

Simulation-Driven Evolutionary Motion Parameterization for Contact-Rich Granular Scooping with a Soft Conical Robotic Hand

Yongliang Wang^{1,2}, Cristian C. Beltran-Hernandez^{*,1}, Tomoya Takahashi¹, and Masashi Hamaya¹

Abstract— Tool-based scooping is vital in robot-assisted tasks, enabling interaction with objects of varying sizes, shapes, and material states. Recent studies have shown that flexible, reconfigurable soft robotic end-effectors can adapt their shape to maintain consistent contact with container surfaces during scooping, improving efficiency compared to rigid tools. These soft tools can adjust to varying container sizes and materials without requiring complex sensing or control. However, the inherent compliance and complex deformation behavior of soft robotics introduce significant control complexity that limits practical applications. To address this challenge, this paper presents the development of a physics-based simulation model of a deformable soft conical robotic hand that captures its passive reconfiguration dynamics and enables systematic trajectory optimization for scooping tasks. We propose a novel physics-based simulation approach that accurately models the soft tool’s morphing behavior from flat sheets to adaptive conical structures, combined with an evolutionary strategy framework that automatically optimizes scooping trajectories without manual parameter tuning. We validate the optimized trajectories through both simulation and real-robot experiments. The results demonstrate strong generalization and successfully address a range of challenging tasks previously beyond the reach of existing approaches. Videos of our experiments are available online: <https://sites.google.com/view/scoopsh>

I. INTRODUCTION

Scooping is an instinctive and essential human skill, allowing us to efficiently handle materials such as fluids and granules in tasks ranging from ladling soup and collecting peas at the dining table to excavating soil at construction sites [1]–[6]. In robotics, scooping has the potential to greatly benefit daily life, particularly in food preparation and assistive feeding across homes, hospitals, and restaurants [7]. However, despite advances in robotics, stable and adaptive autonomous scooping has received limited attention. The task remains challenging due to the complex interactions between the end-effector and dynamic materials. Most robotic scooping tools are rigid and fixed in shape, lacking the passive compliance needed for smooth, precise motion across diverse containers [8]. By contrast, humans naturally adapt their choice of tools to container geometry and fill level, an adaptability that robotic systems still struggle to achieve.

Although scooping trajectories may appear similar at first glance, achieving successful execution is challenging under different environmental settings. Current research on robotic

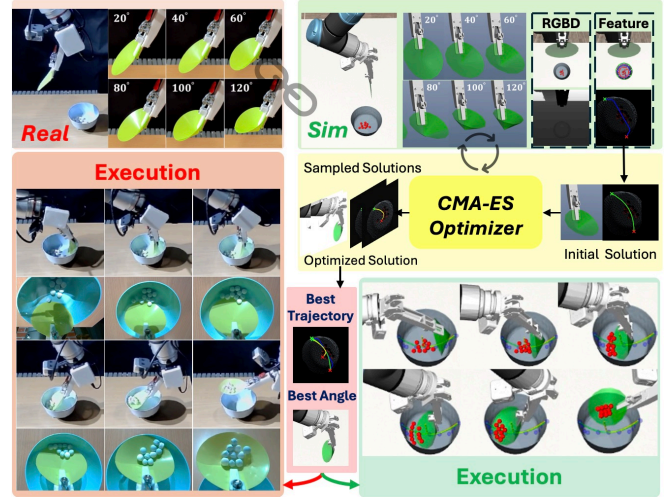


Fig. 1. **Contact-rich granular scooping system:** The framework couples real-world execution (left) with its digital twin in simulation (right). From RGB-D perception and feature extraction, container geometry is abstracted into seed trajectories. In simulation, the covariance matrix adaptation evolution strategy (CMA-ES) optimizes both trajectory and hand roll angle, producing the best solution. This optimized strategy is then directly transferred to the real soft hand without additional tuning, as demonstrated in the 10-ball scooping task.

scooping mainly addresses three aspects of environment design: tools, objects, and containers [9], [10]. **Tools:** Most studies employ rigid spoons or other fixed-shape utensils designed primarily for food handling, with only a few works exploring deformable tools tailored to specific assistive-feeding scenarios. **Objects:** Scooped materials range from solid particles to granular media and fluids, but in many trajectory-generation approaches material properties play a secondary role compared to geometry and contact. **Containers:** Experimental setups are often simplified to open boxes or regular bowls, whereas narrow, deep, or irregular containers pose substantially greater challenges and remain underexplored. In summary, completing a scooping task requires answering three questions: *What tool is used? What material is being scooped? Where is the operation performed?* In this paper, we primarily focus on the first and third aspects.

For generating scooping trajectories, many scooping methods rely on geometric representations. Some assume a standard container with predefined trajectories, later generalized [11], while others use vision to extract features and map them to handcrafted motion primitives. Though robust, these approaches are hand-engineered and inflexible. Learning-

This work was supported by the JST-Mirai Program Grant Number JPMJMI21G2, Japan.

¹ OMRON SINIC X Corporation, Tokyo, Japan.

² Department of Artificial Intelligence, Bernoulli Institute, Faculty of Science and Engineering, University of Groningen, The Netherlands

* Corresponding author:

cristian.beltran [at] sinicx.com

based methods [12] offer adaptability but demand large amounts of real or human-collected data. Limited work addresses simulation, as modeling physics, tools, and material interactions is difficult. Thus, building an end-to-end framework from simulation to trajectory generation remains an open challenge. From prior work, three key limitations can be identified: (1) Most approaches assume fixed environment settings, for example, a spoon of fixed size, a container of fixed geometry, and a fixed quantity of material, making generalization to varied environments difficult; (2) Existing tools are not designed for adaptability, even though real-world scooping tasks require handling diverse materials and containers; and (3) The lack of accurate simulation models for both tools and materials hinders efficient data collection for optimization and learning. A recent study [13] has proposed a tool with the potential to address the first two limitations, but the challenge of realistic simulation remains. Our work primarily focuses on tackling this remaining gap.

