidean distance [19]. The inputs to Algorithm 1 are the initial vehicle state $x_I \in \mathbb{R}^d$, the goal vehicle state $x_G \in \mathbb{R}^d$, the robot dynamic model $\mathcal{M}_r$, the obstacle model $\mathcal{M}_O$, the confidence β, and the W-distance threshold value θ. The output of the algorithm is the W-Safe path $Z = \{x_I, x_2, \ldots, x_G\}$.

To create a tree T, Algorithm 1 generates a random sample, finds the nearest node in the tree, and simulates control of the vehicle from the parent node to the sample with intermediate states. The state distribution and obstacle distributions are simulated via $\mathcal{M}_r$ and $\mathcal{M}_O$, and then intermediate states are checked for safety with confidence $1-\beta$. If the sample and intermediates are safe, as explained next, the sample is added to the tree. At intermediate state $x(j)$, we have the empirical state distribution $\hat{\mathbb{P}}_1^N(j)$ sampled from $\mathbb{P}_1(j)$, and the N_O empirical obstacle distributions $\hat{\mathbb{P}}_{2,k}^M(j)$ sampled from $\mathbb{P}_{2,k}(j)$ for $k \in [0, N_O]$. $W_p(\hat{\mathbb{P}}_1^N(j), \hat{\mathbb{P}}_{2,k}^M(j))$ for each obstacle is found through [20], based upon [21], then Proposition 1 is used in Method 1: WCheck to ensure,

$$W_p(\hat{\mathbb{P}}_1^N(j), \hat{\mathbb{P}}_{2,k}^M(j)) - \varepsilon_1^N - \varepsilon_2^M \geq \theta, \tag{10}$$

as shown in Fig. 2, so that $W_p(\mathbb{P}_1(j), \mathbb{P}_{2,k}(j)) \geq \theta$ for each obstacle k. New samples are added to T until $x_G \in T$.

In contrast to RRT, $N_O + 1$ distinct distribution trees, $T_{\mathbb{P}1}$ and $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$, keep track of $\mathbb{P}_1$ and $\mathbb{P}_{2,1}, \ldots, \mathbb{P}_{2,N_O}$ at each vertex in T. The vertices of $T_{\mathbb{P}1}$ and $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$ are distributions, and the edges are identical to those in T. Once a parent node $x_{\text{parent}} \in T$ is found for a new sample x_{sample}, the distributions $\mathbb{P}_1^{x_{\text{parent}}}$ and $\mathbb{P}_{2,k}^{x_{\text{parent}}}, \forall\, k \in \mathbb{N}_{N_O}$ corresponding to x_{parent} can be quickly found by accessing the same indices in $T_{\mathbb{P}1}$ and $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$. If x_{sample} is added to T, the distributions

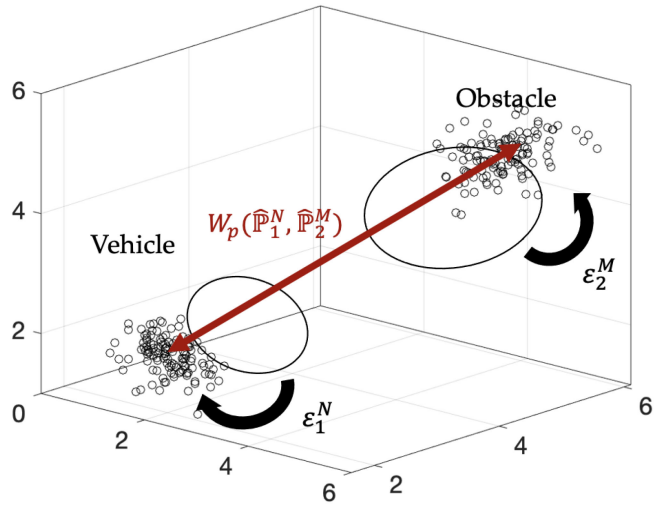

Fig. 2. Discrete W-distance between robot and obstacle distributions $W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_2^M)$ shown compared to maximally-poor distributional sampling errors ε_1^N and ε_2^M, with continuous distribution locations approximated by ovals.

$\mathbb{P}_1^{x_{\text{sample}}}$ and $\mathbb{P}_{2,1}^{x_{\text{sample}}}, \ldots, \mathbb{P}_{2,N_O}^{x_{\text{sample}}}$ corresponding with that sample are added to $T_{\mathbb{P}1}$ and $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$.

