

# Uncertain Pose Estimation during Contact Tasks using Differentiable Contact Features

Jeongmin Lee\*, Minji Lee\* and Dongjun Lee

Department of Mechanical Engineering, IAMD and IOER, Seoul National University

Email: {ljmlgh, mingg8, djlee}@snu.ac.kr

**Abstract**—For many robotic manipulation and contact tasks, it is crucial to accurately estimate uncertain object poses, for which certain geometry and sensor information are fused in some optimal fashion. Previous results for this problem primarily adopt sampling-based or end-to-end learning methods, which yet often suffer from the issues of efficiency and generalizability. In this paper, we propose a novel differentiable framework for this uncertain pose estimation during contact, so that it can be solved in an efficient and accurate manner with gradient-based solver. To achieve this, we introduce a new geometric definition that is highly adaptable and capable of providing differentiable contact features. Then we approach the problem from a bi-level perspective and utilize the gradient of these contact features along with differentiable optimization to efficiently solve for the uncertain pose. Several scenarios are implemented to demonstrate how the proposed framework can improve existing methods.

## I. INTRODUCTION

Contact has always been considered the challenging part of robot manipulation. Unlike free-space motion, contact constraints are complex to model, complicated to numerically solve, and difficult to find an appropriate strategy to handle well. As a result, learning-based methods have been widely adopted in this field, with many impressive results to date [3, 29, 15]. Learning-based methods are essentially sampling-based methods with forward-directed results. That is, they involve collecting data from the actions, analyzing the results, and learning how to produce the best results. But they are data-dependent, often produce noisy results, and generalization is difficult. Some techniques such as domain randomization [39] are often utilized, yet it is deemed still necessary to develop more structured and reliable methods.

From this perspective, the topic of differentiable physics has recently emerged. By building differentiable formulation, gradient-based methods can replace many of the sampling requirements, improving generalization performance and efficiency. As a result, these techniques have proven to be useful in a variety of applications, including trajectory optimization [9, 14], policy gradient [47], system identification [21] and design optimization [46]. However, the use of differentiable modeling in contact-intensive tasks that require responding to uncertain environments, such as robot assembly and placement, has not been well addressed.

In this paper, we present a novel differentiable framework which estimates the uncertain pose during contact tasks from sensor measurements. Our framework has a wide range of

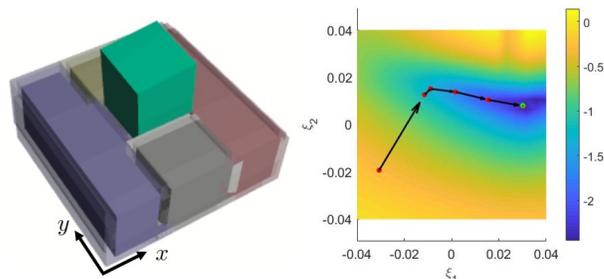


Fig. 1: Graphical abstracts illustrating our differentiable pose estimation during contact. Left: A peg-in-hole task performed in a hole with pose uncertainty along the  $x$  and  $y$  directions. Right: Visualization of the differentiable cost landscape and the gradient-based optimization process utilizing force/torque sensor information acquired through interactions (Green dot: true uncertainty parameter).

applications, from simple external impact localization to interactive manipulation such as peg-in-hole assembly. The main contribution of this paper is: 1) we devise a new geometry representation based on a prescribed support function which guarantees to provide differentiable contact features and their efficient computation algorithm; and 2) an efficient bi-level solution scheme based on differentiable optimization for uncertain pose estimation problem. The proposed methods are validated against both in simulation and experiment, demonstrating the efficacy of our differentiable framework for contact tasks.

The rest of the paper is organized as follows. Previous studies related to our work are reviewed in Sec. II with some preliminary materials presented in Sec. III. Sec. IV presents a formal formulation of the uncertain pose estimation problem in contact. Then in Sec. V, our novel prescribed support function based geometry model for differentiable contact feature is presented. Sec. VI describes bi-level solution scheme for the estimation, based on the geometry model provided in Sec. V and differentiable optimization. Various implementation scenarios with evaluations are provided in Sec. VII, followed by concluding remarks and discussions in Sec. VIII.

## II. RELATED WORKS

### A. Differentiable Contact Formulation

Many existing studies [9, 12, 8, 14] use collision proxies as simple shapes (point, sphere, plane, etc.). To our best knowledge, attempts to utilize more general geometry have begun to take place very recently. First, the scope geometry

\*equal contribution

is extended to convex primitives (e.g., cylinder, cone, padded polygon) in [40] by utilizing implicit differentiation on conic optimization. In [41] and [13], neural network-based implicit functions such as a signed distance field (SDF) or neural radiance field (NeRF [27]) are used. However, accurate modeling of contact between the fields is not well-developed and often rely on query point sampling [41, 20]. This can lead to reduced applicability and may generate an excessive number of contacts. In [28], an approach using randomized smoothing with implicit differentiation of Gilbert-Johnson-Keerthi (GJK [10]) optimality condition is proposed. However, the gradient may not be consistent with the underlying geometry and may still be myopic. Instead, we propose to define the object shape through a prescribed support function which provides a direct parametric representation of convex geometry and allows for the exact computation of contact features. Moreover, theoretical issues on degeneration is addressed, which have not been dealt in previous studies.

### B. Uncertainty Handling in Interaction

Multiple studies have explored the identification of uncertainty in interaction, using a range of sensors. From visual sensor measurements, 6D pose [6, 45] or inertial parameters [26] estimation can be utilized in online during tasks. However, vision sensors have limitations in that occlusion can occur, they cannot cover the entire robot body, and are difficult to achieve the high accuracy required for contact-intensive tasks such as peg-in-hole [16].

Therefore, other sensors such as proprioceptive sensor, force/torque (FT) sensor, or more recently vision-based tactile sensors have also widely used. In many works, encoding sensing measurements for use in manipulation heavily rely on learning-based frameworks. For example, [22] combines vision and FT sensor information using self-supervised learning. In [16], a certain action is performed to acquire FT measurements when contact occurs, and the plotted results are passed through neural network to estimate of the peg pose. For tactile sensor, the work in [42] estimates the pose of grasped object using neural network and [13] perform tracking of extrinsic contact between object and environment based on neural contact fields. Similarly, [38] performs global localization of the finger and object to a larger object and a long horizon. These methods are data-dependent and may require re-learning as the use case expands. Our work can be combined with these approaches to better exploit the dynamic and kinematic structures, thus improving performance and generalizability.

There also exist some model-based methods to estimate certain information during contact. In [11], a comprehensive survey is provided, but how to deal with object geometry in tandem is rarely addressed. Studies that address geometry and sensor information together rely primarily on sampling strategies. For instance, contact particle filter (CPF) [25, 44] presents the way for external contact localization using proprioceptive sensors or force sensors. Object grasp pose estimation method is also conducted in [36] on the extension of CPF. Similarly, [43] presents the Bayesian framework for multi-

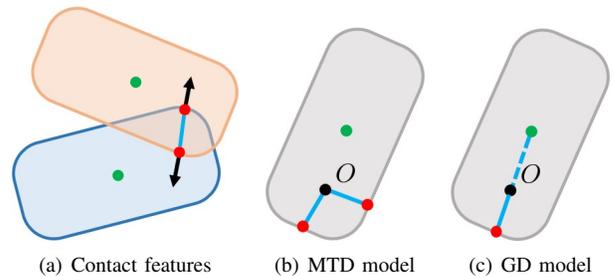


Fig. 2: Comparison of the minimum translation distance (MTD) model and growth distance (GD) model contact features. Minkowski sum is represented by the gray area. The penetration depth is indicated by the blue segment, the contact witness points by the red points, and the contact normal by the black arrows.

modal fusion. These have limitations in that handling multiple contacts is difficult or time-consuming and is inefficient owing to the limitation of the sampling-based method, which only utilizes forward-directed results. Recently, [24] and [18] develop the optimization based extrinsic contact sensing frameworks using various structured constraints. In comparison to above works, we aim for a differentiable formulation that can be applied to more general geometric types.

## III. PRELIMINARY

### A. Support Function

For a convex set  $\mathcal{C} \subset \mathbb{R}^3$ , the support function  $h : \mathbb{R}^3 \rightarrow \mathbb{R}$  is defined as

$$h(x) = \max_{s(x) \in \mathcal{C}} x^T s(x) \quad (1)$$

where  $x \in \mathbb{R}^3$  and  $s(x) \in \mathcal{C}$  is the farthest point in the  $x$  direction among the points in  $\mathcal{C}$ , called the support point. Rather than calculating the support function for a given geometry, in this paper, we define the geometry of the object using a prescribed support function.

### B. Contact Features

The contact features we refer to in this paper are penetration depth, witness points, and contact normal (see Fig. 2(a)). For general convex shapes, the minimum translation distance (MTD) model is widely adopted [10, 31, 35] to define contact features. The model computes the closest point on the boundary of the Minkowski sum [34] from the origin. However as depicted in Fig. 2(b), the closest point may have a non-unique solution. The non-uniqueness occurs commonly under deep penetration and sharp geometry. Although this issue is typically not so critical in simulation because it allows only a small amount of penetration, it is not so for us, as we are aiming for differentiable framework for general manipulation programming, for which such non-uniqueness can pose a serious issue.

