Stochastic Implicit Neural Signed Distance Functions for Safe Motion Planning under Sensing Uncertainty

Carlos Quintero-Peña, Wil Thomason, Zachary Kingston, Anastasios Kyrillidis and Lydia E. Kavraki

***Abstract*— Motion planning under sensing uncertainty is critical for robots in unstructured environments, to guarantee safety for both the robot and any nearby humans. Most work on planning under uncertainty does not scale to high-dimensional robots such as manipulators, assumes simplified geometry of the robot or environment, or requires per-object knowledge of noise. Instead, we propose a method that *directly* models sensor-specific aleatoric uncertainty to find safe motions for high-dimensional systems in complex environments, without exact knowledge of environment geometry. We combine a novel implicit neural model of stochastic signed distance functions with a hierarchical optimization-based motion planner to plan low-risk motions without sacrificing path quality. Our method also explicitly bounds the risk of the path, offering trustworthiness. We empirically validate that our method produces safe motions and accurate risk bounds and is safer than baseline approaches. A version of this paper has been accepted to be published at ICRA 2024.**

I. Introduction

Robots in unstructured environments must reliably plan safe (*i.e.*, collision-free) motions using only uncertain, noisy sensor percepts. For robots in human-oriented environments (*e.g.*, home or assistive robotics), this capability is crucial—as unsafe motions may hurt humans—and challenging, as these robots are often high degree-of-freedom (DoF) manipulators. Reliable safety under uncertainty requires not only producing plans that are unlikely to collide, but also providing evidence that plans are trustworthy. Moreover, for practical use, planners need to efficiently support complex environments without knowledge of the true environment geometry.

However, most work on motion planning under uncertainty makes simplifying assumptions about robot or environment geometry (*e.g.*, point robots or environments with only known, simple geometry) [1–5], does not scale to high DoF systems, or places strict assumptions on the distributions of noise (*e.g.*, only translational noise, segmented to individual objects or normally distributed) [1, 6, 7].

In contrast, we introduce a method for reliable, safe motion planning for high DoF systems under sensing uncertainty that directly models inherent sensor noise without placing assumptions on the environment. We propose to quantify the aleatoric uncertainty of the sensor with an implicit model of the stochastic signed distance fields between the robot's links and points in the environment, conditioned on the robot's configuration. By explicitly modeling this uncertainty, we can both compute safe paths given only noisy sensing and approximately bound the remaining risk of collision.

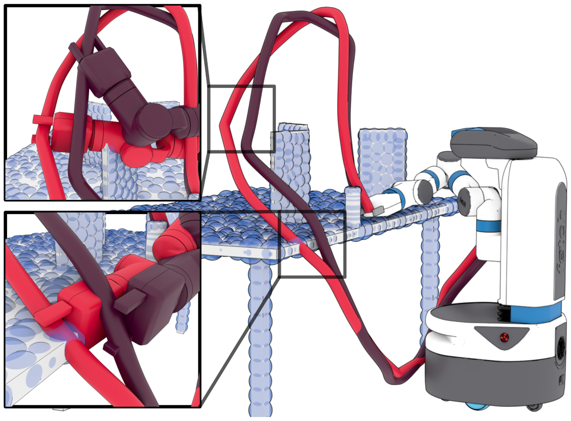

Fig. 1. Simulated motion planning problem under sensing uncertainty. The environment is composed of noisy points (blue spheres) to be avoided. The robot must plan to grasp the cylinder without colliding with the table or objects. Our method transforms a candidate path (red) into a safe path (purple) by solving a sequence of optimization problems that account for sensing uncertainty. Cutouts show parts of the path transformation: the arm is pushed away from noisy regions to attain safer behavior.

Specifically, we contribute (1) a variational inference perspective on modeling stochastic signed distance fields for motion planning (inspired by [8]), used to learn (2) an implicit neural model of sensor-specific noisy egocentric distance, which we incorporate in (3) a novel chance-constrained inverse kinematics (IK) formulation, allowing us to create (4) a hierarchical planner that produces minimal risk motions (with respect to the learned distance model and an uncertainty-agnostic initial motion plan) in realistic environments. Our learned model directly predicts distribution parameters for noisy distance measurements to arbitrary points in the environment, allowing it to capture the aleatoric uncertainty of the sensor in question without assuming that noise is segmented to the level of individual objects or requiring knowledge of object geometry. We empirically validate that our model correctly predicts both distance values and their uncertainty, and that our planner finds motion plans that are both safe (*i.e.*, minimize risk) and reliable (*i.e.*, the predicted risk matches or conservatively upper-bounds the empirically measured probability of collision). We further compare our planner to a commonly used baseline and show that, despite longer planning times, we produce significantly safer and higher-quality plans.

II. Preliminaries

We consider a robot with n controllable joints and configuration space $\mathcal{Q} \subseteq \mathbf{R}^n$. We assume that the environment is rep-

All authors are affiliated with the Department of Computer Science, Rice University, Houston TX, USA {carlosq, wbthomason, zak, anastasios, kavraki}@rice.edu. This work was supported in part by NSF RI 2008720, NSF ITR 2127309 for the Computing Research Association CIFellows Project, and Rice University Funds.

resented by noisily measured 3D points corresponding to the external surfaces of objects. Point clouds [9] are an example of such a representation. These coordinates are usually computed from depth information from, *e.g.*, a RGB-D camera or LiDAR, which is subject to imperfect measurements and other sources of errors. The distance from the sensor to an object's surface can be modeled as a random variable with Gaussian distribution [10]. This source of sensing uncertainty tends to dominate in the settings we consider; the robot's proprioception (*i.e.*, via joint encoders) is typically much less noisy.

