

Quasi-static Soft Fixture Analysis of Rigid and Deformable Objects

Yifei Dong and Florian T. Pokorny

Abstract—We present a sampling-based approach to reasoning about the caging-based manipulation of rigid and a simplified class of deformable 3D objects subject to energy constraints. Towards this end, we propose the notion of soft fixtures extending earlier work on energy-bounded caging to include a broader set of energy function constraints, such as gravitational and elastic potential energy of 3D deformable objects. Previous methods focused on establishing provably correct algorithms to compute lower bounds or analytically exact estimates of escape energy for a very restricted class of known objects with low-dimensional configuration spaces, such as planar polygons. We instead propose a practical sampling-based approach that is applicable in higher-dimensional configuration spaces, but only produces a sequence of upper-bound estimates that, however, appear to converge rapidly to actual escape energy. We present 8 simulation experiments demonstrating the applicability of our approach to various complex quasi-static manipulation scenarios. Quantitative results indicate the effectiveness of our approach in providing upper-bound estimates for escape energy in quasi-static manipulation scenarios. Two real-world experiments also show that the computed normalized escape energy estimates appear to correlate strongly with the probability of escape of an object under randomized pose perturbation¹.

I. INTRODUCTION

Fixturing, which is the task of restraining the poses of a rigid object under noise and external perturbation, forms a key part of robotic manipulation and serves as an essential step towards achieving robust and dexterous manipulation [1]. Classical grasping approaches, such as form and force closure [2], utilize point contacts to attain complete control over the pose of a grasped object, but are restricted to rigid objects and are susceptible to grasp instability issues arising from noise in perception or external disturbances. Furthermore, these approaches only capture a small subset of strategies humans employ to restrain the poses of an object - for example, when we utilize gravity in combination with support surfaces from tools and our hands to interact with soft, deformable or fragile objects such as clothes or food items. In this work, we take the first steps towards an extension of fixturing to “soft fixtures”, where “soft” indicates both that we consider not just “hard” collision constraints but also “soft constraints” formulated in terms of maximal thresholds of a suitable energy function and that the framework is applicable to soft objects. Our approach generalizes the idea of energy-bounded cages of rigid bodies

- where a rigid object is constrained to a bounded path component by collisions and a simple potential energy function. Unlike past works [3], [4] on energy-bounded cages, which employed volumetric configuration space (C-space) representations and were limited to the gravitational potential energy of simple 2D rigid objects, we show that a sampling-based approach to the study of soft fixtures can be applied with more general energy functions and even challenging settings involving both deformable and rigid bodies. The proposed “soft fixture”-based approach can, in our view, be seen as a starting point for an additional avenue towards robotic manipulation that, like classical form and force closure, is fully physics-based and explainable, but opens the possibility to describe and analyze novel interaction scenarios involving both rigid and deformable objects.

The primary contributions of this work² are: (1) An extension of caging-based manipulation to soft fixtures capable of capturing a novel class of rigid and deformable object manipulation scenarios involving potential energy constraints; (2) The introduction of sampling-based algorithms for the estimation of upper bounds of escape energy of soft fixtures that appear to converge rapidly; (3) A first analysis of escape energy in practical quasi-static manipulation scenarios involving diverse rigid and deformable objects, supported by physical experiments that assess the correlation between normalized escape energy and grasp stability.

II. RELATED WORK

Caging Grasps The original notion of caging, introduced by Kuperberg [24], addresses the problem of preventing a polygon from escaping arbitrarily far away using a set of fixed point obstacles, ensuring that the polygon’s path-component in the collision-free C-space remains bounded. Building on this idea, Rimon et al. [25], [26] extended caging to robotic grasping and developed a theory for caging-based grasps for planar objects. In [27], [28], methods for finding two-finger cage formations of planar polygons and 3D polyhedra by means of contact-space search were determined. Additionally, Rodriguez et al. [29] demonstrated that caging grasps can be regarded as a viable means of achieving immobilizing grasps via finger closure. Subsequent works, such as those by [30], [31], have applied caging grasps to certain classes of 3D rigid and deformable objects, analyzing topological and geometric features. However, these methods are limited to objects with specific features, such as holes or double forks. Makapunyo et al. [32] pioneered

The authors are with the Division of Robotics, Perception and Learning, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden, {yifeid, fpokorny}@kth.se. Funded by the European Commission under the Horizon Europe Framework Programme project SoftEnable, grant number 101070600, <http://softenable.eu/>

¹For additional material, visit <https://sites.google.com/view/softfixture/>.

