FRoGGeR: Fast Robust Grasp Generation via the Min-Weight Metric

Albert H. Li†, Preston Culbertson‡, Joel W. Burdick‡, and Aaron D. Ames†,‡

***Abstract*— Many approaches to grasp synthesis optimize analytic quality metrics that measure grasp robustness based on finger placements and local surface geometry. However, generating feasible dexterous grasps by optimizing these metrics is slow, often taking minutes. To address this issue, this paper presents FRoGGeR: a method that quickly generates robust precision grasps using the *min-weight metric*, a novel, almost-everywhere differentiable approximation of the classical ϵ grasp metric. The min-weight metric is simple and interpretable, provides a reasonable measure of grasp robustness, and admits numerically efficient gradients for smooth optimization. We leverage these properties to rapidly synthesize collision-free robust grasps—typically in less than a second. FRoGGeR can refine the candidate grasps generated by other methods (heuristic, data-driven, etc.) and is compatible with many object representations (SDFs, meshes, etc.). We study FRoGGeR's performance on over 40 objects drawn from the YCB dataset, outperforming a competitive baseline in computation time, feasibility rate of grasp synthesis, and picking success in simulation. We conclude that FRoGGeR is fast: it has a median synthesis time of 0.834s over hundreds of experiments.**

I. Introduction

The success of data-driven methods for grasp synthesis has fundamentally changed manipulation in recent years. Traditional methods [1], [2] for grasp synthesis focused largely on optimizing analytic quality metrics (e.g., the *largest inscribed ball metric* [3], [4], [5], which we call the ϵ *metric* as in [6]), which use hand-object contact points to measure a grasp's robustness to external perturbations. However, these methods suffer some drawbacks in practice: traditional metrics are often hard to optimize and may be non-differentiable, meaning grasps must be synthesized with slower sampling-based methods. Further, they are sensitive to the object geometry (i.e., the surface location and normals), which requires detailed object models to be constructed offline, e.g., by using object scanning rigs [7], [8].

To address these shortcomings, a number of authors consider data-driven methods for grasp synthesis (for a detailed survey, see [9]) that seek to learn grasping policies or metrics depending solely on raw sensor data, such as RGB images or point clouds. A wide variety of approaches have emerged, including using supervised learning to train CNNs to estimate grasp quality from images [10], identifying class-level keypoints to find features appropriate for manipulation [11], or learning a generative model conditioned on depth images [12]. These methods, however, have focused nearly exclusively on generating antipodal grasps for parallel-jaw grippers, or power grasps for dexterous hands.

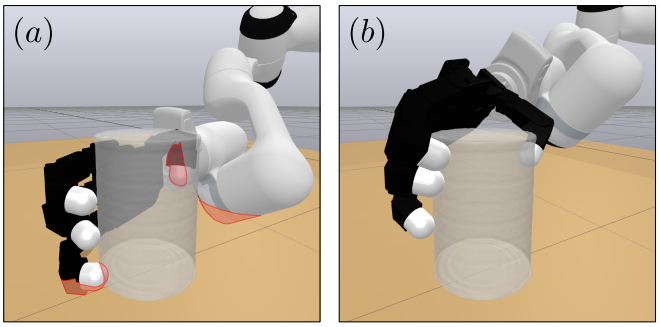

Fig. 1: FRoGGeR quickly refines infeasible multi-finger dexterous grasps into kinematically-feasible, collision-free grasps using gradient-based nonlinear optimization. We leverage the *min-weight metric* presented in Sec. II to maximize refined grasp robustness. **(a)** A candidate grasp q_0 with collisions between the object, robot, and table (highlighted red). **(b)** The feasible and robust refined grasp q^*. In this example, the entire process—from sampling to refinement—took just 0.816 seconds. The median computation time over all experiments was 0.834 seconds, compared to minutes required by methods like GraspIt! [1], [13], [14].

In this work, we consider the problem of quickly refining an initial pose for a dexterous hand into a robust precision grasp for a particular object. Compared to power grasps, precision grasps are more useful for manipulation tasks that require delicate or accurate movements, such as tool use or bin packing. Our goals are twofold. First, we seek to generate these grasps quickly (in seconds rather than minutes for current methods [1], [13], [14]) while enforcing kinematic and collision constraints. Second, we seek to balance common trade-offs of grasp synthesis methods in terms of performance, speed, and interpretability.

A. Contributions

The main contribution of this work is the formulation of grasp synthesis/refinement as a nonlinear optimization problem that leverages a novel, almost-everywhere differentiable approximation of the ϵ metric: the *min-weight metric*. The end result is FRoGGeR, a framework for fast robust grasp generation. The optimization problem underlying FRoGGeR can, due to the properties of the min-weight metric, be solved efficiently using commercial solvers. Additionally, FRoGGeR allows us to explicitly enforce kinematic and collision constraints on generated grasps while harnessing the speed of gradient-based optimization.

We aim for our work to be compatible with existing methods in the grasping community. For instance, while FRoGGeR can synthesize grasps with no prior knowledge, it may also refine infeasible or suboptimal grasps computed by learning-based methods. Further, this paper represents object

† A. H. Li and A. D. Ames are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA, {alberthli, ames}@caltech.edu.

‡ P. Culbertson, J. W. Burdick, and A. D. Ames are with the Department of Civil and Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125, USA, {pculbert, jwb}@caltech.edu.

geometry implicitly using *signed distance fields* (SDFs), which means we can leverage existing work that learns SDFs of objects from sensor data [15]. To allow the use of mesh-based object representations, we also present practical approximations of the SDF, its gradient, and its Hessian computable using only the mesh.

In summary, our contributions are as follows:

- FRoGGeR: a **f**ast **ro**bust **g**rasp **ge**nerato**r** built on the *min-weight metric*, a novel, almost-everywhere differentiable approximation of the ϵ grasp metric with a numerically efficient gradient;
- a practical procedure based on nonlinear optimization with an open-source implementation that generates feasible grasps on the order of seconds; and
- numerical experiments and simulations comparing grasps generated by our method to prior work, thereby demonstrating the speed of the proposed approach.

The open-source implementation of FRoGGeR is available at: github.com/alberthli/frogger.

B. Related Work

The majority of recent work in the grasping literature concerns *parallel-jaw* grasps, which admit simple parameterizations due to the low number of DOFs and ease of control [10], [16], [17], [18], [9]. Multifinger, dexterous grasping introduces numerous challenges, as the grasp parameterization must specify the states of each finger, and the high dimensionality of this representation demands more fine-grained control and better sensing. Moreover, the complex kinematics and potential for self-collisions complicate the search for feasible grasps. In turn, the problem of synthesizing dexterous grasps, particularly using data-driven methods, has received far less attention than parallel-jaw grasps.

Many classical methods for multifinger grasp synthesis ignored kinematic and collision constraints and only optimized for contact location by leveraging analytic grasp metrics [2], [19]. The GraspIt! simulator explicitly considers these constraints but simplifies the optimization problem by searching a lower-dimensional space of "eigengrasps" using simulated annealing [1]. This approach has several downsides, including the need for the user to define eigengrasps for new hands (which is highly non-trivial), and slow computational speed in general. Overall, analytic metrics have two main drawbacks: (i) their usage is typically slow, often due to non-smoothness, and (ii) they demand high-fidelity estimates of object geometry and contact locations, diminishing their efficacy, especially on novel objects [6].

In response to these limitations, numerous authors sought to develop data-driven methods for dexterous grasping. Existing approaches include discriminative models, (i.e., those that seek to estimate the quality of a particular grasp), and generative models, which seek to directly generate grasps for novel objects, conditioned on the object geometry or perceptual data. Among these, some only enforce hand kinematics and check collisions post-hoc [6], [20]; others only optimize for contact points and check kinematic feasibility post-hoc [21]; others learn grasp pre-shapes rather than reasoning about contact [22], [23]. We refer the reader to [9] for a detailed survey of data-driven approaches to grasp synthesis.

One reason for these limitations is the difficulty in casting the nonlinear constrained grasp optimization problem in a computationally tractable way. Among methods addressing this challenge, collision constraints are typically penalized instead of enforced, leading synthesized grasps to have high amounts of infeasible interpenetration [18], [14], [24]. Other attempts at solving the unrelaxed problem are computationally prohibitive (e.g., minutes to hours in [25]).

