SpringGrasp: Synthesizing Compliant, Dexterous Grasps under Shape Uncertainty

Sirui Chen, Jeannette Bohg, C. Karen Liu
Department of Computer Science, Stanford University
{ericcsr, bohg, karenliu}@cs.stanford.edu

***Abstract*—Generating stable and robust grasps on arbitrary objects is critical for dexterous robotic hands, marking a significant step towards advanced dexterous manipulation. Previous studies have mostly focused on improving differentiable grasping metrics with the assumption of precisely known object geometry. However, shape uncertainty is ubiquitous due to noisy and partial shape observations, which introduce challenges in grasp planning. We propose SpringGrasp planner, a planner that considers uncertain observations of the object surface for synthesizing compliant dexterous grasps. A compliant dexterous grasp could minimize the effect of unexpected contact with the object, leading to a more stable grasp with shape-uncertain objects. We introduce an analytical and differentiable metric, SpringGrasp metric, that evaluates the dynamic behavior of the entire compliant grasping process. Planning with SpringGrasp planner, our method achieves a grasp success rate of 89% from two viewpoints and 84% from a single viewpoints in experiment with a real robot on 14 common objects. Compared with a force-closure-based planner, our method achieves at least 18% higher grasp success rate.**

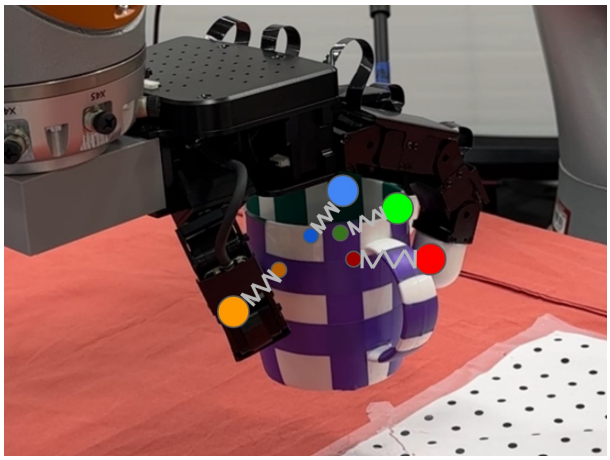


Fig. 1: Compliant grasp as virtual spring attached from finger to object

I. Introduction

Grasping with multi-fingered dexterous hands is an essential manipulation capability that enables a robot to interact with everyday objects - especially those with irregular shapes, such as different fruits and bottles. Previous work addresses the dexterous grasp planning problem mainly from two perspectives: data-driven grasp generation [34, 4] and optimization based grasp generation using differentiable grasp criteria such as force closure [28, 51] or min-weight [24].Most optimization based methods aims to obtain a set of stable contact points on the object surface for each fingertip, which requires a detailed object model to be available. However, perception in the real world usually relies on noisy sensors with limited viewpoints, resulting in uncertainty about the object shape that poses significant challenges to previous works. Data-driven method can work with partial and noisy observation[4, 34], but most of them focus on simplified problem of choosing a few fixed grasp candidates without optimizing for object and grasp stability.

To grasp objects from noisy and partial observations in the real world, we argue that every component in the grasping pipeline should consider the impact of shape uncertainty. The object perception model needs to quantify shape uncertainty from the sensor data [12, 11, 16], the grasp planner needs to leverage uncertain object information to make optimal plans [26, 30], and finally the grasp controller also needs to be compliant to tolerate unexpected contact between fingertips and the object due to shape uncertainty. An optimization based method that can efficiently optimize compliant pregrasps while considering object shape uncertainty is essential for achieving such tasks.

We propose SpringGrasp, a planner that considers uncertain observations of the object surface for synthesizing compliant dexterous grasps. A complaint grasp consists of an optimized pregrasp hand pose and optimized per-finger impedance controls. Unlike the conventional definition of a grasp, we view a compliant grasp as a dynamic process starting from the pregrasp pose, grasping the object according to the per-finger impedance controls, and moving the object along with the finger motion until a stable equilibrium is reached. As shown in Fig.1, a compliant grasp in contact can be modelled as virtual springs connected to the object. Critical to the optimization of SpringGrasp, we introduce an analytical and differentiable metric to evaluate whether the compliant grasping process can lead to a force-closure grasp at equilibrium. In contrast to existing methods that solve for contact locations, our optimized compliant grasp allows for direct control over force magnitudes, force directions and impedance parameters. As such, the unexpected contacts with the object can be minimized, leading to more stable manipulation with uncertain object shapes.

We evaluate our method on a real robotic platform consisting of an Allegro hand and RGB-D cameras. We demonstrate that our method can grasp objects from noisy and partial observations with at least 18% higher success rate

compared to baselines. Our method can even achieve 84% grasp success rate with a single depth camera input thanks to our optimized compliant grasp. We release our grasp planner in a python package at https://github.com/Stanford-TML/SpringGrasp_release.

II. Related Work

A. Optimization based methods for dexterous grasp

Synthesizing precision grasps by solving an optimization problem has been extensively studied over the years. Most research integrates an analytical grasp metric into the objective function or constraints of the optimization problem [38], utilizing either gradient-based or sampling-based optimizers. Recent studies have focused on optimizing robust and kinematically reachable contact points between the hand and object, using either an explicit mesh model of the object [51, 24] or an implicit signed distance function (SDF) [49, 28]. However both, a detailed mesh or SDF are challenging to obtain in the real world due to noisy sensors and partial observability.

The design of the grasp metric is crucial, as it determines both the quality of the synthesized grasp and the optimization runtime. Recent works [28, 24] have demonstrated that gradient-based optimization can significantly accelerate the grasp optimization process. Therefore, our comparison focuses on differentiable grasp metrics, among which the most widely used is the differentiable force closure criterion [35, 51, 28]. Given a set of contact locations $\boldsymbol{p}_i$ and normals $\boldsymbol{n}_i$, this criterion assesses whether there exists a set of nonzero contact forces $\boldsymbol{F}_i$ at each fingertip that can achieve force and torque equilibrium $\sum_i F_i = 0, \sum_i \boldsymbol{r} \times F_i = 0$ within the Coulomb friction cone $\boldsymbol{F}_i \cdot \boldsymbol{n}_i > \frac{1}{\sqrt{1+\mu}}||\boldsymbol{F}_i||$. An alternative metric, the Min-weight metric [24], also measures the robustness of a grasp under force perturbations and speeds up convergence during optimization. However, evaluating force closure involves solving a convex optimization problem, which is not analytically solvable. Therefore, optimizing a grasp with the force closure criterion and under kinematics constraints often necessitates bi-level optimization, which is typically slow to evaluate and challenging to optimize using off-the-shelf gradient-based optimizers [24, 28, 41]. DexGraspNet [49] employs an analytical criterion to approximate force closure, but this does not guarantee force closure of the solution. As an alternative to bi-level optimization, there are also many works that sample initial pregrasps using various heuristics and then either optimize a final grasp from this starting point or do rejection sampling based on the force closure criterion. We refer to [4] for more details. Our method uses a novel compliant grasp formulation that models the grasp as a dynamic process and a novel differentiable grasp metric that measures contact feasibility during the process and force closure at the equilibrium state.

B. Dexterous grasping under shape uncertainty

In many scenarios, a robot does not have access to accurate object models and must rely on perceiving the object through its depth sensors. The resulting perceived object model often contains shape uncertainties due to sensor noise and partial observability. As demonstrated for example in [26, 12, 13], grasping a shape-uncertain object with a dexterous hand requires a system that considers shape uncertainty arising from perception.

1) Grasping shape uncertain objects with compliance: Compliance plays a crucial role in enabling robots to robustly grasp and manipulate objects with varying geometries [1, 40, 14]. Numerous studies have demonstrated this capability using passive compliant multi-fingered hands [37, 25, 43, 40], where compliance is derived from the mechanical design. However, passive compliance often leads to a reduction in the number of actuated degrees of freedom and poses challenges in accurately modeling kinematics. Both of these issues make it challenging to synthesize and execute precision grasps. In contrast, active compliant multi-fingered hands [53, 36, 26, 22] address these limitations by simulating compliance through control of motorised joints. Many previous works have focused on power grasps with passive compliance [37, 1, 25, 40, 14]. However, in this work we are interested in compliant precision grasps such as considered in for example [26, 36, 50]. Among these works [26] adapt joint level compliance based on surface variance to achieve a precision grasp while [50, 36] focus on how to control the object compliantly after a precision grasp. Our work focus on generating grasps using Cartesian space impedance control with optimized gains. At the same time, impedance control with different gains optimized for different tasks has been shown effective in contact rich manipulation tasks and robot locomotion [5, 3], while [5] also consider changing the impedance through time, Our method automatically determined appropriate controller gains per grasp and object. Instead of picking impedance controller gain manually as in [3] or choosing by reinforcement learning in [5], our method optimize controller gains based on each object's uncertainty and geometry.

2) Modelling shape uncertain objects: Objects with shape uncertainty have often been represented as probabilistic occupancy maps in environmental mapping problems [15, 18]. The study in [7] develops a probabilistic camera model and utilizes a set of samples to represent objects with shape uncertainty. However, both grid maps and sample-based representations are memory-intensive, posing challenges in detailed 3D object representation. To overcome this, recent methods have employed probabilistic implicit functions, such as Gaussian Process Implicit Surfaces (GPIS) [12, 2, 13, 30] or Probabilistic Signed Distance Functions (pSDF) [11]. Studies like [26, 2] use GPIS with thin plate kernel functions to smoothly extrapolate the unobserved surface. Rather than directly representing objects with implicit functions, [42] employs Gaussian processes to filter out noisy points, modeling the object using only points with low uncertainty. More recently, various shape completion models have utilized deep neural networks to complete object shapes with greater detail [8, 20, 54, 6]. Given the potential ambiguity in object shapes from limited views, [54, 6] employ generative models to provide multiple plausible guesses based on point clouds from a single view. Similar to

[26], our method uses GPIS to model uncertain object shape from a partially observed point cloud.