In contrast, the growth distance (GD) model, first proposed in [30], computes the growth factor that two objects “touch” each other, i.e.,

$$\min_{\sigma} \text{s.t. } \mathcal{C}_1(\sigma) \cap \mathcal{C}_2(\sigma) \neq \emptyset \quad (2)$$

where  $\sigma \in \mathbb{R}^+$  is the growth factor and  $\mathcal{C}(\sigma)$  is an increased convex set by the growth factor around a given center. The model was intended to convert contact detection processes from polyhedral objects to linear programming, but we are more interested in the fact that it always guarantees uniqueness of solution [51, 40]. This uniqueness can be easily identified using the property that the problem is equivalently substituted by the ray casting problem [51, 50] for Minkowski sum (see also Fig. 2(c)).

### C. Implicit Function Theorem

Consider the multi-variable equation:

$$F(x, y) = 0$$

where  $x \in \mathbb{R}^{n_x}$ ,  $y \in \mathbb{R}^{n_y}$ , and  $F : \mathbb{R}^{n_x+n_y} \rightarrow \mathbb{R}^{n_y}$  is the continuously differentiable function. Then the local solution mapping between  $x$  and  $y$  is unique and continuously differentiable satisfying

$$\frac{dy}{dx} = - \left( \frac{\partial F}{\partial y} \right)^{-1} \frac{\partial F}{\partial x} \quad (3)$$

if the partial Jacobian  $\frac{\partial F}{\partial y}$  is non-singular. The implicit function theorem enables the use of a function between multiple variables based on an implicit relation.

## IV. PROBLEM FORMULATION

The main purpose of this paper is to develop the differentiable and general-purposed framework for uncertain pose estimation in interaction. We define the basic structure of the problem as follows:

*Problem 1 (Uncertain Pose Estimation in Contact):* Given the measurement  $\gamma \in \mathbb{R}^{n_\gamma}$ , estimate uncertain pose parameter  $\xi \in \mathbb{R}^{n_\xi}$  through following optimization problem:

$$\begin{aligned} \min_{\xi, f \in \mathcal{C}} \quad & \frac{1}{2} \left\| \gamma - \sum_{k=1}^{m(\xi)} P_k(\xi) f_k \right\|_{\Sigma^{-1}}^2 \\ \text{s.t.} \quad & g_k(\xi) \geq 0, \quad (g_k(\xi))^+ f_k = 0 \quad \forall k \end{aligned} \quad (4)$$

where  $m$  is the number of collision,  $g_k \in \mathbb{R}$ ,  $f_k \in \mathbb{R}^3$ ,  $P_k \in \mathbb{R}^{n_\gamma \times 3}$  are the gap, contact force, and contact mapping matrix (to the measurement) for the  $k$ -th contact. Note that  $P_k$  can be expressed as a Jacobian matrix related to the contact witness points and normal. Also,  $\|\cdot\|_{\Sigma^{-1}}^2$  is the Mahalanobis distance defined under the covariance matrix  $\Sigma$ ,  $(\cdot)^+ = \max(\cdot, 0)$ , and  $\mathcal{C}$  denotes the friction cone set:

$$\begin{aligned} \mathcal{C} &= \mathcal{C}_1 \times \cdots \times \mathcal{C}_m \\ \mathcal{C}_k &= \{f_k \mid \mu_k f_{k,n} \geq \|f_{k,t}\|\} \end{aligned} \quad (5)$$

with  $\mu, n, t$  being the friction coefficient<sup>1</sup>, subscripts for the normal and tangential direction.

Here, the measurement  $\gamma$  is typically the FT or joint torque sensor value. It can also be a stack of measurements rather than a single measurement. Problem 1 can be interpreted as

<sup>1</sup>In practice, it is difficult to accurately know the friction coefficient value, so the rough upper value is mainly used.

finding the most likely pose and contact force that minimizes the residual of the sensor measurements under several constraints, including the friction cone, non-penetration, and the complementarity constraint that ensures the contact force only acts when the gap is not bigger than zero.

Problem 1 can be seen as a generalization of the problem in [25] to deal with the geometry of multiple objects, multiple contact interactions, and various types of uncertainty. Moreover, by including additional cost in (4), it can be combined with other sensor information (e.g., vision) as well as dynamics condition (see also Sec. VI-D). As a result, it has wide-ranging applications in robotics including grasp pose identification, object tracking, and external impact localization and is easily extensible. However, there are several challenges to solving a problem: 1) the problem is nonlinear with multiple complementarity constraints, and 2) the differentiability of  $m, g, P$  is ambiguous, making it difficult to find a proper gradient direction to optimize.

The following sections describe how to address this problem by making it differentiable. We begin by introducing a geometric representation that enables us to represent  $g$  and  $P$  in a differentiable manner.

## V. DIFFERENTIABLE CONTACT FEATURES VIA PRESCRIBED SUPPORT FUNCTION

The computation of differentiable contact features in primitive shapes (e.g., sphere, plane) is simple, but its application is limited. This section will describe a versatile and efficient scheme based on a prescribed support function for common convex geometry. The method will later be extended to broader non-convex geometries by using a set of convex geometries.

### A. Prescribing Support Function

Following theorem motivates us to model the geometry using a prescribed support function.

*Theorem 1 ([34]):* If  $h : \mathbb{R}^3 \rightarrow \mathbb{R}$  is a sublinear function that satisfies:

$$\text{Positive homogeneity: } h(\lambda x) = \lambda h(x) \quad \forall \lambda \geq 0, x \in \mathbb{R}^3$$

$$\text{Subadditivity: } h(x + y) \leq h(x) + h(y) \quad \forall x, y \in \mathbb{R}^3$$

then there is a unique convex body corresponding to the support function.

This theorem implies the one-to-one relationship between a sublinear function and corresponding convex body.

The question remained is then how to define the prescribed form of the support function. We first consider the set of vertices i.e.,  $v_1, \dots, v_n \in \mathbb{R}^3$ . This vertex set can be determined by the user or obtained from data such as mesh or point cloud. As it will be generalized under SE(3) transformation in Sec. V-B), here we assume that the origin is inside the convex hull of the vertices. Then we can easily find that the support function of the geometry defined as a convex hull is written as

$$h(x) = \max(v_1^T x, \dots, v_n^T x) \quad (6)$$

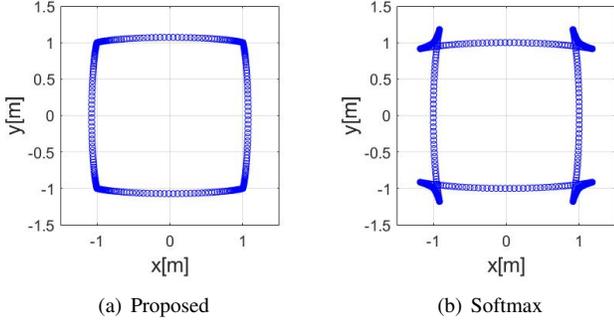


Fig. 3: Comparison of geometry obtained by the proposed support function and the naive softmax support function based on exponential. Vertex set is defined as  $\{[1, 1], [1, -1], [-1, 1], [-1, -1]\}$ .

which is discontinuous. Instead of the max operator, we use a smoothed version of (6) for differentiable contact feature computation. The proposed function form is as follows:

$$h(x) = \left( \sum_{i=1}^n \{ \max(v_i^T x, 0) \}^p \right)^{\frac{1}{p}} \quad (7)$$

where  $p > 2$ . Equation (7) is similar to the  $p$ -norm function, but the  $\text{abs}(\cdot)$  is replaced by  $\max(\cdot, 0)$ , which naturally culls negative elements. Then Theorem 2 summarizes an important property of (7).

*Theorem 2:* Given vertex set  $v_1, \dots, v_n$ , the function (7) is sublinear and twice-differentiable on  $\mathbb{R}^3 \setminus \mathbf{0}$ .

*Proof:* Positive homogeneity is trivial. Subadditivity can be shown as

$$\begin{aligned} h(x) + h(y) &= \left( \sum_{i=1}^n \{ (v_i^T x)^+ \}^p \right)^{\frac{1}{p}} + \left( \sum_{i=1}^n \{ (v_i^T y)^+ \}^p \right)^{\frac{1}{p}} \\ &\geq \left( \sum_{i=1}^n \{ (v_i^T x)^+ + (v_i^T y)^+ \}^p \right)^{\frac{1}{p}} \\ &\geq \left( \sum_{i=1}^n \{ (v_i^T (x + y))^+ \}^p \right)^{\frac{1}{p}} \\ &= h(x + y) \end{aligned}$$

using the Minkowski inequality, where  $\max(\cdot, 0)$  is simplified as  $(\cdot)^+$ . Therefore, the function is sublinear. Twice-differentiability can be easily verified by using the fact that

$$\sum_{i=1}^n \{ (v_i^T x)^+ \}^p > 0$$

for  $x \in \mathbb{R}^3 \setminus \mathbf{0}$  as the origin is inside the vertex set. ■

The properties in Theorem 2 is crucial, as it ensures that any (7) always corresponds to some convex geometry - note from Fig. 3 that other classes of support function are not necessarily able to do so. Fig. 4 depicts various smoothed geometries generated by the support function (7). We can find that smoothness of the geometry can be easily adjusted using  $p$  while retaining convexity and differentiability.

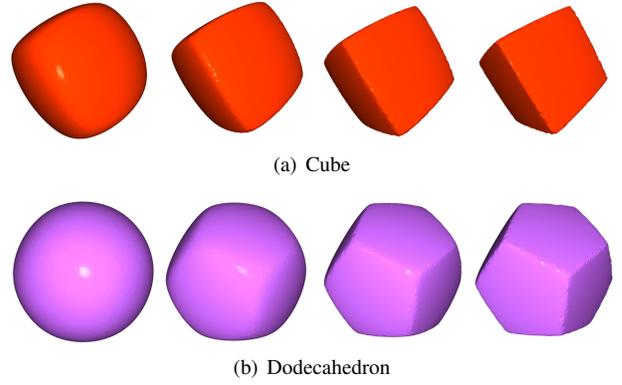


Fig. 4: Visualization of geometries represented by the prescribed support function (7). From left to right,  $p = 5, 10, 20, 40$  are used.