Under certain conditions, W-Safe RRT is a probabilistically complete planner, and the statement of probabilistic completeness follows.

Theorem 2: Let $\mathbb{P}_{\text{obs},k}(\ell)$, be the distributions of obstacles k at time ℓ, for $k \in \mathbb{N}_{N_O}$ and $\ell \in [0, \infty)$, $\mathbb{P}_{\text{rob}}^0$ be initial robot configuration, and B_{goal} be a ball around the goal configuration in a compact configuration space Q. Suppose that $\forall\, k$ the support of each $\mathbb{P}_{\text{obs},k}(\ell)$ remains in a ball $B_{\text{obs},\,k}(\ell)$. Let $S \subset Q$ be $\bigcup_{k=1}^{N_O} S_k$, with S_k such that $\forall\, \ell, B_{\text{obs},k}(\ell) \subset S_k$. Assume that for each ℓ there exist controls such that the robot distribution is transformed into $\mathbb{P}_{\text{rob}}(\ell)$, and assume the support of this distribution remains in a compact ball $B_{\text{rob}}(\ell)$. Assume there is a path $\mathbb{P}_{\text{rob}}(\ell)$ from $\mathbb{P}_{\text{rob}}^0$ to B_{goal} such that the final robot distribution mean $\mu_{\text{rob}} \in B_{\text{goal}}$ and $\forall\, \ell$, the set distance $\text{dist}_p(B_{\text{rob}}(\ell), S) > \delta$,[1] for some $\delta > 0$. Then, if we take N, M samples such that Eq. 3 from Proposition 1 holds, W-Safe RRT returns a path from $\mathbb{P}_{\text{rob}}^0$ to B_{goal} with probability 1 that satisfies $W_p(\mathbb{P}_{\text{rob}}(\ell), \mathbb{P}_{\text{obs},k}(\ell)) > \delta - \varepsilon_1^N - \varepsilon_2^M$, for all $k \in \mathbb{N}_{N_O}$ and $\ell \in [0, \infty)$ with confidence $1 - \beta$.

Proof: The proof follows from the probabilistic completeness of RRT [17] and Proposition 1. These results can be combined because the sampling processes w.r.t. configurations in Q and that of obtaining samples of each distribution are independent. From the probabilistic completeness of RRT, the algorithm will produce a path from $\mathbb{P}_{\text{rob}}^0$ to B_{goal} such that $\forall\, \ell$, $\text{dist}_p(B_{\text{rob}}(\ell), S) > \delta$, for some $\delta > 0$ with probability 1. Note that $\forall\, k$ and $\forall\, \ell$,

$$W_p(\hat{\mathbb{P}}_{\text{rob}}(\ell), \hat{\mathbb{P}}_{\text{obs,k}}(\ell)) \geq \text{dist}_p(B_{\text{rob}}(\ell), B_{\text{obs,k}}(\ell)) > \delta,$$

implying via Proposition 1,

$$W_p(\mathbb{P}_{\text{rob}}(\ell), \mathbb{P}_{\text{obs,k}}(\ell)) \geq \delta - \varepsilon_1^N - \varepsilon_2^M,$$

[1]Here, $\text{dist}_p(A, B) = \inf\{\|x - y\|^p \mid x \in A, y \in B\}$.

Algorithm 1: Wasserstein-Safe RRT.

