

Differentiable Contact Dynamics for Stable Object Placement Under Geometric Uncertainties

Linfeng Li¹, Gang Yang¹, Lin Shao¹ and David Hsu^{1,2}

Abstract—From serving a cup of coffee to positioning mechanical parts during assembly, stable object placement is a crucial skill for future robots. It becomes particularly challenging under geometric uncertainties, *e.g.*, when the object pose or shape is not known accurately. This work leverages a *differentiable simulation model* of contact dynamics to tackle this challenge. We derive a novel gradient that relates force-torque sensor readings to geometric uncertainties, thus enabling uncertainty estimation by minimizing discrepancies between sensor data and model predictions via gradient descent. Gradient-based methods are sensitive to initialization. To mitigate this effect, we maintain a *belief* over multiple estimates and choose the robot action based on the current belief at each timestep. In experiments on a Franka robot arm, our method achieved promising results on multiple objects under various geometric uncertainties, including the in-hand pose uncertainty of a grasped object, the object shape uncertainty, and the environment uncertainty.

Index Terms—Manipulation Planning, Contact Modeling

I. INTRODUCTION

FUTURE robots should play an integral role in aiding humans with everyday tasks. A key capability is to stably place objects in different environments. Such tasks, ranging from serving dishes on a table to carefully rearranging delicate items, might appear simple but are challenging due to the required accuracy. For instance, to avoid spills when serving a full cup of coffee, the robot must place the cup precisely on the saucer without dropping or collision: inaccuracy in a couple of millimeters may cause spilling (Fig. 1). This challenge extends beyond mere motion control, as visual or depth sensors have errors ranging from a few millimeters to centimeters [1], and we rarely know the exact geometries of the object and the environment. Accounting for such geometric uncertainties is critical for robust robotic manipulation.

Stable placement is one instance of contact-rich manipulation, where geometric uncertainty has long been a challenge [2]; exploring stable placement therefore offers insights into this challenge. In contact-rich manipulation, objects interact through the establishment or cessation of contact, leading to hybrid dynamics where each contact mode dictates a unique

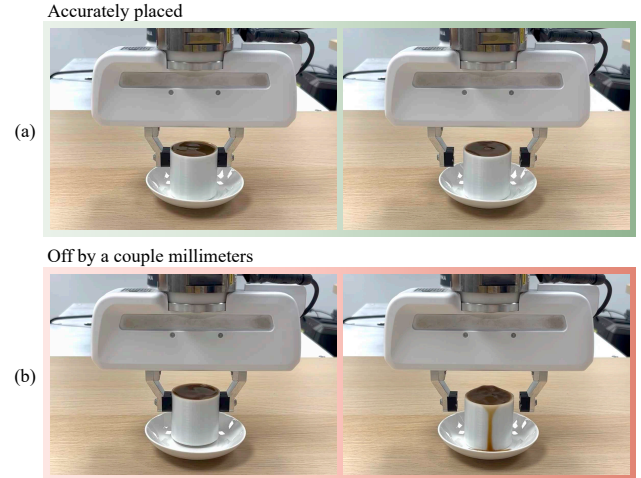


Fig. 1. Place a full cup of coffee on a saucer (left) and then release it (right). (a) When the coffee cup is stably placed, *i.e.*, the bottom surface of the cup is well aligned the saucer, there is no spill after release. (b) If the gripper releases right after a sensed contact without proper estimation of geometric uncertainty, the coffee is spilled.

smooth dynamics. For example, the dynamics of placing a coffee cup changes upon making contact with a saucer, with these dynamics’ boundaries shaped by the objects’ geometries, such as the saucer’s shape. Because of the hybrid dynamics, contact-rich manipulation is sensitive to geometric uncertainty, where “small changes in the geometry of parts can have a significant impact on fine-motion strategies [3].”

Differentiable simulation offers a promising approach to robotic manipulation under uncertainty, as it enables parameter estimation by aligning simulated and measured sensor readings through gradient-based optimization [4]–[6]. However, existing differentiable simulators do not provide gradients of contact wrenches with respect to geometric parameters, and therefore cannot estimate uncertain geometries.

We derive a novel gradient that relates the predicted force-torque readings to the geometric parameters, a feature underexplored in existing differentiable rigid-body simulators. This gradient enables the estimation of uncertain geometry by aligning simulated and measured force-torque data through gradient descent. We implemented this feature on top of the Jade simulator [7], whose generality allows us to address a broad range of geometric uncertainties. We verify the efficacy of the derived gradient using the stable object placement task. To mitigate the sensitivity of gradient-based methods to initialization, we maintain a belief over multiple geometric parameters. At each timestep, we update the belief using the

Manuscript received: July 17, 2025; Revised: October 25, 2025; Accepted: November 22, 2025.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers’ comments. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-014).

¹School of Computing, National University of Singapore, Singapore. linfeng@comp.nus.edu.sg, ygang@u.nus.edu, linshao@nus.edu.sg, dyhsu@comp.nus.edu.sg.

²Smart Systems Institute, National University of Singapore, Singapore.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

recent history, and choose the robot action based on the current belief. We evaluated our method with a Franka Research 3 robot arm under various geometric uncertainties, including in-hand pose uncertainty of the grasped object, the object’s shape uncertainty, and the environment’s shape uncertainty. Our method achieved promising results, compared with a trigger-based heuristic policy and a particle filter–based policy. Finally, we demonstrate our method’s practicality by using it to place a full cup of coffee.

This paper primarily focuses on the geometric uncertainty. Other forms of uncertainty, such as sensor inaccuracies or control imprecision, are addressed through calibration.

II. RELATED WORK

A. Robotic Stable Placement

There are various studies on robotic stable placement with haptic feedback [8]–[11]. However, existing work often assumes a specific type of geometric uncertainty: the object will be placed on a large flat surface [8], [9], the initial contact between the object and the surface has to be a point contact with upward contact normal [10], the test cases are similar to the training data [11], *etc.* Our method applies to a broader range of geometric uncertainties, including the uncertainty in the grasped object’s in-hand pose, the object’s shape, and the environment’s shape. These three types cover the geometric uncertainties considered in prior works, making our approach more general and adaptable.

B. Manipulation Under Geometric Uncertainty

Geometric uncertainty has long been a challenge in manipulation [2]. For a particular setup, a manipulation policy can be manually crafted [9], [12], [13] or learned [11], [14]–[17]; nevertheless, their generalizations are limited by the data diversity or the assumptions that guide the policy-crafting. For example, in a learning-based approach for stable placement [11], the object’s bottom surface and the environmental support surface are assumed to be flat and parallel. Our approach alleviates these restrictions by following the model-based approach.