To address these limitations, particularly the challenge of building accurate models in simulation, we propose a complete framework that integrates the design of a soft deformable gripper with its simulation model in MuJoCo (see Fig. 1). This modeling approach leverages the unique properties of MuJoCo and can serve as inspiration for simulating other soft robotic tools. Furthermore, we map scooping actions from visual information and employ an evolution-based optimization system to efficiently explore the trajectory parameter space, converging to locally optimal parameters that maximize scooping success. In our experiments, we first optimize scooping parameters in simulation and then validate the resulting trajectories on a real robot, demonstrating minimal sim-to-real gap. The primary contributions of this paper are summarized as follows:

- We develop a physics-based MuJoCo simulation model of a soft deformable hand for scooping and validate it by optimizing motion parameters in simulation and transferring directly to a real robot, achieving reliable scooping with minimal sim-to-real gap.
- We propose a framework that maps visual information to primitive scooping motions. Raw trajectories are generated from features of the environment, such as container shape and size, and then optimized using an evolutionary strategy to improve scooping success.

II. RELATED WORK

A. Robotic Scooping Strategies

Recent research has explored a wide range of robotic scooping methods for handling diverse materials. Successful scooping depends on several key factors, the scooping task can be broadly divided into three aspects: **tool design and utilization**, which examines different scooping implements and their suitability for specific applications; **object properties**, which consider how material type, shape, and physical state affect scooping performance; and **motion generation strategies**, which focus on how robots plan and execute effective scooping trajectories.

1) *Robotic Scooping Tools*: A spoon is the most common tool for scooping. In robotics research, small rigid spoons are typically employed, especially in feeding-related studies. For example, [3] demonstrated a real-world robot executing trained scooping primitives in KitchenPR2 across varying bowls, spoons, and spoon poses, while [2] introduced L2D2, where humans taught robots to scoop by sketching trajectories on workspace images. A broad range of works [5], [12], [14], [15] further explored spoon-based scooping in diverse contexts, including contact-rich tasks, granular media, and deformable substances, highlighting the central role of spoons in robotic scooping research. In addition to small spoons, larger soup spoons have been investigated. [1] used a stainless steel soup spoon in food scooping tasks, while [16] showcased spoon-based scooping within a tool-design framework. Other studies extended spoon usage to granular and liquid media: [4] attached a rigid spoon to manipulate granular substances, [10] adopted a large soup spoon for water-based scooping of floating objects, and [17] equipped a robot with a soup spoon to mix soil components. Beyond spoons, other rigid tools were also explored. [18] studied water scooping with bowls and buckets, [9] introduced a tendon-driven Gripper with scoop-shaped fingertips. Soft tools for scooping also attracted growing interest, though such designs remained rare due to their mechanical complexity. [6] presented a 3D-printed utensil with origami-inspired artificial muscles that flexibly switched between gripping and scooping for different food textures. Similarly, [7] introduced a kirigami-based soft spoon capable of deforming into a bowl to wrap, contain, and release food during robot-assisted feeding. In summary, while robotic scooping has been studied with a variety of tools, most works focus on rigid spoons and devote little attention to tool design itself. Few approaches attempt to combine rigidity, softness, and deformability to achieve versatile and practical scooping across diverse tasks. Our work seeks to address these limitations.

2) *Scooped Objects*: Research on robotic scooping can also be categorized by the properties of the target objects. Many studies focus on small solid particles. For example, [1] trained on food items such as rice, beans, and chocolate balls of varying sizes, shapes, and weights, demonstrating strong generalization to unseen items. Similarly, rice was used in [11], while cereals were considered in [2], [3], [12]. Other works extended to larger discrete items like beans [3], [12], granular media such as sand [4], and even fine powders [5]. Beyond rigid particles, some research addressed complex food materials with sticky or hard-to-cut properties [6], [9]. Fluid scooping had also been investigated, including water [18] and sauces [16]. Applications extended further to soil, stones, and mining, as in autonomous excavation [19], as well as deformable plastic-like substances [14].

3) *Scooping Motion Generation*: Research on robotic scooping spans geometric, learning-based, and interactive approaches. Geometric methods construct trajectories from shape-based representations [11], while general motion generation targets robustness across varied terrains [19]. Data-

driven methods dominate: imitation learning from demonstrations enables category-level generalization [12], long-horizon food acquisition [15], motion primitives [5], and sketch-based inputs [2]. Reinforcement learning refines strategies via trial and error [18], and behavior cloning has been applied to food scooping [1]. Advanced architectures extend these directions, including VLMs for tool design [16], diffusion policies for scooping [12], and transformers for tactile perception and granular media modeling [3], [4], [14]. Our proposed physics-based simulation environment can greatly benefit existing approaches, particularly those that are data-intensive and require substantial demonstration data, by providing a realistic and cost-effective platform for trajectory generation and optimization.

B. Soft and Deformable Tools in Robotic Manipulation

Rigid tools have clear limitations when handling diverse or delicate scooping tasks, motivating researchers to investigate soft and hybrid alternatives. For example, [20] proposed a soft-rigid hybrid gripper actuated by pneumatic muscles, enabling dexterous multi-DOF grasping. Similarly, [21] introduced grippers that combined rigid fingers with switchable soft adhesives. Extending this multi-modal functionality, [22] presented a soft robotic gripper with an active suction palm and rotating grasping surfaces. Bio-inspired approaches further broadened the design space. [23] introduced a fish-mouth-inspired origami gripper for underwater grasping and scooping, demonstrating robustness in handling marine creatures. However, when faced with containers of varying shapes and sizes, even soft or hybrid tools often failed to complete scooping tasks. To address these challenges, researchers began exploring deformable tools. [24] presented a teleoperation framework combining a tactile dexterous hand with multimodal perception to achieve complex in-hand manipulations. In parallel, [8] proposed a soft underactuated hand with passive compliance and integrated sensing, and successfully demonstrated tasks such as fabric display and page turning. [13] introduced a flexible reconfigurable end-effector for powder scooping, which we adopt as the tool in this paper. These works highlight the potential of deformable tools to extend robotic manipulation capabilities beyond the limits of rigid and soft designs.