Input: $x_I, x_G, \mathcal{M}_r, \mathcal{M}_O, \beta, \theta$
Output: $Z : x_I, x_2, \ldots, x_G$
1: Initialize Tree T with vertex x_I, and distribution trees $T_{\mathbb{P}1}$ and $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$ using $\mathcal{M}_O$, (1), and (2)
2: **while** $x_G \notin T$ **do**
3: Randomly draw state x_{new}
4: Find $x_{\text{parent}} \in T$, the nearest vertex to x_{new}
5: Find $\mathbb{P}_1^{x_{\text{parent}}} \in T_{\mathbb{P}1}$ and $\mathbb{P}_{2,1}^{x_{\text{parent}}}, \ldots, \mathbb{P}_{2,N_O}^{x_{\text{parent}}} \in T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$ corresponding with x_{parent}
6: Using $\mathcal{M}_r$, advance state from x_{parent} to x_{new} with intermediate states $x(1), \ldots, x(j)$
7: Using $\mathcal{M}_O$, advance distributions from $\mathbb{P}_1^{x_{\text{parent}}}, \mathbb{P}_{2,1}^{x_{\text{parent}}}, \ldots, \mathbb{P}_{2,N_O}^{x_{\text{parent}}}$ to $\mathbb{P}_1^{x_{\text{new}}}, \mathbb{P}_{2,1}^{x_{\text{new}}}, \ldots, \mathbb{P}_{2,N_O}^{x_{\text{new}}}$ with intermediate distributions $\mathbb{P}_1(1), \ldots, \mathbb{P}_1(j)$, $\{\mathbb{P}_{2,1}(1), \ldots, \mathbb{P}_{2,1}(j)\}, \ldots, \{\mathbb{P}_{2,N_O}(1), \ldots, \mathbb{P}_{2,N_O}(j)\}$
8: Unsafe = False
9: **for** $i = 1 : j$ **do**
10: W_p = WCheck ($\mathbb{P}_1(i), \{\mathbb{P}_{2,1}(i), \ldots$
11: $\ldots, \mathbb{P}_{2,N_O}(i)\}, \beta, N, M$)
12: **if** $W_p < \theta$ **then**
13: Continue
14: **else**
15: Unsafe = True, Break
16: **end if**
17: **end for**
18: **if** Unsafe **then**
19: Continue
20: **else**
21: Add x_{new} to T, add $\mathbb{P}_1^{x_{\text{new}}}$ to $T_{\mathbb{P}1}$
22: Add $\mathbb{P}_{2,1}^{x_{\text{new}}}, \ldots, \mathbb{P}_{2,N_O}^{x_{\text{new}}}$ to $T_{\mathbb{P}2,1}, \ldots, T_{\mathbb{P}2,N_O}$
23: **end if**
24: **end while**
25: Trace T from x_G to x_I to extract path
26: Return path $Z : x_I, x_2, \ldots, x_G$

Method 1: WCheck, From Algorithm 1 Line 9.

Input: $\mathbb{P}_1(i), \{\mathbb{P}_{2,1}(i), \ldots, \mathbb{P}_{2,k}(i)\}, \beta, N, M$
Output: W_p
1: Truncate $\mathbb{P}_1(i), \{\mathbb{P}_{2,1}(i), \ldots, \mathbb{P}_{2,k}(i)\}$ to $\rho_1, \rho_{2,1}, \ldots, \rho_{2,k}$ as defined in (8)
2: Calculate C_* via (5)
3: Calculate ε_1^N and $\varepsilon_{2,1}^M, \ldots, \varepsilon_{2,k}^M$ via (9)
4: Sample $\hat{\mathbb{P}}_1^N = \{\hat{\xi}_1, \ldots \hat{\xi}_N\}$ from $\mathbb{P}_1(i)$
5: Sample $\hat{\mathbb{P}}_{2,l}^M = \{\hat{\zeta}_1^l, \ldots \hat{\zeta}_M^l\}$ from $\mathbb{P}_{2,l}(i)$ for each $l \in \mathbb{N}_k$
6: Find W-distance $W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_{2,l}^M)$ between $\{\hat{\xi}_1, \ldots \hat{\xi}_N\}$ and $\{\hat{\zeta}_1^l, \ldots \hat{\zeta}_M^l\}$ for each $l \in \mathbb{N}_k$
7: Return $\inf_{l \in \mathbb{N}_k}(W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_{2,l}^M) - \varepsilon_1^N - \varepsilon_{2,l}^M)$ as per (10)

with confidence $1-\beta$.

IV. Results and Discussion

In this section, we show simulations and results that are performed in a three dimensional environment run with MATLAB ver. R2020b on a machine with an Intel i5-4690 K CPU, 32 GB RAM, and an AMD Radeon R9 290X GPU. Algorithm 1 is compared against a baseline minimum encompassing sphere method that extends PCC-RRT to handle the same assumptions, modeling, and scenarios as W-Safe RRT. The goal is to compare W-Safe RRT to this baseline, with and without forced distributional obstacle errors, in a variety of environments.