In this work, we do not address the perception-based challenges of analytic metrics and assume knowledge of the object's geometry and pose. Instead, we focus on mitigating their slow speed with the view that, despite their drawbacks, these metrics still provide a useful framework for grasping that is agnostic to robot model, object representation, and quality of available data. To that end, we build on prior works that formulate differentiable approximations of analytic force closure measures to recover robust grasps on any multifinger arm/hand system using bilevel optimization.

These prior works propose methods of varying complexity, including solving and differentiating a sequence of linear programs (LPs) [26], a sequence of semidefinite programs (SDPs) [27], a sum of squares program [28], or a single SDP that only approximates force closure [24], [18]. In contrast, we propose in Sec. II a single LP whose optimal value mathematically indicates force closure and whose maximization empirically yields robust grasps. Besides bilevel methods, other methods exist that optimize the full grasp configuration by formulating grasp synthesis as a *maximum a posteriori* inference problem with variants that do and do not consider collision [29], [30], [22].

The method of Wu *et al.* [31] is most similar to ours as they also propose solving a bilevel optimization program with smooth collision constraints. However, instead of optimizing for robustness, they solve a feasibility problem and impose a force closure constraint parameterized as a quadratic program while training a conditional variational autoencoder (CVAE) to output performant initial grasps. We compare FRoGGeR's formulation to theirs in Sec. IV.

C. Preliminaries

We assume a fixed-base, fully-actuated serial manipulator and dexterous hand with n_c fingers contacting the object. Denote by n the total DOFs of the system and $q \in \mathcal{Q} \subset \mathbb{R}^n$ the generalized positions. We let $FK_i(q)$ and $J_i(q)$ denote the forward kinematics and Jacobian of prescribed contact point i (we can relax this by computing a parameterization of the fingertip geometries, but do not for simplicity). Define the hand Jacobian as $J_h = \text{blkdiag}(J_1, \dots, J_{n_c})$. We aim to manipulate a rigid object denoted $\mathcal{O}$ with surface $\partial\mathcal{O}$ and body frame $\{O\}$. The pose of a frame $\{B\}$ with respect to a frame $\{A\}$ is expressed $T_{AB} \in SE(3)$, and let $R_{AB} \in SO(3)$ represent the relative position and orientation of $\{B\}$ with respect to $\{A\}$. The world frame is denoted $\{W\}$.

We refer to the pair (q^*, T_{WO}) as a *grasp*, where q^* is a *feasible* configuration (i.e., no collisions and valid hand-

object contact). In this work, we will model the fingers as point contacts with friction. We can thus define $G(q)$, the *grasp map*, which maps a vector of contact forces expressed in their local contact frames, $F_C \in \mathbb{R}^{3n_c}$, to wrenches in the object frame $w_O \in \mathbb{R}^6$, i.e., we can write $w_O = G(q)F_C$.

We assume a Coulomb friction model, so there is no slip if contact forces remain in the *friction cone*, i.e., if $\|F_C^t\| \leq \mu F_C^n$, where F_C^t and $F_C^n > 0$ denote the tangent and normal components respectively. We assume a pyramidal friction cone approximation [32] with n_s sides and let $m = n_c n_s$ denote the total number of associated *basis wrenches* forming the subset $\mathcal{W}$ of the *grasp wrench space* $\mathscr{W} \subseteq \mathbb{R}^6$. We assume at least 7 affinely independent basis wrenches and let them form the columns of the wrench matrix $W(q) \in \mathbb{R}^{6\times m}$.

We say a grasp is *force closure* if it can resist arbitrary disturbance wrenches in any direction, which is implied if the origin of the grasp wrench space $\mathscr{W}$ lies in the convex hull of $\mathcal{W}$, denoted $\text{conv}(\mathcal{W})$ [4]. For a thorough treatment of grasping fundamentals, we refer the reader to [32, Ch. 5].

II. The Min-Weight Grasp Metric

This section introduces the *min-weight metric*, a simple non-binary indicator of force closure we use as an optimization objective. Specifically, we treat it as a differentiable proxy for the ϵ metric, which measures the robustness of force closure grasps by reporting the radius of the largest origin-centered ball inscribed in $\text{conv}(\mathcal{W})$ [3], [4].

In the sequel, we assume that $\text{int}(\text{conv}(\mathcal{W})) \neq \emptyset$. To check whether $0 \in \text{conv}(\mathcal{W})$, we can solve the following linear feasibility problem [33] over variables $\alpha = (\alpha_1, \ldots, \alpha_m)^\top$, where $\mathbb{1}_m \in \mathbb{R}^m$ is the vector of 1s:

$$\begin{aligned} \text{find} \quad & \alpha && \text{(1a)} \\ \text{subject to} \quad & W\alpha = 0 && \text{(1b)} \\ & \mathbb{1}_m^\top \alpha = 1 && \text{(1c)} \\ & \alpha \succeq 0. && \text{(1d)} \end{aligned}$$

That is, $0 \in \text{conv}(\mathcal{W}) \iff \exists \alpha^*$ satisfying (1b)-(1d), which is equivalent to the existence of an equilibrium wrench.

A. The Min-Weight Metric $\ell^*(q)$ and its Properties

The key idea of the min-weight metric is to relax constraint (1d) by allowing negative weights α. If the minimum weight in α is non-negative, the feasibility problem is satisfied, so $0 \in \text{conv}(\mathcal{W})$. This motivates the following LP:

$$\begin{aligned} \ell^*(q) = \underset{\alpha\in\mathbb{R}^m,\, \ell\in\mathbb{R}}{\text{maximize}} \quad & \ell && \text{(2a)} \\ \text{subject to} \quad & W(q)\alpha = 0 && \text{(2b)} \\ & \mathbb{1}_m^\top \alpha = 1 && \text{(2c)} \\ & \alpha \succeq \ell \mathbb{1}_m. && \text{(2d)} \end{aligned}$$

Thus, $\ell^*(q)$ and $\nabla \ell^*(q)$ are defined even when a grasp is not force closure (i.e., when $\ell^* < 0$ in the case of non-physical "pulling" contact forces). This admits a procedure to make grasps force closure via smooth optimization made possible by the dependence of $\ell^*(q)$ on q via the wrench matrix $W(q)$ in constraint (2b). The following result formally relates ℓ^* and force closure status.

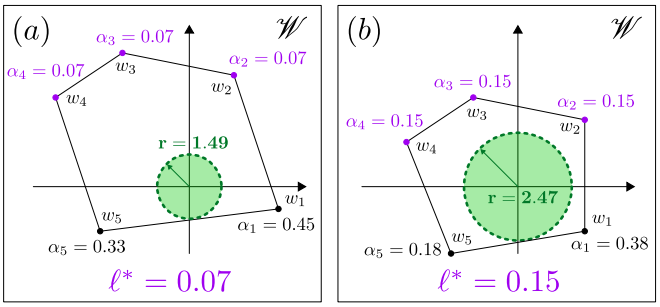

Fig. 2: Toy examples of the min-weight metric. The weights α_i are associated with wrenches $w_i \in \mathcal{W} \subseteq \mathscr{W}$. The minimum weights are highlighted purple. The largest inscribed ball about the origin in each convex hull is highlighted green. We empirically observe that the minimum weight ℓ^* is strongly correlated with the radius of this ball, i.e., the ϵ metric.

Theorem 1. *If $\mathcal{W}$ contains a subset of 7 affinely independent basis wrenches, then problem* (2) *always has a feasible solution. Further, the optimal solution (α^*, ℓ^*) satisfies*

(i) ***[Non-Force Closure]*** $\ell^*(\mathcal{W}) < 0 \iff 0 \notin \text{conv}(\mathcal{W})$,
(ii) ***[Robust Closure]*** $\ell^*(\mathcal{W}) > 0 \iff 0 \in \text{int}(\text{conv}(\mathcal{W}))$,
(iii) ***[Only Equilibrium]*** $\ell^*(\mathcal{W}) = 0 \iff 0 \in \partial\,\text{conv}(\mathcal{W})$,

where ∂S denotes the boundary of a set S.

Proof. We first prove the feasibility claim. Since ℓ is unconstrained, it suffices to show that there always exists α satisfying (2b) and (2c). Let $W' \in \mathbb{R}^{6\times 7}$ denote a submatrix of W with 7 affinely independent columns and α' the associated weights in α. Set all other weights in α to 0. Let $\overline{W}' = \begin{bmatrix}(W')^\top & \mathbb{1}_7\end{bmatrix}^\top \in \mathbb{R}^{7\times 7}$ and $\bar{0} = \begin{bmatrix}0^\top & 1\end{bmatrix}^\top \in \mathbb{R}^7$.