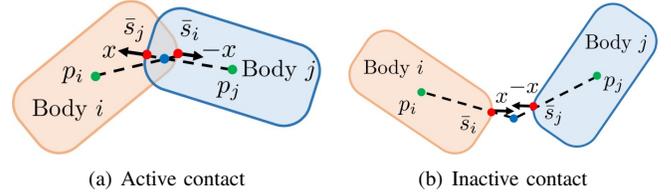


Fig. 5: Visualization of the condition in (10). Support points (red points) on both bodies extended by the growth factor should meet exactly (blue point).

### B. Support Point and SE(3) Transformation

From the definition of support function (1), support point  $s(x)$  can be derived as follows:

$$s(x) = s(x) + x^T \frac{ds}{dx} = \frac{dh}{dx} \quad (8)$$

since  $x^T \frac{ds}{dx} = 0$  holds from the homogeneity. Note that the support point can be easily obtained since  $h(x)$  in (7) is easy to differentiate. By computing support points (8) for various  $x$  direction, we can visualize the corresponding shape of geometry.

The support function  $h(x)$  and the point  $s(x)$  are defined for the geometry that includes the origin. Such geometric representation can be generalized to arbitrary poses through SE(3) transformations. Given  $h$  and configuration vector  $q \in \mathbb{R}^7$  (i.e., position and quaternion), the support function  $\bar{h}$  and support point  $\bar{s}$  for  $q$  and  $x$  can be derived as follows:

$$\begin{aligned} \bar{h}(q, x) &= h(R(q)^T x) + p(q)^T x \\ \bar{s}(q, x) &= R(q)s(R(q)^T x) + p(q) \end{aligned} \quad (9)$$

where  $p(q) \in \mathbb{R}^3$  and  $R(q) \in \text{SO}(3)$  are the translation and rotation by  $q$ . This transformation (9) is essentially equivalent to converting  $x$  to the object local coordinate to obtain  $s(R(q)^T x)$  and then converting it back to the global coordinate. It can be easily verified that  $\bar{f}$  also satisfies the property in Theorem 2, and further twice-differentiable for  $q$ .

### C. Contact Feature Computation

We compute the contact features based on the GD model described in Sec. III-B. However as also mentioned in [51], the

---

**Algorithm 1** Contact Feature Solver
 

---

Initialize  $x, \sigma$  using IE procedure  
 Compute  $F, J$  for initialized value by (10),(11)  
 Initialize trust region radius  $\delta_{tr}$   
**while** not converge **do**  
   Compute Newton step:  $\Delta_{gn} = -J^{-1}F$   
   Compute Cauchy step:  $\Delta_{ca} = -\beta J^T F$   
   Find dogleg step  $\Delta_{dog}$  by  $\Delta_{gn}, \Delta_{ca},$  and  $\delta_{tr}$  [33]  
   Update  $F, J$  under propagated point by  $\Delta_{dog}$   
   Update  $\delta_{tr}$  [33]  
   Update  $x, \sigma$  using  $\Delta_{dog}$  if the step accepted  
**end while**  
 Compute differentiation by (12) and (13)

---

methods for functional surfaces rather than discrete geometries are quite limited. In [51], using the equivalence of the GD model and ray shooting problem, a method based on the internal expanding procedure is presented. In [40], optimization (2) for convex primitives is formulated via conic optimization and solved using primal-dual interior-point method. Here, combined with our geometry definition described above, we present an efficient and robust algorithm to solve GD model and its differentiation. The key concept is to solve the GD model as an unconstrained nonlinear equation by exploiting the support function (7).

1) *Nonlinear equation:* Our unconstrained formulation employs the solution variables as  $x$  (i.e., normal vector for support function input) and growth factor  $\sigma \in \mathbb{R}$ , resulting in 4 dimensions. Then the conditions that the solution must satisfy are: 1) the two support points of each body corresponding to  $x$  coincide exactly when extended to  $\sigma$ ; and 2) the normal vector  $x$  has unit norm. Fig. 5 visualizes the equivalence of these conditions and the growth distance model, both for active (penetrated) and inactive (separated) contact cases. The conditions described above can be formulated by the following nonlinear equation: for given bodies  $i$  and  $j$ :

$$F(x, \sigma, q) = \begin{bmatrix} \sigma(\bar{s}_i - \bar{s}_j) + (1 - \sigma)(p_i - p_j) \\ \|x\|^2 - 1 \end{bmatrix} = 0 \quad (10)$$

where  $\bar{s}_i = \bar{s}_i(x, q_i), \bar{s}_j = \bar{s}_j(-x, q_j)$  and  $p = p(q)$ . The contact detection process is then reduced to solve (10) with respect to  $x, \sigma$  given the configuration  $q_i$  and  $q_j$ . Note that the formulation is of fixed dimension (i.e., 4) regardless of the number of vertices used. See Appendix B for the statements on uniqueness of the solution.

2) *Newton solver:* Theorem 2 ensures that  $h$  is twice-differentiable everywhere. Therefore we can always compute the Jacobian of  $F$  in (10) as follows:

$$J = \begin{bmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial \sigma} \end{bmatrix} = \begin{bmatrix} \sigma \left( \frac{d\bar{s}_i}{dx} + \frac{d\bar{s}_j}{dx} \right) & y \\ 2x^T & 0 \end{bmatrix} \quad (11)$$

$$y = R_i s_i (R_i^T x) - R_j s_j (-R_j^T x)$$

and (11) can be applied to Newton-type algorithm to solve nonlinear equation in (10). Specifically, we utilize the trust-

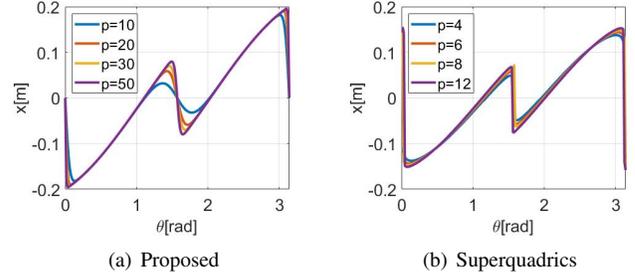


Fig. 6: Witness point change plots for two geometric modelings according to rotation angle.

region-dogleg method [33] to achieve stable convergence property. Due to the simple structure of (7),  $\frac{d\bar{s}}{dx}$  is also very easy to compute, much like  $s$  (see Appendix C for detailed derivation). Consequently (and also due to its low-dimensionality),  $J$  can be computed and solved in a highly efficient manner.

3) *Initialization:* Despite the fact that the trust-region-based method ensures the stability of algorithm, determining a good initial point is critical to practical performance. With a good initialization, the Newton-based iteration is known to have quadratic convergence. We find that the outcome of the first iteration of the internal expanding (IE) procedure presented in [51] is useful as an initial point. See Sec. VII-A for detailed results.

#### D. Feature Differentiation

After obtaining the contact features, the differential values can be computed and used to obtain the gradients for  $P$  and  $g$  from (4). The conciseness of our GD model solver also makes the process of obtaining contact feature differentiation very efficient. Applying implicit differentiation to the nonlinear equation (10), we get

$$\frac{\partial F^*}{\partial q} + J^* \begin{bmatrix} \frac{dx^*}{dq} \\ \frac{d\sigma^*}{dq} \end{bmatrix} = 0 \quad (12)$$

where the superscript  $*$  denotes the value at the solution. As  $J^*$  is only a  $4 \times 4$  matrix (and its factorization have already been computed in the solver step), we can obtain  $\frac{dx^*}{dq}$  and  $\frac{d\sigma^*}{dq}$  (i.e., differentiation of contact normal and growth factor) very efficiently. Moreover, differentiation of witness points are simply computed as

$$\frac{d\bar{s}_i^*}{dq} = \frac{\partial \bar{s}_i^*}{\partial q} + \frac{\partial \bar{s}_i^*}{\partial x} \frac{dx^*}{dq}, \quad \frac{d\bar{s}_j^*}{dq} = \frac{\partial \bar{s}_j^*}{\partial q} + \frac{\partial \bar{s}_j^*}{\partial x} \frac{dx^*}{dq} \quad (13)$$

where  $\frac{\partial \bar{s}_i^*}{\partial x}$  and  $\frac{\partial \bar{s}_j^*}{\partial x}$  are already available from the solver. Overall contact feature computation and differentiation procedure is summarized in Alg. 1.

#### E. Analysis on Degeneration

As we can see in (12),  $J^*$  should be non-singular in order to avoid a degenerated situation. Although the degeneration problem has not been well considered in previous studies, it must be addressed in order to ensure the smooth relation between variables using the implicit function theorem in

Sec. III-C. Without this consideration, pathological cases can arise as demonstrated in [5]. In this paper, we theoretically analyze the condition to avoid degeneration for our proposed framework. We start by making the following assumption.

*Assumption 1:*  $\forall x \in \mathbb{R}^3 \setminus \mathbf{0}$ , there exists at least 3 linearly independent vertices such that  $v_i^T x > 0$ .

This assumption is typically satisfied for shapes that require a sufficient number of vertices to define their geometry, but may not hold for very simple shapes such as a tetrahedron with four vertices. To satisfy the assumption in such cases, additional vertices can be added to the edges of the shape. Based on this, we present the following lemma:

*Lemma 1:*  $\frac{\partial \bar{s}}{\partial x}$  is a positive semi-definite matrix. Moreover, its rank is 2 under Assumption 1.