The chosen configuration space $Q \subseteq \mathbb{R}^3$ is $Q = \{(x,y,z)|x,y,z \in [0,10]\}$. The following vehicle model dynamics and control policy,

$$x(t+1) = A(x(t) - x_{\text{new}}) + Bu(t) + x_{\text{new}}\ ,\ x(0) = x_{\text{parent}},$$

$$A = \begin{bmatrix} .9 & -.05 & .1 \\ .05 & .9 & -.1 \\ 0 & .08 & .85 \end{bmatrix},\ B = \begin{bmatrix} .85 & .2 & 0 \\ -.15 & .85 & .1 \\ -.1 & .1 & .9 \end{bmatrix},$$

$$u(t) = 0.4\frac{-K(x(t) - x_{\text{new}})}{\|u(t)\|},$$

$$K = \begin{bmatrix} 2.36 & -0.51 & 0.18 \\ 0.45 & 1.93 & -0.27 \\ 0.21 & -0.18 & 1.55 \end{bmatrix},$$

are implemented from x_{parent} to x_{new} each time a random state is drawn. The control policy performs reference tracking for controlling to x_{new}. The constant matrix K can be any gain matrix such that the closed loop system is stable.

A single, spherical obstacle is used in the initial simulation results. For each path planning problem, the true obstacle location $\mathcal{O}(t)$ is chosen to be a ball of radius 0.5 centered randomly near the center of Q and drifting in a random direction. The obstacle uncertainty model is,

$$\mathbb{P}_{2,1}(t) = \mathcal{N}\left(\mathcal{O}^C(t), \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}\right),$$

where $\mathcal{O}^C(t) \in \mathbb{R}^3$ is the geometric center of $\mathcal{O}(t)$. The state uncertainty model, $\mathbb{P}_1(t)$, centered at state (x,y,z), is,

$$\mathbb{P}_1(t) = \mathcal{N}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}, \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}\right).$$

For sampling, the support of the distribution in each dimension is an interval centered at x, y, or z, with length 1 in all dimensions. The start state, x_I, and the goal state, x_G, are chosen uniformly randomly to be on opposite sides of the obstacle, forcing the path to interact with the obstacle, as shown in Fig. 3. For all simulations except the drift error percentage study, each

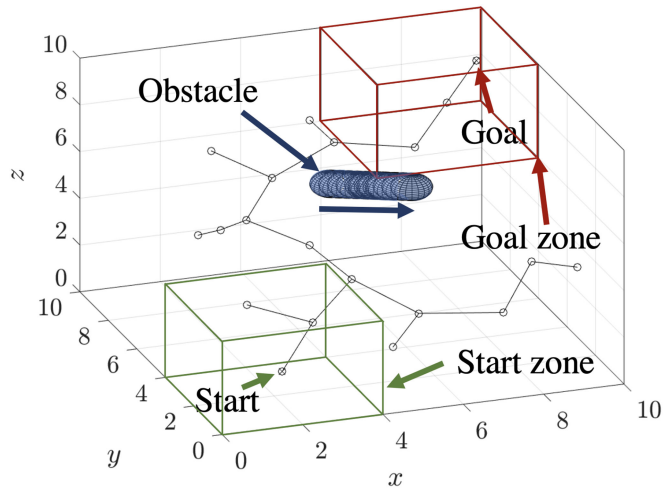

Fig. 3. Tree of probabilistically feasible states found by W-Safe RRT shown within a random opposite-corner start and goal location path planning problem.

Method 2: CompCheck.

Input: $\mathbb{P}_1(j), \mathbb{P}_2(j), N, M$
Output: s
1: Sample $\{\hat{\xi}_1, \ldots \hat{\xi}_N\}, \{\hat{\zeta}_1, \ldots \hat{\zeta}_M\}$ from $\mathbb{P}_1(j), \mathbb{P}_{2,1}(j)$
2: Find minimum encompassing spheres characterized by c_ξ, r_ξ and c_ζ, r_ζ, using Welzl's Algorithm
3: **if** $\|c_\xi - c_\zeta\| < r_\xi + r_\zeta$ **then**
4: $s =$ false
5: **else** $s =$ true
6: **end if**

algorithm assumed a drift vector $\hat{\gamma}$ with 10 percent error from the true drift γ. In simulation, the maximum distance between x_{new} and the closest parent state $x_{\text{parent}} \in T$ is capped, and the random state x_{new} draw is chosen to be toward the goal state x_G with 0.3 probability. The W_p-distance is calculated with $p = 1$ and $d = 3$, so that $p < d/2$. To use higher p values within the W_p-distance, the state can be lifted into higher dimensions to achieve $d > 3$.