In "normal" motion planning, we seek a collision-free path ρ connecting the initial robot configuration $q_{start} \in \mathcal{Q}$ to a goal region $\mathcal{Q}_{goal} \subset \mathcal{Q}$, *i.e.*, $\rho : [0, 1] \rightarrow \mathcal{Q}_{free}$, $\rho(0) = q_{start}, \rho(1) \in \mathcal{Q}_{goal}$. The goal of motion planning under sensing uncertainty is to find a path that is safe despite imperfect sensing information. More specifically, we want a path whose probability of collision (*i.e.*, *risk of collision*) is no larger than a given threshold Δ. This problem can be formulated as the chance-constrained optimization problem Prob. 1:

$$\begin{aligned} \min_{q_0,\dots,q_T} \quad & f(q_{0:T}) && \text{(Prob. 1)} \\ \text{s.t.} \quad & q_0 = q_{\text{start}},\ q_T \in \mathcal{Q}_{goal}, \\ & q^l \le q_t \le q^u,\ t \in [0, \dots, T], \\ & \Pr\left(\bigwedge_t q_t \in \mathcal{Q}_{free}\right) \ge 1 - \Delta, && \text{(1a)} \end{aligned}$$

where $q_0, \dots, q_T$ are waypoints of a discretized path, f is the objective function (*e.g.*, to encourage smooth, short paths), and q^l and q^u are lower and upper joint limits. (1a) is a chance constraint enforcing that the probability of having no collisions *along the path* remains above the threshold. Unfortunately, this probability cannot be expressed in a tractable form suitable for optimization.

III. Related Work

A. Motion Planning under environmental uncertainty

Collision chance constraints, or constraints on the probability that a robot's trajectory collides with a noisy environment, have been successfully used for safe motion planning under uncertainty by a wide range of work. Chance constraints are typically determinized to keep the planning problem tractable. These deterministic reformulations are then used by either optimization [1–3, 11] or sampling [4, 5, 12]-based motion planners to generate provably safe trajectories. Similarly, we also reformulate and enforce chance constraints to guarantee a desired maximum risk of collision. For example, Blackmore et al. [2, 11] create a disjunctive convex optimization problem that can be solved with branch-and-bound; Luders et al. [4] build a tree-like planner that validates states against the reformulated constraints. Summers [5] uses a similar idea for non-Gaussian uncertainty and moment-based ambiguity sets of distributions. Finally, Dawson et al. [1] propose a differentiable surrogate risk for manipulator robots and convex obstacles under Gaussian translational uncertainty that is guaranteed to never underestimate the true risk, enforced by constraints in a nonlinear program. Many of these methods rely on simplified robot shapes, *e.g.*, point robots [2–5], obstacle shapes, *e.g.*, polyhedral [2, 4, 5] or convex [1, 3], and the noise model, *e.g.*, additive Gaussian noise on obstacle positions [1, 2, 4]. In contrast, our method is designed for high-DoF robots and complex, noisy scenes, where point-robot assumptions are insufficient and strict assumptions on the noise distribution may not hold.

When reformulating chance constraints, most methods allocate equal risk for every waypoint and/or obstacle in the path to make the problem tractable [2–5]. However, this strategy can lead to overly conservative solutions, since robot configurations that are far from noisy obstacles will still be forced to satisfy difficult risk bounds. A few works have considered non-uniform risk allocation, either by formulating multi-stage optimization problems [13], iteratively penalizing and relaxing risky waypoints from previous solutions [7], or using differentiable surrogate risks encoded as variables in a nonlinear optimization problem [1]. Similar to [3, 11], our method enforces joint chance constraints by using the union bound (Boole's inequality) and solving a set of individual chance constraints. However, in our method, the risk bound of individual chance constraints for all obstacle-link pairs are decision variables in our optimization formulation.

Other methods design certificates that verify that a path is safe under a noise model. Berg et al. [14] assess path safety by assuming a linear-quadratic controller with Gaussian uncertainty (LQG-MP). Several candidate paths are generated using a sampling-based planner and the best is chosen for execution. Axelrod et al. [15] certify a path as safe for a given level of risk if the robot's swept volume does not intersect a set of unsafe regions. Park and Manocha [16] design probabilistic collision checkers for non-Gaussian distributions, which they use in an optimization-based planner to encourage safety. Quintero-Peña et al. [6] also handle non-Gaussian distributions by solving a robustly formulated sequential convex programming problem. Dai et al. [7] generate candidate paths, propagate the uncertainty along the path using LQG-MP and estimate the resulting risk of collision via numerical integration. Our proposed approach also generates risk-agnostic candidate paths which it then transforms into safe paths by solving a sequence of convex optimization problems.

B. Implicit Representations and Uncertainty Quantification

Recent machine learning advances have produced efficient implicit neural representations of spatial information, such as Neural Radiance Fields (NeRFs) [17] and Signed Distance Fields [18, 19]. Robotics researchers have used these representations to learn multi-object dynamics [20], as manipulation planning constraints [21], to achieve reactive robot manipulation [22] and to perform visual-only robot navigation [23]. Beyond their compact, efficient storage [23], these representations are advantageous for planning due to their continuous representation of geometry [22–24] and ability to be learned directly from sensor data [25]. Recent work has also investigated quantifying the uncertainty of a learned

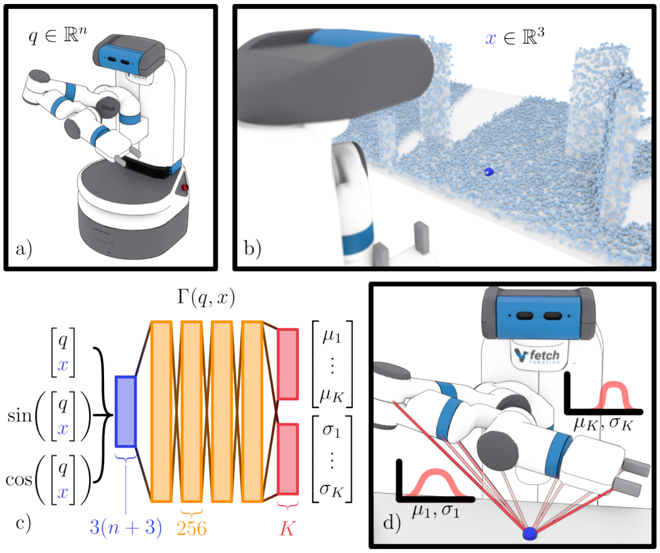
$q \in \mathbb{R}^n$
a)
$x \in \mathbb{R}^3$
b)
$\Gamma(q, x)$
$\begin{bmatrix} q \\ x \end{bmatrix}$
$\sin\left(\begin{bmatrix} q \\ x \end{bmatrix}\right)$
$\cos\left(\begin{bmatrix} q \\ x \end{bmatrix}\right)$
$3(n+3)$
256
K
$\mu_1 \ldots \mu_K$
$\sigma_1 \ldots \sigma_K$
c)
fetch
μ_K, σ_K
μ_1, σ_1
d)