3) Dexterous grasp planning under surface uncertainty: Using above mentioned object modelling, some works plan grasps based on shape completed objects [47, 29]. Other works [26, 30, 23] plan grasps with probabilistic object model and consider uncertainty explicitly. Such work typically necessitates an object model that can quantify uncertainty at various positions, such as GPIS [26]. Research by [30, 26] aims to minimize the impact of uncertainty in grasp planning. Both studies penalize making contact in regions of high variance. Additionally, [26] scales down the force applied at each fingertip in proportion to the uncertainty at the contact point. Under shape uncertainty, achieving precise contact at a planned location is challenging. Some studies refine the contact location and force after the fingertip makes contact with the object [26, 45]. Others iteratively explore the object's shape and re-plan the grasping pose using tactile probing [13, 2]. However, these approaches require additional information from tactile sensors, which are expensive and add fragility to dexterous multi-finger hands. Employing a pre-grasp hand pose before actual grasping can prevent accidental contact with the object and enhance the stability of the subsequent grasp [46, 21, 27]. [27] demonstrates that a carefully chosen pre-grasp can even facilitate grasping multiple loosely piled objects. While most studies rely on manually defined pre-grasp poses without specific planning [21, 10], research like [27, 46] plans pre-grasps using a force closure-based grasp planner with an inflated object model. In our work, we approximate object shape using GPIS based on point clouds from either one or multiple views. Different from previous works, we propose a new grasp metric SpringGrasp metric that can optimize pregrasp when considering uncertainty and adapt impedance gains according to object uncertainty and geometry.

III. Method

We introduce an optimization-based grasp planner, SpringGrasp planner,capable of grasping objects with shape uncertainty. Our planner is designed to directly interface with objects represented as point clouds, either sourced from depth cameras or generated by a shape completion network. We formulate an optimization that solves for a compliant grasp which involves a pregrasp pose and per-finger impedance controls that take into account the potential motion of the object during the process of grasping. Critical to the optimization, we introduce a novel analytical and differentiable metric, SpringGrasp metric,to evaluate whether the compliant grasp can reach a stable equilibrium. In addition, our optimization takes into account the uncertainty of the object surface when planning the compliant grasp. Fig. 2 illustrate the process of planning and executing a compliant pre-grasp using SpringGrasp planner.

A. Compliant grasp formulation

The compliant grasp is defined as $\mathcal{G} = \{\boldsymbol{p}_i(t_0), \boldsymbol{o}_i, k_i\}, i \in \{1, 2, ..., m\}, m \geq 3$, which include a set of m fingertip initial contact locations on object surface $\boldsymbol{p}_i(t_0) \in \mathcal{R}^3$ and target locations $\boldsymbol{o}_i \in \mathcal{R}^3$ both in the world frame and a set of controller gains $k_i \in \mathcal{R}$.

For each finger, starting from the initial state contact position $\boldsymbol{p}_i(t_0)$, we control the position of the fingertip using a Cartesian-space impedance controller moving toward $\boldsymbol{o}_i$ with a gain k_i and damping coefficient $2\sqrt{k_i}$. We do not require that the contacts of initial state at $\boldsymbol{p}_i(t_0)$ to form a force-closure grasp. Instead, we expect the pose of the object to change over time in a dynamic process until an equilibrium is reached at t_{eq}. We assume that fingertips will move with the object without slippage during this dynamic process from t_0 to t_{eq}. Fig. 3 shows a 2D example of our compliant grasping process of a triangle.

The effect of each fingertip on the object can be simulated as a virtual spring-damper system attached to the object between $\boldsymbol{p}_i(t)$ and $\boldsymbol{o}_i$. As such, we model the compliant grasping process as a simultaneous motion of the object and springs toward the equilibrium state through time. We assume that the contact points in the object frame $\boldsymbol{p}_i^o = \boldsymbol{R}_0^T(\boldsymbol{p}_i(t_0) - \boldsymbol{t}_0)$ are fixed over time (i.e. no sliding), where $\boldsymbol{t}_0, \boldsymbol{R}_0^T$ are the known translation and rotation from the world to the object frame at t_0.

Since we are interested to know whether a force-closure grasp can be formed at t_{eq}, we need to obtain the fingertip positions $\boldsymbol{p}_i(t_{eq})$. A naive way to obtain $\boldsymbol{p}_i(t_{eq})$ is to forward simulate the grasping process from t_0 to t_{eq}. However, introducing a numerical forward simulation process to the optimization adds significant computational cost and numerical instability due to gradient back-propagation[33, 52]. Our key insight is that we can express the fingertip positions at equilibrium by the transformation of the object $\boldsymbol{R}_{\text{eq}}, \boldsymbol{t}_{\text{eq}}$:

$$\boldsymbol{p}_i(t_{eq}) = \boldsymbol{R}_{eq}\boldsymbol{R}_0^T\boldsymbol{p}_i(t_0) + \boldsymbol{t}_{eq} - \boldsymbol{R}_{eq}\boldsymbol{R}_0^T\boldsymbol{t}_0, \tag{1}$$

and formulate a Wahba's problem[48] which solves for the optimal rigid transformation of the object to minimize total kinetic energy stored in each virtual spring with gain k_i:

$$\boldsymbol{t}_{\text{eq}}, \boldsymbol{R}_{\text{eq}} = \arg\min_{\boldsymbol{t},\boldsymbol{R}} \sum_i^m \frac{1}{2} k_i ||\boldsymbol{R}\boldsymbol{R}_0^T(\boldsymbol{p}_i(t_0) - \boldsymbol{t}_0) + \boldsymbol{t} - \boldsymbol{o}_i||_2^2. \tag{2}$$

Wahba's problem can be solved analytically with a modified Kabsch algorithm [19, 31]. Alg. 1 shows detailed implementation of the analytical solver. Combining Equation 1 and Equation 2, we can now express $\boldsymbol{p}_i(t_{eq})$ as a function of $\boldsymbol{p}_i(t_0)$, $\boldsymbol{o}_i$, and k_i.

B. SpringGrasp metric

With an analytical expression of $\boldsymbol{p}_i(t_{eq})$, we can now define a metric that determines whether the fingertips at equilibrium can form a force-closure grasp. For each fingertip, we define the contact margin $\epsilon_i(t)$ as the angular difference between the direction of force $\boldsymbol{f}_i(t)$ and the surface normal $\boldsymbol{n}_i(t)$ at the contact in the world frame:

$$\epsilon_i(t) = -\frac{\boldsymbol{f}_i(t)}{||\boldsymbol{f}_i(t)||_2^2} \cdot \boldsymbol{n}_i(t) - \frac{1}{\sqrt{1+\mu^2}}, i \in \{1, 2, ..., m\}, \tag{3}$$

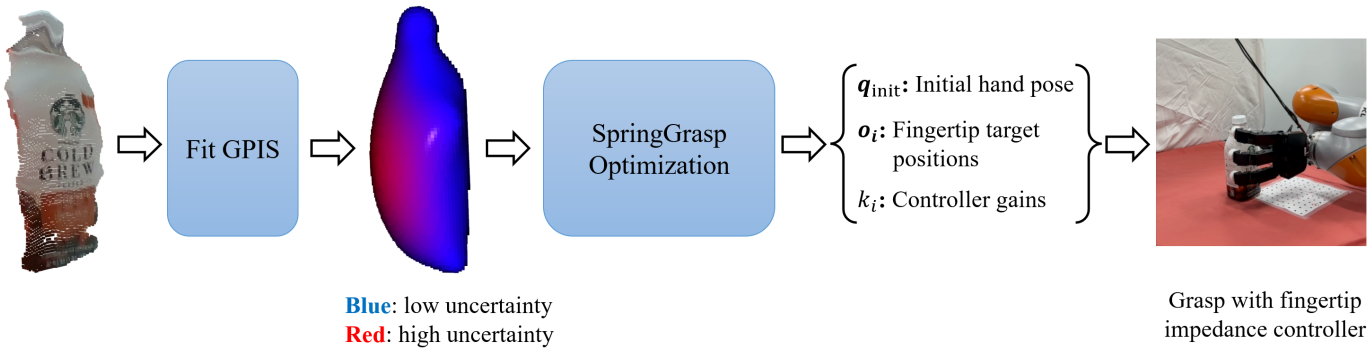


Fig. 2: Process for generating grasps from a partial point cloud.

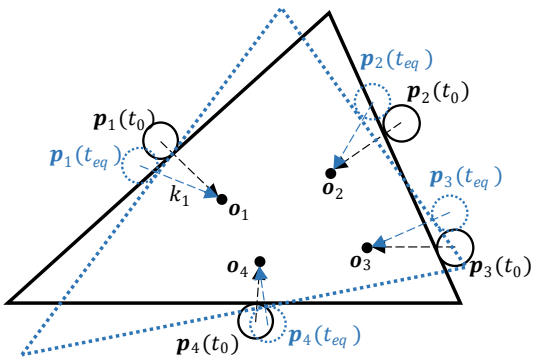


Fig. 3: A compliant grasp $\mathcal{G}$ on a 2D triangle at t_0 and at t_{eq}. The black solid line shows initial fingertip positions and the object pose, the blue dashed line shows fingertip positions and the object pose at equilibrium. $\boldsymbol{p}_i$ is the fingertip contact position and $\boldsymbol{o}_i$ is the target position.

Algorithm 1: Modified Kabsch algorithm

Input: $\boldsymbol{p}_i(t_0), \boldsymbol{o}_i, k_i, i \in \{1, 2, ..., m\}$
Output: $\boldsymbol{t}_{\text{eq}}, \boldsymbol{R}_{\text{eq}}$
$\boldsymbol{c}_p \leftarrow \frac{1}{m}\sum_i \boldsymbol{p}_i(t_0)$
$\boldsymbol{c}_o \leftarrow \frac{1}{m}\sum_i \boldsymbol{o}_i$
$\boldsymbol{P} \leftarrow \text{Stack}(\{\boldsymbol{p}_i(t_0) - \boldsymbol{c}_p\})$; /* mx3 matrix */
$\boldsymbol{O} \leftarrow \text{Stack}(\{\boldsymbol{o}_i - \boldsymbol{c}_o\})$; /* mx3 matrix */
$\boldsymbol{K} \leftarrow \text{Stack}(\{k_i\})$; /* 1xm matrix */
$\boldsymbol{H} \leftarrow (\boldsymbol{K}\boldsymbol{P})^T(\boldsymbol{K}\boldsymbol{O})$
$\boldsymbol{U}, \boldsymbol{V} \leftarrow SVD(\boldsymbol{H})$
$\boldsymbol{R} \leftarrow \boldsymbol{V}\boldsymbol{U}^T$
if $det(\boldsymbol{R}) < 0$ **then**
  $V[:, -1] \leftarrow -V[:, -1]$; /* prevent mirroring */
end
$\boldsymbol{R}_{\text{eq}} \leftarrow \boldsymbol{V}\boldsymbol{U}^T$
$\boldsymbol{t}_{\text{eq}} \leftarrow \frac{1}{\sum_i k_i}\sum_i k_i(\boldsymbol{o}_i - \boldsymbol{R}_{\text{eq}}\boldsymbol{p}_i(t_0))$

where the contact force, expressed in the world frame, is modelled as the virtual spring force:

$$\boldsymbol{f}_i(t) = k_i(\boldsymbol{o}_i - \boldsymbol{p}_i(t)) - 2\sqrt{k_i}\dot{\boldsymbol{p}}_i(t), i \in \{1, 2, ..., m\}. \quad (4)$$

If the contact margins on all fingertips are non-negative during the grasping process, that is, $\epsilon_i(t) \geq 0, i \in 1, 2, ..., m, t \in [t_0, t_{\text{eq}}]$, the object can be successfully grasped. However, when the difference between $\boldsymbol{R}_{\text{eq}}$ and $\boldsymbol{R}_0$ (i.e. the object rotation during the compliant grasping process) is small, enforcing the non-negative contact margins at the initial and equilibrium states, $\epsilon_i(t_0) \geq 0$ and $\epsilon_i(t_{eq}) \geq 0$, guarantees that all states in between also have non-negative contact margins (See Appendix A for detailed analysis).