A. Algorithm Performance

For the baseline comparison algorithm, the Method 1 WCheck is swapped out for the comparison Method 2 CompCheck. In Method 2, both the vehicle and obstacle distributions, $\mathbb{P}_1(j)$ and $\mathbb{P}_{2,1}(j)$, respectively, are sampled to create two circumspheres, which are then checked for overlap. The circumspheres are found through Welzl's Algorithm [22], which is chosen for simplicity. If there is overlap, the state is not safe with respect to the obstacle. This algorithm, which is an extension of PCC-RRT, includes similar simulated dynamics, particle sampling and re-sampling from distributions, and a probabilistic check to admit robot states to the RRT-style tree. It is modified to accept obstacle uncertainty representations instead of a known convex obstacle polytope by changing the comparison between two distributions to be via Welzl's Algorithm.

TABLE I
ALGORITHM PERFORMANCE

	Ball Method	W-Safe RRT
Average Time	5.345 s	12.786 s
Ave. Path Length	14.89 u	15.23 u
Ave. No. Nodes	40.38	43.76
Nominal Check	997/1000	1000/1000
Adversarial Check	681/1000	998/1000

The algorithms are evaluated by comparing returned paths with reality in two ways, nominally and adversarially. The nominal evaluation analyzes the returned path ensuring it avoids the 0.5 radius spherical obstacle. The adversarial evaluation assumes that at any state, $x_i \in Z$, each obstacle will have shifted one unit from $\mathcal{O}(t)$ toward x_i to try to block the path. The post processing evaluations assess the safety of returned paths and therefore algorithm performance. Paths are not discarded by the algorithms based on the evaluations.

Table I records the number of correctly returned paths out of 1000 path planning problems, the average time to solve each problem, the average path length of each solution, and the average number of nodes per solution. Algorithm 1 using Method 1 took an average of 12.786 seconds to solve each problem and returned 998 safe paths when checked adversarially. Method 2 took an average of 5.345 seconds and returned 681 safe paths when checked adversarially. This time penalty is largely due to calculating the discrete W-distance, $W_p(\hat{\mathbb{P}}_1^N, \hat{\mathbb{P}}_{2,1}^M)$ in Algorithm 1, which requires solving an optimization problem (discrete optimal transport reduces to a linear program.) The W-method is more robust in the face of an adversarial path check.

Additionally, Method 1 allows W-Safe RRT to return a path with the probabilistic guarantee on distributional uncertainty given in Proposition 1, while Method 2 does not. The adversarial check analyzes algorithm resiliency to adversarial distributional modeling errors, where obstacles exist outside of possible distribution sampling ranges and toward the state in question. W-Safe RRT outperforms the ball method when returned paths are checked adversarially.

Any finitely supported distribution can be used to represent uncertainty, and poor distribution choice is accounted for by W-Safe RRT. Additionally, when a known state is projected forward with a dynamic model, unknown future events impact the state. W-Safe RRT accounts for distributional uncertainty that results from dynamic model errors, noise model errors, and unpredictable future disturbances.

B. Multi-Obstacle Performance

We compare the algorithms in a random start and goal environment with 3 randomly placed and drifting obstacles, as shown in Fig. 4. Obstacle drift information is known to each algorithm with 10% error. Table II records the number of correctly returned paths out of 1000 as well as the averages of relevant metrics for both algorithms. With the adversarial check on obstacles, when compared to the single obstacle case, the ball method dropped from 681 safe paths to 344 while W-Safe RRT dropped from 998 to 990, and both algorithms take about three times as long to run. W-Safe RRT shows vastly improved performance over the baseline in uncertain multi-obstacle environments.