Fig. 2. Our stochastic implicit neural signed distance representation uses **a)** a robot configuration q and **b)** one noisy point x as input. **c)** Inputs go through a positional encoding layer and then through 4 fully connected layers of size 256. Finally, two separate layers of size K output the mean and standard deviation parameters of **d)** each link's distribution modeling the noisy signed distance conditioned on q, x.

model [26–28]. Methods to estimate both aleatoric and epistemic uncertainty have been proposed in the computer vision [8, 29, 30] and reinforcement learning [26–28] literatures. This is important to enable the design of uncertainty-aware algorithms for downstream tasks. For example, Shen et al. [8, 31] propose probabilistic frameworks that attempt to capture uncertainty in a NeRF for synthetic novel view and depth-map estimation. Similarly, our method takes a probabilistic approach to quantifying aleatoric sensing uncertainty. However, our proposed neural representation also fuses kinematic information about a robot with spatial information to produce a robot-configuration-conditioned probabilistic distance model.

IV. Safe Motion Planning with a Stochastic Neural Representation

In this work, we assume that information about the environment is captured through a sensor as noisy 3D points, akin to a point cloud. This noise is aleatoric from the perspective of the planner as it stems from immutable properties of the sensor and is irreducible. We propose to quantify this aleatoric sensing uncertainty through a stochastic implicit neural representation that models noisy signed distances between the environment and the robot geometry. Our neural representation, inspired by [22], captures not only geometric information about the environment (as in work based on NeRFs [20, 23] or SDFs [21, 24, 25]), but also kinematic information about the robot itself, which makes it suitable for motion planning for manipulation. We find safe paths despite sensing errors by using this representation in a novel *hierarchical motion planner*, instead of directly attempting to reformulate and solve Prob. 1. Our planner first finds a candidate path using only the noisy sensed points (without knowledge of their noise), and then uses this candidate path and a user-provided bound on the risk of collision to compute a safe path. The following sections describe our aleatoric sensing representation and planning framework.

A. Stochastic Neural Implicit Signed Distance Representation

Koptev et al. [22] propose an implicit neural representation that models the signed distance between each robot link and arbitrary points in space. The neural representation learns $\Gamma : \mathcal{Q} \times \mathbb{R}^3 \to \mathbb{R}^K$ comprising K related mappings $\Gamma_k : \mathcal{Q} \times \mathbb{R}^3 \to \mathbb{R}$. Each $\Gamma_k(q, x)$ is the minimum distance function for the k-th robot link ($1 \leq k \leq K$), evaluated at the 3D point x when the robot is in configuration q. This representation is useful for motion planning for manipulation due to 1) representing distances to arbitrary points in the workspace without depending on specific geometry and 2) its gradients point away from obstacles in *configuration space*.

Inspired by this representation, we propose to learn a distribution over signed distance functions modeled as the random variable S, such that the distance between each robot link and points in the workspace is modeled as a Gaussian random variable. We want to learn the posterior of S conditioned on a training set $\mathcal{T}$ consisting of a finite collection of robot configurations q_i, 3D points x_i and per-link noisy signed distance values d_i^k, *i.e.*, $\mathcal{T} = \{(\{d_i^k\}_{k=1}^K, q_i, x_i)\}_{i=1}^N$.

We formulate the problem using a Bayesian approach [8] to compute the posterior $\Pr(S|\mathcal{T})$. Note that explicitly computing this posterior is intractable since it would require the computation of the evidence *i.e.*, the marginal density of the observations. Instead, we approximate it using variational inference (VI), where a parametric distribution $\psi_\theta(S)$ approximates the true distribution. The goal of VI is to find the parametric distribution that is closest to the true distribution, measured via their KL divergence [32]. As the KL divergence is not computable because it requires the evidence, VI typically optimizes the evidence lower bound (ELBO) [32]. For our problem, the VI formulation is:

$$\min_{\theta} \underbrace{\mathbb{E}_{\psi_\theta(S)} \log\left(\frac{\psi_\theta(S)}{p(S)}\right)}_{\text{KL-divergence prior}} - \underbrace{\mathbb{E}_{\psi_\theta(S)} \log\left(\Pr\left(\mathcal{T} \mid S\right)\right)}_{\text{Log likelihood}} \tag{2a}$$

where the first term is the KL divergence between ψ_θ and a prior $p(S)$ on the signed distance field, to encourage densities close to the prior, and the second term is the negative training set likelihood over the approximate posterior ψ_θ, which will choose parameters θ that best explain the observed data.