C. Optimizing compliant grasp with known object surface

With the SpringGrasp metric, we can formulate an optimization to solve for a compliant grasp. We first describe the formulation for grasping an object with known surface in this subsection, and extend it to grasping an object with uncertain surface in the next subsection. We define our SpringGrasp metric energy as

$$E_{\text{sp}} = -\sum_i^m \log(\epsilon_i(t_0) + 1) - \sum_i^m \log(\epsilon_i(t_{\text{eq}}) + 1) \quad (5)$$

Let the surface of the object be a Signed Distance Function (SDF) $d(\boldsymbol{x})$, which maps a 3D point $\boldsymbol{x}$ to the signed distance between $\boldsymbol{x}$ and the object surface. We solve for a compliant grasp $\mathcal{G} = \{\boldsymbol{p}_i(t_0), \boldsymbol{o}_i, k_i\}, i \in \{1, 2, ..., m\}$ that minimizes the distance between fingertips to the object surface, $E_{\text{dist}} = |d(\boldsymbol{p}_i(t_0))|$, while optimizing the SpringGrasp metric energy:

$$\underset{\{\boldsymbol{p}(t_0)\},\{\boldsymbol{o}\},\{k\}}{\arg\min} \quad E_{\text{sp}} + E_{\text{dist}}, \quad (6)$$

where $\{x\}$ is a shorthand for the set $x_i, i \in \{1, 2, \cdots, m\}$.

The logarithm formulation in E_{sp} (Equation 5) encourages the optimizer to focus on fingertips with negative margins instead of improving fingertips that already have positive margins. In rare cases when margin is below -1, Appendix. E shows how we handle them with clipping and adding auxiliary energy terms.

D. Pregrasp with uncertain object surface

The optimization described in Equation 6 assumes that the object surface is precisely known, which is often not true in the real-world scenarios. To model objects with shape uncertainty, we use the widely adopted *Gaussian Process Implicit Surface* (GPIS) to represent the object. GPIS uses Gaussian process regression to approximate a closed SDF from observed data.

Given a query point $\boldsymbol{x}$ in world coordinates, we can compute the expected mean and variance $(d_\mu(\boldsymbol{x}), d_\sigma(\boldsymbol{x}))$ of the signed distance d between $\boldsymbol{x}$ and the object surface. The probability density of $\boldsymbol{x}$ to be on the object surface can be expressed as:

$$p(d(\boldsymbol{x}) = 0|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi d_\sigma(\boldsymbol{x})^2}} e^{-\frac{d_\mu(\boldsymbol{x})^2}{2d_\sigma(\boldsymbol{x})^2}} \tag{7}$$

When such surface uncertainty is present, it is hard to enforce initial contact to happen exactly at $\boldsymbol{p}_i(t_0)$ without making unintended contacts that cause the movement of the object. To mitigate this issue, our method extends the grasping process backward in time to model a pregrasp that places fingertips outside of the object surface before contact. The pregrasp can be represented as fingertip locations in the Cartesian space or a hand pose in the joint configuration space. We choose the latter because defining decision variables in the joint configuration space allows us to enforce hand kinematic constraints without the complexity of differentiating through an inverse kinematic process. Therefore, following definition in [21, 9], we define a pregrasp $\boldsymbol{q}_{\text{init}} \in \mathcal{R}^{22}$ as a wrist 6D pose and joint angles of all fingers such that fingers are not in contact with the object. We assume that a fingertip location $f_{\text{FK}}(\boldsymbol{q}_{\text{init}}, i)$ can be linearly extrapolated from the vector $\boldsymbol{o}_i - \boldsymbol{p}_i(t_0)$ with the extrapolation coefficient $c \in (0, 1]$:

$$\boldsymbol{p}_i(t_0) = c(f_{\text{FK}}(\boldsymbol{q}_{\text{init}}, i) - \boldsymbol{o}_i) + \boldsymbol{o}_i, \tag{8}$$

where the extrapolation coefficient c adjusts the relative distance between surface to the target $\boldsymbol{o}_i$ and the distance between the surface to the pregrasp fingertip. When c is larger, the pregrasp fingertips are closer to the surface. We determine c manually $c = 0.7$ for all our experiments.

A desired pregrasp should result in contact locations that have high probability to be the on the surface of the object. In addition, the contact location should have a prominent probability comparing to points in its neighborhood as we shown in Fig. 4. We define a line segment $\tau_i(\alpha), \alpha \in [0, 1]$ for each contact neighborhood, where $\tau_i(0) = f_{\text{FK}}(\boldsymbol{q}_{\text{init}}, i)$, $\tau_i(1) = \boldsymbol{o}_i$, and $\tau(1-c) = \boldsymbol{p}_i(t_0)$. We then define an objective term that encourages a desired pregrasp:

$$E_{\text{uncer}} = \sum_i^m \int_0^1 p(d(\boldsymbol{\tau}_i(\alpha)) = 0) - p(d(\boldsymbol{\tau}_i(1-c)) = 0) d\alpha \tag{9}$$

To avoid computing integration, we use K equally spaced sample points along the trajectory to approximate p.d.f and the energy term is computed as:

$$E_{\text{uncer}} = \sum_i^m \sum_k^K p(d(\boldsymbol{\tau}_i(\Delta\alpha k)) = 0) - p(d(\boldsymbol{\tau}_i(1-c)) = 0) \tag{10}$$

E. Optimizing compliant grasp with uncertain object surface

With the introduction of pregrasp, $\boldsymbol{q}_{\text{init}}$, we augment the decision variables of the optimization to be $\boldsymbol{q}_{\text{init}}$, $\boldsymbol{p}_i(t_0)$, $\boldsymbol{o}_i$, and k_i, where $i \in \{1, 2, ..., m\}$. In practice, we drop $\boldsymbol{p}_i(t_0)$ since it can be expressed as a function of $\boldsymbol{q}_{\text{init}}$ and $\boldsymbol{o}_i$ (Equation 8).

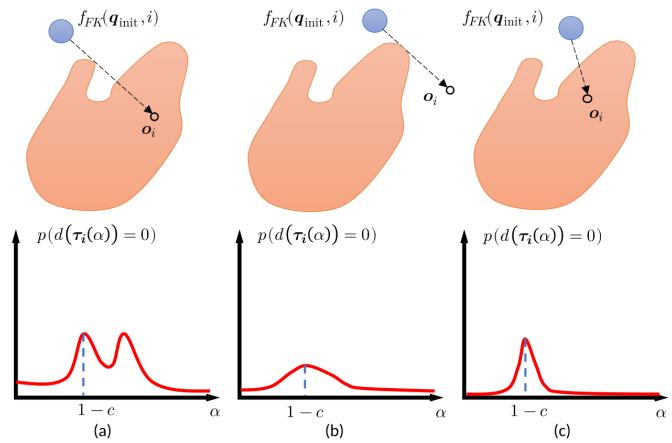


Fig. 4: Shape of p.d.f. for different line trajectory, we prefer (c) instead of (a) and (b) as contact is mostly likely to happen at the expected time and location.

In addition to E_{sp}, E_{dist}, and E_{uncer}, we introduce four more objective terms to improve the quality of the compliant grasp.

1) Low gains: To further increase the compliance of the grasp, we encourage the controller gains to be as low as possible:

$$E_{\text{gain}} = \sum_i^m k_i^2 \tag{11}$$

2) Target position: Ideally, the target location $\boldsymbol{o}_i$ should locate inside the object to encourage stability at equilibrium.

$$E_{\text{tar}} = \sum_i^m d(\boldsymbol{o}_i) \tag{12}$$

3) Collision avoidance: To ensure the grasp is collision free, we penalize collision between fingers, between the hand and the object, and between the hand and the table. We approximate collision geometry of the hand using 16 spheres with the centers defined as $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{16}\}$ in the world frame which can be evaluated by $f_{\text{FK}}(\boldsymbol{q}_{\text{init}})$ analytically. The radii of the spheres are defined manually as $\{r_1, \cdots, r_{16}\}$ (Details in Appendix B). For each self-collision pair (i, j) the collision penalty is defined as:

$$E_{\text{self}}^{\text{i,j}} = \begin{cases} \frac{1}{\text{dist}(\boldsymbol{s}_i, \boldsymbol{s}_j)} & \text{if dist}(\boldsymbol{s}_i, \boldsymbol{s}_j) \leq r_i + r_j \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The collision between the hand and the object is penalized by

$$E_{\text{ho}} = \sum_i \begin{cases} \frac{1}{d(\boldsymbol{s}_i)} & \text{if } d(\boldsymbol{s}_i) \leq r_i \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

and between the hand and the table by

$$E_{\text{ht}} = \sum_i \begin{cases} \frac{1}{\boldsymbol{s}_i^z} & \text{if } \boldsymbol{s}_i^z \leq r_i \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

w_{sp}	w_{dist}	w_{uncer}	w_{gain}	w_{tar}	w_{col}	w_{reg}	w_{force}
200	10000	20	0.5	1000	1	10	200

Table I: Weights of different energy terms in Eq 18

where $\boldsymbol{s}_i^z$ indicates the height of the sphere. Putting it together, the collision penalty term is defined as

$$E_{\text{col}} = \sum_{i,j}^{\text{collision pairs}} E_{\text{self}}^{\text{i,j}} + E_{\text{ho}} + E_{\text{ht}}, \tag{16}$$

where collision pairs are defined in Appendix B. As most collision spheres have the same radius, we don't normalize the energy for radius.

4) Regularization: We also add some regularization to shape the optimization. Less object movement during the compliant grasping process is encouraged by minimizing the difference between $\boldsymbol{p}_i(t_0)$ and $\boldsymbol{p}_i(t_{eq})$. We also regularize the joint angles of $\boldsymbol{q}_{\text{init}}$ to match a neutral reference pose $\boldsymbol{q}_{\text{ref}}$ designed manually. To prevent degenerate solution with zero force at each contact point at equilibrium, we also encourage a minimal force exerted by each fingertip:

$$\begin{aligned} E_{\text{reg}} &= ||\boldsymbol{q}_{\text{ref}} - \boldsymbol{q}_{\text{init}}||_2^2 + \sum_i^m ||\boldsymbol{p}_i(t_0) - \boldsymbol{p}_i(t_{\text{eq}})||_2^2 \\ E_{\text{force}} &= w_{\text{force}} \sum_i^m \min(f_{\text{min}}, ||\boldsymbol{f}_i||_2) \end{aligned} \tag{17}$$

where $f_{\text{min}} = 2$ and $w_{\text{force}} = 200$.