The columns of W' are affinely independent in $\mathbb{R}^6$ if and only if the columns of $\overline{W}'$ are linearly independent in $\mathbb{R}^7$ [34, Exercise 1.1]. Thus, $\overline{W}'$ is invertible, so we can always find α' satisfying $\overline{W}'\alpha' = \bar{0}$. Equivalently, $W'\alpha' = 0$ and $\mathbb{1}_7^\top \alpha' = 1$, implying $W\alpha = 0$ and $\mathbb{1}_m^\top \alpha = 1$.

Proof of (i). By feasibility problem (1), $\ell^* < 0$ implies (1) has no solution, so equivalently, $0 \notin \text{conv}(\mathcal{W})$.

Proof of (ii). $\ell^*(\mathcal{W}) > 0$ if and only if $\exists \alpha \succ 0$ such that $W\alpha = 0$. Since we assume $\text{conv}(\mathcal{W})$ has nonempty interior, its relative interior is its interior, so $0 \in \text{int}(\text{conv}(\mathcal{W}))$ if and only if $\exists \alpha \succ 0$ such that $W\alpha = 0$ [34, Exercise 3.1].

Proof of (iii). Follows immediately from (i) and (ii). □

Thereom 1 states that under mild assumptions, the sign of ℓ^* indicates force closure, justifying its maximization. Heuristically, very negative values of ℓ^* indicate a grasp is far from force closure while very positive values indicate the origin lies well within $\text{conv}(\mathcal{W})$ (see Fig. 2). This motivates using ℓ^* as an approximate measure of robustness.

Since $\ell^* \leq 1/m$, the *normalized min-weight metric* $\bar{\ell}^* = m\ell^*$ is well-defined, allowing us to specify the constraint $\bar{\ell}^* \geq k_\ell$, where $k_\ell \in [0, 1]$ is a lower bound on the desired grasp robustness. In experiments, we use $k_\ell = 0.3$.

We note that while using ℓ^* to measure force closure is theoretically justified, its use as a proxy for the ϵ metric is not, since $\ell^* \gg 0$ does not guarantee a large ball is contained in $\text{conv}(\mathcal{W})$. Nevertheless, empirically, we find that ℓ^* and

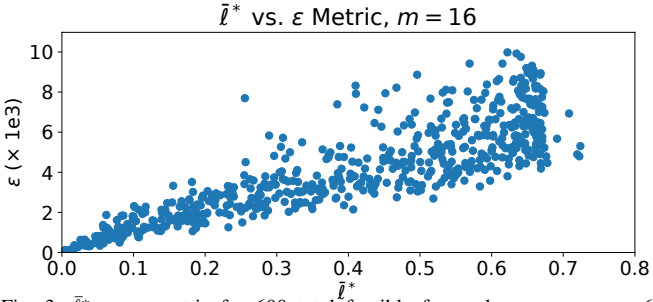

Fig. 3: $\bar{\ell}^*$ vs. ϵ metric for 600 total feasible force closure grasps on 6 objects from the YCB dataset [8] generated using FRoGGeR for $m = 16$ basis wrenches. We report a Pearson correlation of 0.865 and observe a clear improvement in the worst-case ϵ value as $\bar{\ell}^*$ increases. Because $\bar{\ell}^*$ is bounded above, we typically observe a ceiling in its value (here, ≈ 0.7) which degrades the correlation as grasp robustness increases. However, we see that ℓ^* is very informative in the regime where grasps are barely force closure (bottom left), i.e., where they are least robust.

the ϵ metric are strongly correlated and maximizing ℓ^* improves a lower bound on the ϵ value (see Fig. 3).

Finally, as in many classical metrics, ℓ^* is not invariant to the object frame [5, Ch. 13.5]. While methods exist that address this [19], we do not explore them in this work.

B. Computing $\nabla \ell^*(q)$ with Differentiable Optimization

We compute $\nabla \ell^*(q)$ where it is defined using implicit differentiation of the KKT conditions [35] and exploit the resulting structure to compute it quickly.

For brevity, let $x = (\alpha, \ell)$ and express (2) as

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^\top x && \text{(3a)} \\ \text{subject to} \quad & A_{eq} x = b_{eq} && \text{(3b)} \\ & A_{in} x \preceq 0. && \text{(3c)} \end{aligned}$$

Let λ and ν denote the Lagrange multipliers associated with inequality and equality constraints respectively. As in [35], we write the stationarity, primal feasibility, and complementary slackness conditions for (2):

$$H = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} := \begin{bmatrix} c + A_{in}^\top \lambda + A_{eq}^\top \nu \\ \lambda \odot (A_{in} x) \\ A_{eq} x + b_{eq} \end{bmatrix}, \tag{4}$$

where $\odot$ denotes the Hadamard product. Solving the system $H = 0$ is necessary and sufficient to solve any LP since it is convex and always satisfies Slater's condition. Let $D_{(\cdot)} f$ denote the Jacobian of a vector-valued function f with respect to variables $(\cdot)$. Since $H = 0$ at the optimal solution, by implicit differentiation,

$$\begin{aligned} & D_q H(x^*, \lambda^*, \nu^*, q) = 0 \\ \implies & D_{(x,\lambda,\nu)} H(x^*, \lambda^*, \nu^*, q) D_q (x^*, \lambda^*, \nu^*)(q) \\ & \quad + D_q H(x^*, \lambda^*, \nu^*, q) = 0. \end{aligned} \tag{5}$$

We can compute $D_q H$ explicitly or with autodifferentiation through the primitive wrench matrix $W(q)$, which is derived from the grasp map $G(q)$, with details deferred to our open-source implementation. We compute $\nabla \ell^*(q)$ for the case where $D_{(x,\lambda,\nu)} H$ is invertible and apply the result without further justification like a subgradient in the singular case (similar to the non-differentiable case in [36]).

In particular, $\nabla \ell^*(q)^\top$ is given by the last row of $D_q x^*(q)$, which we can compute via the following result:

Proposition 1 (Gradient Exploit)**.** *Let $(\cdot)^\dagger$ denote the Moore-Penrose pseudoinverse. Then,*

$$D_q x^*(q) = \begin{bmatrix} D_q \alpha^*(q) \\ \nabla \ell^*(q)^\top \end{bmatrix} = \begin{bmatrix} \operatorname{diag}(\lambda^*) A_{in} \\ A_{eq} \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ D_q H_3 \end{bmatrix}. \tag{6}$$

Proof. See App. A. □

Proposition 1 allows efficient computation of $\nabla \ell^*(q)$, especially when m is "small" ($\lesssim 40$). For example, when $m = 16$ (e.g., using a square pyramidal approximation for a 4-fingered hand), we find that $\ell^*(q)$ and $\nabla \ell^*(q)$ can be computed together in about $3.9ms$ with `cvxpylayers` [36] on an Nvidia A6000 GPU, whereas using our exploit, they are computed in $0.18ms$ on an Intel i9 CPU, a $22\times$ speedup.

III. The FRoGGeR Formulation

A. The Grasp Refinement Problem

FRoGGeR *refines* a candidate grasp configuration q_0 into a locally optimal one q^* by solving the following nonlinear bilevel optimization program (recalling that $\bar{\ell}^* = m\ell^*$):

$$\begin{aligned} \underset{q}{\text{maximize}} \quad & \ell^*(q) \\ \text{subject to} \quad & q_{\min} \preceq q \preceq q_{\max} \\ & \bar{\ell}^*(q) \geq k_\ell \\ & FK_i(q) \in \partial\mathcal{O},\ i = 1, \ldots, n_c \\ & \text{No (non-finger/object) collision.} \end{aligned} \tag{FRoGGeR}$$

To enforce joint limits, we constrain the robot configuration q to lie between minimum and maximum values $q_{\min}$ and $q_{\max}$. Further, we enforce that the fingertips lie on the object surface $\partial\mathcal{O}$ and that no rigid bodies are interpenetrating.