C. Simulation for Soft Robotic Systems

Simulation is fundamental for advancing soft robotics, enabling controlled experimentation, design exploration, and policy learning. Differentiable simulation has been proposed as a means to jointly optimize robot design and control through gradient-based methods [25]. To support evaluation, benchmarks such as SoftGym [26] provide standardized tasks for deformable object manipulation. Platforms like *Elastica* [27] extend simulation to continuum arms. Sim-to-real approaches have also been explored in the context of soft robotic wrists for insertion tasks [28]. In tactile sensing, finite element simulation has been applied to soft-bubble sensors for contact and force estimation [29], while

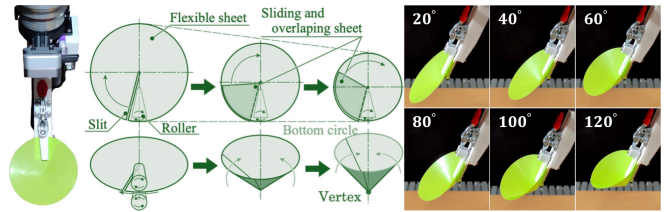


Fig. 2. The real soft-hand [13] rolling from 20° to 120° in 20° increments.

differentiable methods have been developed for efficient gel-based surface tactile simulation [30]. Origami simulators based on rigid-panel models [31], [32] capture folding well and inspire origami/kirigami robotic tools, but mainly serve for visualization. Our work instead integrates deformable tool simulation into a contact-rich scooping framework, enabling optimization and sim-to-real transfer. Together, these works establish a foundation for developing simulators tailored to soft robotic systems. However, very few studies have developed soft, deformable tools specifically for scooping. Such tools are also uncommon in real-world applications, and accurately modeling them in simulation remains particularly challenging. The absence of robust simulation models limits the use of advanced learning methods, constraining efforts to explore, validate, and scale deformable-tool-based scooping.

III. METHOD

A. System Overview

While novel soft and deformable tools have been developed, building simulation models for manipulation tasks remains challenging, alongside the difficulty of collecting large-scale real-world data. This study employs the soft-hand (SH) mechanism originally proposed in [13], shown in Fig. 2. The mechanism is modeled in MuJoCo, where an RGB-D camera captures the environment and FastSAM segments the target container [33]. Segmentation results form the basis for a raw scooping trajectory, which is subsequently optimized in simulation via CMA-ES, iteratively refining motion parameters according to interaction dynamics. The trajectories optimized in simulation are directly transferred to the physical robot for validation of real-world effectiveness.

B. Soft Hand: Mechanical Design and Simulation

1) *Mechanical Design of the Soft Hand:* The SH incorporates a flexible, conical structure that conforms to various container geometries through passive deformation, ensuring consistent contact without the need for complex force sensing or machine learning-based control strategies. Its reconfigurable mechanism further allows size adjustment, enabling efficient scooping across a wide range of container types. The SH is constructed from a circular, flexible sheet, a supporting frame, rollers, a drive shaft, and a motor. The prototype is mounted at a 45° angle relative to the manipulator to facilitate insertion into deep containers. The end-effector is offset 150mm from the motor to extend its reach. A 0.2mm thick polypropylene sheet (initial diameter 100mm) is used as the flexible tool. The roller assembly

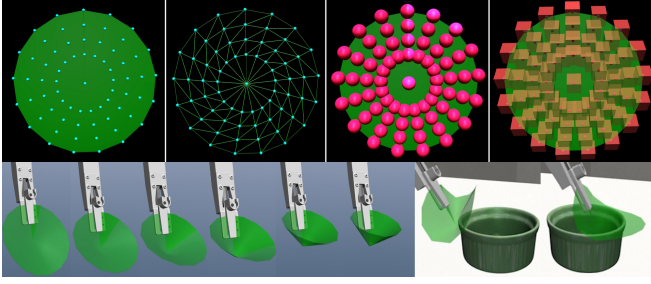


Fig. 3. The top row shows, from left to right, the Model View, Wireframe View, Mesh View, and Inertia View. The bottom row illustrates the hand rolling from 20° to 120° in 20° increments, highlighting its soft and deformable properties.

consists of a driven nitrile rubber roller and a free polyacetal roller, which together pinch the sheet to generate motion (Fig. 2).

2) *MuJoCo Simulation Model*: To replicate the behavior of the SH hand in simulation, we employ the MuJoCo physics engine, which provides physics-based contact dynamics, continuous-time physics, and support for deformable structures [34], [35]. A central challenge in this work lies in modeling a soft, funnel-like sheet that can deform realistically under roller actuation. While MuJoCo provides examples of deformable objects such as jelly, cloth, or fluid, these demonstrations are typically passive and lack mechanisms for precise, task-oriented control. In contrast to rigid-body approximations commonly employed in dexterous hand simulation, our approach combines a custom-generated geometric mesh with a staged, constraint-based control strategy. This integration enables both realistic passive compliance and controllable deformation. To the best of our knowledge, this is the first method to achieve such a controllable deformable-hand model within the MuJoCo simulator.

a) *Geometry Construction*: The top row of Fig. 3 shows how the flexible sheet was represented in simulation. The sheet is modeled as a network of connected triangles arranged in rings, using the `trimesh` tool. A section of the sheet is cut away to match the real scooping tool, which needs an opening to create a funnel shape when scooping. By adjusting the number of rings, the spacing between points, and the size of the cutout, the simulation strikes a balance between precision and stability. Seventy-two points on the sheet’s face are matched with small spheres in the simulation, and these spheres are locked to their corresponding sheet points with constraints (equality *weld*) that prevent movement between them. This approach ensures that simulated contact, like the tool touching container surfaces, happens exactly at the edge of the sheet, while the sheet itself can bend and flex realistically.