Putting it all together, we arrive at our final optimization of a compliant grasp that aims to grasp an object with uncertain shape:

$$\begin{aligned} \underset{\boldsymbol{q}_{\text{init}},\{\boldsymbol{o}\},\{k\}}{\arg\min} \quad & w_{\text{sp}}E_{\text{sp}} + w_{\text{dist}}E_{\text{dist}} + w_{\text{col}}E_{\text{col}} + w_{\text{uncer}}E_{\text{uncer}} \\ & + w_{\text{gain}}E_{\text{gain}} + w_{\text{tar}}E_{\text{tar}} + w_{\text{reg}}E_{\text{reg}} + w_{\text{force}}E_{\text{force}} \end{aligned} \tag{18}$$

The weight of each object term is shown in Table I. The final objective function is fully differentiable and is optimized with off-the-shelf gradient-based optimizers.

IV. Experiments

In our experiment, we will demonstrate that optimizing controller gains is important for grasping objects under shape uncertainty. To this end, we compare the proposed approach to a baseline that uses bilevel optimization with the force closure criterion (similar to [49, 28]). For this baseline, the controller gains have to be picked and it does not consider object shape uncertainty. We also systematically study the impact of different levels of shape uncertainty on the performance of our approach. Additionally, we perform an ablation of the objective function with and without considering object shape uncertainty. Our experiments demonstrate that our approach outperforms the baselines by 18-27% and that not taking uncertainty into account in the objective function leads to a significant performance drop (9%). Through an additional ablation, we will show that the impact of the pregrasp formulation on grasp performance is significant. Finally, we discuss the property of allowing object movement in the SpringGrasp metric and how it is beneficial for grasp planning. Fig. 8 shows an example grasp per object in our dataset.

Fig. 5: Real robot setup showing the Kuka iiwa arm equipped with a left Allegro hand and the configuration of the three RGB-D cameras.

A. Evaluation metric

Unless otherwise states, we use grasp success rate in real robot experiment as the evaluation metric for all our experiments. To evaluate whether a grasp is successful or not, the object is lifted 5cm after grasping. We use two criteria to assess the success of a grasp: 1) Whether the object is lifted. 2) Whether there is sliding between fingertips and the object. If a grasp satisfies both criteria, it is considered a successful grasp. If the object is lifted but slides during the process, the grasp is considered as partially successful. For each object, we attempt five grasps over different object poses. We then compute the grasp success rate as $\frac{\#\text{successful}+0.5\#\text{partial}}{5}$. We also report the average success rate over all objects.

B. Experiment setup

1) Hardware setup: We use three realsense RGB-D cameras located around the table (see Fig. 5). The robot arm is Kuka iiwa R820 and is equipped with a left Allegro hand. All the devices are connected to a PC with a i7-13700K CPU and RTX3090 GPU, which is also used for grasp optimization and motion planning for the arm.

2) Software setup: For robot control, our Allegro hand impedance controller is adapted from the AllegroHand KDL [17] and our Kuka Arm controller is adapted from iiwaPy [39]. Our SpringGrasp planner is written in Pytorch and we solve the optimization using RMSProp. After obtaining a grasp, we use curobo [44] to plan a collision free arm trajectory to the desired wrist pose.

C. Baseline

For our baseline, we use bilevel optimization with a differentiable force closure energy term similar to [49, 28]. We use this baseline to show the advantage of optimizing the controller

Fig. 6: All objects used for experiments

gains with the SpringGrasp metric compared to force closure bilevel optimization where the user has to manually tune the gains. The energy function of this baseline is adapted from our method with E_{sp} changed into the force closure metric energy E_{fc}. More details can be found in the Appendix. C. To directly compare the performance between using the SpringGrasp or the force closure metric, we compute the target location $\boldsymbol{o}_i$ with $i \in \{1, 2, ..., m\}$ where m is the number of fingers and pregrasp hand pose $\boldsymbol{q}_{\text{init}}$ based on the force $\boldsymbol{F}_i(t_0)$ that is the result of the inner loop of the bilevel optimization problem and pick a controller gain k_i:

$$\boldsymbol{o}_i = \boldsymbol{p}_i(t_0) + \frac{\boldsymbol{F}_i(t_0)}{k_i}, i \in \{1, 2, ..., m\}$$

$$\boldsymbol{q}_{\text{init}} = \text{IK}(\frac{1}{c}\{\boldsymbol{p}_i\} - \frac{1-c}{c}\{\boldsymbol{o}_i\}), c \in [0, 1]$$

c_i are the same clearance coefficients as used in our method.

D. Comparison with Baseline without Optimized Gains

To demonstrate the importance of optimizing the controller gains in robotic grasping, we conducted experiments using two sets of gains in the baseline. We refer to these sets as either Bilevel (high), Bilevel (low) or Bilevel(heuristic). For Bilevel (high), we set the gains to $k_1, k_2, k_3 = 160, k_4 = 320$. For Bilevel (low), we used $k_1, k_2, k_3 = 80, k_4 = 160$, which is close to the average value output by our approach. For Bilevel(heuristic), we set the gains inverse proportional to variance of expected contact location as $k_i = \frac{\alpha}{d_\sigma(\boldsymbol{p}_i)}$, which is similar to the setting in [26]. For this experiment, we estimate the object shape using point clouds collected from two camera views that are then input to GPIS.

The result of the comparison between our method and the baseline is shown in Table II. With input point clouds from two viewpoints, our method achieved an overall grasp success rate of 89%, which is 24% higher than that of Bilevel (low) and 27% higher than Bilevel (high). Compared to Bilevel (heuristic), our method also achieves an 18% higher success rate. The performance difference between our method and Bilevel (low) was more pronounced with long and thin objects, such as boxes and bananas. For instance, the success rate for grasping a banana dropped from 80% to 40%, and for a box from 100% to 60%. When compared with Bilevel (high), the difference was more significant with non-convex objects like bananas and cars. Although both types of failures were mitigated in Bilevel (heuristic), there is still a significant success rate difference when grasping thin and long objects as well as non-convex objects.

The results suggest that there is no universal set of gains that works for all scenarios. This is illustrated in Fig. 7. On the one hand, low gains can result in an unstable equilibrium where even small perturbations are sufficient to spin the object out of the grasp. High gains can position the target locations $\mathbf{o_i}$ close to the zero level set of the GPIS. This can result in the finger not making contact with the true object surface that may be behind the estimated surface. SpringGrasp allows to also optimize the controller gains and therefore to avoid unstable equilibria or missed contacts.

E. Robustness to Varying Levels of Shape Uncertainty

To study how robust our method is against varying levels of shape uncertainty, we evaluate grasp success rate of our method using point clouds recorded from either three, two or one viewpoint. For each setting, we pick the viewpoint that provides the most information about the object (Camera 1 and 2 for experiments with two viewpoints and Camera 3 for single viewpoint experiments). As shown in Tab. II, grasp success rate slightly drop from 91% to 89% when changing from three to two viewpoints. The success rate drops to 84% when only using point cloud from a single viewpoint. Contradict to our expectation, for larger object such as the Cheezit box, the grasp success rate does not drop significantly when changing from two to one viewpoint. As shown in Fig. 9, GPIS estimates the true shape of the object fairly well een from one viewpoint and preserves necessary geometry information for successful grasping. For smaller objects, our method is quite robust to a reduction of information.

F. Ablation study

In our method, we hypothesize that both considering uncertainty and using a pregrasp is essential to achieve a successful

	Ours (3 views)	Ours (2 views)	Ours (1 view)	w/o uncertainty (1 view)	w/o pregrasp (1 view)	Bilevel-high (2 views)	Bilevel-low (2 views)	Bilevel-heu (2 views)
Mustard bottle	90%	**100%**	80%	70%	90%	50%	60%	60%
Lego	90%	70%	80%	70%	60%	70%	50%	50%
Pyramid	**100%**	**100%**	**100%**	80%	70%	70%	70%	80%
Campell can	**100%**	**100%**	**100%**	90%	**100%**	90%	90%	80%
Cheezit box	80%	**90%**	80%	60%	**90%**	30%	40%	80%
Mug	**90%**	80%	80%	70%	80%	70%	60%	80%
Orange	**100%**	**100%**	90%	**100%**	**100%**	90%	80%	**100%**
Coffee bottle	80%	**90%**	**90%**	70%	**90%**	50%	60%	60%
Spam	**80%**	60%	60%	60%	60%	40%	60%	50%
Plane	**100%**	70%	70%	80%	80%	60%	70%	70%
Car	90%	**100%**	70%	70%	70%	60%	80%	60%
Banana	**100%**	80%	90%	60%	70%	30%	40%	60%
Pear	**100%**	**100%**	90%	**100%**	80%	80%	80%	80%
Small box	**100%**	**100%**	**100%**	70%	70%	80%	60%	90%
Total	**93%**	89%	84%	75%	79%	62%	65%	71%

Table II: Grasp success rate over 5 trials per object. Note that we can have successful and partially successful grasps. *Ours* refers to SpringGrasp that takes as input point clouds from either 3, 2 or 1 viewpoint. *w/o uncertainty* and *w/o pregrasp* are ablations of our method when either removing E_{uncer} or setting $c = 1$ respectively in Eq. 8. Both use a single view point cloud as input. *Bilevel-high*, *Bilevel-low* and *Bilevel-heu* refer to the bilevel optimization baseline with high gains, low gains, or heuristically selected gains. They all use point clouds from two views as input. Our method significantly outperforms the baseline as well as the ablations even under maximum uncertainty in object shape.

grasp under object shape uncertainty. We justify our design with an ablation study on these two components. To understand the importance of the term E_{uncer} in Eq. 10, we evaluate the grasp success rate on all objects with or without this term in the objective function. Note that for this experiment, we assume as input an object point cloud from a single view which is the most common scenario for robotic grasping. As shown in Tab. II, the grasp success rate drops from 84% to 75% when removing E_{uncer} from Eq. 10 demonstrating the effectiveness of considering uncertainty in our framework. We also observed a typical failure mode when not considering uncertainty visualized in Fig. 10. The optimized grasp may make contact with regions on the GPIS surface that have high uncertainty and therefore tend to be far off from the true object surface. This scenario results in missing or unexpected contacts which adversely affects grasp success rate.