Model-based methods hold the promise of generalizing to different setups. For simpler tasks, a robust sequence of actions can be planned to achieve the goal under all possible variations of the uncertain geometry [3], [18]–[21]. Alternatively, the problem can be modeled as a partially observable Markov decision process [22]–[24]. However, these methods only utilize binary haptic sensor feedback (touch *vs.* no-touch) or simple termination conditions (*e.g.*, *y*-force larger than a threshold); this limits their applicability to more complex tasks.

Many prior works estimate uncertain geometry using pre-defined probing procedures, where robot actions follow constrained motion patterns, such as following a fixed trajectory or choosing a location on a plane to move downwards until a force threshold is met [25]–[35]. These probing procedures are tailored to specific types of uncertainties, limiting their generalization, and the quality of estimation heavily depends on their design.

By using a general differentiable contact dynamics model, we estimate various geometric uncertainties in a unified way:

minimizing the discrepancy between the model prediction and the multi-dimensional continuous-value haptic sensor readings.

C. Differentiable Contact Dynamics

Differentiable physical models have been shown to be useful in estimating the uncertain parameters [4]–[6]; the discrepancy between the model prediction and the observation is minimized using the gradients. We take a similar approach in the context of rigid-body contact dynamics.

To estimate the uncertain geometry, we need the gradients of contact forces with respect to the uncertain geometric parameters. Prior differentiable rigid-body simulators do not provide this capability [36]–[43]. Lee *et al.* propose a contact feature that is differentiable w.r.t. the object pose [34]; a differentiable contact feature is an important intermediary step, but the feature itself does not suffice to predict the contact force. We derived and implemented the gradients of contact forces with respect to the uncertain geometric parameters on top of the Jade simulator [7].

III. PROBLEM FORMULATION

Given an initial guess of the uncertain geometry, our method takes the robot’s proprioception and the 6-axis force-torque (FT) sensor feedback as input; it outputs the robot action to approach the goal configuration to stably release the target object.

We consider three entities in our model: the end-effector frame, the object to be placed, and the environment. The object and the environment are represented as triangle meshes. We model them using a differentiable simulator

$$\mathbf{x}_{t+1}, \mathbf{y}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t; \boldsymbol{\theta}), \quad (1)$$

where $\mathbf{x}_t \in \text{SE}(3) \times \mathbb{R}^6$ is the robot state consisting of the end-effector’s pose and twist, $\mathbf{u}_t \in \text{SE}(3)$ is the robot action of reference pose, which is converted into a control wrench by a lower-level impedance controller, and $\mathbf{y}_t \in \mathbb{R}^6$ is the predicted contact wrench between the object and the environment expressed in the end-effector frame. We always use the subscript t in $(\cdot)_t$ to denote the timestep. The uncertain geometry is represented by the parameter $\boldsymbol{\theta} \in \mathbb{R}^P$, which includes parameters of i) the relative pose between the object and the end-effector, ii) the mesh coordinates of the object, and iii) the mesh coordinates of the environment. The users have the flexibility to define the elements of $\boldsymbol{\theta}$ based on their specific placement task requirements. The dynamics function f is differentiable with respect to \mathbf{x}_t , \mathbf{u}_t , and $\boldsymbol{\theta}$.

After the robot grasps the target object, we start with an initial guess $\boldsymbol{\theta}^{\text{init}} \in \mathbb{R}^P$ of the unknown groundtruth geometry $\boldsymbol{\theta}^{\text{gt}} \in \mathbb{R}^P$. We assume $\boldsymbol{\theta}^{\text{gt}}$ to be a constant. For example, this assumption implies a non-slip grasp, ensuring that the uncertain in-hand pose of the object does not change during execution.

To stably place the grasped object, the robot needs to release the object at a goal

$$\mathbf{x}^{\text{goal}} \in \mathcal{G}(\boldsymbol{\theta}^{\text{gt}}) \quad (2)$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

determined by the unknown groundtruth θ^{gt} , where \mathcal{G} maps a geometric parameter to a set of goal states. For example, to place a cup on a table, $\mathcal{G}(\theta^{\text{gt}})$ includes all configurations where bottoms of the cup are aligned with the table surface.

We assume a deterministic policy

$$\mathbf{u}_t = \pi(\mathbf{x}_t, \theta) \quad (3)$$

that takes the robot state and geometric parameter as inputs and outputs the robot action \mathbf{u}_t . When the groundtruth geometry is used, this policy drives the robot to the stable release configuration, *i.e.*, $\lim_{t \rightarrow \infty} \mathbf{x}_t \in \mathcal{G}(\theta^{\text{gt}})$ for the closed-loop dynamics $\mathbf{x}_{t+1}, \mathbf{y}_{t+1} = f(\mathbf{x}_t, \pi(\mathbf{x}_t, \theta^{\text{gt}}); \theta^{\text{gt}})$. This assumption reflects that when the geometry is known, we can leverage existing methods for stable placement. In general, this policy can be implemented using motion planning methods [44]–[46], or learned conditioned on the groundtruth geometry [47]. In this work, we use heuristic policies that interpolate from the current state to the goal.

Our method relies on the FT sensor to measure the contact wrench between the object and the environment $\mathbf{y}_t^{\text{gt}} \in \mathbb{R}^6$ at timestep t . By minimizing the error between the prediction \mathbf{y}_t and the measurement \mathbf{y}_t^{gt} through gradient descent, we expect the estimated geometric parameter $\hat{\theta}$ to approach θ^{gt} , and $\pi(\mathbf{x}_t, \hat{\theta})$ drives the robot to the stable release configuration.

We assume uncertainties other than the geometric ones are compensated for by calibration.

IV. GRADIENTS OF GEOMETRIC PARAMETERS

To differentiate the error between the model prediction \mathbf{y}_t and the measurement \mathbf{y}_t^{gt} with respect to the geometric parameter θ , we require the gradient $\partial \mathbf{y}_{(\cdot)} / \partial \theta$ from the simulator (1). However, this feature is absent in most existing differentiable rigid-body simulators. To address this limitation, we extend the Jade simulator to support differentiation with respect to geometric parameters [7]. Our key design choice is to model contact features not only as functions of configuration but also as functions of the geometric parameter. In this section, we detail the derivation and implementation of this gradient.

A. Robot States and Actions

The robot state $\mathbf{x}_t = \langle \mathbf{q}_t, \dot{\mathbf{q}}_t \rangle$ consists of the pose $\mathbf{q}_t \in \text{SE}(3)$ and twist $\dot{\mathbf{q}}_t \in \mathbb{R}^6$ of the end-effector. We model the end-effector and grasped object as a floating rigid body controlled by a wrench $\boldsymbol{\tau}_t \in \mathbb{R}^6$, which is computed from the robot action \mathbf{u}_t . For example, if \mathbf{u}_t is the reference pose of a proportional controller [48], the control wrench is given as $\boldsymbol{\tau}_t = \text{ControllerGain} \cdot \text{PoseError}(\mathbf{u}_t, \mathbf{q}_t)$, where PoseError computes the 6-dimensional pose difference vector between two poses [49], [50].