b) *Dynamics and Parameterization*: The SH sheet is modeled as a mass–spring–damping system, where each mesh node obeys

$$m_i \ddot{\mathbf{x}}_i = \mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}} - c \dot{\mathbf{x}}_i \quad (1)$$

where m_i is the mass associated with node i , \mathbf{x}_i is the position vector of node i , $\ddot{\mathbf{x}}_i$ and $\dot{\mathbf{x}}_i$ are its acceleration and velocity, respectively, $\mathbf{f}_i^{\text{int}}$ denotes the internal elastic force on node i arising from stretching and bending, $\mathbf{f}_i^{\text{ext}}$ is the external force applied to node i (e.g., gravity or contact with the container), and c is the Rayleigh damping coefficient. The elastic energy of the sheet is given by

$$E = \frac{1}{2} \sum_{(i,j)} k_s (\|\mathbf{x}_i - \mathbf{x}_j\| - l_{ij}^0)^2 + \frac{1}{2} \sum_{(p,q)} k_b (\theta_{pq} - \theta_{pq}^0)^2 \quad (2)$$

where \mathbf{x}_i and \mathbf{x}_j are the positions of nodes i and j , k_s and k_b denote the stretching and bending stiffness coefficients, l_{ij}^0 is the rest length of edge (i, j) , and θ_{pq}^0 is the rest dihedral angle at hinge (p, q) . The material parameters are selected to approximate the properties of the nitrile sheet. The surface density ρ_s is chosen so that the simulated total mass matches the physical prototype. The stiffness coefficients (k_s, k_b) are tuned to balance container conformity with sufficient structural integrity. The damping coefficient c is introduced to suppress spurious oscillations during simulation. In practice, seventy-two peripheral vertices are exposed and coupled to colocated spherical geoms via equality *weld* constraints, which act as proxies for contact and actuation while preserving the compliance of the underlying deformable model.

c) *Control*: To enable controllable deformation, the sheet boundary is discretized into vertices, each associated with a small proxy geom. As illustrated in the bottom row of Fig. 3, based on observations from the physical prototype, we establish mapping rules that determine which vertex pairs should be linked to reproduce a given funnel angle. This mapping captures the correspondence between geometric connections and the resulting global deformation. For robustness, the activation of these constraints is scheduled in stages. This staged strategy ensures smooth transitions, prevents over-constraining, and yields reliable control of the deformable sheet while preserving its compliance.

C. Problem Formulation

The objective is to enable a soft, deformable robotic hand to perform scooping across diverse containers, adapting its shape and size much like humans intuitively select tools. To address this, we construct consistent simulation environments and formulate the task as an optimization problem in trajectory generation under interaction dynamics, where both the motion and the hand’s rolling angle are decision variables. All environments are generated from real-world setups into a simulation, making our framework akin to a digital twin system. This allows the optimized trajectories to transfer to real robots with minimal sim-to-real gap.

1) *Observation*: The input is an RGB-D image \mathcal{I} , from which we extract container keypoints $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 0, \dots, M\}$ where M is the number of keypoints and construct a seed path that connects anchor points (e.g., `topN` \rightarrow center \rightarrow `topS`), where top denotes the container’s top rim. Illustrated in Fig. 4.

2) *Scooping Action Parameterization*: We aim to optimize the trajectory and the rolling angle of the SH jointly.

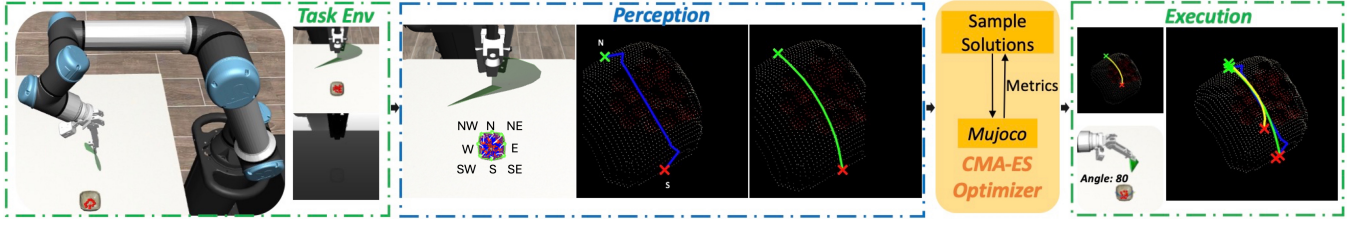


Fig. 4. Overall framework of our system: from RGB-D perception and ring-compass abstraction, through seed trajectory generation and parameterization, to trajectory optimization with CMA-ES, enabling transfer of optimized scooping strategies from simulation to the real robot.

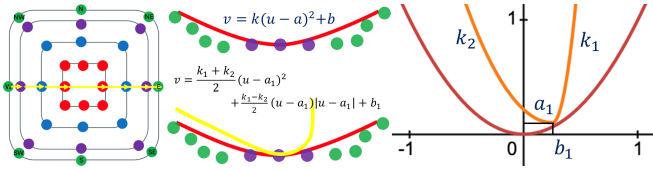


Fig. 5. From left to right: anchor skeleton, parabolic fit in uv -plane, and piecewise-parabola trajectory.

a) Scoop Trajectory Analysis: Fig. 5 shows that we fit a local motion plane using singular value decomposition (SVD), which is equivalent to principal component analysis (PCA) on the point set. This defines a right-handed coordinate frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$ with basis vectors spanning the plane and a normal vector \mathbf{n} . Any 3D point can then be projected into local coordinates (u, v) .