To analyse the benefit of optimizing a pregrasp, we compare the grasp success rate of our method with different pregrasp coefficients $c = 1.0$ and $c = 0.7$. A pregrasp becomes a grasp (i.e. the fingertips are in contact with the estimated object surface) if the pregrasp coefficient $c = 1.0$. From Tab. II, we found that the grasp success rate decreases from 84% to 78% if we don't optimize for a pregrasp in which the fingertips are slightly offset from the estimated object surface. Among all objects and poses, the grasp success rate change most significantly if the object is not at a stable pose before grasping. In this case, it is most vulnerable to being perturbed by unexpected contact when the hand is reaching for the grasp pose. Fig. 11 shows an example where the box is standing on its side and is perturbed by premature contact resulting in it tipping over before the optimized grasp can be acquired.

multiplier	w_{sp}	w_{dist}	w_{uncer}	w_{gain}	w_{tar}	w_{col}	w_{reg}	w_{force}
0.5	0.80	**0.83**	0.77	0.77	0.73	**0.93**	0.77	**0.90**
1	0.83	**0.83**	**0.83**	**0.83**	**0.83**	0.83	0.83	0.83
2	**0.90**	0.77	**0.83**	**0.83**	0.77	0.83	**0.90**	0.83

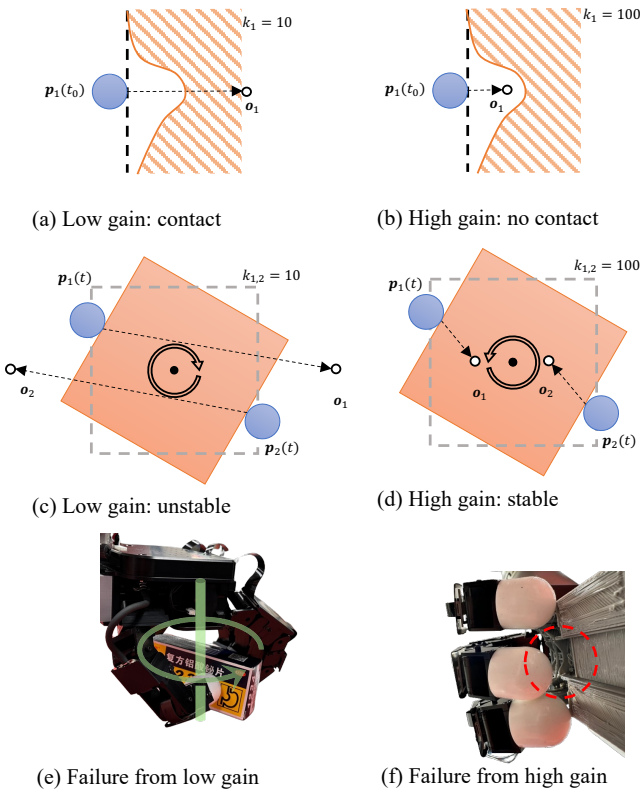
Table III: Grasp success rate with different scaling factors on different weights

G. Hyperparameter analysis

Tab. I lists the weights we used for the energy terms in Equation 18. Here we study how the selection of different weights affects the performance of our grasp planner. For each weight, we scale it by two multipliers, 0.5 and 2, and measure the grasp success rate respectively. We keep all other parameters unchanged when scaling the parameter of interest. We experiment with three distinctive objects, a pear, mustard bottle, and a Cheez-it box. The results shown in Tab. III show that in general changing weights of different energy terms moderately will not cause significant changes in the grasp success rate, though some parameters are more sensitive than others such as w_{tar} and w_{reg}.

H. Computation time

We report the time taken to optimize grasps from 7 initial guesses and single initial guesses for both our method and baseline on both CPU and GPU in Tab. IV. Each time is measured as an average of 5 experiments. In general, our method optimizes faster than the bilevel baseline. When scaling up the number of seeds, the main computation bottleneck is forward kinematics computation provided by [32]. Notice that due to implementation limitations in differentiable robot model[32], grasp planning runs faster on CPU than GPU.



(a) Low gain: contact

(b) High gain: no contact

(c) Low gain: unstable

(d) High gain: stable

(e) Failure from low gain

(f) Failure from high gain

Fig. 7: Illustration of baseline failure modes. The fingertip needs to apply the same force despite different controller gains (either $k_i = 10$ for (a,c) or $k_i = 100$ for (b,d)). For shaping uncertain object (top row - the dashed line indicates the expected object surface; the solid curve is the actual surface), using low gains k_i can ensure making contact with the object as shown in (a), instead of not making contact as shown in (b). For smaller objects (middle row), when the object is perturbed (represented by the solid box) from its equilibrium (represented by the dashed box) at time t, using higher gains can ensure that the fingertip can always generate an object torque to compensate a perturbation as shown in (c) instead of continuing to twist the object in the direction of the perturbation as shown in (d). In real robot experiments (bottom row), the box was twisted in (e) due to an unstable equilibrium while in (f) the middle finger fails to attach to the object because the target location is outside the object.

time(s)	Ours (7)	Ours (1)	Bilevel (7)	Bilevel (1)
CPU	15.58	7.02	34.97	9.17
GPU	23.44	14.22	41.47	15.28

Table IV: Optimization time in second. Results of our methods and bilevel baseline using 1 and 7 seeds are represented as Ours (1), Ours (7), Bilevel (1), and Bilevel (7) respectively.

I. The Benefit of Modeling Object Movement during Grasp Acquisition

An additional property of the proposed SpringGrasp metric is that we model grasping as a dynamic process and allow object movement during grasping as we shown in Sec.III-A. In this section, we will present evidence that allowing object movement can help to get around kinematic and joint torque limits when directly trying to establish a force closure grasp. For example, if a finger cannot apply the force required by the optimizer due to the current joint configuration and joint torque limits, the robot could first move the object into a new pose that leads to a joint configuration which allows a stable grasp. To test this hypothesis, we setup a 2D simulated experiment with two 2D triangles that are grasped by a three-fingered hand. For each triangle, we generate three different contact configurations on the triangle surface and sample 100 triangle poses per triangle and contact configuration. In each experiment, we optimize grasps with our metric as follows: as in this experiment, we only care about kinematic constraints and joint torque limits, we optimize target locations $\boldsymbol{o}_i$ and gains k_i while contact locations $\boldsymbol{p}_i$ remain the same according to the three contact configurations we picked. Note that the wrist of the hand remains at a fixed location. Of the generated grasps, we only keep those that are feasible and can apply the desired force (with or without triangle movement) as verified in simulation. Of those feasible grasps, we evaluate how many of them are in force closure and in which each finger can apply enough force under joint torque constraints without moving the triangle (more details in Appendix. D). Grasps that allow the fingers to directly achieve force closure without triangle movement are only a subset of all feasible grasps. This means that there is no feasible grasp that could form force closure without triangle movement but fails to obtain a feasible compliant grasp. From Fig. 12, only around 85% of feasible grasps can directly achieve force closure without triangle movement. This demonstrates that there is a significant portion (15%) of object poses that can benefit from object movement to avoid such kinematic and joint torque limits. Using the SpringGrasp metric, we can also deliberately control the object movement during optimization. Using an extra energy term $E_{\text{pose}} = \sum_i ||\boldsymbol{p}_i^{\text{des}}(t_{\text{eq}}) - \boldsymbol{p}_i(t_{\text{eq}})||_2$ to replace $\sum_i ||\boldsymbol{p}_i(t_0) - \boldsymbol{p}_i(t_{\text{eq}})||_2$ in Eq. 17, we can encourage object and fingertips to move in some desired direction, such as up by 1cm. For example, in Fig. 13 we set $\boldsymbol{p}_i^{\text{des}}(t_{\text{eq}}) = \boldsymbol{p}_i(t_0) + [0, 0, 0.01]$ which could be used to minimize undesired contact with the table during grasping.

V. Conclusion & Future Work

We present an optimization-based system that can generate pregrasp grasps from noisy and partial point cloud observation. Benefiting from the SpringGrasp metric, pregrasp formulation, and consideration of uncertainty, our method could grasp a diverse set of objects with an 89% success rate from 2 views and 84% grasp success rate from single view in a real robot experiment. When grasping objects with our method, the most common failure happens when the object being grasped is heavier than 0.3kg. This is shown in the experiment with the fully loaded spam can (0.4kg). As our method doesn't model gravity explicitly, its ability to grasp heavy objects is limited. A natural next step would be to integrate gravity and uncertainty into object mass into our grasp planner. Moreover, the GPIS model suffers from modeling objects with thin walls such as knives or baskets due to the challenge of distinguishing points from the inner surface and outer surface and effectively sampling the interior point of the object. Future work could focus on decomposing the object geometry and extracting part of the object that is easy to grasp from the point cloud such

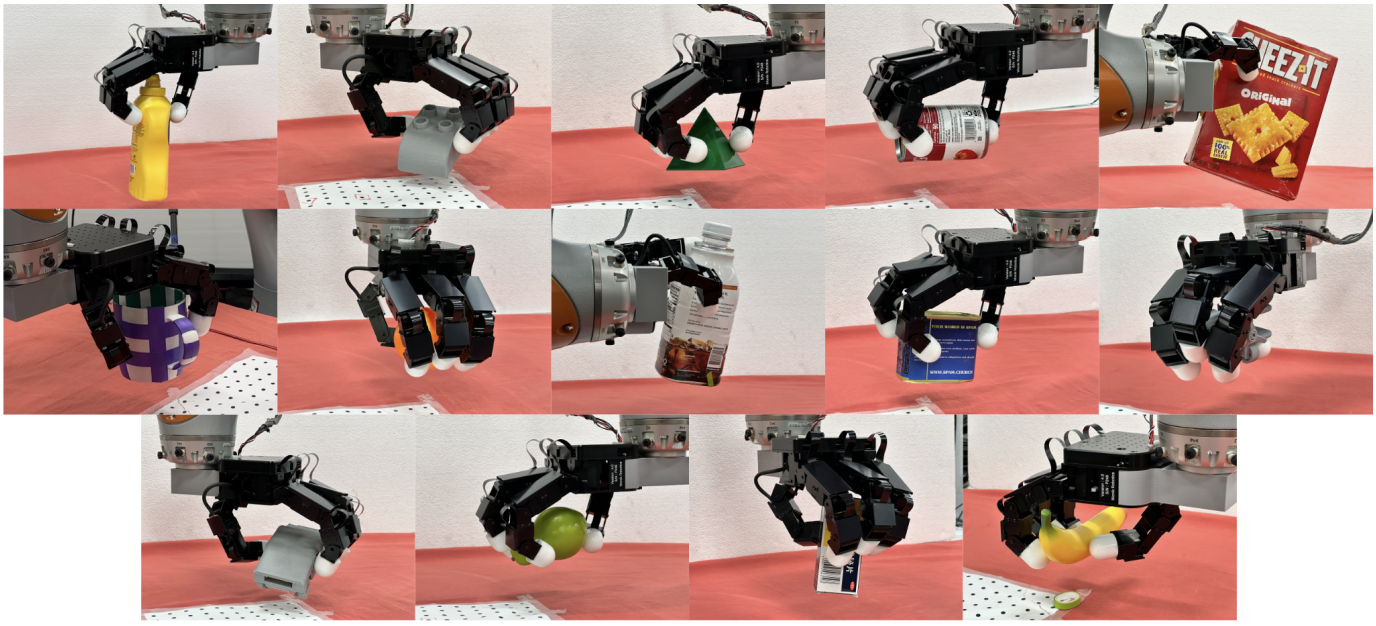
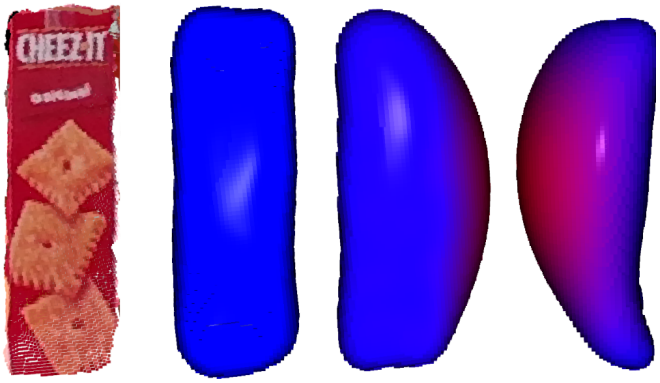