B. Collision-Free Forward Dynamics

We start with the simple case where no contact change occurs during each forward simulation step. We represent contacts as points between object pairs. Given \mathbf{q} and θ , the collision engine detects all contact features, each as a tuple with respect to objects A and B

$$\langle \mathbf{p}^A(\mathbf{q}; \theta), \mathbf{p}^B(\mathbf{q}; \theta), \mathbf{v}^A, \mathbf{v}^B, \mathbf{n}(\mathbf{q}; \theta) \rangle_i, \quad (4)$$

where the subscript i denotes the i -th contact feature, $\mathbf{p}^A, \mathbf{p}^B \in \mathbb{R}^3$ are the coordinates of witness points on objects A and B , $\mathbf{v}^A, \mathbf{v}^B \in \mathbb{R}^3$ are the velocities of witness points, and $\mathbf{n} \in \text{S}(3)$ is the unit normal vector of contact surface pointing from object A to object B . When the collision happens, we have $\mathbf{p}^A = \mathbf{p}^B$, meaning the witness points coincide and become contact points.

Given the detected C contact features (4), the simulator predicts the position \mathbf{q}' , velocity $\dot{\mathbf{q}}'$ and contact impulse $\boldsymbol{\lambda} \in \mathbb{R}^{3C}$ after applying the control force $\boldsymbol{\tau}$ for the time duration Δ using the forward dynamics

$$\mathbf{q}', \dot{\mathbf{q}}', \boldsymbol{\lambda} = \text{FD}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \Delta; \theta) \quad (5)$$

under the assumption that the contact features do not change during Δ , *i.e.*, there is no collision during Δ . Let Δ_s be the duration between two timesteps t and $t+1$, we have $\mathbf{q}_{t+1}, \dot{\mathbf{q}}_{t+1}, \boldsymbol{\lambda}_t = \text{FD}(\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t, \Delta_s; \theta)$, and the predicted contact wrench is $\mathbf{y}_{t+1} = \mathbf{J}^\top(\mathbf{q}_t; \theta) \boldsymbol{\lambda}_t$, where $\mathbf{J} \in \mathbb{R}^{6 \times 3C}$ is the contact Jacobian that depends on (4).

We solve for $\boldsymbol{\lambda}$ in (5) via a linear complementarity problem $\boldsymbol{\lambda} = \text{LCP}(\mathbf{A}, \mathbf{b})$: finding $\langle \boldsymbol{\lambda}, \mathbf{a} \rangle$ such that $\boldsymbol{\lambda} \geq 0$, $\mathbf{a} \geq 0$, $\boldsymbol{\lambda}^\top \mathbf{a} = 0$, and $\mathbf{a} = \mathbf{A}\boldsymbol{\lambda} + \mathbf{b}$ [7], [41], [51]. Here we highlight that the LCP parameters $\mathbf{A} = \mathbf{A}(\mathbf{J})$ and $\mathbf{b} = \mathbf{b}(\mathbf{J}, \Delta)$ depend on \mathbf{J} and Δ .

However, the assumption of no contact change during the forward simulation step (5) rarely holds. We alleviate this assumption in the next subsection.

C. Collision Between Two Timesteps

If there is a collision between timesteps t and $t+1$ that changes the contact features (4), we can detect the exact time-of-impact $toi \in [0, 1)$ as a ratio of Δ_s when two rigid bodies collide. The general forward dynamics

$$\mathbf{q}_{t+1}, \dot{\mathbf{q}}_{t+1} = \text{FD}_{\text{full}}(\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t, \Delta_s; \theta) \quad (6)$$

might involve multiple calls of collision-free forward dynamics [7]. In the case of 2 calls, we have

$$\begin{aligned} \mathbf{q}_{t+toi}, \dot{\mathbf{q}}_{t+toi}, \boldsymbol{\lambda}_t &= \text{FD}(\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t, toi\Delta_s; \theta), \\ \mathbf{q}_{t+1}, \dot{\mathbf{q}}_{t+1}, \boldsymbol{\lambda}_{t+toi} &= \\ &\text{FD}(\mathbf{q}_{t+toi}, \dot{\mathbf{q}}_{t+toi}, \boldsymbol{\tau}_t, (1-toi)\Delta_s; \theta). \end{aligned}$$

We have so far defined the forward dynamics of the simulator (1). Prior works [7], [41] have implemented the gradients $\partial \text{FD} / \partial \mathbf{q}$, $\partial \text{FD} / \partial \dot{\mathbf{q}}$, $\partial \text{FD} / \partial \boldsymbol{\tau}$, and $\partial \text{FD} / \partial \Delta$. However, our method needs $\partial \mathbf{y}_{(\cdot)} / \partial \theta$ from the simulator (1). We elaborate on its derivation in the next subsection.

D. Differentiating Wrench wrt Geometric Parameters

Since in our problem setup, \mathbf{q}_t is the pose of the robot's end-effector, the contact wrench impulse between the grasped object and the environment is given as

$$\mathbf{y}_{t+1} = \begin{cases} \mathbf{J}_{t+toi}^\top \boldsymbol{\lambda}_{t+toi} + \mathbf{J}_t^\top \boldsymbol{\lambda}_t & \text{collision,} \\ \mathbf{J}_t^\top \boldsymbol{\lambda}_t & \text{no collision.} \end{cases} \quad (7)$$

By applying the chain rule, the gradient of the geometric parameter $\partial \mathbf{y}_{t+1} / \partial \theta$ involves the differentiation of contact impulses $\partial \boldsymbol{\lambda}_{(\cdot)} / \partial \theta$ and the Jacobian $\partial \mathbf{J}_{(\cdot)}^\top / \partial \theta$, where $\partial \boldsymbol{\lambda}_{(\cdot)} / \partial \theta$ involves $\partial \mathbf{J}_{(\cdot)}^\top / \partial \theta$, $\partial \mathbf{J}_{(\cdot)} / \partial \theta$ and $\partial toi / \partial \theta$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

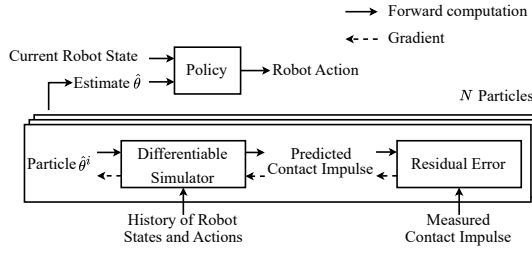


Fig. 2. Our overall pipeline. At every timestep, we update the estimated geometry and plan a robot action from the estimation.