We assume that ψ_θ can be factored as the product of independent Gaussian densities, $\psi_\theta^k(d|q, x)$, representing the distance functions for each robot link k. This assumption simplifies our analysis and allows us to efficiently solve the VI problem, but introduces additional conservatism to our chance-constrained formulation (see Sec. IV-C). We mitigate the impact of this conservatism by dynamically allocating the risk for each link at each timestep while still satisfying the joint risk threshold. These densities are jointly modeled as a neural network, $\Gamma(q, x)$, that outputs the parameters of ψ_θ, $\{\mu_1, \sigma_1, \ldots, \mu_K, \sigma_K\}$ (see Fig. 2 for network architecture). The second term in Eq. (2a) is computed in closed form using

Algorithm 1: Chance-Constrained Hierarchical Motion Planner

```
input : $q_{start}, \mathcal{Q}_{goal}, \Xi, \Delta$
output: $\rho^s$, $\Delta_{T-1}$
1  $\rho^c \leftarrow$ MotionPlan $(q_{start}, \mathcal{Q}_{goal}, \Xi)$;
2  $\Delta_0 \leftarrow \Delta$, $q^s_0 \leftarrow q^c_0$;
3  for $j \leftarrow 0, \ldots, T-1$ do
4      if $(\Delta q, \delta, \gamma) \leftarrow$ CCIKOPT $(q^c_{j+1}, q^s_j, \Delta_j)$ then
5          $q^s_{(j+1)} \leftarrow q^s_j + \Delta q$;
6          $\Delta_{j+1} \leftarrow \Delta_j - \gamma$;
7      else
8          break;
9  if $j < T-1$ then
10     return false;
11 return $[q^s_0, \ldots, q^s_T], \Delta_{T-1}$;
```

the likelihood of the Gaussian distribution. For the first term we assume that (similar to ψ_θ) the prior can be factored as a product of Gaussians, $p^k(d)$ with parameters $\{\mu^p_k, \sigma^p_k, \}$. The KL divergence between these distributions can be computed analytically as:

$$\begin{aligned} \mathrm{KL}\left(\psi_\theta(S)||p\,(S)\right) &= \sum_{k=1}^{K} \sum_{x\in\mathbb{R}^3} \sum_{q\in\mathbb{Q}} \mathrm{KL}\left(\psi^k_\theta(S|x,q)||p^k\,(S)\right) \\ &\approx \sum_{k=1}^{K} \sum_{i} \frac{\sigma^2_{k,i} + (\mu^p_{k,i} - \mu_{k,i})^2}{2{\sigma^p_{k,i}}^2} - \log \sigma_{k,i} \end{aligned}$$

In practice, we use a fixed number of samples to remove the dependency of the approximate posterior on x and q.

B. Chance-Constrained Hierarchical Planning

We propose a hierarchical motion planner to generate safe robot motions, described in Alg. 1. First, an off-the-shelf motion planner [33–37] is used to find a candidate path ρ^c in the noisy sensed environment Ξ (line 1). For each waypoint of ρ^c (line 3), we solve a chance-constrained IK problem (CCIKOPT, line 4) to compute the motion to the *next* waypoint. We use the pose of the robot's end-effector at q^c_{j+1} as a soft constraint for the j-th IK problem, encouraging solutions close to the original path. We accumulate the risk allocated to each waypoint to ensure that it does not exceed the bound Δ for the total path (line 6). Each IK problem is allowed up to the full remaining risk available, and returns an upper bound on the risk allocated to the corresponding waypoint. This method can be seen as using ρ^c as *guidance* for the sequence of IK problems, while flexibly accommodates the allowable risk bounds to compute a safe path, ρ^s. We extend the IK formulation of [22] to the chance-constrained IK setting in Prob. 2:

$$\begin{aligned} \min_{\boldsymbol{\Delta q},\boldsymbol{\delta}} \quad & \boldsymbol{\Delta q}^{\boldsymbol{T}} Q \boldsymbol{\Delta q} + \boldsymbol{\delta}^\top D \boldsymbol{\delta} & \text{(Prob. 2)} \\ \text{s.t.} \quad & q^l \leq q^s_j + \boldsymbol{\Delta q} \leq q^u \\ & \mathrm{FK}(q^s_j) + J^\top(q^s_j)\boldsymbol{\Delta q} = \mathrm{FK}(q^c_{j+1}) + \boldsymbol{\delta} \\ & \Pr\left(\bigwedge_{r,k} -\nabla\Gamma^\top_{k,r}\boldsymbol{\Delta q} \leq \Gamma_{k,r} - r_r\right) \geq 1 - \Delta_j & \text{(3a)} \end{aligned}$$

with decision variables $\boldsymbol{\Delta q}$ and $\boldsymbol{\delta}$. $\boldsymbol{\Delta q}$ corresponds to the robot motion between q^s_j and q^s_{j+1}; $\boldsymbol{\delta}$ is a vector of slack variables that provide flexibility on the pose of the end-effector. We minimize a quadratic function of the decision variables to encourage small motions that end close to the end-effector pose from ρ^c. Constraint (3a) is the joint chance constraint requiring the risk of collision to remain under a given threshold Δ_j for all robot links $1 \leq k \leq K$ and noisy points $1 \leq r \leq R$, modeled as spheres of radius r_r. In its deterministic version [22], these constraints force $\boldsymbol{\Delta q}$ to align with $-\nabla\Gamma_{r,k}$ (the negative gradient of the signed distance with respect to q) when $\Gamma_{k,r} = \Gamma_k(q^s_j, x_r) - r_r$ becomes small, i.e., when the robot gets too close to an obstacle. In our approach, $\Gamma_{k,r}$ are random Gaussian variables, and we enforce that the probability that this constraint is satisfied is above a given threshold. Next we describe our reformulation of the constraint to make the problem tractable.

C. Reformulation of the Chance-Constrained IK Problem

For simplicity, let $\boldsymbol{x}^\top = [\boldsymbol{\Delta q}^\top, \boldsymbol{\delta}^\top]$, $A = \mathrm{diag}([Q,\ D])$, $B = [J^\top(q^s_j),\ -I]$, $c^\top_{k,r} = [-\nabla\Gamma^\top_{k,r},\ 0]$, $g_{k,r} = \Gamma_{k,r} - r_r$, and $b = \mathrm{FK}(q^c_{j+1}) - \mathrm{FK}(q^s_j)$. Note that, here, $\nabla\Gamma_{k,r}$ is the deterministic gradient of the stochastic SDF $\Gamma_{k,r}$, evaluated at q. We also use the following facts:
(4a) $\Pr\left(\bigwedge_i A_i\right) \geq 1 - p \iff \Pr\left(\bigvee_i \bar{A}_i\right) \leq p$,
(4b) $\Pr\left(\bigvee_i A_i\right) \leq p \Longleftarrow \sum_i \Pr\left(A_i\right) \leq p$,
(4c) $\sum_i \Pr\left(A_i\right) \leq p \Longleftarrow \Pr\left(A_i\right) \leq p_i,\ \forall i, \sum_i p_i \leq p$.