*Proof:* See Appendix D. ■

Based on the lemma, following theorem can be established:

*Theorem 3:*  $J^*$  is non-singular under Assumption 1.

*Proof:* First,  $x$  cannot be 0 at the solution. Now suppose that  $J^*$  is singular, therefore for a nonzero vector  $z = [z_1, z_2]^T$ ,  $z_1 \in \mathbb{R}^3, z_2 \in \mathbb{R}$ ,  $J^* z = 0$  holds i.e.,

$$\sigma \left( \frac{d\bar{s}_i}{dx} + \frac{d\bar{s}_j}{dx} \right) z_1 + z_2 y = 0 \quad (14)$$

$$x^T z_1 = 0 \quad (15)$$

By multiplying  $x^T$  to equation (14), we obtain:

$$x^T \left( \sigma \left( \frac{d\bar{s}_i}{dx} + \frac{d\bar{s}_j}{dx} \right) z_1 + z_2 y \right) = z_2 (x^T y) = 0$$

holds. From the definition of support point,  $x^T y > 0$  holds, therefore we get  $z_2 = 0$ . Now  $z_1$  is supposed to be a non-zero vector and perpendicular to  $x$  from (15). Also from the positive semi-definite property in Lemma 1, we have  $\frac{d\bar{s}_i}{dx} z_1 = 0$ , which means the row space of  $\frac{d\bar{s}_i}{dx}$  must be perpendicular to both  $x$  and  $z_1$ . Because  $x$  and  $z_1$  are perpendicular to each other, this contradicts the condition that the rank is 2. Therefore  $z$  cannot be a non-zero vector, which means  $J^*$  is non-singular. ■

The theorem provides assurance that a degenerated situation can be avoided, given certain assumptions. This property is generally applicable as the assumptions do not impose significant limitations on its usage. For demonstration, we conduct a simple experiment that plots the change in the witness point according to the rotation angle of the figure in 2D (see Appendix E for illustration and details). As depicted in Fig. 6, modeling using superquadrics always induces degeneration (i.e., non-smoothness) even though the parametric equation is smooth and strictly convex. Our method, on the other hand, is always smooth and exhibits a stiffening pattern as  $p$  increases.

*Remark 1:* Non-singular property of  $J^*$  is also useful in terms of Newton-based solver (Alg. 1), as it guarantee that the limit point of the sequence satisfies  $\|F\| = 0$ .

## VI. BI-LEVEL OPTIMIZATION SOLVER

Combined with geometry modeling described above, in this section, we present the overall gradient-based solution scheme of the estimation problem (4).

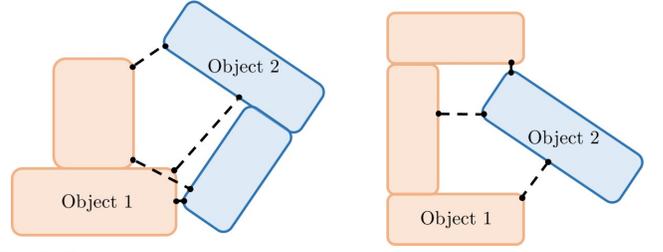


Fig. 7: Example of convex decomposition for collisions between two objects. The number of collision  $m = 4$  for the left and  $m = 3$  for the right.

### A. Predefined Number of Contact

From differentiable contact feature suggested in Sec. V,  $P(\xi)$  and  $g(\xi)$  are already differentiable. Despite this, differentiability of the overall problem is unclear because the number of contacts  $m(\xi)$  can change discretely. We address this issue by keeping the total number of collisions constant. When two (possibly non-convex) interacting objects are present, we decompose them into  $m_1$  and  $m_2$  convex geometries, respectively. Each convex geometry is represented by the method in Sec. V, therefore only a single collision occurs between them constituting different objects. Accordingly, we can predefine the collision number as constant, i.e.,  $m(\xi) = m = m_1 m_2$ . See Fig. 7 for illustrative examples. Note that defined contacts are not necessarily active. We can suppress contact forces for inactive contact by imposing the constraint  $(g_k(\xi))^+ f_k = 0$ .

### B. Differentiable Low-level Optimization

1) *Smoothing and solving:* For the fixed  $\xi$ , problem (4) reduces to find the optimal contact force  $f^*$  as

$$\min_{f \in \mathcal{C}} \frac{1}{2} \|\gamma - P(\xi) f\|_{\Sigma^{-1}}^2 \quad \text{s.t.} \quad (g_k(\xi))^+ f_k = 0$$

which is a second-order cone programming (SOCP). Here, the constraint  $(g_k(\xi))^+ f_k = 0$  is only a  $\mathcal{C}^0$  function as it includes max operator. For better smoothness, we replace it by a quadratic penalty term in the cost:

$$\min_{f \in \mathcal{C}} \frac{1}{2} \|\gamma - P(\xi) f\|_{\Sigma^{-1}}^2 + \frac{k_0}{2} \|D_g^+(\xi) f\|^2 \quad (16)$$

where  $k_0$  is a penalty coefficient,  $P = [P_1, \dots, P_m]$ , and  $D_g^+ = \text{blkdiag}(g_1^+ I_3, \dots, g_m^+ I_3)$ . Then the cost is  $\mathcal{C}^1$  function for  $\xi$ , as each  $(g_k(\xi))^+$  is squared. The problem (16) is still SOCP and compare to quadratic programming (QP) [4], it can impose the friction cone without linearization, making it more preferable. Optimality conditions of (16) can be written as

$$H f + b = J_c^T \lambda \quad (17)$$

$$0 \leq \lambda_k \perp c_k \geq 0 \quad \forall k \quad (18)$$

where  $\perp$  denotes the complementarity,  $H \in \mathbb{R}^{3m_c \times 3m_c}$ ,  $b \in \mathbb{R}^{3m_c}$ , and  $c_k \in \mathbb{R}$  are defined as

$$H = P^T \Sigma^{-1} P + k_0 (D_g^+)^2$$

$$b = -P^T \Sigma^{-1} \gamma$$

$$c_k = \mu f_{n,k} - \|f_{t,k}\|$$

where  $(\xi)$  is omitted for simplicity,  $\lambda = [\lambda_1, \dots, \lambda_m] \in \mathbb{R}^m$  is the Lagrange multiplier, and  $J_c \in \mathbb{R}^{m \times 3m}$  is the Jacobian  $\frac{dc}{df}$ . We can see that  $c_k$  is non-smooth at  $f_k = 0$ , implying that singularity can occur. Indeed, the solution of  $f_k = 0$  is often obtained, particularly in inactive contact. To relax this issue, we propose to use the following smoothed  $c_k$  instead:

$$c_k = \mu f_{n,k} - \sqrt{f_{t_1,k}^2 + f_{t_2,k}^2} + \epsilon \quad (19)$$

where  $\epsilon \in \mathbb{R}^+$  is the small positive value. Our smoothing scheme has several advantages. First, as the problem is still strictly convex, its solution set is always singleton. Also, as (19) is still analytic, we can resolve the problem efficiently using projection based methods. Specifically, we utilize the projected Gauss-Seidel (PGS) method [17] which is widely used in physics simulation. In practice, the problem is solved reliably and efficiently as PGS iteration converges to a solution in a small number of iterations.

2) *Differentiation*: To utilize the gradient method in the high-level optimization, the derivative of the solution of the low-level optimization with respect to the target parameter  $\xi$  is required. Based on the differentiable contact features in Sec. V, differentiating (17) and (18) with respect to the parameter  $\xi$  is possible, therefore

$$H \frac{df^*}{d\xi} + \frac{dH}{d\xi} f^* + \frac{db}{d\xi} = J_c^T \frac{d\lambda}{d\xi} + D_\Lambda \frac{df^*}{d\xi} \quad (20)$$

$$\frac{d\lambda_k}{d\xi} c_k + \lambda_k \frac{dc_k}{d\xi} = 0 \quad \forall k \quad (21)$$

can be obtained at the optimal solution  $f^*$  of (16) where  $D_\Lambda = \text{blkdiag} \left( \lambda_1 \frac{d^2 c_1}{df_1^2}, \dots, \lambda_m \frac{d^2 c_m}{df_m^2} \right)$ . Here we can classify (21) into two cases:  $c_k = 0$  and  $c_k > 0$ :

$$\begin{aligned} c_k = 0 : \quad & \lambda_k \frac{dc_k}{d\xi} = \lambda_k J_{c,k} \frac{df_k^*}{d\xi} = 0 \\ c_k > 0 : \quad & \frac{d\lambda_k}{d\xi} = 0 \end{aligned}$$

This allows us to exclude the components of  $\lambda$  that correspond to inactive constraints (i.e.,  $c_k > 0$ ) and reduce (20) and (21) into following form:

$$\begin{bmatrix} H - D_\Lambda & -J_{c,r}^T \\ \Lambda_r J_{c,r} & 0 \end{bmatrix} \begin{bmatrix} \frac{df^*}{d\xi} \\ \frac{d\lambda_r}{d\xi} \end{bmatrix} = \begin{bmatrix} -\frac{dH}{d\xi} f^* - \frac{db}{d\xi} \\ 0 \end{bmatrix} \quad (22)$$

where  $\lambda_r$  and  $J_{c,r}$  are the reduced Lagrange multiplier and Jacobian, respectively, and  $\Lambda_r$  is a diagonal matrix with the diagonal entries being the elements of  $\lambda_r$ . In situations with a positive definite  $H$  and no  $\lambda_k$  that simultaneously satisfy  $\lambda_k = 0$  and  $c_k = 0$ , the equation is solvable (See Appendix F for more details). Otherwise, the least-squares solution can be employed instead [2].