To derive $\partial toi / \partial \theta$, we consider the dynamics at timestep $t + toi$ as discussed in [7]. Suppose a tiny geometry variant $\delta \theta$ causes coordinate variants $\delta \mathbf{p}^A, \delta \mathbf{p}^B$ and makes the contact points separate along the normal direction, it'll take

$$\delta toi = \frac{(\delta \mathbf{p}^B - \delta \mathbf{p}^A) \cdot \mathbf{n}}{(\mathbf{v}^B - \mathbf{v}^A) \cdot \mathbf{n}}$$

longer for them to collide and vice versa. Denote $v_n = (\mathbf{v}^B - \mathbf{v}^A) \cdot \mathbf{n}$ as the relative normal velocity. We have

$$\frac{\partial toi}{\partial \theta} = \frac{\mathbf{n}^\top}{\text{clamp}(v_n, \varepsilon)} \left(\frac{\partial \mathbf{p}^B}{\partial \theta} - \frac{\partial \mathbf{p}^A}{\partial \theta} \right) \Big|_{\mathbf{q}=\mathbf{q}_{t+toi}}, \quad (8)$$

where $\varepsilon > 0$ is a small positive constant for clamping to avoid division by zero. When there is no collision, we treat Δ as a constant in (6).

The gradients of Jacobian $\partial \mathbf{J}^\top / \partial \theta$ and $\partial \mathbf{J} / \partial \theta$ are always calculated in the context of $\partial (\mathbf{J}^\top \mathbf{z}) / \partial \theta$ and $\partial (\mathbf{J} \mathbf{z}') / \partial \theta$, with \mathbf{z} and \mathbf{z}' being arbitrary constant vectors. In our setting, the i -th column $\mathbf{J}^{i^\top} = [(\mathbf{n}^i \times \mathbf{p}^i)^\top \mathbf{n}^{i^\top}]^\top$ of \mathbf{J}^\top is defined by the location \mathbf{p}^i and normal \mathbf{n}^i of the corresponding contact feature (4). Then, $\mathbf{J}^\top \mathbf{z} = \sum_i \mathbf{J}^{i^\top} z^i$ and

$$\frac{\partial (\mathbf{J}^\top \mathbf{z})}{\partial \theta} = \sum_i \left[\begin{array}{c} (\partial \mathbf{n}^i / \partial \theta) \times \mathbf{p}^i + \mathbf{n}^i \times (\partial \mathbf{p}^i / \partial \theta) \\ (\partial \mathbf{n}^i / \partial \theta) \end{array} \right] z^i,$$

where $z^i \in \mathbb{R}$ is the i -th element of \mathbf{z} . We derive $\partial (\mathbf{J} \mathbf{z}') / \partial \theta$ in a similar manner.

For our current implementation, the user needs to specify the callback functions to calculate the derivatives of the contact features $\mathbf{p}(\cdot)$ and $\mathbf{n}(\cdot)$ with respect to the geometric parameter θ for each URDF file. We expect future simulators to provide automatic routines in the collision engine.

V. STABLE OBJECT PLACEMENT

In this section, we demonstrate how our newly derived gradients can address geometric uncertainties in stable object placement. At each timestep, we use gradient descent to refine the estimate of the uncertain geometry and update the robot's action accordingly. This process repeats for T steps before the robot releases the object. Throughout execution, the estimate approaches the groundtruth geometry, guiding the robot toward a stable placement.

Algorithm 1 Procedure to update a single geometric parameter and its cost using the history.

```

1: procedure GEOMETRY-UPDATE(  $\hat{\theta}_{t-1}, \mathbf{x}_{t-H:t}, \mathbf{u}_{t-H:t}, \mathbf{y}_{t-H+1:t}^{\text{gt}}$  )
2:   Initialize:  $\theta \leftarrow \hat{\theta}_{t-1}$ .
3:   for number of iterations do
4:     Compute the residual  $r_t(\theta)$  defined in (10).
5:      $\text{gradient} \leftarrow \partial r_t / \partial \theta$ .
6:     if  $\text{gradient} = 0$  then
7:       return  $\hat{\theta}_t = \theta, c = r_t(\theta)$ .
8:     end if
9:      $\theta \leftarrow \theta - \text{learningRate} \cdot \text{clipGrad}(\text{gradient})$ .
10:  end for
11:  return  $\hat{\theta}_t = \theta, c = r_t(\theta)$ .
12: end procedure

```

Algorithm 2 Procedure to update the belief using the history and our gradient-based estimator.

```

1: procedure BELIEF-UPDATE(  $\text{bel}_{t-1}, \mathbf{x}_{t-H:t}, \mathbf{u}_{t-H:t}, \mathbf{y}_{t-H+1:t}^{\text{gt}}$  )
2:   Initialize:  $\text{bel}_t \leftarrow \emptyset$ .
3:   for parameter  $\theta$  in  $\text{bel}_{t-1}$  do
4:      $\text{updatedParticle} \leftarrow \text{GEOMETRY-UPDATE}(\theta, \mathbf{x}_{t-H:t}, \mathbf{u}_{t-H:t}, \mathbf{y}_{t-H+1:t}^{\text{gt}})$ 
5:      $\text{bel}_t \leftarrow \text{bel}_t \cup \{\text{updatedParticle}\}$ 
6:   end for
7:   return  $\text{bel}_t$ .
8: end procedure

```

A. Maintaining A Single Estimate

We begin with the case of maintaining a single estimate $\hat{\theta}_t$, initialized as $\hat{\theta}_0 = \theta^{\text{init}}$. At each timestep t , we update the estimate $\hat{\theta}_{t-1}$ to $\hat{\theta}_t$ using data from the past $H + 1$ timesteps, including the robot states $\mathbf{x}_{t-H:t}$, robot actions $\mathbf{u}_{t-H:t}$, and measured contact wrench $\mathbf{y}_{t-H+1:t}^{\text{gt}}$. Specifically, we refine the estimate by minimizing the discrepancy between the predicted contact wrench $\mathbf{y}_{t-H+1:t}$ and the force-torque (FT) sensor measurements $\mathbf{y}_{t-H+1:t}^{\text{gt}}$.

Given a geometric parameter θ , we use the history data to get the predicted contact wrench

$$\mathbf{y}_{t-H+1:t} = \text{Rollout}(\mathbf{x}_{t-H:t-1}, \mathbf{u}_{t-H:t-1}, \theta), \quad (9)$$

where Rollout is defined by applying the simulator (1) iteratively: $(\cdot), \mathbf{y}_{t'+1} = f(x_{t'}, u_{t'}; \theta)$ for $t' = t - H, t - H + 1, \dots, t - 1$.

We define the discrepancy as the residual error between the measurement and the prediction

$$r_t(\theta) := r(\mathbf{y}_{t-H+1:t}^{\text{gt}}, \mathbf{y}_{t-H+1:t}) \in \mathbb{R}, \quad (10)$$

where $r \geq 0$ measures the distance between two sequences of contact wrenches; when the two sequences are identical, the residual is zero.