To express these constraints mathematically, we first parameterize $\partial\mathcal{O}$ as the 0-level set of a twice-differentiable SDF $s : \mathbb{R}^3 \to \mathbb{R}$, which reports the distance of query points $x \in \mathbb{R}^3$ to $\partial\mathcal{O}$, with $s(x) < 0$ for all points in $\mathcal{O}$:

$$s(x) = \begin{cases} -\operatorname{dist}(x, \partial\mathcal{O}), & x \in \mathcal{O}, \\ +\operatorname{dist}(x, \partial\mathcal{O}), & x \notin \mathcal{O}. \end{cases}$$

Second, we consider every possible pair of geometries we would like to prevent from colliding and parameterize the collision status using the differentiable constraints $\sigma\left(o_A^{(j)}, o_B^{(j)}; q\right),\ j = 1, \ldots, n_p$, where n_p is the number of collision pairs and σ is an SDF between two geometries, at least one of whose state depends smoothly on q. For pair j, we enforce a minimum safety margin of $d_j > 0$ unless it is a finger-object pair, for which we allow a small amount of interpenetration by specifying $d_j < 0$. Thus, we can express optimization program (FRoGGeR) formally as

$$\begin{aligned} \text{maximize} \quad & \ell^*(q) && \text{(7a)} \\ \text{subject to} \quad & q_{min} \preceq q \preceq q_{max} && \text{(7b)} \\ & \bar{\ell}^*(q) \geq k_\ell && \text{(7c)} \\ & s(FK_i(q)) = 0,\ i = 1, \ldots, n_c && \text{(7d)} \\ & \sigma\left(o_A^{(j)}, o_B^{(j)}; q\right) \geq d_j,\ j = 1, \ldots, n_p. && \text{(7e)} \end{aligned}$$

B. Gradients of the Constraint Functions

To use gradient-based methods, we must compute the gradients of the objective and each constraint. The gradient of constraint (7d) is immediately given by $J_i^\top(q)\nabla s(FK_i(q))$ for $i = 1, \dots, n_c$. For constraint (7e), we compute for each pair of geometries (o_A, o_B) the *witness points* (p_A, p_B). If the pair is colliding, then the witness points are the two points of furthest penetration. If the pair is not colliding, then they are the two closest points. Let $I_c = 1$ indicate collision of a pair and $I_c = 0$ otherwise. Then,

$$\begin{aligned} &\sigma(o_A, o_B; q) = (-1)^{I_c+1} \|p_A - p_B\| \\ \implies &\nabla_q \sigma(o_A, o_B; q) = (-1)^{I_c} \left(J_B^\top - J_A^\top\right) \hat{n}_{AB}, \end{aligned} \tag{8}$$

where J_A and J_B are the Jacobians at witness points A and B and $\hat{n}_{AB}$ is the unit vector from p_A to p_B.

We use `Drake` [37] to compute witness points for all geometry pairs in a scene. To speed up computation, we represent nonconvex bodies as a union of convex polytopes computed using V-HACD [38]. To reduce the amount of checked pairs, `Drake` culls distant pairs using a broadphase algorithm and we set the associated gradients to 0. When two geometries have exactly 0 signed distance, the gradient may not be defined since $p_A - p_B = 0$. In this case, we use the previous value of $\hat{n}_{AB}$, which is initialized randomly.

C. Object Surface Representations

The SDF representation of objects is convenient for reasoning about collision and also geometric properties, since the outward-pointing surface normal at a point $p \in \partial\mathcal{O}$ is given by $\nabla s(p)$ and principal curvatures can be computed from the Hessian $\nabla^2 s(p)$. However, supplying the true object SDF s is non-trivial. Some approaches learn this representation [15], [30], while most avoid learning by using the object's mesh or a point cloud [10], [21], [28]. Since these representations are all widely used, it is desirable for grasp synthesis methods to be compatible with any of them.

In the case of a learned or analytical SDF, computing the requisite gradients can be done via autodifferentiation. Further, if provided a dense enough point cloud, the Poisson surface reconstruction algorithm can return a watertight mesh [39]. Therefore, we focus on the case of meshes.

We assume that there exists a true smooth SDF s and denote the approximation computed with the object mesh as $\tilde{s}$. To compute $\tilde{s}$ and $\nabla\tilde{s}$, we use the open-source signed distance query provided by `open3d` that also computes the closest point on a mesh to any query point $p \in \mathbb{R}^3$ [40]. Then, given a closest point p' to p, the gradient is simply

$$\nabla\tilde{s}(p) = \text{sign}(\tilde{s}(p)) \frac{p - p'}{\|p - p'\|}, \tag{9}$$

where when $\tilde{s}(p) = 0$, $\nabla\tilde{s}(p)$ is the mesh normal at p'.

To compute whether a grasp is (robustly) force closure, we must compute the grasp map $G(q)$, which depends on the contact frames associated with fingertip positions $p \in \mathbb{R}^3$ [4]. The normal component of each contact frame is the inward-pointing surface normal, i.e., $\hat{n}(p) = -\nabla s(p)$. Therefore, to differentiate any objective or constraint that depends on measures of force closure, we require $\nabla^2 s(p)$.

However, when the object is parameterized as a mesh, the surface is piecewise flat, so $\nabla^2\tilde{s}(p) \equiv 0$ wherever it is defined even if $\nabla^2 s(p) \neq 0$. Other works present methods of varying complexity to compute $\tilde{s}$ and $\nabla\tilde{s}$ that involve solving a quadratic program or deep learning, but do not consider the problem of computing the Hessian of s [28] [18].

Here, we propose a coarse but efficient approximation. Let $p_1 = p \in \mathbb{R}^3$ be an arbitrary query point. Fix a small constant $\delta > 0$, randomly select 2 unit vectors denoted $d_2, d_3 \in \mathbb{R}^3$, and define $p_i = p_1 + \delta d_i$ for $i = 2, 3$. By (9), we have $\nabla\tilde{s}(p_i)$ for all $i = 1, 2, 3$. Finally, fix $d_1 = \nabla s(p_1)$.

The directional derivative of $\nabla s(p)$ in a direction v is

$$\nabla^2 s(p)[v] = \lim_{\delta \to 0} \left(\nabla s(p + \delta v) - \nabla s(p)\right)/\delta. \tag{10}$$

Further, for twice-differentiable functions, we must have $\nabla^2 s(p)[v] = \nabla^2 s(p)v$. Thus, using our perturbation directions d_i and a finite-difference approximation of the directional derivatives, $y_i = (\nabla s(p_1 + \delta d_i) - \nabla s(p_1))/\delta$, we can write a system of equations to estimate $\nabla^2 s(p)$,

$$\Sigma \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}, \tag{11}$$

by solving for $\Sigma \in \mathbb{R}^{3\times3}$. We note $y_1 = 0$ since $\nabla s(p)$ is unchanging (close to the surface) along $\nabla s(p)$ and d_1, d_2, d_3 are linearly independent with probability 1. Σ denotes an initial estimate for $\nabla^2 s(p)$ that may not be symmetric, so we simply choose $\nabla^2\tilde{s}(p) = (\Sigma + \Sigma^\top)/2$.

The quality of our estimate $\nabla^2\tilde{s}(p)$ is sensitive to the choice of δ and the properties of the mesh. Empirically, we find that setting δ to be roughly 10 times the average mesh edge length yields accurate enough gradients for grasp refinement. Using `open3d`, we find the time to estimate all of $\tilde{s}$, $\nabla\tilde{s}$, and $\nabla^2\tilde{s}$ is on the order of 0.1ms.

IV. Experiments

We describe the high-level experimental setup and defer a detailed discussion to App. B-E. We use the 7-DOF Franka Research 3 and 16-DOF 4-fingered Allegro hand. The system is mounted on a flat tabletop and each target object is spawned with a fixed initial pose over all trials for repeatability, since the arm/hand configuration is allowed to vary arbitrarily.

We evaluate the robustness of FRoGGeR by executing 20 "shaky pickups" per object in simulation using `Drake`. To do so, we generate a pick trajectory where the end-effector is lifted $10cm$ in $1s$ and then held for $1.5s$. We add sinusoidal perturbations to this trajectory with amplitude $3mm$ and varying frequency in all spatial axes $0.25s$ after the pick begins until the end of the simulation. A pick fails if either (1) the object rotates by more than 30° or if the object deviates from the pick trajectory by more than $7.5cm$ at any point; or (2) the total grasp synthesis time exceeds 1 minute.