### C. High-level Optimization Solver

By substituting the obtained low-level solution  $f^*$  and handling the gap constraint  $g_k(\xi) \geq 0$  as penalty functions, we can formulate the high-level problem as

$$\min_{\xi} \frac{1}{2} \|\gamma - P(\xi) f^*\|_{\Sigma^{-1}}^2 + \frac{k_1}{2} \sum_{k=1}^m ((-g_k(\xi))^+)^2 \quad (23)$$

---

### Algorithm 2 Uncertain Pose Estimation in Contact

---

```

Initialize  $\xi_1, \dots, \xi_N$  by sampling
for  $i = 1$  to  $N$  do
  while not converge do
    Calculate  $f_i^*$  with  $\xi_i$  (16)
    Calculate  $\frac{df_i^*}{d\xi_i}$  by solving (22)
    Calculate the gradient of the cost function of (23)
    Update  $\xi_i$  using Gauss-Newton algorithm
  end while
end for
Determine the best  $\xi^*$  among  $\xi_1^*, \dots, \xi_N^*$ 

```

---

where  $k_1$  is the penalty coefficient to penalize penetration between objects. As we can obtain the gradient of  $f^*$ , (23) is now a non-linear least squares problem with differentiable error terms. Hence, we can use off-the-shelf algorithms such as the Gauss-Newton method to solve the problem, which also shows good convergence in practice.

Since the problem (23) is non-convex, there can be multiple local minimum. To enhance the ability of our gradient-based algorithm to discover global minimum, we adopt a strategy of sampling the initial pose parameters and selecting the optimal value from among them after optimization. The overall procedure of our differentiable uncertainty estimation is summarized in Alg. 2.

### D. Augmentation

The nonlinear least squares problem (23) can be extended by including various additional costs that reflect different aspects of the problem being solved. Some examples are as follows:

1) *Prior*: Prior knowledge of the uncertain pose parameters may be known in many cases. The following simple Gaussian prior cost can be added in this case:

$$\frac{1}{2} \|\xi - \xi_p\|_{\Sigma_p^{-1}}^2$$

where  $\xi_p$  is the prior of  $\xi$  and  $\Sigma_p$  is the covariance matrix.

2) *Bound constraint*: The bound constraint can be introduced to limit the range of uncertainty. In this case, penalty function can be utilized:

$$\frac{1}{2} \|(-\xi + \xi_l)^+\|_{\Sigma_l^{-1}}^2 + \frac{1}{2} \|(\xi - \xi_u)^+\|_{\Sigma_u^{-1}}^2$$

where  $\xi_l, \xi_u$  are the lower, upper bound of  $\xi$  and  $\Sigma_l, \Sigma_u$  are the (typically low) covariance matrix.

3) *Motion model*: The pose parameters can sometimes be estimated over multiple time intervals. In such cases, a motion model can be introduced to better estimate the pose parameters. See Sec. VII-D for an example.

## VII. RESULTS AND EVALUATIONS

In this section, various simulation and experiment results are presented to validate the proposed framework.

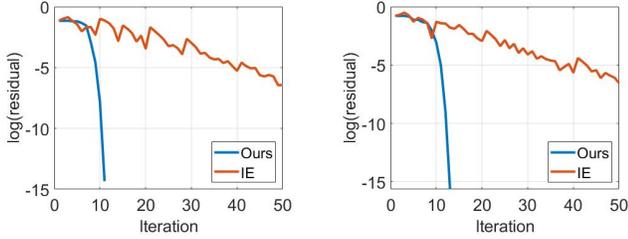


Fig. 8: Residual plots for IE and our method for 2 benchmark cases.

Solver		IE			Ours		
Max Iteration		30	60	90	10	15	20
A-M	AT ↓	50.30	85.75	134.8	23.66	27.29	29.27
	MLR ↑	3.905	5.587	6.271	5.262	8.109	9.954
M-S	AT ↓	32.83	65.08	85.00	15.95	22.95	24.21
	MLR ↑	4.813	5.785	6.629	4.103	7.554	9.578
S-A	AT ↓	45.62	75.05	108.9	22.11	24.63	26.97
	MLR ↑	4.962	5.645	5.998	5.099	8.043	9.729

TABLE I: Evaluation results for two contact feature (with its differentiation) solvers. A, M, S are abbreviations for Apple, Mustard, and Sponge, respectively. AT: average computation time ( $\mu$ s), MLR: residual converted using  $-\log(\cdot)$  before being averaged, therefore bigger is better).

### A. Collision Detection

We conduct benchmark tests to verify the usefulness of our geometric representations and contact feature computation methods. We implement the baseline algorithm for GD model as state-of-the-art internal expanding (IE) algorithm [51, 50]<sup>2</sup>, which is proven to be better than GJK based method. We employ 3 types of object from YCB dataset (Apple, Mustard, and Sponge, see Appendix G for the images). Note that our support function based geometric modeling applies to both. With the residual defined as (10), the termination condition is set based on its norm reaching  $10^{-10}$ . The performance is recorded under various max iteration number.

Comparison results in Table I demonstrate that our method outperforms the IE algorithm. It achieves faster and more accurate convergence, typically within 20 iterations. Fig. 8 illustrates the convergence behavior, with our method showing quadratic convergence after a few iterations, while the IE algorithm exhibits first-order convergence. This showcases the advantage of our Newton-type method utilizing the differential value of contact features.

### B. External Contact Localization

External contact localization problem [25], that determines where the contact occurred on the robot arm, is one of the basic examples of Problem 1. In this case, the uncertain parameter  $\xi \in \mathbb{R}^3$  is the collision point, and the measurement  $\gamma \in \mathbb{R}^7$  is obtained from the joint torque sensor. We assume that

<sup>2</sup>While combination of IE with convex cone projection [49] was also proposed, we find that using IE alone is more suitable for our 3D cases.

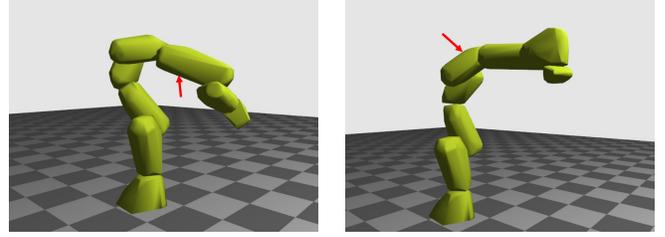


Fig. 9: 7-DoF manipulator where each link consists of differentiable collision geometry

Noise		Low			High		
Methods		PF	AGD	Ours	PF	AGD	Ours
Arm 5	AT ↓	8.94	7.86	4.10	9.00	8.03	3.95
	MLE ↑	1.94	2.43	3.29	1.84	1.91	2.05
	MLC ↑	2.21	3.47	5.02	1.65	2.05	2.21
Arm 6	AT ↓	9.46	7.64	3.95	9.38	8.04	3.93
	MLE ↑	2.13	3.17	3.67	1.99	2.12	2.19
	MLC ↑	1.99	3.90	5.03	1.75	2.16	2.39
Arm 7	AT ↓	13.1	12.0	5.27	13.2	11.7	5.39
	MLE ↑	2.27	3.55	4.08	2.16	2.44	2.37
	MLC ↑	2.05	4.33	5.29	1.62	2.15	2.18

TABLE II: Evaluation results for the external contact localization. AT: average computation time (ms), MLE/MLC: position error (m) and cost value converted using  $-\log(\cdot)$  before being averaged, therefore bigger is better.

contact occurs at a single point on a given link<sup>3</sup>, therefore  $m = 1$ . Existing contact localization algorithms rely heavily on sampling and retraction of points on the mesh, which is computationally expensive. On these, we verify the efficacy of our differentiable framework here.

For the test cases, Franka Emika Panda [1] is used, while its links are represented by a convex hull of CAD data, as shown in Fig. 9. Two baseline algorithms are employed for comparison in our study. The first algorithm is a particle filter (PF)-based method widely used in the literature [19, 25]. In each iteration, every particle is updated based on the outcome of low-level problems and subsequently projected onto the mesh. The second baseline algorithm is a more recent approach that utilizes an approximated gradient descent (AGD) combined with low-level problem differentiation [32]. Here the gradient is approximated, as certain terms are disregarded. Additionally, a projection step to the mesh is still required since the derivative of contact features such as the gap and normal vector is unavailable. Total 1000 trials are conducted and for each trial, a random force is applied to a random position on a link, and the accuracy and computation time are recorded. For the contact particle filter method, we use 100 particles and iterated for 50 times for convergence. For other two methods (AGD and ours) an initial 10 randomly sampled points from the surface of geometry are used as initial

<sup>3</sup>Here, the contact is point-geometry contact, while the preceding contents mainly describe geometry-geometry contact. However, the problem is still a subset of Problem 1.

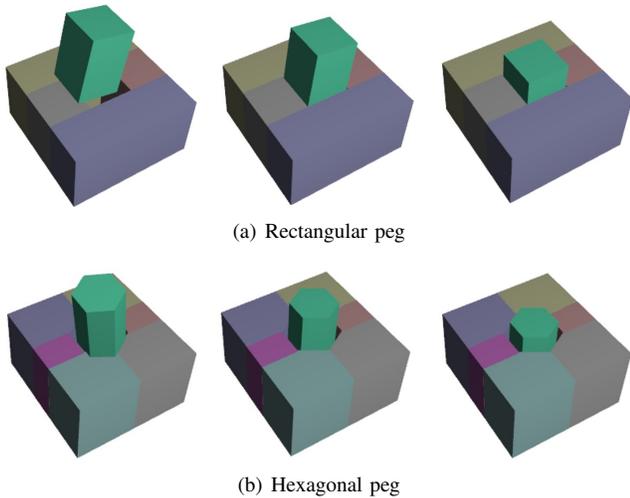


Fig. 10: Snapshots of simulation results of peg-in-hole manipulation using our uncertain pose estimation framework in online. Different colors are used to represent convex-decomposed shapes.

guesses. Also, for both of the baseline methods, the low-level optimization is performed using the same PGS-based approach employed in our method.