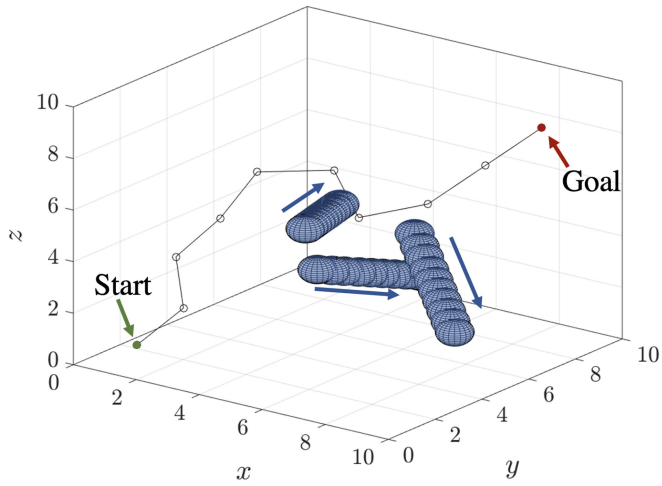

Fig. 4. Returned path from W-Safe RRT shown navigating a three obstacle environment with random constant drift values.

TABLE II
THREE OBSTACLE ALGORITHM PERFORMANCE

	Ball Method	W-Safe RRT
Average Time	16.53 s	45.12 s
Ave. Path Length	14.65 u	16.27 u
Ave. No. Nodes	39.41	51.46
Nominal Check	935/1000	1000/1000
Adversarial Check	344/1000	990/1000

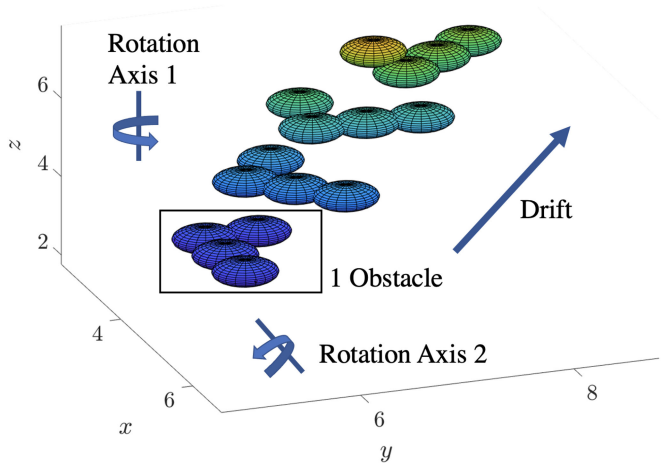

Fig. 5. A single L-shaped non-convex randomly rotating and translating obstacle moving across a 3D environment.

C. Non-Convex and Rotating Multiple Obstacle Case

We compare the algorithms in a random start and goal environment with three randomly placed, drifting, and rotating non-convex obstacles. Obstacles tested were L-shaped, rotate on multiple world-frame axes, and rotate between a fifth and a half of a full rotation during each simulation, as shown in Fig. 5. Obstacle drift information is again known to each algorithm with

TABLE III
ROTATING NON-CONVEX OBSTACLE PERFORMANCE

	Ball Method	W-Safe RRT
Average Time	20.48 s	62.71 s
Ave. Path Length	14.77 u	16.36 u
Ave. No. Nodes	40.25	47.03
No. Safe Paths	663/1000	867/1000

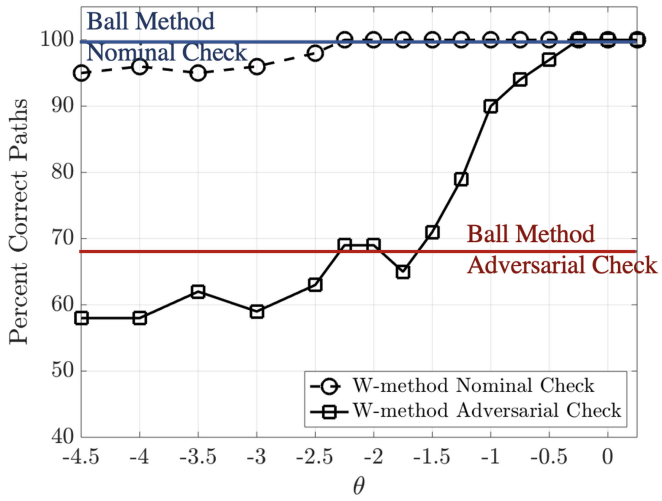

Fig. 6. Illustration of variance to the risk parameter θ and its effect on percent safe paths.