Fig. 8: Grasping different object with single depth image input



(a) cheez-it box (b) three views (c) two views (d) single view

Fig. 9: GPIS reconstructed surfaces from different number of viewpoints. The color on the surface indicates variance of a surface point where hotter colors correspond to a larger variance.

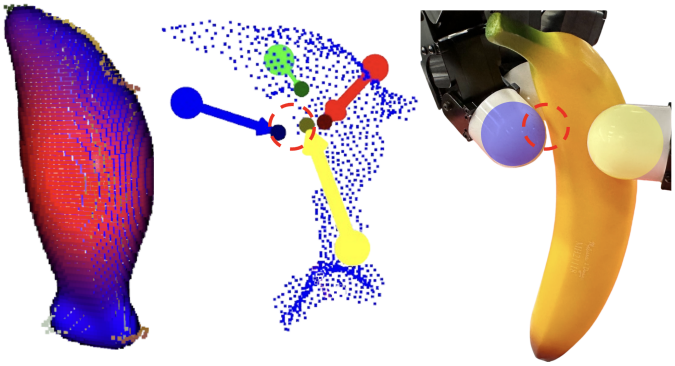


Fig. 10: The left figure illustrates the uncertain surface modeled by GPIS. The central figure depicts the anticipated trajectory of the fingertips in the predicted pregrasp. The right figure shows that the tip of the ring finger (blue) does not make contact with the actual object surface, resulting in a partial grasp failure. For visualization, we color visible thumb and ring finger in the right figure to match the colors in the middle figure.

as the handle of the knife and basket. Another interesting direction could be using our grasp planner to generate a dataset and train a deep neural network to directly predict compliant grasp, given a partial observation of the object.

VI. Acknowledgement

This project is supported by NSF:FRR:2153854 and NSF:FRR:2342246. We also thank Albert Wu for the insightful discussion regarding math notation and Purvi Goel for recording a voiceover for our submission video.

References

[1] Aditya Bhatt, Adrian Sieler, Steffen Puhlmann, and Oliver Brock. Surprisingly robust in-hand manipulation: An empirical study. In *Robotics: Science and Systems*, 2021.

[2] Marten Björkman, Yasemin Bekiroglu, Virgile Högman, and Danica Kragic. Enhancing visual perception of shape through tactile glances. In *IROS*, 2013.

[3] Amy A Blank, Allison M Okamura, and Louis L Whitcomb. Task-dependent impedance and implications for upper-limb prosthesis control. *The International Journal of Robotics Research*, 33(6):827–846, 2014.

[4] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014. doi: 10.1109/TRO.2013.2289018.

[5] Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833, 2011.

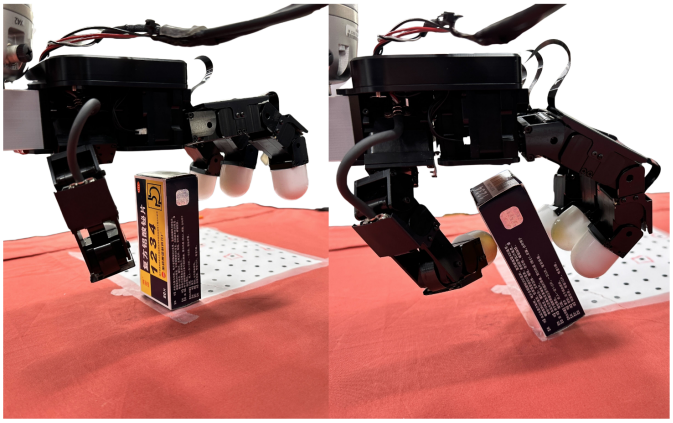


Fig. 11: Grasp failure when not using pregrasp optimization. The object tips before the optimized grasp can be acquired.

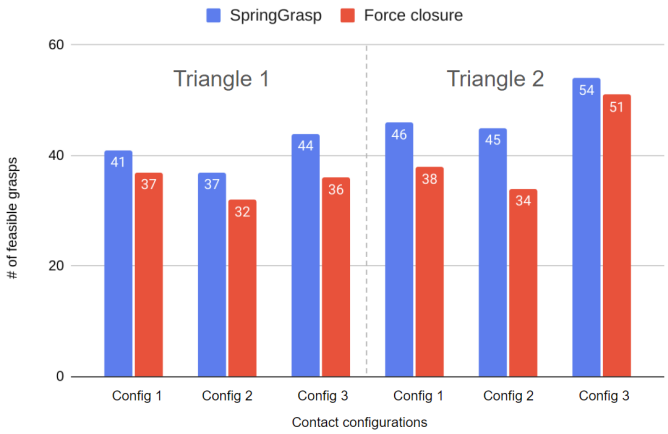


Fig. 12: (Blue) Number of feasible grasps generated by our method for 100 object poses per contact configuration. (Red) Number of feasible force closure grasps that do not require object movement for 100 object poses per contact configuration.

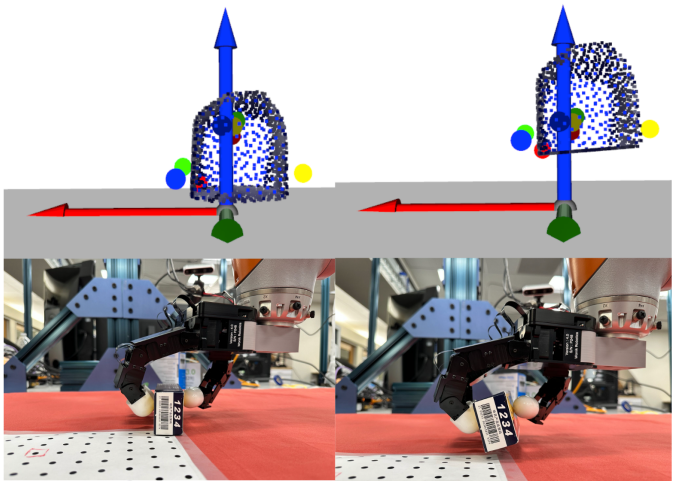


Fig. 13: Object move up by 1cm as anticipated during grasp planning.

[6] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023.

[7] Vassilios N. Christopoulos and Paul Schrater. Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[8] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017.

[9] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896. IEEE, 2023.

[10] Zhen Deng, Bin Fang, Bingwei He, and Jianwei Zhang. An adaptive planning framework for dexterous robotic grasping with grasp type detection. *Robotics and Autonomous Systems*, 140:103727, 2021.

[11] Wei Dong, Qiuyuan Wang, Xin Wang, and Hongbin Zha. PSDF fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction. In *ECCV*, 2018.

[12] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *ICRA*, 2011.

[13] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. Uncertainty aware grasping and tactile exploration. In *ICRA*, 2013.

[14] Clemens Eppner, Raphael Deimel, José Álvarez Ruiz, Marianne Maertens, and Oliver Brock. Exploitation of environmental constraints in human and robotic grasping. *The International Journal of Robotics Research*, 34(7): 1021–1038, 2015. doi: 10.1177/0278364914559753. URL https://doi.org/10.1177/0278364914559753.

[15] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[16] Kshitij Goel, Nathan Michael, and Wennie Tabib. Probabilistic point cloud modeling via self-organizing gaussian mixture models. *RA-L*, 2023.

[17] Solak Gokhan. Allegro hand kdl. https://github.com/ARQ-CRISP/allegro_hand_kdl, 2020.

[18] Bing-Jui Ho, Paloma Sodhi, Pedro Teixeira, Ming Hsiao, Tushar Kusnur, and Michael Kaess. Virtual occupancy grid map for submap-based pose graph slam and planning in 3d environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2175–2182. IEEE, 2018.

[19] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4):629–642, 1987.

[20] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi

Le. Pf-net: Point fractal network for 3d point cloud completion. In *CVPR*, pages 7662–7670, 2020.

[21] Daniel Kappler, Lillian Chang, Markus Przybylski, Nancy Pollard, Tamim Asfour, and Rüdiger Dillmann. Representation of pre-grasp strategies for object manipulation. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 617–624, 2010. doi: 10.1109/ICHR.2010.5686321.

[22] Moslem Kazemi, Jean-Sebastien Valois, J. Andrew Bagnell, and Nancy S. Pollard. Robust object grasping using force compliant motion primitives. In Nicholas Roy, Paul Newman, and Siddhartha S. Srinivasa, editors, *Robotics: Science and Systems VIII, University of Sydney, Sydney, NSW, Australia, July 9-13, 2012*, 2012. doi: 10.15607/RSS.2012.VIII.023. URL http://www.roboticsproceedings.org/rss08/p23.html.

[23] Albert H. Li, Preston Culbertson, and Aaron D. Ames. PONG: probabilistic object normals for grasping via analytic bounds on force closure probability. *CoRR*, 2023.

[24] Albert H. Li, Preston Culbertson, Joel W. Burdick, and Aaron D. Ames. Frogger: Fast robust grasp generation via the min-weight metric. *IROS*, 2023.