²This work is an extension of our workshop contribution [5].



Fig. 1: Soft fixture scenarios: (1) radishes held together with a rubber band, (2) putting on a mask, (3) hooking a frozen fish with a fishing hook, (4) catching a starfish with a bowl, (5) scooping a fish with a shovel, (6) grasping a chain of transfusion bags, (7) manipulating a Rubik cube with two grippers, (8) snap lock mechanisms. The examples include textures and meshes by [6]–[23].

the concept of partial caging, characterizing it as a finger arrangement allowing limited escape movements. Satoshi et al. [33] further explored partial caging concerning circular objects within a planar hand. Varava et al. [34] defined a clearance-based partial cage quality metric and employed deep neural networks for efficient quality estimation from images of 2D partial cages. Gravity caging is proposed in [35] to derive placing areas where objects could be partially caged and held under gravitational force. Mahler et al. [3], [4] introduced energy-bounded caging of 2D rigid objects, extending the caging paradigm with gravitational potential energy constraints and utilizing a cell-based decomposition of C-space in combination with persistent homology. Advancing along this trajectory, we show that it is possible to incorporate generalized potential energies and complex 3D rigid and deformable objects with 6-18 dimensional C-spaces using the sampling-based approach we propose here.

Sampling-based Planning for Manipulation Sampling-based motion planning methods encompass methods like PRMs [36] and RRTs [37], alongside asymptotically optimal approaches such as RRT* and PRM* [38]. Notably, Batch Informed Trees (BIT*) [39] strikes a balance between graph-search and sampling-based techniques, prioritizing search by potential solution quality akin to A*, while maintaining asymptotic optimality like RRT* [40]. Building upon BIT*, our method optimizes a customized cost function that accelerates finding a solution minimizing the maximal value along a feasible path. A similar problem was defined in [41] and solved with a bottleneck tree. Manipulation problems are often approached using sampling-based motion planners [42]. For instance, Schmitt et al. [43] address manipulation tasks as a multi-modal planning problem in high-dimensional C-space, utilizing a PRM*-based approach for offline manipulation planning. Shome et al. [44] enhance computational scalability for dual-arm systems without explicitly constructing a roadmap in the composite C-space of both arms. CBiRRT [45], [46] explores constraint manifolds in manipulation and automates motion primitive generation by exploring lower-dimensional C-space manifolds [47].

III. PRELIMINARIES AND DEFINITIONS

In the following, we introduce a notion of potential energy for rigid and deformable objects that we shall study in this work. We then consider the problem of estimating the minimum energy required for a bounded 3D object $\mathcal{O} \subset \mathbb{R}^3$ to escape from its initial configuration subject to

fixed environmental constraints, including support surfaces of finite size, such as a table-top or static robot grippers and in-hand tools.