We find the updated geometric parameter that minimizes the residual error $\hat{\theta}_t = \arg \min_{\theta} r_t(\theta)$ by gradient descents from $\hat{\theta}_{t-1}$. We summarize the estimation procedure in Algorithm 1.

Then, we compute the robot action using the policy $\mathbf{u}_t = \pi(\mathbf{x}_t, \hat{\theta}_t)$. Finally, we send \mathbf{u}_t to the robot for execution.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

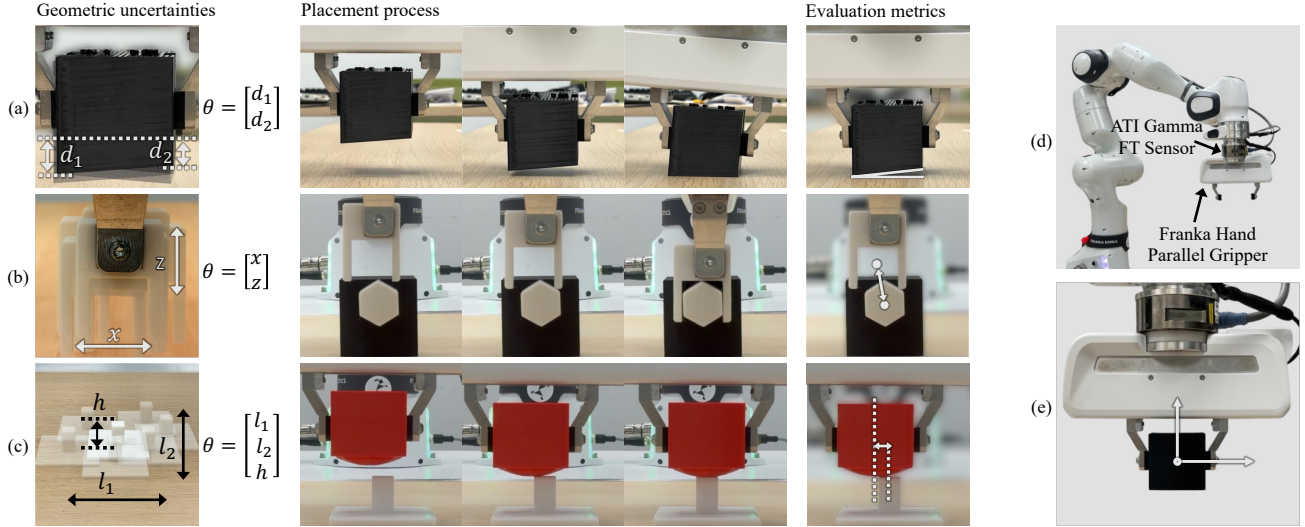


Fig. 3. Experimental setups. (a)-(c) Visualization of the task setups for (a) **shape**, (b) **pose** and (c) **env**: the left column shows the unknown geometric parameters, the middle columns show an ideal placement process, and the right column visualizes the distance to goal when releasing. (d) The ATI Gamma sensor and Franka Hand mounted on a Franka Research 3 arm. (e) The end-effector frame in which the motion and forces are represented.

B. Multiple Initializations

Gradient descents only find the local minimum and are sensitive to the initialization. To mitigate this, we keep track of a belief over multiple geometric parameters over time.

We represent the belief bel_t as a set of N particles, with each particle being a tuple of the value and the cost of the parameter

$$bel_t = \left\{ \left\langle \hat{\theta}_t^1, c_t^1 \right\rangle, \left\langle \hat{\theta}_t^2, c_t^2 \right\rangle, \dots, \left\langle \hat{\theta}_t^N, c_t^N \right\rangle \right\}, \quad (11)$$

where $\hat{\theta}_t^i$ is the i -th parameter at timestep t , and $c_t^i = r_t(\hat{\theta}_t^i) \geq 0$ is the cost of that parameter. To initialize the belief bel_0 , we set the value $\hat{\theta}_0^i = \theta^{\text{init}} + \epsilon$ with ϵ sampled from a distribution of the perception noise, and the cost c_0^i to be an arbitrary positive constant, for $i = 1, 2, \dots, N$.

At every timestep t , we first update the belief from bel_{t-1} to bel_t using the history of the past $H+1$ timesteps, including the states $\mathbf{x}_{t-H:t}$, actions $\mathbf{u}_{t-H:t}$, and measurements $\mathbf{y}_{t-H+1:t}^{\text{gt}}$ (see Algorithm 2). Then, we select the value of the particle with the minimum cost as the estimate $\hat{\theta}_t$, and compute the robot action using the policy $\mathbf{u}_t = \pi(\mathbf{x}_t, \hat{\theta}_t)$. Finally, we send \mathbf{u}_t to the robot for execution. We repeat this process for T steps before the robot releases the object. The overall pipeline is illustrated in Fig. 2.

VI. EXPERIMENT

A. Tasks

We evaluate our proposed method on 3 tasks with different kinds of geometric uncertainty: **shape**, **pose**, and **env**. The geometric uncertainty and placement processes of the 3 tasks are illustrated in Fig. 3.

In **shape**, the robot needs to stably place a cube on the table surface. The goal set \mathcal{G} in (2) is defined by all configurations in which the bottom surface of the cube is aligned with the table surface. The cube’s left and right walls have uncertain

heights, defined by the deviations from the nominal values d_1 and d_2 .

In **pose**, the robot needs to stably place a wrench on a hex-shaped part. The goal set \mathcal{G} in (2) contains a single configuration in which the wrench is aligned with the part. The wrench’s in-hand pose has uncertainty, represented as the deviations x and z .

In **env**, the robot needs to place a block with a curved bottom surface on a pillar, whose top square surface is parallel to the table surface; the square’s side length is 15 mm. The goal set \mathcal{G} in (2) contains a single configuration in which the center of the object’s bottom surface coincides with that of the pillar’s top surface. The horizontal locations and height of the pillar have uncertainty, represented as the deviations from the nominal values l_1 , l_2 and h .

In all tasks, the nominal values of the uncertain geometry are 0, and are used as the initial guess $\theta^{\text{init}} = 0$. Each task has 10 test cases of different groundtruth geometric parameters $\theta^{\text{gt}} \neq \theta^{\text{init}}$. The groundtruth geometric parameters θ^{gt} are obtained from manual calibration and CAD models of the 3D-printed parts. We set the duration $T = 50$ and the history $H = 5$ timesteps for all tasks. At the initial robot configuration, the gripper is above a *nominal* goal defined by the *nominal* geometric parameter $\mathcal{G}(\theta^{\text{init}})$.

B. Setup

We carry out the experiments on a Franka Research 3 (FR3) arm. We mount an ATI Gamma FT sensor on the flange of the FR3 arm, and mount the Franka Hand parallel gripper to the FT sensor. See Fig. 3(d).