[25] Haoran Li, Christopher J. Ford, Matteo Bianchi, Manuel G. Catalano, Efi Psomopoulou, and Nathan F. Lepora. Brl/pisa/iit softhand: A low-cost, 3d-printed, underactuated, tendon-driven hand with soft and adaptive synergies. *RA-L*, 2022.

[26] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics Auton. Syst.*, 2016.

[27] Yuyang Li, Bo Liu, Yiran Geng, Puhao Li, Yaodong Yang, Yixin Zhu, Tengyu Liu, and Siyuan Huang. Grasp multiple objects with one hand. *arXiv preprint arXiv:2310.15599*, 2023.

[28] Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *RA-L*, 2022.

[29] Jens Lundell, Francesco Verdoja, and Ville Kyrki. Robust grasp planning over uncertain shape completions. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pages 1526–1532. IEEE, 2019. doi: 10.1109/IROS40897.2019.8967816. URL https://doi.org/10.1109/IROS40897.2019.8967816.

[30] Jeffrey Mahler, Sachin Patil, Ben Kehoe, Jur van den Berg, Matei T. Ciocarlie, Pieter Abbeel, and Ken Goldberg. GP-GPIS-OPT: grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In *ICRA*, 2015.

[31] F Landis Markley and Daniele Mortari. How to estimate attitude from vector observations. In *Astrodynamics Specialist*, 1999.

[32] Franziska Meier, Austin Wang, Giovanni Sutanto, Yixin Lin, and Paarth Shah. Differentiable and learnable robot models. *arXiv preprint arXiv:2202.11217*, 2022.

[33] Miguel Angel Zamora Mora, Momchil Peychev, Sehoon Ha, Martin Vechev, and Stelian Coros. Pods: Policy optimization via differentiable simulation. In *International Conference on Machine Learning*, pages 7805–7817. PMLR, 2021.

[34] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, Dieter Fox, and Akansel Cosgun. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 2023.

[35] Van-Duc Nguyen. Constructing force-closure grasps. *The International Journal of Robotics Research*, 1988.

[36] Martin Pfanne, Maxime Chalon, Freek Stulp, Helge J. Ritter, and Alin Albu-Schäffer. Object-level impedance control for dexterous in-hand manipulation. *RA-L*, 2020.

[37] Steffen Puhlmann, Jason Harris, and Oliver Brock. Rbo hand 3: A platform for soft dexterous manipulation. *IEEE Transactions on Robotics*, 38(6):3434–3449, 2022.

[38] Máximo A. Roa and Raúl Suárez. Grasp quality measures: review and performance. *Auton. Robots*, 2015.

[39] M. Safeea and P. Neto. Model-based hardware in the loop control of collaborative robots: Simulink and python based interfaces. *International Journal of Computer Integrated Manufacturing*, 2023.

[40] Cosimo Della Santina, Cristina Piazza, Giorgio Grioli, Manuel G. Catalano, and Antonio Bicchi. Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. *IEEE Transactions on Robotics*, 34(5): 1141–1156, 2018. doi: 10.1109/TRO.2018.2830407.

[41] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

[42] Nicolas Sommer, Miao Li, and Aude Billard. Bimanual compliant tactile exploration for grasping unknown objects. In *ICRA*, 2014.

[43] Hannah Stuart, Shiquan Wang, Oussama Khatib, and Mark R Cutkosky. The ocean one hands: An adaptive design for robust marine manipulation. *The International Journal of Robotics Research*, 36(2):150–166, 2017.

[44] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. Curobo: Parallelized collision-free robot motion generation. In *ICRA*, 2023.

[45] Taro Takahashi, Toshimitsu Tsuboi, Takeo Kishida, Yasunori Kawanami, Satoru Shimizu, Masatsugu Iribe, Tetsuharu Fukushima, and Masahiro Fujita. Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control. In *2008 IEEE International Conference on Robotics and Automation*, pages 264–271. IEEE, 2008.

[46] Dongying Tian, Xiangbo Lin, and Yi Sun. Adaptive motion planning for multi-fingered functional grasp via

force feedback. *arXiv preprint arXiv:2401.11977*, 2024.

[47] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017.

[48] Grace Wahba. A least squares estimate of satellite attitude. *SIAM review*, 1965.

[49] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023.

[50] Thomas Wimbö"ck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23, 2012. doi: 10.1177/0278364911416526. URL https://doi.org/10.1177/0278364911416526.

[51] Albert Wu, Michelle Guo, and Karen Liu. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. In *CoRL*, 2022.

[52] Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *ICLR*, 2021.

[53] Zhuozhi Zhang, Jie Zhang, Qi Luo, Chih-Hong Chou, Anran Xie, Chuanxin M Niu, Manzhao Hao, and Ning Lan. A biorealistic computational model unfolds human-like compliant properties for control of hand prosthesis. *IEEE Open Journal of Engineering in Medicine and Biology*, 2022.

[54] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021.

APPENDIX

A. Analysis of contact margins

This section intends to justify why the feasibility of the initial and target state according to the proposed SpringGrasp metric is a good heuristic to indicate feasibility of the entire dynamic process. For simplicity of the analysis, we consider a problem with three fingertips that make contact with a 2D object as shown in Figure 14. In this system, the pose of the object can be described by $\boldsymbol{s}(t) = \{x(t), y(t), z(t), \theta(t)\}$. If the damping coefficients of each virtual spring are sufficiently large, no oscillation will happen during the dynamic process. This means that for any time, we can express it as convex combination of initial state and target state with a blending coefficient $\beta(t), t \in [t_0, t_{\text{eq}}]$ where $\beta(t_0) = 0$ and $\beta(t_{\text{eq}}) = 1$:

$$\boldsymbol{s}(t) = (1 - \beta(t))\boldsymbol{s}(t_0) + \beta(t)\boldsymbol{s}(t_{\text{eq}})$$

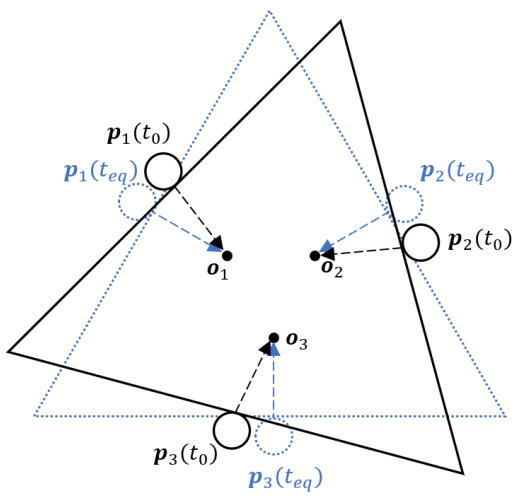


Fig. 14: Three fingertips making contact with a 2D triangle. Fingertips apply force toward $\boldsymbol{o}_i$ and move with the object from $\boldsymbol{p}_i(t_0)$ to $\boldsymbol{p}_i(t_{\text{eq}})$

.

1) Translation only: We define the angle between the force vector $\boldsymbol{f}_i(t)$ and surface normal vector $\boldsymbol{n}_t(t)$ as $\alpha_i(t)$. To examine the temporal evolution of $\alpha_i(t)$, we initially consider a scenario involving only translation. Assuming the object undergoes a rigid translation $\boldsymbol{t}$ towards the equilibrium point, the initial force vector is given by:

$$\boldsymbol{f}_i(t_0) = k_i(\boldsymbol{o}_i - \boldsymbol{p}_i(t_0))$$

where k_i is controller gain of fingertip. At any subsequent time, the force vector will be:

$$\boldsymbol{f}_i(t) = k_i(\boldsymbol{o}_i - \boldsymbol{p}_i(t_0) - \beta(t)\boldsymbol{t}) = \boldsymbol{f}_i(t_0) - k_i\beta(t)\boldsymbol{t}$$

indicating that, during translation, the force vector changes from $\boldsymbol{f}_i(t_0)$ to $\boldsymbol{f}_i(t_0) - k_i\boldsymbol{t}$. The normal vector remains constant as translation does not affect the direction of the normal vector. Consequently, $\alpha_i(t)$ stays within the bounds defined by $\alpha_i(t_0)$ and $\alpha_i(t_{\text{eq}})$. Provided that the force directions at the initial and equilibrium states fall within the friction cone, the force direction at any transient state will also lie within this cone.

2) Including rotation: When the dynamic grasping process involves rotation, the feasibility of the initial and equilibrium state does not guarantee the feasibility of the transient states. Consider a triangle rotating around its center $\boldsymbol{c}$ (see Figure 15). We choose a target location $\boldsymbol{o}_i$, where the distance between $\boldsymbol{c}$ and $\boldsymbol{o}_i$ is $\frac{\sqrt{2}}{2}r$, with r denoting the shortest distance from $\boldsymbol{c}$ to any edge of the triangle. Let the triangle rotate around its center until the force aligns with the surface normal vector. Then, $\alpha_i(t)$ initially increases to a maximum of $\alpha_{\text{max}} = \frac{\pi}{4}$ when $\boldsymbol{f}_i(t)$ is perpendicular to $\boldsymbol{o}_i - \boldsymbol{c}$, and then decreases to 0 at t_{eq}. If initially, angle $\angle \boldsymbol{p}_i(t_0)\boldsymbol{c}\boldsymbol{o}_i$ is less than $\frac{\pi}{4}$,$\alpha_i(t)$ will monotonically decrease and the force direction will always lies between the initial and equilibrium force directions. Bringing $\boldsymbol{o}_i$ closer to $\boldsymbol{c}$ reduces the change in $\alpha_i(t)$ for the same rotation angle on the object orientation $\theta(t)$, necessitating a greater rotation on $\theta(t)$ for $\alpha_i(t)$ to reach its maximum. Thus, adding E_{tar} to the energy function Eq.18 encourages the target location to stay deep inside the object, which allows more rotation on the object. Adding E_{reg} to Eq. 18 regulates movement during the dynamic process and reduces the rotation during the dynamic process. In practice, the feasibility of transient states can usually be inferred from the feasibility of initial and equilibrium states.