The three anchors (τ_{opN} , center, τ_{opS}) are projected into the local plane and used to fit a parabola

$$v = Au^2 + Bu + C \quad (3)$$

We convert it to vertex form

$$v = k(u - a)^2 + b, \quad k = A, \quad a = -\frac{B}{2A}, \quad b = C - \frac{B^2}{4A} \quad (4)$$

and denote the seed by parameters (k, a, b) . The endpoints $(u_{\text{start}}, v_{\text{start}})$ and $(u_{\text{end}}, v_{\text{end}})$ are also recorded.

To allow more flexibility, the optimized trajectory is modeled as a continuous, piecewise parabola with potentially different curvatures on either side of the vertex:

$$v(u; k_1, k_2, a_1, b_1) = \frac{k_1 + k_2}{2}(u - a_1)^2 + \frac{k_1 - k_2}{2}(u - a_1)|u - a_1| + b_1 \quad (5)$$

This ensures C^1 continuity while allowing asymmetric curvature. $\bar{\mathbf{p}}$ denotes the reference point in world coordinates. The world-space curve is then obtained by mapping back via

$$\mathbf{x}(u) = \bar{\mathbf{p}} + u\mathbf{e}_1 + v(u)\mathbf{e}_2 \quad (6)$$

Since our scooping task always takes place in the x - z or y - z plane, constraining the z -position is meaningful. Therefore, the curve is trimmed so that its endpoints align with the seed anchors, and it is then resampled into n evenly spaced waypoints along its arc length, which provides a consistent parameterization for optimization and execution.

b) Soft Hand Roll Angle and Rotation of End-effector:

From the container observation, we obtain an initial estimate of the SH roll angle $angle_{\text{initial}}$ related to the size of the container. This value serves as an initial angle, while the optimizer explores variations by adding a continuous offset $\Delta angle$. The optimizer evaluates candidate roll angles by interacting with the MuJoCo simulation environment, enabling identification of the most effective roll angle for the scooping task. Since the roll angle directly affects the orientation of the sheet funnel's edge relative to the end effector (Fig. 3), to maintain a consistent sheet orientation relative to the ground, the optimized roll angle is remapped into an appropriate rotation range for the end effector. This ensures that both insertion into and withdrawal from the container remain feasible.

For consecutive waypoints $(\mathbf{x}_i, \mathbf{x}_{i+1})$, we compute the motion direction to define a tilt angle $tiltdeg_i$. Pitch of the end-effector is then obtained as a function of $tiltdeg_i$ and the discrete SH roll angle $angle_i$:

$$pitch_i = m(tiltdeg_i; angle_i) \quad (7)$$

where $m(\cdot; \cdot)$ is the piecewise-linear band mapping.

Fig. 3 shows that six discrete roll angles are defined for the SH. Specifically, the roll angle is discretized into buckets:

$$\mathcal{A} = \{20^\circ, 40^\circ, 60^\circ, 80^\circ, 100^\circ, 120^\circ\} \quad (8)$$

$$\tilde{a}_i = \min(\max(angle_{\text{initial}} + \Delta angle, 20^\circ), 120^\circ) \quad (9)$$

$$angle_i = \operatorname{argmin}_{a \in \mathcal{A}} |a - \tilde{a}_i| \quad (10)$$

where $angle_{\text{initial}}$ denotes the prior roll angle and $\Delta angle$ is the decision variable determined by the optimizer.

3) Evaluation Metrics: As the optimizer interacts with the simulation at each trial, we define an objective function based on three key aspects: the volume of material successfully scooped, the amount of overflow spilled outside the container, and whether unintended collisions occur between the hand and the container. These criteria jointly capture both efficiency and feasibility of the scooping motion.

D. Scooping Trajectory Generation

1) Visual Perception: To initialize scooping, the system first detects the target container and estimates its geometry. Using FastSAM [33], object masks are extracted from RGB

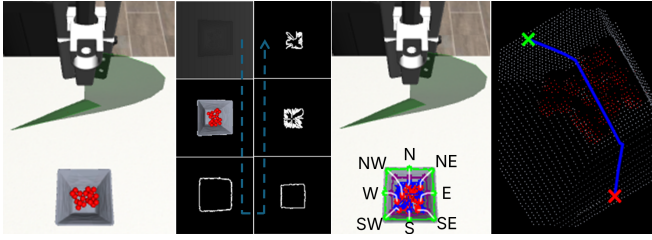


Fig. 6. Perception pipeline: RGB-D input to ring-compass representation for trajectory planning.

images and fused with depth data to reconstruct a partial height field in world coordinates. This field is vertically bounded by observed heights, and several iso-height surfaces are defined to approximate the container’s functional regions: base, walls, and rim. At each iso-height level, we extract the closed contour formed by the mask–height band intersection. From each contour, eight equi-angular “compass” points are sampled with respect to the centroid, yielding a structured set of control points $p_{r,d}$, where r indexes the ring level and d the directional label. Connecting points with identical directions across levels produces eight vertical profiles that approximate the container’s inner surface from base to rim (see Fig. 6). This ring–compass representation offers a compact geometric abstraction of the container, capturing both global radial symmetry and local variations in rim curvature and wall inclination. These features act as geometric priors for trajectory planning, guiding insertion and deformation control of the hand.

2) *Generation of Initial Trajectories*: Given the ring-compass skeleton $\{p_{r,d}\}$ extracted from perception, we generate a parameterized waypoint sequence to guide the end-effector from free space to insertion and lift-out. As shown in Fig. 7, a compass direction $d^* \in \{N, NE, E, SE, S, SW, W, NW\}$ is first selected based on task heuristics such as approach side, occlusion, or arm reachability. Along this direction, the vertical profile

$$\mathcal{P}_{d^*} = [p_{\text{bottom},d^*}, p_{\text{mid1},d^*}, p_{\text{mid2},d^*}, p_{\text{top},d^*}]$$

provides four seed points from base to rim, which are connected into a piecewise linear path. Each waypoint is assigned an end-effector orientation, initialized as described in Sec. III-C. The initial angle of the deformable hand is set according to the container size. Following Sec. III-C, the scooping trajectory is formulated as an optimization problem.