We ensure the end-effector executes slow, low-acceleration motions and remains near static equilibrium during contact, so the interaction can be regarded as quasi-static. Hence, we assume the FT sensor measures the sum of the gripper weight and the contact wrench between the object and the environment. We compensate for the gripper weight and transform

Algorithm 3 Procedure to update the belief using the history and the particle filter (PF) baseline.

```

1: procedure BELIEF-UPDATE-PF(  $bel_{t-1}$ ,  $\mathbf{x}_{t-H:t}$ ,  $\mathbf{u}_{t-H:t}$ ,
 $\mathbf{y}_{t-H+1:t}^{\text{gt}}$  )
2:   Initialize:  $bel_t \leftarrow \emptyset$ ,  $\overline{bel}_t \leftarrow \emptyset$ .
3:   for parameter  $\theta$  in  $bel_{t-1}$  do
4:      $\theta \leftarrow \theta + \epsilon$ , where  $\epsilon$  is sampled from a zero-mean normal
     distribution.
5:     Compute the residual  $c \leftarrow r_t(\theta)$  defined in (10).
6:      $\overline{bel}_t \leftarrow \overline{bel}_t \cup \{(\theta, c)\}$ .
7:   end for
8:    $bel_t \leftarrow \text{LOW-VARIANCE-SAMPLER}(\overline{bel}_t)$ , with unnormal-
     ized weights defined in (12).
9:   return  $bel_t$ .
10: end procedure

```

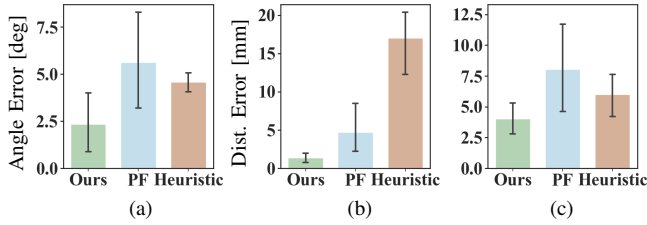


Fig. 4. Our method has smaller distance to the goal errors when the robot releases. We plot the means and 95% confidence intervals over 10 test cases. For (a) **shape**, the errors are equivalent to angle errors. For (b) **pose** and (c) **env**, the errors are equivalent to distance errors.

the measured wrench into the end-effector frame (Fig. 3(e)) as \mathbf{y}_t^{gt} , using the transformation between the FT sensor’s sensing plate and the end-effector frame obtained from their mechanical drawings. When inertial effects are significant, a generalized momentum observer can be used to estimate the contact wrench between the object and the environment [52].

We assume the operational-space dynamics [53] and impedance control [54], [55] for the system (1), with \mathbf{u}_t being the reference equilibrium pose. The generalized bias forces are compensated. Since during the 3 placement tasks, the robot moves in a small space, we approximate the operational-space inertia as a constant matrix.

We implement the policy (3) as a heuristic that interpolates from the current robot state \mathbf{x}_t to the closest goal state in $\mathcal{G}(\theta)$. We have verified that when $\theta = \theta^{\text{gt}}$, this policy indeed drives the robot to a goal $\mathbf{x}^{\text{goal}} \in \mathcal{G}(\theta^{\text{gt}})$.

Our differentiable simulation does not run in real time. Since the tasks are quasi-static, we pause estimation and hold the reference pose while waiting for the next robot action. Each action takes an average of 3.36 seconds. We expect real-time performance to be achievable with a more optimized implementation. Currently, our implementation reloads the URDF each time the geometric parameter is updated (Line 9 in Algorithm 1); enabling in-memory updates could eliminate this overhead. Simulation speed is also limited by reliance on PyTorch’s auto-differentiation, which incurs Python-C++ communication at each step [7], [41], [56]; combining multiple steps per call can reduce this cost. Real-time differentiable contact simulation has been demonstrated in locomotion planning [46], and we believe similar performance is feasible for geometric uncertainty estimation.

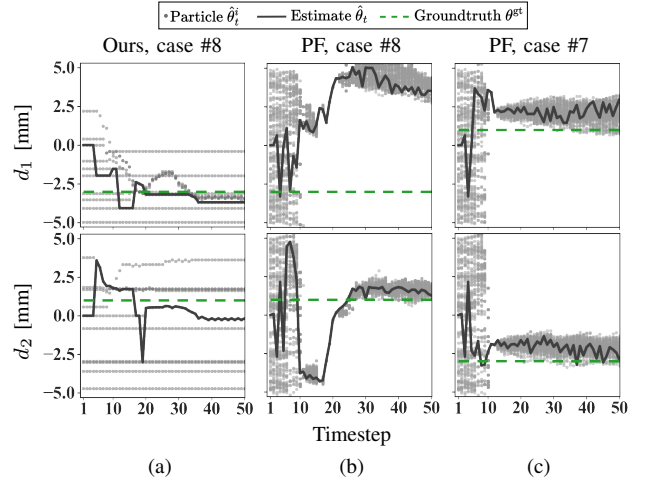


Fig. 5. PF might suffer from “particle starvation”. We plot estimation over time from 2 test cases of the **shape** task. In the same test case, ours (a) has good estimate, while PF (b) diverges on d_1 due to “particle starvation.” (c) In a different test case, PF maintains good estimate.

C. Baselines

Granted that we can design an optimal policy for a particular task [57], here we consider two baselines that are applicable in all scenarios. The first baseline **PF** replaces our gradient-based estimator by a particle filter [58]. Similar to the prior works [32], [59], we add a small Gaussian noise during the prediction update, and the unnormalized weight of the i -th particle is given as

$$\text{Weight}(\hat{\theta}_t^i) = \exp(-\beta c_t^i) \quad (12)$$

with β being a positive constant. We implement the low variance sampler as described in [58]. The PF implementation is summarized in Algorithm 3.

The second baseline is a trigger-based **heuristic** policy. The robot moves downwards until the measured contact wrench is larger than a threshold. This baseline demonstrates the performance of a compliant controller that does not explicitly reason about the geometric uncertainties. When there is no geometric uncertainty $\theta^{\text{init}} = \theta^{\text{gt}}$ or there is only uncertainty in heights, this heuristic policy is sufficient to reach a goal in $\mathcal{G}(\theta^{\text{gt}})$ for all tasks.

We use $N = 50$ particles for PF, and $N = 10$ particles in our method. This ensures that the two methods have the same number of total rollouts in each timestep.

D. Evaluation Metric

We measure the distance to goal errors when the robot releases the object. In our particular tasks, instead of measuring the distance between poses [49], [60], we apply simplifications to have consistent physical dimensions. For **shape**, since in all test cases for all methods, the cube touches the table surface, we reduce the distance to goal error to the angle between the cube’s bottom surface and the table surface. For **pose** and **env**, we measure the distance between the final and goal configurations. The evaluation metrics for the 3 tasks are illustrated in Fig. 3.