We rewrite (3a) in simplified notation, then apply **(4a-c)**:

$$\begin{aligned} & \Pr\left(\bigwedge_r \bigwedge_k c^\top_{k,r}\boldsymbol{x} \leq g_{k,r}\right) \geq 1 - \Delta_j, \\ & \Longleftarrow \sum_r \Pr\left(\bigvee_k c^\top_{k,r}\boldsymbol{x} \geq g_{k,r}\right) \leq \Delta_j, & \textbf{(4a-b)} \\ & \Longleftarrow \sum_k \Pr\left(c^\top_{k,r}\boldsymbol{x} \geq g_{k,r}\right) \leq \boldsymbol{y_r}\ \forall r,\ \sum_r \boldsymbol{y_r} \leq \Delta_j, & \textbf{(4b-c)} \\ & \Longleftarrow \Pr\left(c^\top_{k,r}\boldsymbol{x} \geq g_{k,r}\right) \leq \boldsymbol{\gamma_{k,r}}\ \forall r,k,\ \sum_{k,r} \boldsymbol{\gamma_{k,r}} \leq \Delta_j, & \textbf{(4c)} \\ & = \Pr\left(c^\top_{k,r}\boldsymbol{x} \leq g_{k,r}\right) \geq 1 - \boldsymbol{\gamma_{k,r}}\ \forall r,k,\ \sum_{r,k} \boldsymbol{\gamma_{k,r}} \leq \Delta_j \end{aligned}$$

where $\boldsymbol{\gamma_{k,r}}$ is the risk allocated to link k and point r. By properties of the Gaussian CDF [38], we know $\Pr\left(a^\top b \leq c\right) \geq p \iff a^T b - \mu_c + \sigma_c \phi^{-1}(p) \leq 0$ for $c \sim \mathcal{N}(\mu_c, \sigma^2_c)$, where ϕ^{-1} is the inverse CDF of the standard normal distribution. Thus, we can write the following deterministic reformulation for (3a):

$$c_{k,r}^{\top}\boldsymbol{x} - \mu_{k,r} - r_r + \sigma_{k,r}\phi^{-1}(\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}) \leq 0 \;\forall k, r \tag{5}$$

$$\sum_{k,r}(1 - \bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}) \leq \Delta_j \tag{6}$$

where $\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}} = 1 - \gamma_{\boldsymbol{k},\boldsymbol{r}}$. In order to mitigate the conservatism introduced by assuming independent distance function densities and the use of the union bound we intelligently allocate the risk (complement) thresholds $\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}$ for each waypoint so that configurations and robot links that are closer to noisy obstacles are assigned more risk than those that are far. This is in contrast to manually setting these thresholds *e.g.*, by using uniform allocation [2, 3, 15] which would make most of our problems infeasible. This variable risk feature has been introduced before as risk allocation [1, 7, 13]. To increase the chances of finding feasible paths that satisfy the risk threshold Δ we follow two strategies: **1)** we allow each chance-constrained IK problem to use the maximum amount of residual risk up to that point by accumulating the risk along waypoints of the path and **2)** we ask each IK problem to minimize the amount of used risk. We achieve this by letting $\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}$ be decision variables in our optimization problems at the cost of solving harder optimization problems:

$$\begin{aligned}
\min_{\boldsymbol{x},\bar{\boldsymbol{\gamma}}} \quad & \boldsymbol{x}^{\top} A\boldsymbol{x} + h^{\top}\bar{\boldsymbol{\gamma}} && \text{(Prob. 3)}\\
\text{s.t.} \quad & x_l \leq \boldsymbol{x} \leq x^u \\
& \bar{\gamma}_l \leq \bar{\boldsymbol{\gamma}} \leq \bar{\gamma}^u \\
& B\boldsymbol{x} = b \\
& c_{k,r}^{\top}\boldsymbol{x} - \mu_{k,r} - r_r + \sigma_{k,r}\phi^{-1}(\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}) \quad \leq 0 \;\forall k, r \\
& \sum_{r,k}(1 - \gamma_{\bar{\boldsymbol{k}},\boldsymbol{r}}) \leq \Delta_j
\end{aligned}$$

To solve Prob. 3 we propose two approaches namely **Log PWL** and **NLP**. In the former, we provide a conservative reformulation of (5) by noting that, for $0.5 \leq x < 1$, $\sqrt{\pi/8}\log\left(x/(1-x)\right) \geq \phi^{-1}(x)$, allowing:

$$(5) \Longleftarrow c_{k,r}^{\top}\boldsymbol{x} - \mu_{k,r} + \sigma_{k,r}\sqrt{\frac{\pi}{8}}\log\left(\frac{\bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}}{1 - \bar{\gamma}_{\boldsymbol{k},\boldsymbol{r}}}\right) \leq 0 \tag{7}$$

which requires the risk variables to be in $0 < \gamma_{\boldsymbol{k},\boldsymbol{r}} \leq 0.5$. This restriction is reasonable in our context since we are interested in paths with low collision risk. Using (6) and (7) instead of (3a) allows us to solve the problem using efficient tools that approximate the log using piecewise linear functions. The latter directly solves the problem using tools for nonlinear programming such as interior point methods, with no additional approximations. The linear term on the objective function minimizes the amount of risk allocated to the j-th waypoint. It is not hard to prove that both formulations are convex in the domain of interest, allowing us to *globally* minimize (with respect to Γ and ρ^c) the risk of ρ^s if a solution can be found.