3) *Trajectory Optimization via CMA-ES*: CMA-ES is a robust evolutionary strategy for real-valued optimization in continuous domains, well-suited to stochastic, nonlinear, and nonconvex functions [36]. It maintains a population of candidate solutions and updates them through selection, recombination, and mutation. As described in Sec. III-C, our initial trajectory is parameterized, requiring optimization of four parameters to shape the curve and one additional parameter to set the hand roll.

a) *Parameterization*: We optimize five variables: four curve parameters and the hand roll angle. The optimization

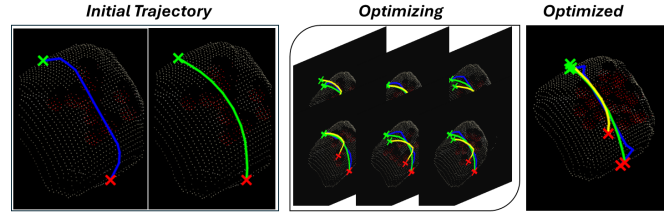


Fig. 7. Generation and refinement of scooping trajectories. From left to right: initial trajectory seeded from the ring-compass skeleton, intermediate optimization process, and final optimized trajectory.

vector is

$$\theta = (k_1, k_2, a_1, b_1, \Delta\text{angle}) \quad (11)$$

Using the single-parabola seed (k, a, b) , we define data-driven bounds:

$$k_1 \in \text{sign}(k) \cdot (2|k|, 20|k|], \quad (12)$$

$$k_2 \in \text{sign}(k) \cdot (0.8|k|, 1.2|k|], \quad (13)$$

$$a_1 \in [a_{\text{lo}}, a_{\text{hi}}], \quad (14)$$

$$b_1 \in (b, b + \Delta b], \quad \Delta b \approx 0.01, \quad (15)$$

$$\Delta\text{angle} \in [-\Delta_{\text{max}}, \Delta_{\text{max}}], \quad \Delta_{\text{max}} \approx 40^\circ \quad (16)$$

where a_{lo} and a_{hi} correspond to the two waypoints before and after the center waypoint of the initial trajectory.

b) *Objective Function*: Each candidate trajectory is simulated in MuJoCo with the SH deformable model at a chosen discrete opening θ . During rollout, task metrics are extracted: (i) *scooped* — number of particles lifted above a container-specific Z threshold; (ii) *overflow* — particles remaining below threshold; (iii) *collision* — indicator if any robot body contacts the container or table.

The scalar loss is defined as

$$\mathcal{L}(\vartheta, \theta) = -w_1 \text{scooped} + w_2 \text{overflow} + w_3 \text{collision} + \lambda \mathbf{I} \quad (17)$$

where \mathbf{I} penalizes trajectories that fail to reach all waypoints.

CMA-ES samples and rebuilds the trajectory, and evaluates \mathcal{L} in simulation. The optimizer returns the best parameters and trajectory, together with diagnostics such as per-term metrics, consistently improving scooping yield while reducing overflow and collisions.

IV. EXPERIMENTS

A. Experimental Setups

As illustrated in Fig. 1, the left shows the real robot and the right the corresponding simulation environment, designed for repeatable scooping and camera observation. Fig. 8 shows that experiments use granular media in containers of varying shapes and sizes (bowls and plates, circular, square, and irregular). The objective is to assess the fidelity of the hand simulation: whether it reproduces the deformation and interaction of the real SH, and whether optimized trajectories transfer with minimal sim-to-real gap. We conducted two types of experiments. First, ablation studies examined which parameters should be optimized. Second, we evaluated the success rate of trajectories optimized in simulation and transferred them to the real robot.

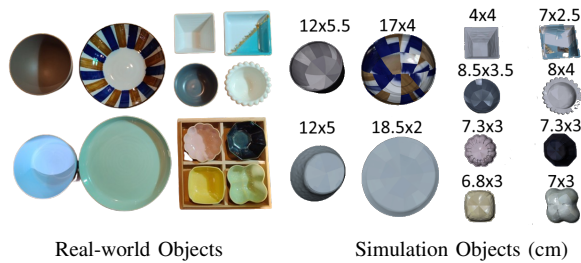


Fig. 8. Containers used in experiments: 12 objects in three sizes (large, medium, small), including regular and irregular shapes. (Width x Height)

TABLE I

SCOOPING RESULTS UNDER DIFFERENT ACTION PARAMETERIZATIONS ACROSS 3 CONTAINER SIZES IN THE 10 BALLS TASK.

Action Parameterization	Parameters	Scoped		
		Big	Medium	Small
Initialization	$(k, a, b, \Delta angle)$	0	0	0
No k_1	$(k_2, a_1, b_1, \Delta angle)$	2	1	1
No k_2	$(k_1, a_1, b_1, \Delta angle)$	6	5	5
No a_1	$(k_1, k_2, b_1, \Delta angle)$	6	7	8
No b_1	$(k_1, k_2, a_1, \Delta angle)$	8	9	9
No $\Delta angle$	(k_1, k_2, a_1, b_1)	8	8	6
Full optimization	$(k_1, k_2, a_1, b_1, \Delta angle)$	10	10	9

B. Simulation Experiments

1) *Ablation Experiments*: Scooping performance is influenced by multiple trajectory model parameters. To assess their relative contributions, we performed ablation experiments in simulation with a standard bowl (Table I). In each variant, one parameter was excluded from optimization: k_1 , k_2 , a_1 , b_1 , or $\Delta angle$. The results highlight that curvature parameters (k_1 , k_2) are the most critical: removing either sharply reduces performance, with success counts dropping to nearly half of the full optimization setting. By contrast, excluding vertex parameters (a_1 , b_1) causes smaller but still noticeable degradation. Excluding the rotation term $\Delta angle$ also reduces scooping efficiency, particularly for small containers. Overall, these results confirm that robust scooping requires jointly tuning all parameters curvature, vertex, and rotation underscoring the importance of full optimization for consistent performance across container sizes.