10% error and no information on obstacle shape and rotation is known. As shown in Table III, W-Safe RRT returned 867 out of 1000 paths as safe paths, at a cost of approximately three times the computation time, while the comparison algorithm returned 663 safe paths. W-Safe RRT outperforms the comparison method in high uncertainty environments with rotating non-convex obstacles.

D. Variance of Risk Parameter

The W-distance threshold value θ can be used as a risk parameter to change the margin of safety, as shown in Fig. 6. Risk is defined as the likelihood that returned paths intersect with obstacles. Risk parameter performance is measured by the nominal and adversarial obstacle checks on large numbers of produced paths at each parameter level. Since the Ball Method is independent of θ, its performance is represented as a line at 68.1% adversarially and 99.7% nominally. As shown in Fig. 6, reducing θ decreases the distributional safety and causes the percentage of safe paths to drop. With a low enough θ, W-Safe RRT can be outperformed by the comparison method. The confidence $1-\beta$ can also act as a risk parameter, and parameter value choice is left as a supervisory decision for the appropriate trade-off between performance and cost in the specific application.

E. Variance of Drift Informational Error

When vehicle observations and assumptions toward obstacle drift vector are erroneous, W-Safe RRT performance and comparison algorithm performance both drop. As the drift error percentage is larger, the performance of W-Safe RRT drops from 99% correct paths to 80% correct paths, as shown in Fig. 7. The performance of the baseline method also sees an overall downward trend, but it is bounded from below at 60%, which could represent the proportion of paths that do not interact with the obstacle at all. With highly erroneous information, W-Safe RRT still outperforms the baseline algorithm.

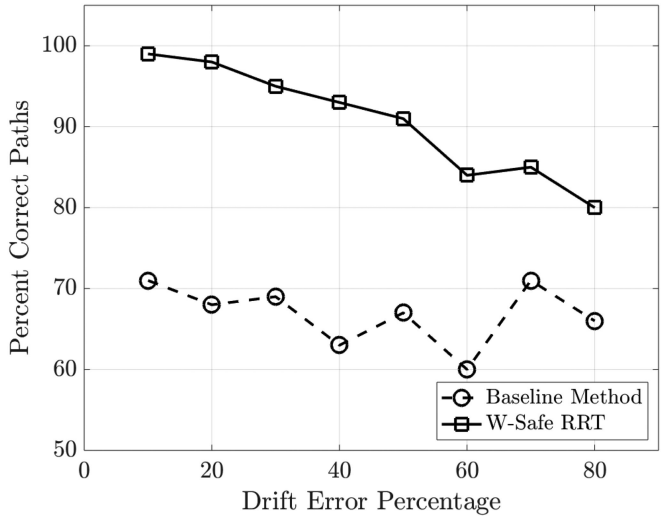

Fig. 7. Illustration of variance to the drift error percentage and its effect on percent safe paths.

F. Comparison With Similar Algorithms and Extensions

The work [16] describes a state-of-the-art path planning tool that uses Wasserstein ambiguity sets around obstacle drift values. The main approach consists of constraining a model predictive control problem to pick state control $u(t)$ that can closely follow an un-safety-constrained reference trajectory. In contrast, W-Safe RRT is an offline pre-planning state estimation algorithm that applies uncertainty models to the vehicle state as well as the obstacle states and only admits nodes to the RRT tree that satisfy Wasserstein distance safety constraints. W-Safe RRT can handle rotating non-convex and convex obstacles with non-linear boundaries and substantially drifting obstacles that are not fully known to the algorithm. In the spirit of sampling-based motion planners, the explicit boundary representation of the obstacles and bounding uncertainty sets is not required; this makes the algorithm generalizable to higher-dimensional configuration spaces. Crucially, W-Safe RRT is tested to return paths with a probabilistic guarantee on distributional sampling error. Lastly, because the MPC constraints in [16] scale with number of samples, the number of obstacles, and time horizon, it becomes computationally more expensive in the face of a large number of samples, substantial drift, and offline use over long time horizons.

Because of the assumptions in [16] that include that (i) the robot state and a convex polytopes of each obstacle are known, and (ii) the uncertainty satisfies known polytope constraints; the fact that W-Safe RRT simulations are performed with rotating

non-convex and substantially drifting obstacles; and the fact that [16] considers a MPC decision making solver that is not readily adaptable to be a pre-planning path finder; a direct comparison in simulation is neither plausible nor meaningful. Performing fair comparisons between algorithms that rely on differing assumptions is an important open question in safe learning and control [23].