V. Evaluation and Results

We evaluate our proposed approach on a $n = 8$ DoF Fetch robot with $K = 11$ links, corresponding to those in the kinematic chain of its end effector (including the torso and fingers). We use PyBullet [39] for collision checking, PyTorch [40] for neural network training, OMPL's Python bindings [41] for planning and Gurobi [42] and IPOPT [43] as our optimizers. All experiments were conducted on an Intel i7-12700K CPU and a RTX2080Ti GPU.

A. Implicit Neural Representation

We parameterize our implicit stochastic distance model as a feed-forward neural network (Fig. 2c). For a n-DoF robot, the network takes an input tensor of size $3 * (n + 3)$ comprising the n values of the robot configuration concatenated with the 3 coordinates of the environment point, as well as the sine and cosine of these values. These trigonometric components serve as a form of positional encoding similar to that used in standard neural radiance fields [17]. The network has a shared core of four fully connected 256-wide layers with rectified linear unit (ReLU) activation. For a robot with K links, the output of these layers is used (independently) with one additional fully connected layer of size $256 \times K$ to predict the mean distance from the environment point to each link's geometry, as well as with another fully connected layer of size $256 \times K$ and a softplus layer of size K to predict the standard deviation of these distances.

We generate a dataset of noisy distance samples from a simulated sensor to train the distance model. Similarly to Koptev et al. [22], we sample a set of robot configurations (Q) uniformly at random. For each configuration, we sample a set of environment points uniformly at random (P_R), a set of environment points *near* to each link (P_N) and a set of environment points *inside* each link (P_I). We compute the true shortest distance between each point and link using PyBullet. We then simulate a set of noisy sensor measurements (NS) with mean at the true distance for each environment point and a fixed standard deviation (σ). In our experiments, $|Q| = 3000$, $|P_R| = 500$, $|P_N| = K * 10$, $|P_I| = K * 20$, $|NS| = 50$, and $\sigma = 2\,\mathrm{cm}$. This results in a total of 2.49 million sampled points, each of which has 50 noisy distance samples. Empirically, this dataset is roughly balanced between points in collision and points in free space.

We train the model on the collected dataset for 500 epochs with an Adam [44] optimizer, learning rate of 1×10^{-4}, and batch size of 512. We verify its performance by predicting distance distributions between robot links and a set of randomly generated 3D points from the waypoints of 1000 discretized paths. The gripper link shows an average error of $1\,\mathrm{cm}$ for mean and $3.7\,\mathrm{mm}$ for standard deviation while the elbow attains $0.7\,\mathrm{mm}$ and $0.3\,\mathrm{mm}$, respectively. Fig. 3 shows the predicted and true distribution parameters for one path, one randomly selected point and these two robot links.

B. Safe Motion Planning with Implicit Neural Representation

We evaluate our proposed approach on a set of simulated tabletop manipulation problems generated using MotionBenchMaker [45]. The Fetch robot needs to plan to grasp an object, avoiding collisions with the table and obstacles upon it (Fig. 1). We create 50 problems by randomly

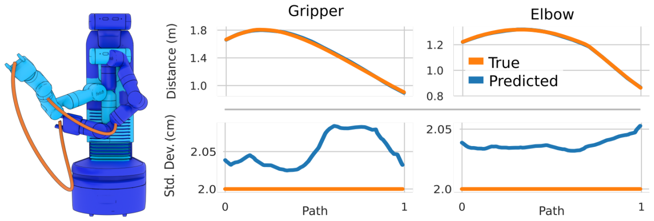

Fig. 3. Comparison between true and predicted probability distribution parameters (top: mean; bottom: standard deviation) for the distance between robot links and a randomly generated point in space for the Gripper (left) and Elbow (right) links along a path of 500 waypoints. Note that (despite the visual gap) the error in standard deviation is small, below 1 mm.

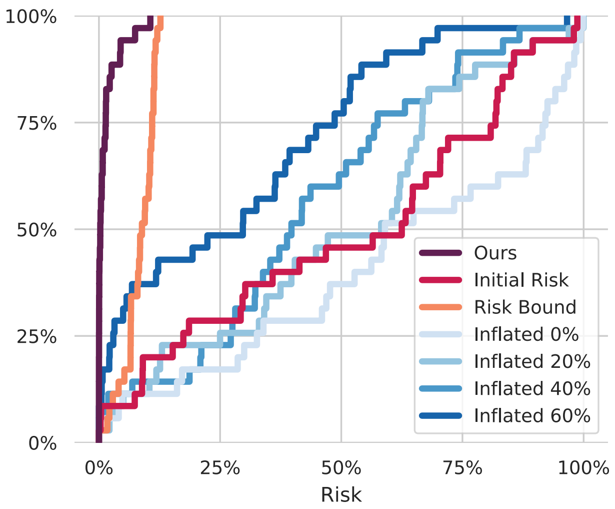

Fig. 4. Estimated CDF of the risk accumulated by each method.

Method	EE Disp. (m)	Path Length (rad.)	Path Risk (%)	Planning Time (s)
Inflated 0%	2.32 ± 0.82	7.91 ± 2.53	60 ± 33	3.51 ± 10.42
Inflated 20%	2.39 ± 0.76	8.34 ± 2.74	48 ± 29	3.61 ± 9.55
Inflated 40%	2.33 ± 0.79	8.27 ± 2.86	42 ± 26	5.85 ± 13.73
Inflated 60%	2.38 ± 0.85	8.57 ± 3.18	28 ± 25	5.69 ± 12.95
Proposed (Log PWL)	2.17 ± 0.62	6.76 ± 2.27	0.95 ± 1.82	31.31 ± 30.07
Proposed (NLP)	2.21 ± 0.47	7.12 ± 1.86	0.03 ± 0.09	67.73 ± 40.31

Table I. Path quality metric mean and standard deviation for each method.

perturbing the positions (± 2.5cm in x, y, z) and orientations ($\pm 15°$) of the objects of a nominal scene and the relative pose of the robot's base (± 10cm in x, y, z and $\pm 90°$) with respect to the table. The environment is represented as a point cloud-like set of noisy 3D spheres of different radii that covers the (unknown to the planner) collision geometries of all objects. We assume that the table's geometry is noise-free while the objects on top are noisily sensed, per Sec. V-A. Note that these problems were designed by Chamzas et al. [45] to be challenging and "realistic" from the motion planning perspective and require the robot to plan long, elaborate paths that need to avoid the table and then dodge collisions with the objects on top.