We remark that our shaking test is more dynamic than others in the literature (e.g. [17], [31]), which either do not shake or classify a shake only as a linear movement in space

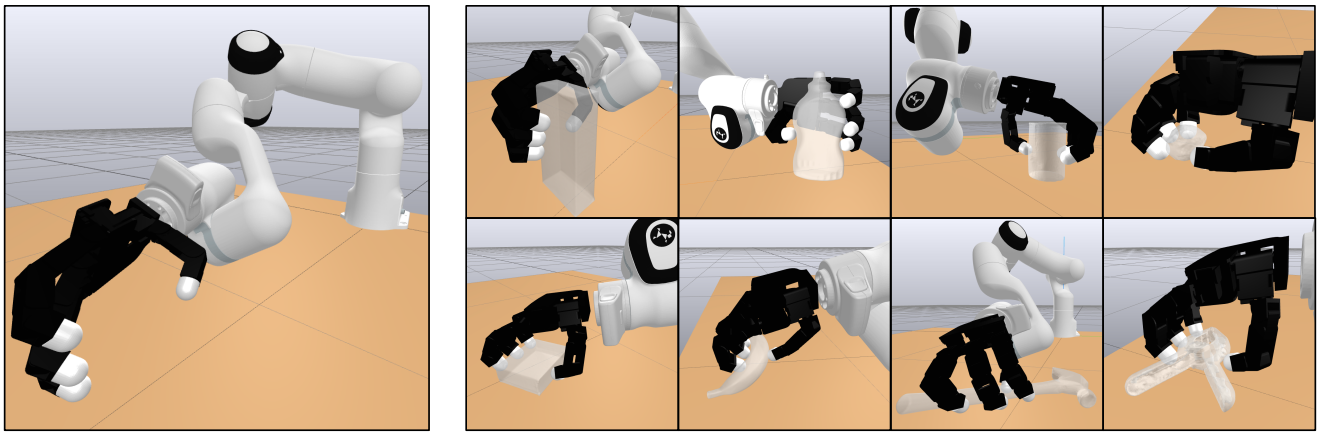

Fig. 4: **Left.** A representative pre-shaped grasp q_0 from our heuristic sampler. The initial width of the grip is determined by the object's bounding box. **Right**. Example refined grasps q^* from our experiments. We can produce robust grasps for highly varying objects that are tall, round, small, flat, long, or otherwise irregular. Top row: sugar box, mustard bottle, soup can, strawberry. Bottom row: pudding box, banana, hammer, large clamp.

with zero gravity. In contrast, we simulate gravity as well as sustained high-frequency perturbations in all directions.

We compare FRoGGeR's performance on the pickup task to a baseline presented by Wu *et al.*, which only enforces force closure without optimizing for robustness [31].

The controller used in simulation is given by $\tau = J_h^\top R_{WC} F_C^* + (I - J_h^\top (J_h^\top)^\dagger)\tau_{\text{joint}}$. F_C^* is computed by solving for the optimal contact forces to resist external wrenches and errors in the object's pose (e.g., [31]). τ_{joint} is the concatenation of arm torques tracking the pick trajectory with hand torques that drive the hand configuration q_h towards the optimized one q_h^*. We project τ_{joint} to the null space of J_h to avoid affecting the fingertip locations.

To obtain initial configurations q_0, we use a heuristic sampler that noisily aligns the palm with the axes of the object's oriented bounding box with probabilities proportional to the box side lengths, motivated by observations of preferred human grasps [41]. We choose a width for the fingertips by computing the width of the appropriate axis of the bounding box. The palm is then placed $4cm$ from the object. To obtain the configuration variables, we solve an inverse kinematics (IK) problem as in [31], but we do not enforce collision constraints or force the fingertips to lie on the object surface. Thus, we only consider infeasible candidate grasps. To solve (7), we use the `NLopt` [42] implementation of SLSQP [43].

Our choice to use a coarse sampling heuristic instead of a more performant method is intentional, as our goal is to evaluate FRoGGeR's robustness to the quality of the initial guess. We control for the resulting decrease in performance by evaluating the relative performance of our method versus the baseline under these conditions.

Thus, we do not evaluate the CVAE sampler from [31]. Further, we found that the quality of CVAE-generated grasps was not consistent for all objects in our dataset and its performance was on par with our heuristic on a small set of test objects. Ultimately, we chose to synthesize 4-finger grasps to capture the full dexterity of the Allegro hand, which are incompatible with the 3-finger CVAE sampler.

The only difference between our method and the baseline is that in (7), FRoGGeR maximizes $\bar{\ell}^*(q)$ with constraint $\bar{\ell}^*(q) \geq 0.3$, while the baseline has no objective and (7c) is replaced with the bilevel force closure equality constraint described in [31]. Otherwise, the same IK routines, sampler, collision geometries, and controller were used.

A. Object Data Processing

We only present results on objects parameterized as meshes. When supplied with analytical SDFs or well-trained deep SDFs, our method was generally both fast and performant. We use meshes to demonstrate our approach on non-smooth object representations and to validate the usefulness of the Hessian approximation from Sec. III-C.

The objects used in our experiments are from a pruned subset of the YCB dataset [8]. First, we removed all objects that were too large, small, or thin to reasonably grasp with 4 fingers from a flat table, as well as deformable or multibody objects. Second, since the YCB meshes are not watertight, we attempted to reprocess them by densely sampling points on each mesh and running Poisson reconstruction. Of these, we removed objects for which we could not produce watertight meshes due to poor data quality (e.g. from transparency, thin walls, etc.). We note that FRoGGeR works even on non-watertight meshes of adequate quality, but we take this step to eliminate the effect of poor meshes on our results. In total, we test on 43 objects belonging to three categories: *spheroids*, like fruits and balls; *boxes/cylinders*, like food containers, cans, or large cups; and *adversarial* objects with irregular geometry, like tools or very flat/long objects.

For simplicity, we set the friction coefficient to be $\mu = 0.7$ for all objects, which is reasonable for the rubbery Allegro fingertips on mostly plastic objects. We also assumed a uniform density of 150 kg/m^3 (as in [17]) and computed masses using the volume of the processed meshes. The optimizer assumed a more conservative friction coefficient of $\mu = 0.5$ and the controller was provided the mass.

B. Results and Discussion

We report values related to the quality of the grasp (pick success, ϵ metric value, and $\bar{\ell}^*$) as well as values regarding the runtime of each method in Table I. We find that overall,

category	method	% converged ↑	% pick success ↑	ϵ ($\times$1e3) ↑	normalized ℓ^* ↑	time per solve (s) ↓	num. solves ↓	total time (s) ↓
sphere	baseline	12.9% (31/240)	83.9% (26/31)	2.7 (1.6, 3.6)	0.32 (0.19, 0.39)	0.67 (0.44, 1.00)	68 (61, 73)	27.2 (13.5, 36.2)
(240 total)	FRoGGeR	**97.9%** (235/240)	**95.3%** (224/235)	**4.9** (4.2, 5.5)	**0.67** (0.55, 0.72)	**0.31** (0.18, 0.49)	**2** (1, 4)	**0.57** (0.30, 1.2)
box/cyl	baseline	52.8% (169/320)	68.6% (116/169)	2.2 (0.1, 3.8)	0.20 (0.09, 0.36)	0.79 (0.41, 1.29)	42 (13, 53)	13.4 (5.5, 31.1)
(320 total)	FRoGGeR	**100%** (320/320)	**81.6%** (261/320)	**4.8** (3.8, 5.9)	**0.58** (0.44, 0.65)	**0.15** (0.09, 0.24)	**3** (2, 5)	**0.87** (0.44, 1.6)
adversarial	baseline	61.7% (185/300)	43.8% (81/185)	1.8 (0.6, 2.9)	0.18 (0.08, 0.29)	0.79 (0.47, 1.25)	29 (10, 55)	12.7 (5.8, 24.5)
(300 total)	FRoGGeR	**100%** (300/300)	**63.0%** (189/300)	**4.3** (3.5, 5.1)	**0.53** (0.42, 0.63)	**0.18** (0.11, 0.30)	**3** (2, 9)	**1.0** (0.49, 3.3)
overall	baseline	44.8% (385/860)	58.0% (223/385)	2.0 (0.8, 3.4)	0.19 (0.09, 0.32)	0.73 (0.44, 1.15)	50 (17, 63)	13.8 (5.8, 29.5)
(860 total)	FRoGGeR	**99.4%** (855/860)	**78.8%** (674/855)	**4.6** (3.7, 5.5)	**0.58** (0.45, 0.66)	**0.21** (0.12, 0.36)	**3** (1, 6)	**0.83** (0.39, 1.9)

TABLE I: **Simulation results.** For each of 43 objects, we try to generate 20 feasible grasps and evaluate them for both methods. In each cell, the top entry is the baseline and the bottom is FRoGGeR. The better result is bolded. ↑ and ↓ denote whether higher or lower values are better. Statistics are reported as the median and interquartile range over converged runs and all times are reported in seconds. A run *converges* if it yields a feasible grasp in under 1 minute. We find that our method almost always quickly converges to a feasible grasp while the baseline succeeds under 45% of the time. Further, over converged grasps, our method outperforms the baseline on a "shaky pickup" task in every category and by 20 percentage points overall. Our method's superior robustness is reflected in its ϵ values, which are about twice as high as the baseline. FRoGGeR's fast convergence is a result of both solving each optimization problem faster as well as requiring significantly fewer attempts before finding a feasible grasp. Note that the total time reported also includes time spent solving IK problems when sampling initial configurations q_0 (unreported, since the IK problem solved is the same for both methods).