As a result, for a state of the art comparison algorithm, we have extended PCC-RRT [6] into the associated minimum encompassing-ball algorithm. This extension takes the same assumptions and information as W-Safe RRT, and can compare multiple uncertainty distributions in a direct way.

W-Safe RRT can be extended to handle known time-varying discrete linear systems by substituting the appropriate equations into $\mathcal{M}_r$. However, known nonlinear systems will result in nonlinear distributional evolutions, which can be handled by linearizing dynamics at each time step. W-Safe RRT can handle continuous-time dynamical systems by taking a small enough discretization step when performing planning and safety checks.

V. Conclusion

Wasserstein Safe RRT leverages probabilistic guarantees on discrete sampling error from [11], [18] to build a probabilistically complete RRT-style path planner that accounts for distributional uncertainty. When the vehicle and the obstacle are both represented through distributions, W-Safe RRT outperforms the PCC-RRT-inspired minimum encompassing ball method in simple convex obstacle environments and in highly uncertain environments with rotating non-convex obstacles. W-Safe RRT carries probabilistic guarantees on returned paths being a certain Wasserstein distance away from obstacles at the cost of a time penalty, and shows robustness in the face of distributional error. Future work includes testing W-Safe RRT with linear approximations of nonlinear dynamics, planning with continuous time dynamics, and planning in crowded static environments.

References

[1] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *Proc. Amer. Control Conf.*, 2017, pp. 4023–4028.

[2] L. Blackmore, H. Li, and B. C. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. Amer. Control Conf.*, 2006, p. 7.

[3] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.

[4] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 723–730.

[5] B. D. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Conf. Guid., Navigation Control*, 2010, Art. no. 8160.

[6] B. D. Luders and J. P. How, "Probabilistic feasibility for non-linear systems with non-Gaussian uncertainty using RRT," *AIAA Infotech, Aerosp.*, 2011, Art. no. 1589.

[7] A. Agha-Mohammadi, S. Chakravorty, and N. Amato, "On the probabilistic completeness of the sampling-based feedback motion planners in belief space," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 3983–3990.

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

[9] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3863–3870, Aug. 2020.

[10] A. Cherukuri and J. Cortés, "Cooperative data-driven distributionally robust optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4400–4407, Oct. 2020.

[11] D. Boskos, J. Cortés, and S. Martínez, "High-confidence data-driven ambiguity sets for time-varying linear systems," *SIAM J. Uncertainty Quantification*, 2021, under review. [Online]. Available: https://arxiv.org/abs/2102.01142

[12] T. H. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 6518–6523.

[13] A. Hota, A. Cherukuri, and J. Lygeros, "Data-driven chance constrained optimization under Wasserstein ambiguity sets," in *Proc. Amer. Control Conf.*, Philadelphia, PA, USA, 2019, pp. 1501–1506.

[14] A. Jasour, W. Han, and B. Williams, "Convex risk bounded continuous-time trajectory planning in uncertain nonconvex environments," 2021, *arXiv:2106.05489*.

[15] S. Singh, Y. Chow, A. Majumdar, and M. Pavone, "A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2905–2912, Jul. 2019.

[16] A. Hakobyan and I. Yang, "Wasserstein distributionally robust motion planning and control with safety constraints using conditional value-at-risk," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 490–496.

[17] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, vol. 2, pp. 995–1001.

[18] D. Boskos, J. Cortés, and S. Martínez, "Data-driven ambiguity sets with probabilistic guarantees for dynamic processes," *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 2991–3006, Jul. 2021.

[19] Z. Littlefield, D. Klimenko, H. Kurniawati, and K. Bekris, "The importance of a suitable distance function in belief-space planning," in *Proc. Robot. Res.*, 2018, pp. 683–700.

[20] U. Yilmaz, "The earth mover's distance," MATLAB central file exchange, 2009. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/22962-the-earth-mover-s-distance

[21] L. G. Y. Rubner and C. Tomasi, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[22] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *New Results and New Trends in Computer Science*. Berlin, Germany: Springer, 1991, pp. 359–370.

[23] L. Brunke *et al.*, "Safe learning in robotics: From learning-based control to safe reinforcement learning," 2021, *arxiv:2108.06266*.