We compare the performance and safety of our approach with a commonly used baseline: inflating the environment's geometry to encourage the computation of paths that maintain larger clearance and have therefore less chances of colliding. We inflate each sphere by increasing its radius by $20\%, 40\%$, or 60%. We also include results of 0% inflation as a baseline to show the performance of a planner that is unaware of the sensing uncertainty. Motion plans for all baselines, as well as the candidate paths used by our method, are computed using RRT-Connect [35], with the environment represented as a union of spheres centered at each sensed workspace point.

We estimate the risk of collision for each computed path using Monte-Carlo sampling with $20,000$ samples, where each sample draws sphere poses from the noisy sensed distribution. For our method, we also show the guaranteed pathwise risk bound (Risk Bound) and estimated risk of collision of the candidate path before optimization (Initial Risk). All problems have a maximum number of 15 attempts to find any valid (*i.e.*, collision-free with respect to the inflated obstacles, for the baselines) plan. The results are shown in Fig. 4.

We note that the uncertainty-unaware planner produces paths with the highly variable risk of collision (an average of 60%), which is likely unacceptable for safety-critical applications. For higher parameter values of the inflated baseline, the estimated risk of collision decreases as expected due to a larger $\mathcal{Q}$-space obstacle region that encourages larger clearance with the true geometry. However, there is no clear relation between the inflation increase and the drop in risk which makes the baselines difficult to tune when a desired level of risk is required (see also Table I). Additionally, we note that success rate (not shown here) for the baseline methods started dropping significantly as the inflation ratio increased, suggesting a potential limit on minimum risk that they can attain for these problems. We give our planner a maximum allowable risk bound of 10% and ask it to return the minimum risk for each waypoint. Our proposed approach can compute paths with significantly lower risk for most problems, starting from risky candidate paths.

For each problem and method we also compute the path length and end-effector displacement as path quality metrics, as well as the time taken by the planner. The results are summarized in Table I. The table shows mean and standard deviation for each method over all 50 problems. The large values of path length and end-effector displacement are evidence of the high complexity of the computed paths due to the challenging motion planning problems. Our method finds paths with lowest end-effector displacement and path length, which is the result of the minimization of motion displacement in our planner. However, our method has planning times that are larger than the baselines. This is due to the large number of link-point pairs at each waypoint along the path—often hundreds or thousands—that create hard optimization problems. Despite this, it is noteworthy that our method can find paths with the lowest collision risk among the baselines without sacrificing path quality. It is noteworthy that our method that uses an NLP solver is capable of finding paths with much lower risk than the one using a the log approximation at the cost of increased planning time.

VI. Concluding Remarks

This paper presents a novel approach to planning under sensing uncertainty for high DoF robots that reliably computes safe paths without strong assumptions on the true environment geometry. Our planner relies on an implicit neural representation trained to capture aleatoric uncertainty arising from the robot's sensor. Our representation does not place assumptions on the environment but instead directly approximates signed distance distributions between the robot and points in space, conditioned on robot configurations. We further show how this representation can be integrated with a hierarchical planner to compute paths with guaranteed bounds on the probability of collision (up to the quality of the model). We have experimentally validated the merits of our approach on challenging, realistic manipulation motion planning problems to show that our method is capable of finding safe paths despite sensing uncertainty without reducing path quality. As future work we will investigate how to further reduce the need for conservative over-approximations in our approach, since this will allow us to solve more tightly constrained motion planning problems, such as robots reaching into shelves and tight boxes. We will also seek to reduce the time taken by our method, in part by applying intelligent constraint subset selection heuristics [46] to simplify the optimization problems, as well as by applying specialized loss functions for learning SDFs, such as enforcing the Eikonal equation. Finally, we will further investigate the need to consider the epistemic uncertainty coming from our neural representation for planning, a problem that has recently gained much attention in machine learning [28].

References

[1] C. Dawson, A. Jasour, A. Hofmann, and B. Williams. "Provably Safe Trajectory Optimization in the Presence of Uncertain Convex Obstacles". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 6237–6244.

[2] L. Blackmore, M. Ono, and B. C. Williams. "Chance-Constrained Optimal Path Planning With Obstacles". In: *IEEE Transactions on Robotics* 27.6 (2011), pp. 1080–1094.

[3] T. Lew, R. Bonalli, and M. Pavone. "Chance-Constrained Sequential Convex Programming for Robust Trajectory Optimization". In: *European Control Conference*. In Press. St. Petersburg, Russia, May 2020.

[4] B. Luders, M. Kothari, and J. How. "Chance constrained RRT for probabilistic robustness to environmental uncertainty". In: *AIAA guidance, navigation, and control conference*. 2010, p. 8160.

[5] T. Summers. "Distributionally Robust Sampling-Based Motion Planning Under Uncertainty". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 6518–6523.

[6] C. Quintero-Peña, A. Kyrillidis, and L. E. Kavraki. "Robust Optimization-based Motion Planning for high-DOF Robots under Sensing Uncertainty". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 9724–9730.

[7] S. Dai, S. Schaffert, A. Jasour, A. Hofmann, and B. Williams. "Chance Constrained Motion Planning for High-Dimensional Robots". In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. 2019.