FRoGGeR outperforms the baseline in terms of pick success by 20 percentage points and yields ϵ values that are roughly twice as high. We find that ℓ^* is a noisy but accurate predictor of grasp success.

We also found that overall, FRoGGeR was $\sim$16$\times$ faster at generating grasps than the baseline, a result of $\sim$3$\times$ faster single solve times and $\sim$15$\times$ fewer number of solves required to produce a feasible grasp. This is consistent with the observation by Wu *et al.* that their method struggles to converge to feasible solutions when q_0 is infeasible, which requires an expensive IK pre-solve [31]. In contrast, FRoGGeR retains superior speed and feasibility rate even with a coarse IK procedure, which suggests that our formulation is also robust to poor candidate grasps. In particular, only 5 runs (all on one object) timed out using our method, whereas over half of the runs timed out for the baseline.

One explanation for this gap is that the baseline force closure equality constraint's gradient vanishes at force closure, which yields a constraint geometry that is difficult to satisfy. Since we do not demand that q_0 satisfies (7d), we observed that the optimization often terminated unsuccessfully satisfying only one of the equality constraints. In contrast, our constraint (7c) has non-zero gradients even in force closure, which we conjecture is better-posed numerically, and in particular, allows FRoGGeR to converge for a larger set of candidate grasps q_0 than the baseline.

We remark that our reported baseline pick success values are significantly lower than those reported by Wu *et al.* [31], which we attribute to adding shaking to the pick trajectory. When these perturbations were smaller or nonexistent, we typically observed much higher baseline pick success rates, which supports our hypothesis that enforcing only non-robust force closure yields grasps that are brittle in practice.

One of the limitations of our method is that the ϵ metric often prefers grasps where the fingertips lie on edges or corners, since these regions are typically farther from the center of the object (roughly where we place the object frame), yielding larger moment arms. Moreover, these regions allow a grasp to direct forces in "non-robust" directions with little change, drawing solutions to them. This yields unstable grasps in practice, since small deviations in the positions of the fingertips produce large changes in the contact conditions, which commonly occurs in dynamic scenarios.

Edge-seeking behavior was the most common failure mode of both methods, which is reflected by the poor performance on many adversarial objects with less low-curvature area on which to grasp. This behavior was also observed on objects in the box/cyl category, which explains the worse performance compared to spheroids. However, failures often occurred for the baseline even when no fingers were placed on edges.

Finally, we find that the overall performance of both FRoGGeR and the baseline was highly sensitive to the sampled initial conditions. For instance, if the initial width of the fingertips was not guided by object bounding boxes, both methods suffered in terms of runtime and grasp quality, as enforcing surface constraints became harder. This motivates the use of data-driven methods in identifying candidate grasps that may be synergistic with the refinement process.

V. Conclusion and Future Work

We presented FRoGGeR, a fast method for generating robust precision grasps using the min-weight metric ℓ^*, a simple, almost-everywhere differentiable approximation of the ϵ metric. We have demonstrated that ℓ^* is empirically correlated with the ϵ metric, and validated through simulation that using ℓ^* as an optimization objective yields grasps that are more robust to dynamic perturbations than a baseline that only enforces a (non-robust) force closure constraint. Further, we have shown that both the solve time and the feasibility rate of FRoGGeR are superior to that of the baseline.

In the future, we hope to develop methods to combat edge-seeking behavior, such as adding curvature regularization terms to the objective. We also seek to generalize FRoGGeR to allow non-precision grasps and to allow for a non-fixed number of contact points. Finally, we hope to explore better object representations that do not require online mesh construction or analytical SDFs to be provided beforehand.

References

[1] A.T. Miller and P.K. Allen. Graspit! A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.

[2] C. Borst, M. Fischer, and G. Hirzinger. Grasping the Dice by Dicing the Grasp. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 4, pages 3692–3697 vol.3, 2003.

[3] D. G. Kirkpatrick, B. Mishra, and C. K. Yap. Quantitative steinitz's theorems with applications to multifingered grasping. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 341–351, New York, NY, USA, 1990. Association for Computing Machinery.

[4] C. Ferrari and J. Canny. Planning optimal grasps. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295 vol.3, 1992.

[5] Elon Rimon and Joel Burdick. *The Mechanics of Robot Grasping*. Cambridge University Press, 2019.

[6] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311, 2015.

[7] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022.

[8] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, July 2015.

[9] 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, 2022.

[10] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. *ArXiv*, abs/1703.09312, 2017.

[11] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation. *arXiv:1903.06684 [cs]*, October 2019.

[12] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2901–2910, Seoul, Korea (South), October 2019. IEEE.

[13] Ravi Balasubramanian, Ling Xu, Peter D. Brook, Joshua R. Smith, and Yoky Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *2010 IEEE International Conference on Robotics and Automation*, pages 2294–2301, 2010.

[14] Dylan Turpin, Liquan Wang, Eric Heiden, Yun-Chun Chen, Miles Macklin, Stavros Tsogkas, Sven Dickinson, and Animesh Garg. Grasp'D: Differentiable Contact-Rich Grasp Synthesis for Multi-Fingered Hands. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, page 201–221, 2022.

[15] Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and S. Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.

[16] Douglas Morrison, Peter Corke, and J. Leitner. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis ApproachClosing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. *ArXiv*, abs/1804.05172, 2018.

[17] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A Large-Scale Grasp Dataset Based on Simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227, 2021.

[18] 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. *ArXiv*, abs/2210.02697, 2022.

[19] Máximo A. Roa and Raúl Suárez. Grasp quality measures: review and performance. *Autonomous Robots*, 38:65 – 88, 2014.

[20] Umit Rusen Aktas, Chaoyi Zhao, Marek Kopicki, Ales Leonardis, and Jeremy L. Wyatt. Deep Dexterous Grasping of Novel Objects from a Single View. *Int. J. Humanoid Robotics*, 19:2250011:1–2250011:30, 2019.

[21] Lin Shao, Fábio Ferreira, Mikael Jorda, Varun Nambiar, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Oussama Khatib, and Jeannette Bohg. UniGrasp: Learning a Unified Model to Grasp with N-Fingered Robotic Hands. *ArXiv*, abs/1910.10900, 2019.

[22] Qingkai Lu, Mark Van der Merwe, Balakumar Sundaralingam, and Tucker Hermans. Multifingered Grasp Planning via Inference in Deep Neural Networks: Outperforming Sampling by Learning Differentiable Models. *IEEE Robotics and Automation Magazine*, 27(2):55–65, 2020.

[23] Zhenjia Xu, Beichun Qi, Shubham Agrawal, and Shuran Song. AdaGrasp: Learning an Adaptive Gripper-Aware Grasping Policy. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626, 2020.

[24] 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. *IEEE Robotics and Automation Letters*, 7:470–477, 2021.

[25] Min Liu, Zherong Pan, Kai Xu, and Dinesh Manocha. New Formulation of Mixed-Integer Conic Programming for Globally Optimal Grasp Planning. *IEEE Robotics and Automation Letters*, 5:4663–4670, 2019.

[26] Xiangyang Zhu and Jun Wang. Synthesis of force-closure grasps on 3-D objects based on the Q distance. *IEEE Trans. Robotics Autom.*, 19:669–679, 2003.