The Table II shows the result of external contact localization test, on various links under low/high sensor noise. As expected, the particle filter exhibits the lowest performance due to the necessity of conducting the lower-level optimization for each particle and relying on exploration through randomness. Furthermore, the convergence behavior of AGD is inferior to ours because it is limited to first-order methods with an approximated gradient and necessitates projection. In contrast, our method leverages second-order Gauss-Newton optimization with the exact gradient, leading to improved convergence.

### C. Peg-in-Hole

Next, the proposed framework is tested on estimating the uncertain grasp pose (i.e., the pose of the peg with respect to the gripper) in peg-in-hole assembly task. Here the uncertain parameter  $\xi \in \mathbb{R}^3$  is the parameterized grasp pose (see Appendix H for details) and the measurement  $\gamma \in \mathbb{R}^6$  is from the force/torque sensor on gripper. We assume that the gripper and hole poses are known.

The experiment employs two distinct peg geometries: a rectangular prism and a hexagonal prism. The rectangular prism has eight vertices, while the hexagonal prism has twelve vertices. As shown in Fig. 10, the hole is decomposed into a total of 4 and 6 convex geometries, and the predefined numbers of collisions  $m$  are 4 and 6, respectively.

For the evaluation, we first collect simulation data (FT measurement, ground-truth grasp pose) in a contact situation using the original geometry. Here, the data accumulated over three contacts (i.e.,  $\gamma \in \mathbb{R}^{18}$ ) is used. The identification is then performed using the proposed differentiable contact feature, with three initial samples. For the baseline, we implement the particle filter (PF)-based method similar to [36]. The PF solves the high-level problem by using the grasp pose as particles

Noise		Low			High		
Methods		Ours	PF25	PF50	Ours	PF25	PF50
Rect	AT ↓	9.22	44.1	89.1	10.7	43.7	89.3
	MLPE ↑	4.54	1.91	2.08	3.34	1.99	2.22
	MLRE ↑	3.72	1.06	0.94	2.47	0.83	1.16
	MLC ↑	5.74	-0.912	-0.66	2.03	-0.70	-0.19
Hexa	AT ↓	18.6	100	197	16.6	99.4	192
	MLPE ↑	3.71	2.21	2.37	3.87	2.47	2.34
	MLRE ↑	2.58	0.87	1.10	2.50	1.06	1.07
	MLC ↑	3.28	-0.59	0.36	1.66	0.46	0.19

TABLE III: Evaluation results for the peg-in-hole assembly task. AT: average computation time (ms). MLPE/MLRE/MLC: position error (m), rotation error (rad) and cost value converted using  $-\log(\cdot)$  before being averaged, therefore bigger is better.

Noise		Low			High		
$p$		50	60	70	50	60	70
Rect	AT ↓	8.98	9.10	9.22	9.71	9.31	10.7
	MLPE ↑	2.92	3.27	4.54	2.75	3.02	3.34
	MLRE ↑	2.11	2.48	3.72	1.91	2.29	2.47
	MLC ↑	3.43	4.32	5.74	2.03	1.95	2.03
Hexa	AT ↓	15.5	18.2	18.6	16.5	16.1	16.6
	MLPE ↑	2.94	3.19	3.71	3.00	3.15	3.87
	MLRE ↑	1.97	2.13	2.58	2.06	2.03	2.50
	MLC ↑	1.75	2.11	3.28	1.32	1.24	1.66

TABLE IV: Evaluation results under various smoothing parameters. AT: average computation time (ms). MLPE/MLRE/MLC: position error (m), rotation error (rad) and cost value converted using  $-\log(\cdot)$  before being averaged, therefore bigger is better.

with sampling strategy. For the low-level problem for each particle, we take the same methodology of our framework for better performance. Also, the number of particles is 25 (PF25) and 50 (PF50).

The comparison results are summarized in Table III. A total of ten datasets and two different amounts of noise (standard deviations of 0.1 and 0.001) are used. The results clearly demonstrate that the proposed method outperforms the particle filter-based method in terms of accuracy and efficiency. This highlights how the Gauss-Newton algorithm, utilizing gradients, enables rapid convergence to a solution with non-penetration and proper normal/witness points.

Furthermore, a comparative study is conducted by varying the smoothing parameters within our framework. Specifically, the smoothing parameter  $p$  is varied from 50 to 70. The results are presented in Table IV. It can be observed that lower values of  $p$  result in slightly shorter average computation times. Conversely, higher values of  $p$  yield more accurate results as they are closer to the original geometry. Consequently, future investigations could focus on finding fast approximated solutions through proper  $p$ -smoothing and refining them towards the exact geometry under higher values of  $p$ .

Additionally, the estimation process can be performed online, involving repeated trials and data augmentation until the task is completed. Simulation snapshots of the peg-in-hole assembly with online estimation are depicted in Fig. 10. For the video, please refer to our supplementary material. Further visualization results can be found in the Appendix H.

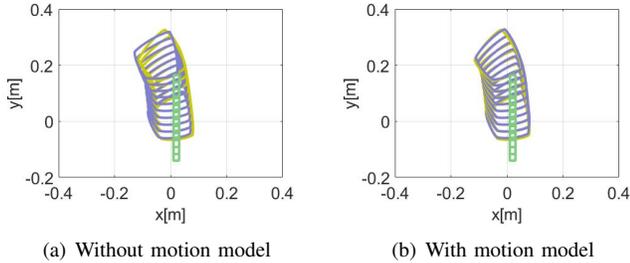


Fig. 11: Comparison result of the blind object tracking performance with/without motion model. Green: End effector. Yellow: ground-truth. Purple: Estimation result.

#### D. Augmentation: Blind Object Tracking

This subsection provides an example of augmenting our framework with other models, as explained in Section VI-D. Specifically, we focus on blind object tracking, which involves tracking an object without relying on visual information during the task. This capability proves beneficial in cluttered environments or areas with limited lighting. To demonstrate blind object tracking, we configure a pushing environment where the interacting objects are represented by convex geometries based on four vertices. It is worth noting that previous studies [48, 37] have tackled similar tasks; however, many of them simplified the shape of the tip to a point. In contrast, our framework allows for a more versatile geometric representation, enabling its applicability to a broader range of end effectors and object shapes. However, still diverse real-world scenarios are remained for future research.

The uncertain parameter  $\xi$  and the FT measurement  $\gamma$  are stack of values for multiple time intervals. Here, we adopt the quasi-static motion model based on limit surface [23, 37] for augmented cost. Note that all components in the model is a function of  $\xi$  and  $f^*$  therefore can be efficiently differentiated. Ground truth data is obtained from the simulation environment and compared to the estimated results. The results are illustrated in Fig. 11. The result from our vanilla cost formulation (23) without motion model exhibits a noticeable bias error. Conversely, when incorporating the motion model, the results demonstrate a substantial improvement in accuracy (reducing the RMSE by 30%).

#### E. Real World Experiment: Dish Placing

We deploy our framework in a dish placement task for experimental validation in the real world. The manipulator is built with Franka Emika Panda and a parallel gripper, and ATI Gamma is utilized as the FT sensor. Three different dishes are used, with a narrow-spaced dish rack. Test is conducted as follow: a human makes the gripper to grasp the dish in an arbitrary pose, and the robot identifies the uncertain grasp pose through interaction with the ground. In the identification process, Alg. 2 is employed while the uncertain grasp parameter is modeled in 3-dimension and the dishes are represented by a smoothed convex hull with a prescribed support function. Following the identification, the

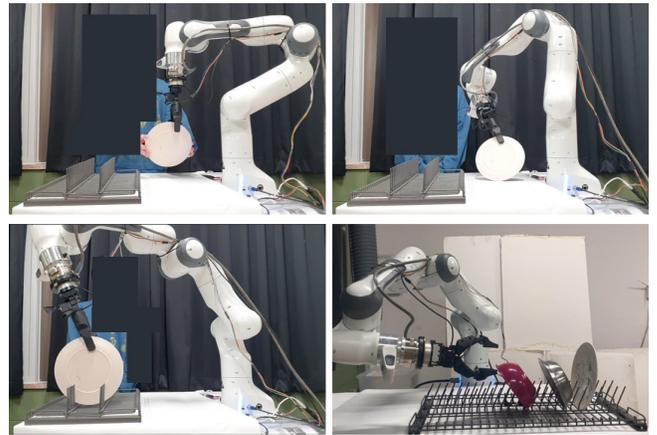


Fig. 12: Experimental demonstration of our framework in dish placing task. Top left: A human gives an arbitrary grasp pose. Top right: The robot estimates the uncertainty through interaction with the ground. Bottom left: Placing succeeded by proper estimation. Bottom right: Three dishes are successfully placed in a row.

placing is carried out by following the pre-planned trajectory. If the grasp pose is not estimated correctly, the placement will fail with a stuck or jamming. Our framework is successfully applied to enable successful performance of dish placement tasks - see Fig. 12 for experiment snapshots. See also our supplementary materials for video and more details.