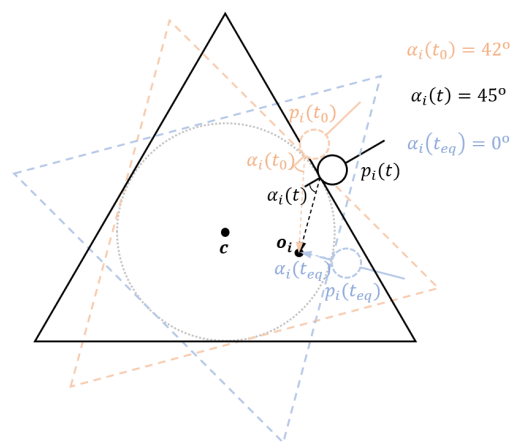


Fig. 15: Triangle rotate around $\boldsymbol{c}$, at time t, the force vector is perpendicular to vector $\boldsymbol{c} - \boldsymbol{o}_i$ and $\alpha_i(t) = \frac{\pi}{4} = 45^o$. In initial state and equilibrium state: $\alpha_i(t_0) = 42^o, \alpha_i(t_{\text{eq}}) = 0$

B. Collision spheres

Here we provide details on how we compute the energy terms $E_{\text{self,ho,ht}}$ in Eq 16 that are related to collisions with the hand itself, the object and table. Fig. 16 shows the placement of spheres that we use to approximate the geometry of the hand. $s_1 \sim s_{11}$ have a radius of 1cm and $s_{13} \sim s_{15}$ have radius of 2cm. For computing self-collision energy E_{self}, we find it is sufficient to check the following collision pairs: $(s_1, s_4), (s_1, s_7), (s_1, s_{10}), (s_4, s_7), (s_4, s_{10}), (s_7, s_{10}), (s_2, s_5), (s_5, s_8)$. When computing the hand object collision energy

Fig. 16: Placement of collision spheres for computing collision loss, each sphere has radius of 0.015

E_{ho} and hand table collision energy E_{ht}, we use all collision spheres.

C. Energy function of baseline method

For our baseline approach, we define the following energy function:

$$E = w_{\text{fc}} E_{\text{fc}} + w_{\text{dist}} E_{\text{dist}} + w_{\text{col}} E_{\text{col}} + w_{\text{reg}} E_{\text{reg}} + w_{\text{uncer}} E_{\text{uncer}}$$

Most terms in this energy function are directly migrated from our method Eq. 18. We remove E_{gain} as gains are choosen independently hence no other variables depends on the gains in the optimization problem. Therefore, the optimizer would drive the controller gains directly to the lowest possible value. We also remove E_{tar} and E_{force} as force and target locations are the result of the inner force closure solver of the baseline and therefore cannot be controlled by the outer loop. We set $w_{\text{fc}} = 200$ and weights of other energy term is the same as Eq. 18. In [5, 4], E_{fc} is approximated by assuming that each fingertip can only apply a force along the contact normal with a fixed magnitude, which accelerates computation. We replace it with a more accurate force closure metric as defined in [6] and compute it using the differentiable convex optimization solver cvxpylayers[1] as:

$$E_{\text{fc}} = \min\left(||\sum_i^n \boldsymbol{f}_i||_2 + ||\sum_i^n \boldsymbol{p}_i \times \boldsymbol{f}_i||_2\right)$$

$$s.t.: \boldsymbol{f}_i \cdot \boldsymbol{n_i} \leq -\frac{1}{\sqrt{1+\mu^2}}||\boldsymbol{f}_i||, \boldsymbol{f}_i \cdot \boldsymbol{n_i} \leq -F_{\text{min}}$$

As the grasps optimized by the baseline method do not involve a dynamic process, we use $\boldsymbol{f}_i$, $\boldsymbol{p}_i$ without the time index. Because the force direction is determined by the force closure solver which cannot be controlled to reduce pregrasp uncertainty, we instead focus on reducing uncertainty at the fingertip contact location $\boldsymbol{p}_i$ and set the uncertainty energy term E_{uncer} as follows:

$$E_{\text{uncer}} = \sum_i d_\sigma(\boldsymbol{p}_i)$$

Where d_σ is variance value function of GPIS as shown in Sec. III-D.

	Triangle 1	Triangle 2
config 1	(0.5,0.0), (0.75,0.5), (0.25,0.5)	(0.5,0.0), (1.0,0.5), (0.5,0.5)
config 2	(0.4,0.0), (0.8,0.4), (0.2,0.4)	(0.4,0.0), (1.0,0.4), (0.4,0.4)
config 3	(0.6,0.0), (0.7,0.6), (0.3,0.6)	(0.6,0.0), (1.0,0.6), (0.6,0.6)

Table V: Summary of fingertip contact configurations

D. Grasp coverage experiment setup

Fig. 17 shows dimension of two triangles and positions of different contact configurations we used in experiment IV-I. To ensure the best coverage of our method during optimization, we randomly initialize 2000 target positions and controller gains when optimizing grasp for each object pose.

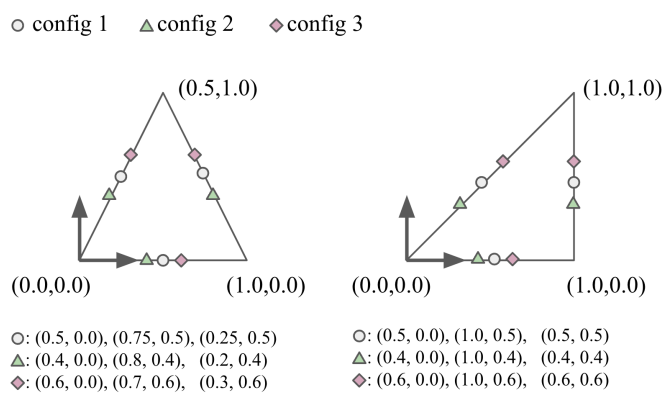


Fig. 17: Specification of two triangles. Vertices are expressed in the local object frame. Contact points in each contact configurations are expressed as different markers.

E. Clipping of spring grasp energy function

Here we illustrate implementation details of our spring grasp energy function E_{eq}. As the value of the contact margins $\epsilon_i(t_0), \epsilon_i(t_{\text{eq}})$ is in the range of $[-2, 1]$, in rare cases if contact margins are below -1 and logarithm mapping becomes undefined, directly clipping the value of the margin will set the gradient of the energy function to zero. Inspired by Leaky Relu [7], we switch to an auxiliary energy to encourage the contact force to stay close to surface normal if contact margins are below -1. Therefore, E_{sp} is defined as:

$$E_{\text{sp}} = -\sum_i^m \begin{cases} \log(\epsilon_i(t_0)+1) & \epsilon_i(t_0) > -1 \\ \log \frac{\boldsymbol{f}_i(t_0)}{||\boldsymbol{f}_i(t_0)||_2^2} \cdot \boldsymbol{n}_i(t_0) & \epsilon_i(t_0) \leq -1 \end{cases}$$
$$-\sum_i^m \begin{cases} \log(\epsilon_i(t_{\text{eq}})+1) & \epsilon_i(t_{\text{eq}}) > -1 \\ \log \frac{\boldsymbol{f}_i(t_{\text{eq}})}{||\boldsymbol{f}_i(t_{\text{eq}})||_2^2} \cdot \boldsymbol{n}_i(t_{\text{eq}}) & \epsilon_i(t_{\text{eq}}) \leq -1 \end{cases}$$

F. Fitting GPIS from partial point cloud

We estimate the true surface of an object from a partial point cloud by fitting a GPIS to it. Following [3], we use three sets of points to fit GPIS: a) Surface points, b) Exterior points, and c) Interior points. We assign a different value and noise level to each set of points.

1) Surface points: We use points from the partial point cloud as surface points. As the object surface is represented by the zero-level set of GPIS, we assign each point in the point cloud the value 0. We set the noise level of each point to be 0.005m.

2) Exterior points: To generate exterior points, we initially determine the axis-aligned bounding box of the surface points and scale it by 120% relative to the surface points' center. In total we have 14 exterior points which are located at each corner of the upscaled bounding box, as well as the midpoint of each bounding box edge. 14 points is sufficient for GPIS to distinguish between outside region and internal region. When fitting the GPIS, we found empirically that setting each exterior point value to be equal to half the length of the longest edge in the scaled bounding box works well. The noise level for each point is set to 0.2 meters.

3) Interior points: We compute interior points through convex combination of all surface points with random weights. To distribute these points evenly, rather than simply assigning weights from a uniform distribution to each surface point, we apply a softmax function to the initial random weights. This approach determines the actual weights for calculating interior points. For every object, we generate 50 interior points using this method. We found empirically that assigning each point a negative value equal to a quarter of the length of the longest edge of the upscaled bounding box works well. The noise level for each point is set at 0.05 meters.

G. Initial guesses for grasp planning

We initialize 7 wrist poses around the center of the oriented bounding box of the observed partial point cloud. The 7 poses consist of 5 with the palm facing the table and 2 with the palm perpendicular to the table. Tab.VI illustrates different initial guesses. Moreover, we always initialize joint angles as the relaxed joint pose defined in [17] and initialize the target position as a halfway point from each fingertip toward their center. Lastly, we use $k_1, k_2, k_3 = 80, k_4 = 160$ as initial controller gains.

H. Partial successes

To provide more information for Tab.II, we show the number of partial successes of each entry in Tab. VII.

wrist pose	position(xyz)	orientation(Euler)
1	-0.05, 0.0, 0.06	0, 0, 0
2	-0.04, 0.03, 0.03	0, 0, -45
3	-0.04, -0.03, 0.03	0, 0, 45
4	0.1, 0.06, -0.05	-90, 90, 0
5	0.0, 0.06, 0.05	-90, 0, 0
6	-0.0, -0.06, 0.03	0, 0, 90
7	0.02, -0.04, 0.03	0, 0, 135

Table VI: Initial wrist position and orientation, the unit of position is meter, and the unit of orientation is degree. We use intrinsic convention for Euler angles.

Appendix References

[A1] Akshay Agrawal, Brandon Amos, Shane T. Barratt, Stephen P. Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Neural Information Processing Systems*, 2019.

[A2] Solak Gokhan. Allegro hand kdl. https://github.com/ARQ-CRISP/allegro_hand_kdl, 2020.

[A3] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics Auton. Syst.*, 2016.

[A4] Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *RA-L*, 2022.

[A5] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023.

[A6] Albert Wu, Michelle Guo, and Karen Liu. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. In *CoRL*, 2022.

[A7] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. 2015.

	Ours (3 views)	Ours (2 views)	Ours (1 view)	w/o uncertainty (1 view)	w/o pregrasp (1 view)	Bilevel-high (2 views)	Bilevel-low (2 views)	Bilevel-heu (2 views)
Mustard bottle	1	0	0	3	1	2	1	1
Lego	1	1	0	1	0	1	3	1
Pyramid	0	0	0	2	1	3	3	2
Campell can	0	0	0	1	0	1	1	2
Cheezit box	0	0	0	2	1	2	3	2
Mug	1	1	2	3	2	1	3	2
Orange	0	2	1	0	0	1	2	0
Coffee bottle	0	0	1	1	1	1	2	2
Spam	2	1	2	2	2	2	2	3
Plane	0	2	1	2	2	4	1	3
Car	1	1	1	3	3	2	2	4
Banana	0	0	1	2	1	1	0	2
Pear	0	0	1	0	2	2	2	2
Small box	0	0	0	1	3	2	2	1
Total	6	8	10	23	19	25	27	27

Table VII: Number of partial success for each experiment.