[27] Hongkai Dai, Anirudha Majumdar, and Russ Tedrake. Synthesis and Optimization of Force Closure Grasps via Sequential Semidefinite Programming. In *International Symposium of Robotics Research*, 2015.

[28] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep Differentiable Grasp Planner for High-DOF Grippers. *ArXiv*, abs/2002.01530, 2020.

[29] Qingkai Lu, Kautilya Chenna, Balakumar Sundaralingam, and Tucker Hermans. Planning Multi-Fingered Grasps as Probabilistic Inference in a Learned Deep Network. In *International Symposium of Robotics Research*, 2018.

[30] Mark Van der Merwe, Qingkai Lu, Balakumar Sundaralingam, Martin Matak, and Tucker Hermans. Learning Continuous 3D Reconstructions for Geometrically Aware Grasping. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11516–11522, 2019.

[31] Albert Wu, Michelle Guo, and C. Karen Liu. Learning Diverse and Physically Feasible Dexterous Grasps with Generative Model and Bilevel Optimization. *ArXiv*, abs/2207.00195, 2022.

[32] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[33] Rafaela Filippozzi, Douglas S. Gonçalves, and Luiz-Rafael Santos. First-order methods for the convex hull membership problem. *European Journal of Operational Research*, 306(1):17–33, 2023.

[34] Arne Brøndsted. *An Introduction to Convex Polytopes*. Graduate Texts in Mathematics. Springer-Verlag, New York, 1982.

[35] Brandon Amos and J. Zico Kolter. OptNet: Differentiable Optimization as a Layer in Neural Networks. In *International Conference on Machine Learning*, 2017.

[36] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems*, 2019.

[37] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019.

[38] Khaled Mamou and Faouzi Ghorbel. A simple and efficient approach for 3D mesh approximate convex decomposition. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3501–3504, 2009.

[39] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. In Alla Sheffer and Konrad Polthier, editors, *Symposium on Geometry Processing*. The Eurographics Association, 2006.

[40] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*, 2018.

[41] Ravi Balasubramanian, Ling Xu, Peter D. Brook, Joshua R. Smith, and Yoky Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *2010 IEEE International Conference on Robotics and Automation*, pages 2294–2301, 2010.

[42] Steven G. Johnson. The NLopt nonlinear-optimization package, 2011.

[43] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht Deutsche Forschungs und Versuchsanstalt fur Luft und Raumfahrt*, 1988.

APPENDIX

This appendix provides the proof of Proposition 1 as well as numerous implementation details. For the most fine-grained explanation of these details, we refer the reader to our open-source implementation at github.com/alberthli/frogger.

A. Proof of Proposition 1

By direct computation, we have

$$D_{(x,\lambda,\nu)}H = \begin{bmatrix} 0 & A_{in}^\top & A_{eq}^\top \\ \text{diag}(\lambda^*)A_{in} & \text{diag}(A_{in}x^*) & 0 \\ A_{eq} & 0 & 0 \end{bmatrix}. \quad (12)$$

For brevity, let $\Omega := D_{(x,\lambda,\nu)}H$ and unless otherwise stated, let functions be evaluated at the optimal primal/dual solution (x^*, λ^*, ν^*). Let

$$\Omega := \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad (13)$$

where for convenience we denote

$$A = 0_{(m+1)\times(m+1)}, \quad B = \begin{bmatrix} A_{in}^\top & A_{eq}^\top \end{bmatrix},$$
$$C = \begin{bmatrix} \text{diag}(\lambda^*)A_{in} \\ A_{eq} \end{bmatrix}, \quad D = \begin{bmatrix} \text{diag}(A_{in}x^*) & 0_{m\times 7} \\ 0_{7\times m} & 0_{7\times 7} \end{bmatrix}.$$

Proof. We have

$$\begin{aligned} \Omega^{-1} &= (\Omega^\top\Omega)^{-1}\Omega^\top \\ &= \begin{bmatrix} C^\top C & C^\top D \\ D^\top C & B^\top B + D^\top D \end{bmatrix}^{-1} \begin{bmatrix} 0 & C^\top \\ B^\top & D^\top \end{bmatrix}. \end{aligned} \quad (14)$$

Observe that

$$\begin{aligned} C^\top D &= \begin{bmatrix} A_{in}^\top \text{diag}(\lambda^*) & A_{eq}^\top \end{bmatrix} \begin{bmatrix} \text{diag}(A_{in}x^*) & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} A_{in}^\top \text{diag}(\lambda^*)\,\text{diag}(A_{in}x^*) & 0_{(m+1)\times 7} \end{bmatrix} \\ &= 0, \end{aligned} \quad (15)$$

where the last equality follows because

$$\text{diag}(\lambda^*)\,\text{diag}(A_{in}x^*) = \text{diag}(\lambda^* \odot (A_{in}x^*)) = 0 \quad (16)$$

by complementary slackness. Letting $P = C^\top C$ and $R = B^\top B + D^\top D$,

$$\begin{aligned} \Omega^{-1} &= \begin{bmatrix} P^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \begin{bmatrix} 0 & C^\top \\ B^\top & D^\top \end{bmatrix} \\ &= \begin{bmatrix} 0 & P^{-1}C^\top \\ R^{-1}B^\top & R^{-1}D^\top \end{bmatrix}, \end{aligned} \quad (17)$$

where we note that $P^{-1}C^\top = C^\dagger$. By substituting (17) into (5), the result immediately follows. □

We remark that if ℓ^* is locally Lipschitz with respect to the constraint matrix parameters A_{eq} and A_{in}, it is differentiable everywhere but a set of measure 0 by a theorem of Rademacher (see [44] for discussion). Further, the gradient is defined when (2) has unique primal/dual optima and in this case, $\left(D_{(x,\lambda,\nu)}H\right)^{-1}$ is defined so $\nabla\ell^*(q)$ is computable [44, Prop. 4.1]. The Lipschitz condition can always be satisfied by removing degenerate constraints, so we assume it.

B. Controller Implementation Details

This section explains the structure of the controller used for the pickup task. Recall that our controller is of the form

$$\tau = J_h^\top R_{WC} F_C^* + (I - J_h^\top (J_h^\top)^\dagger)\tau_{\text{joint}}. \quad (18)$$

We first explain computation of $\tau_{\text{joint}} \in \mathbb{R}^n$. We have that

$$\tau_{\text{joint}} = \tau_{\text{grav}} + \tau_{\text{track}}, \quad (19)$$

where the first term is a gravity compensation torque computed using partial inverse dynamics (i.e., ignoring the inertial/coriolis dynamical terms and assuming quasi-static operation) and the second term is a tracking term with independent components for the arm and the hand. We have

$$\tau_{\text{track}} = \begin{bmatrix} \tau_{\text{arm}} \\ \tau_{\text{hand}} \end{bmatrix}. \quad (20)$$

The arm tracking torques are computed as

$$\tau_{\text{arm}} = -K_{p,\text{arm}}(q_a - q_{a,\text{des}}) - K_{d,\text{arm}}(\dot{q}_a - \dot{q}_{a,\text{des}}), \quad (21)$$

with the gains set to

$$\begin{aligned} K_{p,\text{arm}} &= 500I, \\ K_{d,\text{arm}} &= \text{diag}\left(\left[1,1,1,1,0.1,0.1,0.1\right]\right). \end{aligned} \quad (22)$$

The desired values $q_{a,\text{des}}$ and $\dot{q}_{a,\text{des}}$ are computed via a differential inverse kinematics controller implemented in `Drake` that converts a desired end-effector pose trajectory specified in Cartesian space to joint angles and velocities that can be tracked. We defer those details to [45, Ch. 3.10].

The hand tracking torques are simply given by the following proportional controller:

$$\tau_{\text{hand}} = -k_{p,\text{hand}}(q_h - q_h^*), \quad (23)$$

where q_h^* is the component of the refined configuration q^* corresponding to the hand states and $k_{p,\text{hand}} = 5$. We project all of these torques via the left multiplication of $(I - J_h^\top (J_h^\top)^\dagger)$ to the null space of J_h, which ensures that applying them does not change the contact positions between the hand and object. We note that this projection does not affect the arm torques at all.