## VIII. DISCUSSIONS AND CONCLUSION

In this paper, we propose a novel uncertain pose estimation framework for interactive robot tasks. Essentially, we frame the problem as bi-level optimization and devise a way to solve it based on gradient. Prescribed support function based geometry definition is first presented to make it possible to express differentiable contact features. The definition also comes with an effective solver algorithm and has useful theoretical properties. Then by using the predefined number of contacts and differentiating low-level problems, the original problem is finally transformed into a non-linear least squares problem, which can be solved efficiently using conventional gradient-based methods. Several scenarios are implemented and demonstrate how well our method can outperform currently used sampling-based approaches.

There exists several possible directions for future works. First, our method is mainly to utilize FT or joint torque sensor information, so combination with more diverse sensors will be useful. Specifically, embedding the differentiable nonlinear least square derived in our work to general factor graph optimization form will be an important task. It would also be meaningful to develop a way to handle situations that uncertainty exists in geometry parameters as well as poses. In a similar vein, specific methodologies for extracting prescribed support function from visual information will be an important topic. Finally, since our method essentially consists of model-based optimization, it will be interesting to combine it with learning-based methods by modeling it as a single layer [4].

## ACKNOWLEDGEMENT

This research was supported by Samsung Research and the RS-2022-00144468 of the National Research Foundation (NRF) funded by the Ministry of Science and ICT (MSIT) of Korea.

## REFERENCES

- [1] Franka emika. <https://www.franka.de/>.
- [2] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J Z. Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.
- [3] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.
- [4] B. Amos and J Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145, 2017.
- [5] J. Bolte, T. Le, E. Pauwels, and T. Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. *Advances in neural information processing systems*, 34:13537–13549, 2021.
- [6] Y. Deng, X. and Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox. Self-supervised 6d object pose estimation for robot manipulation. In *IEEE International Conference on Robotics and Automation*, pages 3665–3671, 2020.
- [7] N. Dyn and W. E Ferguson. The numerical solution of equality constrained quadratic programming problems. *Mathematics of Computation*, 41(163):165–170, 1983.
- [8] C D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax—a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.
- [9] 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 Transactions on Graphics*, 39(6): 1–15, 2020.
- [10] E. G Gilbert, D. W Johnson, and S S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, 1988.
- [11] S. Haddadin, A. De Luca, and A. Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.
- [12] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S Sukhatme. Neursim: Augmenting differentiable simulators with neural networks. In *IEEE International Conference on Robotics and Automation*, pages 9474–9481, 2021.
- [13] C. Higuera, S. Dong, B. Boots, and M. Mukadam. Neural contact fields: Tracking extrinsic contact with tactile sensing. *arXiv preprint arXiv:2210.09297*, 2022.
- [14] 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.
- [15] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.
- [16] S. Jin, X. Zhu, C. Wang, and M. Tomizuka. Contact pose identification for peg-in-hole assembly under uncertainties. In *American Control Conference*, pages 48–53, 2021.
- [17] F. Jourdan, P. Alart, and M. Jean. A gauss-seidel like algorithm to solve frictional contact problems. *Computer methods in applied mechanics and engineering*, 155(1-2):31–47, 1998.
- [18] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *IEEE International Conference on Robotics and Automation*, pages 10241–10247, 2022.
- [19] M. C Koval, N. S Pollard, and S. S Srinivasa. Pose estimation for planar contact manipulation with manifold particle filters. *The International Journal of Robotics Research*, 34(7):922–945, 2015.
- [20] 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 and Automation Letters*, 8(5):2780–2787, 2023.
- [21] Q. Le Lidec, I. Kalevatykh, I. Laptev, C. Schmid, and J. Carpentier. Differentiable simulation for physical system identification. *IEEE Robotics and Automation Letters*, 6(2):3413–3420, 2021.
- [22] M. A Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *IEEE International Conference on Robotics and Automation*, pages 8943–8950, 2019.
- [23] K. M Lynch, H. Maekawa, and K. Tanie. Manipulation and active sensing by pushing using tactile feedback. In *IEEE/RSJ international conference on intelligent robots and systems*, volume 1, pages 416–421, 1992.
- [24] D. Ma, S. Dong, and A. Rodriguez. Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements. In *IEEE international conference on robotics and automation*, pages 11262–11268, 2021.
- [25] L. Manuelli and R. Tedrake. Localizing external contact using proprioceptive sensors: The contact particle filter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.
- [26] N. Mavrakis and R. Stolkin. Estimation and exploitation of objects’ inertial parameters in robotic grasping and manipulation: A survey. *Robotics and Autonomous Systems*, 124:103374, 2020.
- [27] 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. *Communications of the ACM*, 65(1):99–106, 2021.
- [28] L. Montaut, Q. L. Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier. Differentiable collision detection: a randomized smoothing approach. *arXiv preprint arXiv:2209.09012*, 2022.
- [29] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112, 2020.
- [30] C. J. Ong and E. G Gilbert. Growth distances: New measures for object separation and penetration. *IEEE Transactions on Robotics and Automation*, 12(6):888–903, 1996.
- [31] J. Pan, S. Chitta, and D. Manocha. Fcl: A general purpose library for collision and proximity queries. In *IEEE International Conference on Robotics and Automation*, pages 3859–3866, 2012.
- [32] T. Pang, J. Umenberger, and R. Tedrake. Identifying external contacts from joint torque measurements on serial robotic arms and its limitations. In *IEEE International Conference on Robotics and Automation*, pages 6476–6482, 2021.
- [33] D. M Rosen, M. Kaess, and J. J Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *IEEE International Conference on Robotics and Automation*, pages 1262–1269, 2012.
- [34] R. Schneider. *Convex bodies: the Brunn–Minkowski theory*. Number 151. Cambridge university press, 2014.
- [35] 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. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.
- [36] A. Sipos and N. Fazeli. Simultaneous contact location and object pose estimation using proprioception and tactile feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.
- [37] S. Suresh, M. Bauza, K-T. Yu, J. G Mangelson, A. Rodriguez, and M. Kaess. Tactile slam: Real-time inference of shape and pose from planar pushing. In *IEEE International Conference on Robotics and Automation*, pages 11322–11328, 2021.
- [38] S. Suresh, Z. Si, S. Anderson, M. Kaess, and M. Mukadam. Midastouch: Monte-carlo inference over distributions across sliding touch. In *Conference on Robot Learning*, 2022.
- [39] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ international conference on intelligent robots and systems*, pages 23–30, 2017.
- [40] K. Tracy, T. A Howell, and Z. Manchester. Differentiable collision detection for a set of convex primitives. *arXiv preprint arXiv:2207.00669*, 2022.
- [41] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Grasp’d: Differentiable contact-rich grasp synthesis for multi-fingered hands. In *European Conference on Computer Vision*, pages 201–221, 2022.
- [42] M. B. Villalonga, A. Rodriguez, B. Lim, E. Valls, and T. Sechopoulos. Tactile object pose estimation from the first touch with geometric contact rendering. In *Conference on Robot Learning*, pages 1015–1029, 2021.
- [43] 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 International Conference on Robotics and Automation*, pages 968–974, 2021.
- [44] S. Wang, A. Bhatia, M. T Mason, and A. M Johnson. Contact localization using velocity constraints. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7351–7358, 2020.
- [45] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018.
- [46] J. Xu, T. Chen, L. Zlokapa, M. Foshey, W. Matusik, S. Sueda, and P. Agrawal. An end-to-end differentiable framework for contact-aware robot design. In *Robotics: Science and Systems*, 2021.
- [47] J. Xu, V. Makovychuk, Y. Narang, F. Ramos, W. Matusik, A. Garg, and M. Macklin. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations*, 2021.
- [48] K-T. Yu, J. Leonard, and A. Rodriguez. Shape and pose recovery from planar pushing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1208–1215, 2015.
- [49] Y. Zheng and C. Chew. Distance between a point and a convex cone in  $n$ -dimensional space: Computation and applications. *IEEE Transactions on Robotics*, 25(6): 1397–1412, 2009.
- [50] Y. Zheng and K. Yamane. Ray-shooting algorithms for robotics. *IEEE Transactions on Automation Science and Engineering*, 10(4):862–874, 2013.
- [51] Y. Zheng, M. C Lin, and D. Manocha. A fast  $n$ -dimensional ray-shooting algorithm for grasping force optimization. In *IEEE International Conference on Robotics and Automation*, pages 1300–1305, 2010.

## A. Additional visualizations

Fig. 13 shows additional visualizations of equations presented in the main contents.

## B. On the uniqueness of the solution of (10)

There are two possible solutions for (10), one with a positive  $\sigma$  and one with a negative  $\sigma$ . In order to ensure appropriate collision detection, the constraint  $\sigma > 0$  is necessary. To impose  $\sigma > 0$ , a few simple but additional steps are required in the Newton step, but we find that those are not really necessary under proper initialization of  $\sigma$  (in our cases, via IE process). Thus, we take an unconstrained approach to the problem.

 C. Derivation of  $\frac{d\bar{s}}{dx}$ 

We will derive  $\frac{d\bar{s}}{dx}$ , as  $\frac{d\bar{s}}{dx}$  is straightforwardly obtained from it. To simplify notation, let us define  $\hat{a}_k = [a_1^k, \dots, a_n^k]^T$  with  $a_i = (v_i^T x)^+$  and  $\tilde{a}_p = \sum \hat{a}_p$ . Then we have

$$s(x) = V(\tilde{a}_p)^{\frac{1}{p}-1} \hat{a}_{p-1}$$

$$\frac{ds}{dx} = (p-1)V \underbrace{\left( (\tilde{a}_p)^{\frac{1}{p}-1} \text{diag}(\hat{a}_{p-2}) - (\tilde{a}_p)^{\frac{1}{p}-2} \hat{a}_{p-1} \hat{a}_{p-1}^T \right)}_A V^T$$

where  $\text{diag}(\cdot)$  denotes the diagonal matrix from a given vector. Note that the actual computation flow computes the  $3 \times 3$  matrix after computing the  $3 \times 1$  vector  $V\hat{a}_{p-1}$ , so the complexity is  $\mathcal{O}(n)$ .