[8] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer. "Stochastic Neural Radiance Fields: Quantifying Uncertainty in Implicit 3D Representations". In: *2021 International Conference on 3D Vision (3DV)*. 2021, pp. 972–981.

[9] A. Kuntz, C. Bowen, and R. Alterovitz. "Fast Anytime Motion Planning in Point Clouds by Interleaving Sampling and Interior Point Optimization". In: *Robotics Research*. Ed. by N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti. Cham: Springer International Publishing, 2020, pp. 929–945.

[10] K. Khoshelham and S. O. Elberink. "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications". In: *Sensors* 12.2 (2012), pp. 1437–1454.

[11] L. Blackmore, H. Li, and B. Williams. "A probabilistic approach to optimal robust path planning with obstacles". In: *2006 American Control Conference*. June 2006.

[12] K. Ekenberg, V. Renganathan, and B. Olofsson. "Distributionally Robust RRT with Risk Allocation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 12693–12699.

[13] M. Ono and B. C. Williams. "Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a Joint Chance Constraint". In: *47th IEEE Conference on Decision and Control (CDC)*. 2008, pp. 427–3432.

[14] J. V. D. Berg, P. Abbeel, and K. Goldberg. "LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information". In: *Proceedings of Robotics: Science and Systems*. Zaragoza, Spain, June 2010.

[15] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez. "Provably safe robot navigation with obstacle uncertainty". In: *The International Journal of Robotics Research* 37.13-14 (2018), pp. 1760–1774.

[16] J. S. Park and D. Manocha. "Efficient probabilistic collision detection for non-gaussian noise distributions". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1024–1031.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.

[18] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[19] P. Liu, K. Zhang, D. Tateo, S. Jauhri, J. Peters, and G. Chalvatzaki. "Regularized Deep Signed Distance Fields for Reactive Motion Generation". In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022). IEEE, 2022.

[20] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. "Learning Multi-Object Dynamics with Compositional Neural Radiance Fields". In: *Proceedings of The 6th Conference on Robot Learning*. Vol. 205. Proceedings of Machine Learning Research. PMLR, Dec. 2023, pp. 1755–1768.

[21] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake. "Learning Models as Functionals of Signed-Distance Fields for Manipulation Planning". In: *Proc. of the Annual Conf. on Robot Learning (CoRL)*. 2021.

[22] M. Koptev, N. Figueroa, and A. Billard. "Neural Joint Space Implicit Signed Distance Functions for Reactive Robot Manipulator Control". In: *IEEE Robotics and Automation Letters* 8.2 (2023), pp. 480–487.

[23] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. "Vision-Only Robot Navigation in a Neural Radiance World". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4606–4613.

[24] M. Kurenkov, A. Potapov, A. Savinykh, E. Yudin, E. Kruzhkov, P. Karpyshev, and D. Tsetserukou. "NFOMP: Neural Field for Optimal Motion Planner of Differential Drive Robots With Nonholonomic Constraints". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10991–10998.

[25] G. S. Camps, R. Dyro, M. Pavone, and M. Schwager. "Learning Deep SDF Maps Online for Robot Navigation and Exploration". In: *arXiv preprint arXiv:2207.10782* (2022).

[26] W. R. Clements, B.-M. Robaglia, B. Van Delft, R. B. Slaoui, and S. Toth. "Estimating Risk and Uncertainty in Deep Reinforcement Learning". In: *arXiv preprint arXiv:1905.09638* (2019).

[27] S. Lahlou, M. Jain, H. Nekoei, V. I. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio. *DEUP: Direct Epistemic Uncertainty Prediction*. 2023.

[28] A. Acharya, R. Russell, and N. R. Ahmed. "Learning to Forecast Aleatoric and Epistemic Uncertainties over Long Horizon Trajectories". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 12751–12757.

[29] A. Kendall and Y. Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 5580–5590.

[30] F. Vasconcelos, B. He, N. M. Singh, and Y. W. Teh. "UncertaINR: Uncertainty Quantification of End-to-End Implicit Neural Representations for Computed Tomography". In: *Transactions on Machine Learning Research* (2023).

[31] J. Shen, A. Agudo, F. Moreno-Noguer, and A. Ruiz. "Conditional-Flow NeRF: Accurate 3D Modelling With Reliable Uncertainty Quantification". In: *Computer Vision – ECCV 2022: 17th European*

Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III. Tel Aviv, Israel: Springer-Verlag, 2022, pp. 540–557.

[32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (Apr. 2017), pp. 859–877.

[33] L. E. Kavraki, P. Svestka, J. .-. Latombe, and M. H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[34] S. M. LaValle, J. J. Kuffner, and Jr. "Rapidly-Exploring Random Trees: Progress and Prospects". In: *Algorithmic and Computational Robotics: New Directions*. 2000, pp. 293–308.

[35] J. J. Kuffner and S. M. LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. Vol. 2. 2000, 995–1001 vol.2.

[36] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. "CHOMP: Covariant Hamiltonian optimization for motion planning". In: *The International Journal of Robotics Research* 32.9-10 (2013), pp. 1164–1193.

[37] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. "Motion planning with sequential convex optimization and convex collision checking". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1251–1270.

[38] A. Prékopa. *Stochastic Programming*. Springer Dordrecht, 1995.

[39] E. Coumans and Y. Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. `http://pybullet.org`. 2016–2021.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[41] I. A. Sucan, M. Moll, and L. E. Kavraki. "The Open Motion Planning Library". In: *IEEE Robot. Autom. Magazine* 19.4 (2012), pp. 72–82.

[42] L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2022.

[43] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* (2006).

[44] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[45] C. Chamzas, C. Quintero-Peña, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki. "MotionBenchMaker: A Tool to Generate and Benchmark Motion Planning Datasets". In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 882–889.

[46] K. Hauser. "Semi-infinite programming for trajectory optimization with non-convex obstacles". In: *The International Journal of Robotics Research* 40.10-11 (2021), pp. 1106–1122.