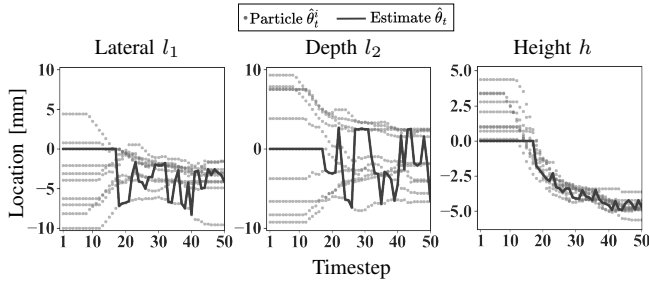


Fig. 6. Estimation of the saucer’s location over time for a successful attempt. The estimate converges as the task progresses, and its variance decreases over time.

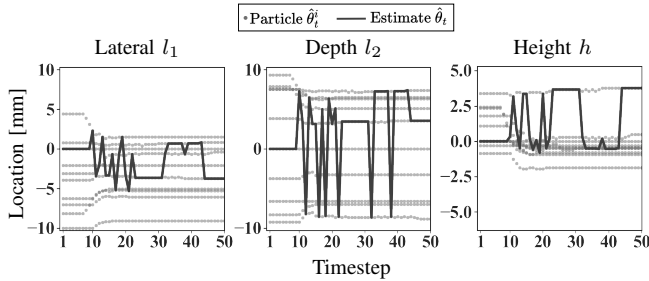


Fig. 7. Estimation of the saucer’s location over time for a failed attempt. Due to poor initialization, most particles remained stagnant throughout the task, resulting in inaccurate placement.

E. Results

Fig. 4 shows the errors over the 10 test cases for the 3 tasks. Our proposed method achieves smaller average errors than the PF and heuristic baselines. Interestingly, PF baseline does not always outperform the heuristic baseline on average, even though it has better best performance. When PF has bad performance, we observe “particle starvation”, where most particles “collapse” to a small range of wrong values [61]. Fig. 5(a)–(b) show the eighth test case from **shape**. In Fig. 5(b), PF’s estimate on d_1 collapsed to the wrong value, while in Fig. 5(a) our gradient-based estimator maintains an estimate closer to the groundtruth. The particle starvation does not always happen. Fig. 5(c) shows PF’s estimate over time in the seventh test case from **shape**, where PF’s estimate is close to the groundtruth.

F. Placing a Full Cup of Coffee

Lastly, we deploy our method to place a full cup of coffee on a saucer (Fig. 8), relaxing several assumptions from Section III. The cup of coffee is approximated as a solid cylinder, and the saucer’s mesh is imprecise. We model the geometric uncertainty as the saucer’s 3D location, $\theta = [l_1 \ l_2 \ h]^T$, shown in Fig. 8(a). Across five trials with varying saucer positions, our method successfully placed the cup without spilling in four of them, while the naive heuristic policy failed in all cases. We plot θ over time in a successful trial in Fig. 6, and a failure case in Fig. 7, where poor initialization led to stagnant particles.

VII. CONCLUSION AND DISCUSSION

We presented a novel gradient that relates FT sensor readings to geometric parameters within a differentiable simulator,

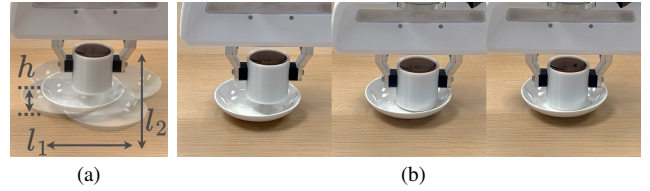


Fig. 8. The coffee placement task. (a) The saucer’s location, represented by $\theta = [l_1 \ l_2 \ h]^T$, is uncertain. (b) During the placement process, as the estimate is refined, the robot adjusts its motion to position the cup closer to the flat center of the saucer.

providing a unified approach to stable object placement under geometric uncertainty. By leveraging this gradient to minimize the discrepancy between sensor measurements and model predictions, we achieve improved performance over the gradient-free particle filter baseline. We demonstrate the effectiveness of our approach across various geometric uncertainties, including the object’s in-hand pose uncertainty, the object’s shape uncertainty, and the environment’s shape uncertainty.

While this work focuses on geometric uncertainties in stable object placement, several directions remain open for future research. Our current approach selects robot actions based on the best estimate using an uncertainty-blind policy (3), without reasoning over the full belief or actively reducing uncertainty. Future work could incorporate full belief reasoning to balance exploration and execution [58]. Additionally, our formulation—which considers a gripper, an object, and a static environment—may extend to other contact-rich tasks under geometric uncertainty. For instance, peg-in-hole insertion can be modeled similarly, with the peg’s in-hand pose and the hole’s location as sources of uncertainty. Beyond geometry, future research could also address other forms of uncertainty, such as friction coefficients or sensor noise. Another improvement is to incorporate robot state predictions and measurements into the estimation to complement the force-torque data. Lastly, while our current implementation is not real-time, we believe real-time performance is achievable through further optimization, as shown in recent work on differentiable locomotion planning [46].

REFERENCES

- [1] Intel, “Intel® RealSense™ Product Family D400 Series Datasheet,” 2020.
- [2] A. Rodriguez, “The unstable queen: Uncertainty, mechanics, and tactile feedback,” *Science Robotics*, vol. 6, no. 54, 2021.
- [3] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, “Automatic synthesis of fine-motion strategies for robots,” *Int. J. Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
- [4] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, “Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids,” in *Int. Conf. on Learning Representations*, 2019.
- [5] P. Mitrano, A. LaGrassa, O. Kroemer, and D. Berenson, “Focused adaptation of dynamics models for deformable object manipulation,” in *IEEE Int. Conf. on Robotics & Automation*, 2023.
- [6] S. Le Cleac’h, H.-X. Yu, M. Guo, T. Howell, R. Gao, J. Wu, Z. Manchester, and M. Schwager, “Differentiable physics simulation of dynamics-augmented neural objects,” *IEEE Robotics & Automation Letters*, vol. 8, no. 5, pp. 2780–2787, 2023.
- [7] G. Yang, S. Luo, Y. Feng, Z. Sun, C. Tie, and L. Shao, “Jade: A differentiable physics engine for articulated rigid bodies with intersection-free frictional contact,” in *IEEE Int. Conf. on Robotics & Automation*, 2024.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