We now explain the computation of the optimal contact forces F_C^*, which is formulated as the solution to the following quadratic program:

$$\begin{aligned} \underset{F_C}{\text{minimize}} \quad & \left\| GF_C - \left(-{}^O w_{\text{des}}\right)\right\|_2^2 \\ \text{subject to} \quad & \Lambda_i F_{C,i} \leq 0,\ i = 1,\dots,n_c \\ & F_{C,i}^n \geq F_{\text{min}}^n,\ i = 1,\dots,n_c \\ & \tau_{\text{lb}} - \tau_{\text{joint,h}} \leq J_i^\top R_{WC} F_{C,i},\ i = 1,\dots,n_c \\ & J_i^\top R_{WC} F_{C,i} \leq \tau_{\text{ub}} - \tau_{\text{joint,h}},\ i = 1,\dots,n_c. \end{aligned}$$

We note that $F_{C,i} \in \mathbb{R}^3$ is the i^{th} contact force in the concatenation of all contact forces F_C.

The QP objective produces applied wrenches on the object as close as possible to counteracting some desired wrench whose force and torque components are expressed in the

object frame. The first constraint represents the pyramidal friction cone constraints (e.g., [31]). As in [31], the second enforces a minimum normal force which we specify as $F^n_{\text{min}} = 1.0$ and we additionally specify that if the object weighs under $0.01kg$, we set $F^n_{\text{min}} = 0.25$. The final two constraints enforce torque limits by ensuring the total applied joint torques from both controller terms in (18) respect the desired limits.

The desired external wrench is computed as follows:

$$^{O}w_{\text{des}} = R_{OW}\left(^{W}w_{\text{grav}} + {}^{W}w_{\text{err}}\right), \tag{24}$$

where $^{W}w_{\text{grav}}$ is the gravitational wrench expressed in the world frame. $^{W}w_{\text{err}}$ is an error wrench computed from the measured error in the object's desired pose. Suppose the desired object pose in the world frame is specified as the tuple $(p_{\text{des}}, R_{\text{des}})$ where we suppress the frame notation for brevity. We convert errors in the pose into a wrench using the formulation provided in [46]:

$$^{W}w_{\text{err}} = \begin{bmatrix} -k_{p,\text{err}}(p - p_{\text{des}}) - k_{d,\text{err}}\dot{p} \\ -k_{R,\text{err}}e_R - k_{\omega,\text{err}}\omega \end{bmatrix}, \tag{25}$$

where

$$e_R = \left(\frac{1}{2}\left(R_{\text{des}}^{\top}R - R^{\top}R_{\text{des}}\right)^{\vee}\right), \tag{26}$$

$(\cdot)^{\vee}$ is the map sending elements of the Lie algebra $so(3)$ to $\mathbb{R}^3$ (see [46]), and ω is the angular velocity of the object computed using numerical differentiation of its orientation. We choose the gains

$$\begin{aligned} k_{p,\text{err}} &= 50, \\ k_{d,\text{err}} &= 5, \\ k_{R,\text{err}} &= 50, \\ k_{\omega,\text{err}} &= 5. \end{aligned} \tag{27}$$

C. Heuristic Sampler Implementation Details

We first fix a convention for the axes of the palm of the hand. Let the x-axis be the outward palm normal and the z-axis be the corresponding axis that points in the direction of the fingers of the hand (for non-anthropomorphic hands, this choice may be arbitrary). The y-axis is then chosen consistently with the right hand rule.

The heuristic sampler consists of the following steps: (1) from the oriented bounding box of the object (which can be computed approximately very quickly using `open3d`), choose an axis with which to align the palm's y-axis up to sign and use the width of this box edge to fix an initial guess for the separation of the hand's fingers; (2) of the two remaining axes, choose one with which to align the palm's x-axis; (3) add rotational noise drawn from the von Mises distribution on the 2-sphere to randomly perturb the palm frame; (4) compute a desired location of the palm frame with respect to the object by placing it roughly $4cm$ from the surface of the object, which is approximated as its bounding box; (5) using the constraints on the palm frame, solve an inverse kinematics problem to recover q_0.

The probability of choosing a given bounding box axis for alignment is proportional to its length. For instance, if the bounding box has side lengths a, b, c, then the probability of choosing the first axis is $a/(a+b+c)$. For very short objects, we only accepted a palm frame whose x-axis approached the object from above to avoid heavy collisions with the tabletop.

D. Additional Dataset Processing Details

Out of non-excluded objects, we ranked the quality of the provided data in order of (i) Google 16k mesh, (ii) Poisson reconstruction, and (iii) TSDF file. For instance, if the 16k mesh was available, we would always prefer to use that as the initial mesh for processing before the Poisson reconstructed mesh. The excluded objects and the exact reasons for their exclusion are listed in Table II.

Object	Reason for Exclusion
`019_pitcher_base`	too big
`022_windex_bottle`	poor model: transparency
`023_wine_glass`	poor model: transparency
`024_bowl`	thin walls
`025_mug`	thin walls
`026_sponge`	deformable, too flat
`028_skillet_lid`	poor model: transparency
`029_plate`	thin walls
`030_fork`	too flat
`031_spoon`	too flat
`032_knife`	too flat
`033_spatula`	too big
`035_power_drill`	too big
`037_scissors`	too flat
`038_padlock`	no file
`039_key`	no file
`040_large_marker`	too small
`041_small_marker`	too small
`042_adjustable_wrench`	too flat
`046_plastic_bolt`	no file
`047_plastic_nut`	no file
`049_small_clamp`	too small
`050_medium_clamp`	too small
`053_mini_soccer_ball`	too big
`059_chain`	multibody
`076_timer`	lost features, not interesting

TABLE II: **YCB Exclusions.**

While flat utensils are generally too flat to be picked up by an Allegro hand from a flat table, we did replace the excluded mug with a teacup from the ShapenetSem dataset [47] with ID `23fb2a2231263e261a9ac99425d3b306` and scaled by a factor of `0.0003874877849382519`3. This cup was added to the adversarial category.

E. Other Experimental Parameters

For all experiments, we used a 4-sided pyramidal approximation of the friction cone. We enforced a minimum safety margin of $1mm$ between every collision geometry pair that was not a fingertip/object pair. For fingertip/object pairs, we allowed interpenetration up to $3mm$.

We selected a specific desired point of contact on each fingertip such that the forward kinematics were fixed. This point was located on each fingertip at an angle of 60° tilted towards the palm, measured from the very tip of each finger. Again, we remark that we can relax the assumption that we use a fixed contact point on the fingertip by parameterizing the surface of each fingertip using an SDF and applying a

similar constraint to (7e) between the fingertip geometries and the objects, with $d_j = 0$. For simplicity, we instead fix this contact point.

We supplied the following constraint tolerances to the optimization solver:

Constraint	Tolerance
joint	1e-2
surface contact	5e-4
collision	1e-3
force closure	1e-5

TABLE III: **Constraint Tolerances.**

We note that the force closure constraint refers to the robustness constraint for our method and the QP equality constraint for the baseline. In the original implementation of the method of [31] (obtained through private correspondence), the authors used a tolerance of 1e-7. In practice, we had to loosen this slightly to obtain a reasonable feasibility rate for their method.

Finally, we use the default rigid body contact model implemented in `Drake` for all of our simulations, the details of which we defer to the software documentation [37].

F. Acknowledgments

We thank Victor Dorobantu for useful discussions involving our proposed Hessian approximation. We thank Ivan D. J. Rodriguez for help with setting up experiments. We thank Wu *et al.* for their thoughtful correspondence concerning their work. We thank Philipp Wu for constructive feedback and comments. Finally, we thank all developers and maintainers of the open-source software that made this work possible (not cited in the main text but used either directly or indirectly: [48], [49], [50], [51], [52]).

REFERENCES

[44] Daniel De Wolf and Yves Smeers. Generalized derivatives of the optimal value of a linear program with respect to matrix coefficients. *European Journal of Operational Research*, 291(2):491–496, 2021.

[45] Russ Tedrake. *"Robotic Manipulation"*. 2022.

[46] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Geometric tracking control of a quadrotor uav on se(3). In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, 2010.

[47] M. Savva, A. X. Chang, and P. Hanrahan. Semantically-enriched 3d models for common-sense knowledge. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 24–31, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society.

[48] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[49] Dawson-Haggerty et al. trimesh, 2019.

[50] QuantEcon Organization. The QuantEcon package., 2023.

[51] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[52] Jia Pan, Sachin Chitta, and Dinesh Manocha. FCL: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3859–3866, 2012.