## D. Proof of Lemma 1

Let us first prove the positive semi-definite property. It is sufficient to show the positive semi-definite property of  $A$ . Consider a  $n$ -dimensional vector  $u = [u_1, \dots, u_n]^T$ . Then

$$u^T A u = \tilde{a}_p u^T \text{diag}(\hat{a}_{p-2}) u - (u^T \hat{a}_{p-1})^2$$

holds. As Cauchy-Schwarz inequality indicates

$$(a_1^p + \dots + a_n^p)(a_1^{p-2} u_1^2 + \dots + a_n^{p-2} u_n^2)$$

$$\geq (a_1^{p-1} u_1 + \dots + a_n^{p-1} u_n)^2$$

it can be confirmed that  $u^T A u \geq 0$ , which means  $A$  is positive semi-definite. Now let us show the rank property. It is well known that  $Au = 0$  holds if and only if  $u^T A u = 0$ , if  $A$  is a positive semi-definite matrix. Then as  $u = V^T x$  holds the equality condition of Cauchy-Schwarz inequality, rank of  $\frac{ds}{dx}$  is lower than 2. Now suppose that there exists  $x'$  such that  $u' = V^T x'$  meets the equality condition. From the assumption, at least three components of  $\hat{a}_1$  are non-zero. Without loss of generality, let us consider  $a_1, a_2, a_3$  are non-zero. Then  $V_{nz}^T x$  and  $V_{nz}^T x'$  must be parallel, with  $V_{nz} = [v_1, v_2, v_3]$ . Finally, as  $V_{nz}$  is full rank from the assumption,  $x'$  is parallel to  $x$ , and we conclude that the rank of  $\frac{ds}{dx}$  is always 2.

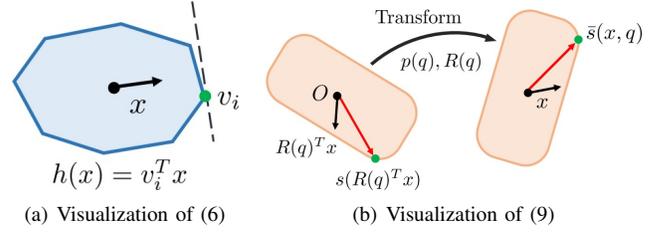


Fig. 13: Visualizations of equations. Left: Support function and point for a vertex set in (6). Right: Support point for SE(3) transformation of body in (9).

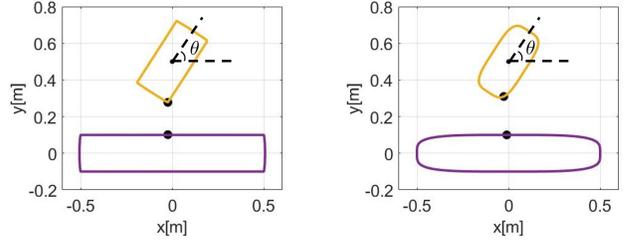


Fig. 14: Illustrations for the degeneration test in Sec. V-E. Witness points (block dots) are recorded as the rotation angle  $\theta$  changes. Left: our support function based modeling. Right: Superquadrics.

## E. Details on Degeneration Test in Sec. V-E

Illustrations for the degeneration test conducted in Sec. V-E is visualized in Fig. 14. Each rectangle shape is represented by 4 vertices in our geometry model. Superquadric model can be written as following equation:

$$\left( \frac{x}{\alpha_1} \right)^p + \left( \frac{y}{\alpha_2} \right)^p = 1$$

where  $p \in \mathbb{R}^+$  is the smoothing parameter similarly to in (7), and  $\alpha_1, \alpha_2 \in \mathbb{R}$  are the size parameters.

## F. Invertibility of (22)

Jacobian and Hessian of  $c_k$  can be written as

$$\frac{dc_k^*}{df_k} = \left[ -\frac{2f_{t_1}}{(f_{t_1}^2 + f_{t_2}^2 + \epsilon)^{\frac{1}{2}}} \quad -\frac{2f_{t_2}}{(f_{t_1}^2 + f_{t_2}^2 + \epsilon)^{\frac{1}{2}}} \quad \mu \right]$$

$$\frac{d^2 c_k^*}{df_k^2} = -\frac{2}{(f_{t_1}^2 + f_{t_2}^2 + \epsilon)^{\frac{3}{2}}} \begin{bmatrix} f_{t_2}^2 & f_{t_1} f_{t_2} & 0 \\ f_{t_1} f_{t_2} & f_{t_1}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We can find that Jacobian is always rank 1, as  $f_k$  cannot be 0 to satisfy  $c_k \geq 0$ . Also, Hessian is always negative semi-definite. We also know that  $\lambda_k \geq 0$ , so  $D_\Lambda$  is negative semi-definite. Thus, if  $H$  is positive definite and  $\lambda_k > 0$ , then  $H - D_\Lambda$  is also positive definite and invertible. Finally, Theorem 2.1 in [7] concludes the invertibility of the problem.

## G. Geometries for Collision Detection Test

Fig. 15 shows the objects utilized in the collision detection test conducted in Sec. VII-A.

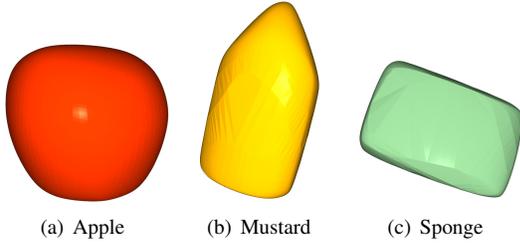
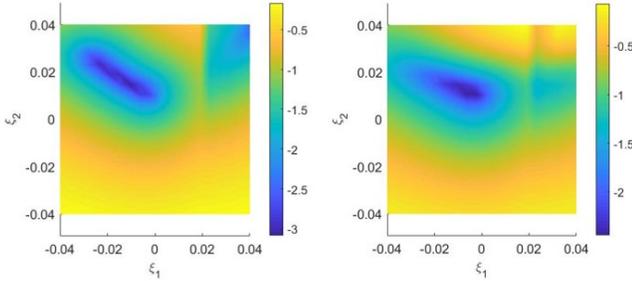
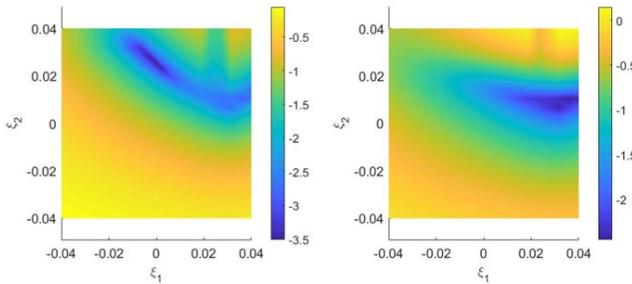


Fig. 15: Images of the objects used in the collision detection benchmark



(a) Scenario1 (Left: after first touch, Right: after second touch)



(b) Scenario2 (Left: after first touch, Right: after second touch)

Fig. 16: Images of the objects used in the collision detection benchmark

#### H. Additional details and results for peg-in-hole task

1) *Cost landscape*: To assess validity of the optimization-based formulation and the impact of multiple interactions, we visualize the cost landscape. For more intuitive interpretation, we assume that the uncertainty exists in the  $x$  and  $y$  positions of the hole, while the grasped peg pose are known (therefore,  $\xi \in \mathbb{R}^2$ ). Also here, rectangular peg is employed.

Fig. 16 illustrates the cost landscape obtained from our differentiable framework for the problem. See also Fig. 1 for the optimization path on the landscape. As depicted, the solution initially obscured with multiple minima, becomes more apparent as interaction is added. This observation highlights the potential of our method in generating interesting results when integrated with active sensing. It suggests that incorporating additional interactions can enhance the identification and clarity of the optimal solution.

2) *Grasp parameterization*: Fig. 17 provides a visualization of how grasping is modeled. In the scenarios described in Sec. VII-C, the pose of the grasped peg can be effectively represented with just 3 parameters, offering intuitive under-

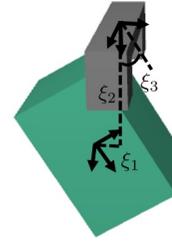


Fig. 17: Parameterization of grasp pose for a rectangular peg. The parameterization is also similarly defined for a hexagonal peg.

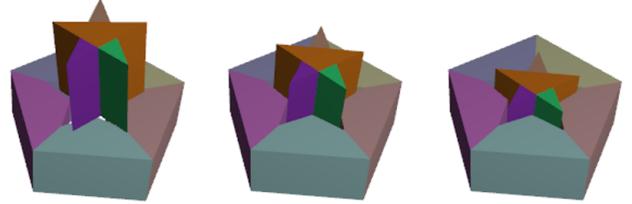


Fig. 18: Snapshots of simulation results of star-shaped peg-in-hole manipulation using our uncertain pose estimation framework in online. Different colors are used to represent convex-decomposed shapes.

standing. However, for more intricate shapes of pegs and grippers, 6 parameters can be employed, while incorporating non-penetration constraints. Exploring scenarios that encompass these complexities would present an intriguing avenue for future research.

3) *Star-shaped geometry*: To test our approach on more complex geometries, we implement a star-shaped peg-in-hole scenario. In this setup, both the peg and the hole are decomposed into five convex geometries, and a total of 25 collisions are pre-defined. We validate the effectiveness of our method in successfully identifying and executing tasks in this scenario, as demonstrated in Fig. 18 and the supplementary video.