C. Real-World Experiments

We optimized trajectories exclusively in simulation on the 10-ball task, where the best solution achieved a perfect outcome, scooping all ten balls. This single optimized trajectory was then directly transferred to the real SH system without any further tuning. Figures 1 and 9 illustrate the experimental setup and action sequence, while Table II reports the quantitative results. In real experiments, the robot scooped on average 7.9 balls in the 10-ball task, 14.9 in the 20-ball task, and 48.25 g in the rice task. Although real-world performance showed some variability due to sensing and actuation noise, one trial in the 10-ball task exactly matched the simulation result, successfully scooping all ten balls. Importantly, despite being optimized only on the 10-ball task, the same trajectory generalized effectively to the 20-ball

and rice tasks, demonstrating strong zero-shot transfer across object types and container conditions. These findings confirm that the digital twin framework enables robust sim-to-real transfer: the deformable hand model faithfully reproduces the key behaviors of the physical system, and trajectories optimized in simulation generalize to real execution with minimal adaptation. Supplementary videos provide additional evidence, illustrating both the scooping process and the high fidelity of the transferred strategies.

V. CONCLUSION

This paper presents a comprehensive framework that integrates an existing soft, deformable conical robotic hand with a physics-based MuJoCo simulation model to enable robust, adaptive scooping across varied containers. The hand's passive compliance enables it to conform to various container geometries, while the physics-based simulation accurately reproduces its deformation and interaction with granular materials. Visual perception is utilized to generate initial scooping trajectories that are then optimized within the simulation environment using the CMA-ES evolutionary strategy. The resulting optimized trajectories transfer effectively to the physical robot with minimal sim-to-real gap.

This approach addresses the limitations of rigid and hand-engineered methods by leveraging soft tools and simulation-driven trajectory optimization, revealing significant potential for advancing robot-assisted manipulation. Future work will explore extending the framework to more complex materials and incorporating learning-based trajectory optimization for broader applications.

REFERENCES

- [1] Y.-L. Tai, Y. C. Chiu, Y.-W. Chao, and Y.-T. Chen, "Scone: A food scooping robot learning framework with active perception," in *Conference on Robot Learning*, pp. 849–865, PMLR, 2023.
- [2] S. A. Mehta, H. Nemlekar, H. Sumant, and D. P. Losey, "L2d2: Robot learning from 2d drawings," *arXiv preprint arXiv:2505.12072*, 2025.
- [3] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Learning compositional models of robot skills for task and motion planning," *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 866–894, 2021.
- [4] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Conference on Robot Learning*, pp. 239–248, PMLR, 2017.
- [5] S. Ruan, W. Liu, X. Wang, X. Meng, and G. S. Chirikjian, "Primp: Probabilistically-informed motion primitives for efficient affordance learning from demonstration," *IEEE Transactions on Robotics*, vol. 40, pp. 2868–2887, 2024.
- [6] Y. R. Song and Y. Luo, "Soda—soft origami dynamic utensil for assisted feeding," in *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*, pp. 1–8, IEEE, 2025.
- [7] M. Keely, B. Franco, C. Grothoff, R. K. Jenamani, T. Bhattacharjee, D. P. Losey, and H. Nemlekar, "Kiri-spoon: A kirigami utensil for robot-assisted feeding," *arXiv preprint arXiv:2501.01323*, 2025.
- [8] C. Zhao, C. Jiang, L. Luo, S. Yuan, Q. Chen, and H. Yu, "Learning thin deformable object manipulation with a multi-sensory integrated soft hand," *IEEE Transactions on Robotics*, 2025.
- [9] L. Franco, E. Turco, V. Bo, M. Pozzi, M. Malvezzi, D. Prattichizzo, and G. Salvietti, "The double-scoop gripper: A tendon-driven soft-rigid end-effector for food handling exploiting constraints in narrow spaces," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4170–4176, IEEE, 2024.
- [10] X. Liu, Y. Zhou, F. Weigend, S. Sonawani, S. Ikemoto, and H. B. Amor, "Diff-control: A stateful diffusion-based policy for imitation learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7453–7460, IEEE, 2024.

TABLE II
SCOOPING RESULTS OVER 10 TRIALS FOR THREE TASKS IN A REAL ROBOT.

Task	Trial 1	2	3	4	5	6	7	8	9	10	Average
10-balls	10	7	8	8	5	9	5	9	9	9	7.9
20-balls	16	16	16	13	15	15	14	14	14	16	14.9
50g-rice (g)	48.23	47.99	48.12	48.27	47.82	47.92	49.06	48.13	48.60	48.38	48.25

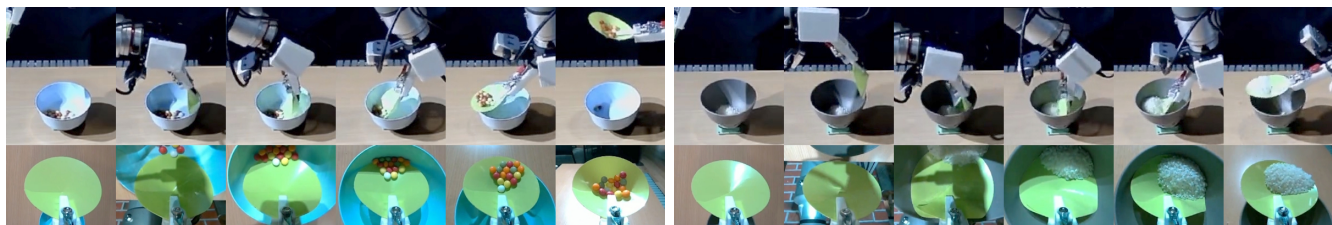


Fig. 9. Action sequence of the SH executing the optimized trajectory transferred from simulation to reality. The figure illustrates key phases of the scooping process, including rolling SH, insertion, and lifting. The left is the 20-ball task, and the right is the 50g-rice task.