- [8] A. Jain and C. C. Kemp, "EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, pp. 45–64, 2010.
- [9] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Trans. on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.
- [10] S. Kim, D. K. Jha, D. Romeres, P. Patre, and A. Rodriguez, "Simultaneous tactile estimation and control of extrinsic contact," in *IEEE Int. Conf. on Robotics & Automation*, 2023.
- [11] K. Ota, D. K. Jha, K. M. Jatavallabhula, A. Kanezaki, and J. B. Tenenbaum, "Tactile estimation of extrinsic contact patch for stable placement," in *IEEE Int. Conf. on Robotics & Automation*, 2024.
- [12] J. Stückler and S. Behnke, "Adaptive tool-use strategies for anthropomorphic service robots," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.
- [13] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham, "Can robots assemble an IKEA chair?" *Science Robotics*, vol. 3, no. 17, 2018.
- [14] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2017.
- [15] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *IEEE Int. Conf. on Robotics & Automation*, 2019.
- [16] S. Jin, X. Zhu, C. Wang, and M. Tomizuka, "Contact pose identification for peg-in-hole assembly under uncertainties," in *American Control Conf.*, 2021.
- [17] X. Zhang, M. Tomizuka, and H. Li, "Bridging the sim-to-real gap with dynamic compliance tuning for industrial insertion," in *IEEE Int. Conf. on Robotics & Automation*, 2024.
- [18] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.
- [19] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, no. 2-4, pp. 210–225, 1993.
- [20] F. Wirmshofer, P. S. Schmitt, W. Feiten, G. v. Wichert, and W. Burgard, "Robust, Compliant Assembly via Optimal Belief Space Planning," in *IEEE Int. Conf. on Robotics & Automation*, 2018.
- [21] S. Chintalapudi, L. Kaelbling, and T. Lozano-Perez, "Bi-level belief space search for compliant part mating under uncertainty," in *Proc. Int. Workshop on the Algorithmic Foundations of Robotics*, 2024.
- [22] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Grasping POMDPs," in *IEEE Int. Conf. on Robotics & Automation*, 2007.
- [23] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty," *Int. J. Robotics Research*, vol. 35, no. 1-3, pp. 244–264, 2016.
- [24] B. Saund, S. Choudhury, S. Srinivasa, and D. Berenson, "The blindfolded traveler's problem: A search framework for motion planning with contact estimates," *Int. J. Robotics Research*, vol. 42, no. 4-5, pp. 289–309, 2023.
- [25] S. Chhatpar and M. Branicky, "Localization for robotic assemblies using probing and particle filtering," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2005.
- [26] Y. Taguchi, T. K. Marks, and H. Okuda, "Rao-Blackwellized particle filtering for probing-based 6-dof localization in robotic assembly," in *IEEE Int. Conf. on Robotics & Automation*, 2010.
- [27] F. von Drigalski, K. Hayashi, Y. Huang, R. Yonetani, M. Hamaya, K. Tanaka, and Y. Ijiri, "Precise multi-modal in-hand pose estimation using low-precision sensors for robotic assembly," in *IEEE Int. Conf. on Robotics & Automation*, 2021.
- [28] B. Saund and D. Berenson, "CLASP: Constrained latent shape projection for refining object shape from robot contact," in *Conference on Robot Learning*, 2022.
- [29] D. Ma, S. Dong, and A. Rodriguez, "Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements," in *IEEE Int. Conf. on Robotics & Automation*, 2021.
- [30] K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation," *Int. J. Robotics Research*, vol. 24, no. 8, pp. 615–630, 2005.
- [31] K. Hertkorn, M. A. Roa, C. Preusche, C. Borst, and G. Hirzinger, "Identification of contact formations: Resolving ambiguous force torque information," in *IEEE Int. Conf. on Robotics & Automation*, 2012.
- [32] A. Sipos and N. Fazeli, "Simultaneous contact location and object pose estimation using proprioceptive tactile feedback," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2022.
- [33] —, "MultiSCOPE: Disambiguating in-hand object poses with proprioception and tactile feedback," in *Robotics: Science & Systems*, 2023.
- [34] J. Lee, M. Lee, and D. Lee, "Uncertain pose estimation during contact tasks using differentiable contact features," in *Robotics: Science & Systems*, 2023.
- [35] U. Thomas, S. Molkenstruck, R. Iser, and F. M. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *IEEE Int. Conf. on Robotics & Automation*, 2007.
- [36] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *Advances in Neural Information Processing Systems*, 2018.
- [37] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "NeuralSim: Augmenting differentiable simulators with neural networks," in *IEEE Int. Conf. on Robotics & Automation*, 2021.
- [38] J. Degraeve, M. Hermans, J. Dambre, et al., "A differentiable physics engine for deep learning in robotics," *Frontiers in neurorobotics*, p. 6, 2019.
- [39] Y. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Scalable differentiable physics for learning and control," in *Int. Conf. on Machine Learning*, 2020.
- [40] —, "Efficient differentiable dimulation of articulated bodies," in *Int. Conf. on Machine Learning*, 2021.
- [41] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints," in *Robotics: Science & Systems*, 2021.
- [42] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, "ADD: Analytically differentiable dynamics for multi-body systems with frictional contact," *ACM Trans. on Graphics*, vol. 39, no. 6, pp. 1–15, 2020.
- [43] T. A. Howell, S. L. Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable simulator for robotics," *arXiv preprint arXiv:2203.00806*, 2022.
- [44] M. Toussaint, K. Allen, K. Smith, and J. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Robotics: Science & Systems*, 2018.
- [45] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Trans. on Robotics*, pp. 1–21, 2023.
- [46] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Trans. on Robotics*, vol. 40, pp. 1617–1629, 2024.
- [47] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Robotics: Science & Systems*, 2021.
- [48] K. J. Åström and R. M. Murray, *Feedback Systems*, 2010.
- [49] K. M. Lynch and F. C. Park, *Modern Robotics*, 2017.
- [50] R. M. Murray, "Proportional derivative (PD) control on the euclidean group," in *Proc. European Control Conf.*, 1995.
- [51] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The linear complementarity problem*, 2009.
- [52] A. de Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *IEEE Int. Conf. on Robotics & Automation*, 2005.
- [53] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [54] N. Hogan, "Impedance control: An approach to manipulation. I - theory. II - implementation. III - applications," *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 1–24, 1985.
- [55] N. Hogan and S. P. Buerger, "Impedance and interaction control 19.1," *Robotics and automation handbook*, pp. 19–1, 2005.
- [56] A. Bilogur, "PyTorch Training Performance Guide - Eager versus graph execution," <https://residentmario.github.io/pytorch-training-performance-guide/jit.html#eager-versus-graph-execution>, 2021.
- [57] M. Khansari, E. Klingbeil, and O. Khatib, "Adaptive human-inspired compliant contact primitives to perform surface–surface contact under uncertainty," *Int. J. Robotics Research*, vol. 35, no. 13, pp. 1651–1675, 2016.
- [58] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2005.
- [59] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2016.
- [60] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.
- [61] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *Int. J. Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.