- [11] D. Das, A. Patankar, N. Chakraborty, C. Ramakrishnan, and I. Ramakrishnan, "Screw geometry meets bandits: Incremental acquisition of demonstrations to generate manipulation plans," *arXiv preprint arXiv:2410.18275*, 2024.
- [12] Q. Yang, M. C. Welle, D. Kragic, and O. Andersson, "S²-diffusion: Generalizing from instance-level to category-level skills in robot manipulation," *arXiv preprint arXiv:2502.09389*, 2025.
- [13] T. Takahashi, C. C. Beltran-Hernandez, Y. Kuroda, K. Tanaka, M. Hamaya, and Y. Ushiku, "Scu-hand: Soft conical universal robotic hand for scooping granular media from containers of various sizes," *arXiv preprint arXiv:2505.04162*, 2025.
- [14] Y. Kageyama, M. Hamaya, K. Tanaka, A. Hashimoto, and H. Saito, "Learning scooping deformable plastic objects using tactile sensors," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pp. 4020–4025, IEEE, 2024.
- [15] A. Bhaskar, R. Liu, V. D. Sharma, G. Shi, and P. Tokekar, "Lava: Long-horizon visual action based food acquisition," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8929–8935, IEEE, 2024.
- [16] C. Lin, H. Yuan, Y. Wang, X. Qiu, T.-H. Wang, M. Guo, B. Wang, Y. Narang, D. Fox, and C. Gan, "Robotsmith: Generative robotic tool design for acquisition of complex manipulation skills," *arXiv preprint arXiv:2506.14763*, 2025.
- [17] N. Moorman, A. Singh, M. Natarajan, E. Hedlund-Botti, M. Schrum, C. Yang, L. Seelam, M. C. Gombolay, and N. Gopalan, "Investigating strategies enabling novice users to teach plannable hierarchical tasks to robots," *The International Journal of Robotics Research*, p. 02783649241301075, 2024.
- [18] Y. Niu, S. Jin, Z. Zhang, J. Zhu, D. Zhao, and L. Zhang, "Goats: Goal sampling adaptation for scooping with curriculum reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1023–1030, IEEE, 2023.
- [19] N. Franceschini, P. Thangeda, M. Ornik, and K. Hauser, "Autonomous excavation of challenging terrain using oscillatory primitives and adaptive impedance control," *arXiv preprint arXiv:2409.18273*, 2024.
- [20] W. Zhu, C. Lu, Q. Zheng, Z. Fang, H. Che, K. Tang, M. Zhu, S. Liu, and Z. Wang, "A soft-rigid hybrid gripper with lateral compliance and dexterous in-hand manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 1, pp. 104–115, 2022.
- [21] S. A. Mehta, Y. Kim, J. Hoegerman, M. D. Bartlett, and D. P. Losey, "Riso: Combining rigid grippers with soft switchable adhesives," in *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 1–8, IEEE, 2023.
- [22] X. Wang, L. Horrigan, J. Pinskiier, G. Shi, V. Viswanathan, L. Liow, T. Bandyopadhyay, J. J. Chung, and D. Howard, "Dexgrip: Multimodal soft gripper with dexterous grasping and in-hand manipulation capacity," in *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*, pp. 1–6, IEEE, 2025.
- [23] H. Guo, J. Huang, I. Zhang, B. Liang, X. Ma, Y. Liu, and J. Zhou, "Fish mouth inspired origami gripper for robust multi-type underwater grasping," *arXiv preprint arXiv:2503.11049*, 2025.
- [24] J. Huang, K. Chen, J. Zhou, X. Lin, P. Abbeel, Q. Dou, and Y. Liu, "Dih-tele: Dexterous in-hand teleoperation framework for learning multiobjects manipulation with tactile sensing," *IEEE/ASME Transactions on Mechatronics*, 2025.
- [25] M. Bächer, E. Knoop, and C. Schumacher, "Design and control of soft robots using differentiable simulation," *Current Robotics Reports*, vol. 2, no. 2, pp. 211–221, 2021.
- [26] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, pp. 432–448, PMLR, 2021.
- [27] N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3389–3396, 2021.
- [28] Y. Fuchioka, C. C. Beltran-Hernandez, H. Nguyen, and M. Hamaya, "Robotic object insertion with a soft wrist through sim-to-real privileged training," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9159–9166, IEEE, 2024.
- [29] J.-C. Peng, S. Yao, and K. Hauser, "3d force and contact estimation for a soft-bubble visuotactile sensor using fem," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5666–5672, IEEE, 2024.
- [30] J. Xu, S. Kim, T. Chen, A. R. Garcia, P. Agrawal, W. Matusik, and S. Sueda, "Efficient tactile simulation with differentiability for robotic manipulation," in *Conference on Robot Learning*, pp. 1488–1498, PMLR, 2023.
- [31] Y. Zhu, M. Schenk, and E. T. Filipov, "A review on origami simulations: from kinematics, to mechanics, toward multiphysics," *Applied Mechanics Reviews*, vol. 74, no. 3, p. 030801, 2022.
- [32] T. Tachi, "Simulation of rigid origami," *Origami*, vol. 4, no. 08, pp. 175–187, 2009.
- [33] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.
- [34] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033, IEEE, 2012.
- [35] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs, et al., "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.
- [36] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE international conference on evolutionary computation*, pp. 312–317, IEEE, 1996.