Articulated Objects with Potential Energy Three types of objects are considered in our problem setting: (1) 3D rigid objects; (2) 3D deformable objects like fish that yield when subjected to external forces and absorb or release energy through a spring-like mechanism; (3) linear deformable objects, such as ropes and elastic bands. With the exception of elastic bands, we model all types as links or point masses interconnected by compliant revolute joints. We denote by $\mathcal{X} \subset \text{SE}(3) \times \mathbb{R}^{n_j}$ the C-space of a 3D object with n_l links connected by n_j revolute joints, each angle constrained to $[a_i, b_i] \subset [-\pi, \pi]$ ($1 \leq i \leq n_j$), with a special rigid object case where $n_j = 0$. An element $\mathbf{x} \in \mathcal{X}$ with components $\mathbf{x} = (\mathbf{r}, \mathbf{q}, \boldsymbol{\alpha})$, encompasses the position \mathbf{r} and orientation (a unit quaternion) \mathbf{q} of the center of mass of the base link of the object and a vector of revolute joint angles $\boldsymbol{\alpha}$. The collision space \mathcal{X}_{obs} indicates a set of configurations for which \mathcal{O} penetrates at least one of the rigid obstacles in the workspace. The free C-space is given by $\mathcal{X}_{\text{free}} = \text{cl}(\mathcal{X} \setminus \mathcal{X}_{\text{obs}})$, where $\text{cl}(\cdot)$ denotes the closure of a set. We assume that torsion springs are affixed at each of the revolute joints, resulting in non-zero elastic potential energy for joint configurations other than those with $\boldsymbol{\alpha} = \mathbf{0}$. We consider the following energy function $E : \mathcal{X} \rightarrow \mathbb{R} \cup \{\infty\}$, combining gravitational and elastic potential energy [48]:

$$E(\mathbf{x}) = \sum_{i=1}^{n_l} m_i g z_i + \sum_{j=1}^{n_j} \frac{1}{2} k_j \alpha_j^2, \quad \mathbf{x} \in \mathcal{X}_{\text{free}}. \quad (1)$$

Here, g denotes the gravitational acceleration constant, and z_i represents the height of the center of mass of the i 'th link in the world coordinates (gravity in $-z$ direction). α_j , k_j denote the angle and the stiffness coefficient of the j 'th joint ($k_j = 0$ is an allowable special case). Configurations in collision space \mathcal{X}_{obs} have infinite energy by convention.

For a closed loop linear elastic deformable object, such as a rubber band, we select n_b equally spaced key points along it and assume these points are loosely interconnected by non-compressible tension springs. Its C-space is approximated by $\mathcal{X} \subset \mathbb{R}^{3n_b}$ with an element $\mathbf{x} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{n_b})$. The energy function is given by

$$E_b(\mathbf{x}) = \sum_{i=1}^{n_b} \frac{1}{2} k_i (\max\{\|\mathbf{r}_i - \mathbf{r}_{i-1}\| - L, 0\})^2, \quad (2)$$

where $\mathbf{x} \in \mathcal{X}_{\text{free}}$, L is the rest length of the springs, \mathbf{r}_0 and \mathbf{r}_{n_b} indicate identical key points along the enclosed loop.

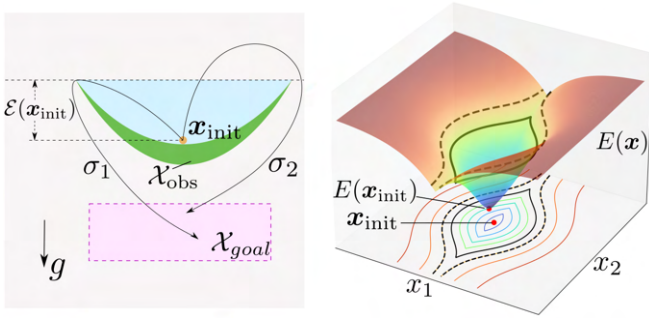


Fig. 2: Left: escape energy $\mathcal{E}(\mathbf{x}_{\text{init}})$ of a point-mass object subject to gravity inside a bowl (green) in a 2D workspace. Right: a synthetic energy function $E(\mathbf{x})$ on a 2D C-space with its contour map to illustrate the concept of a soft fixture. We display the relative increase in energy as relative iso-lines below the energy surface. Note that the two subplots illustrate two different scenarios.

For this scenario, we do not model gravitational effects and mass is not explicitly modeled in the energy function.

Caging and Soft Fixtures We introduce the concept of a soft fixture as an extension of caging to include both rigid and deformable objects subject to potential energy as defined above. Let us first recall the classical notion of a cage.

Definition 1: An object in configuration $\mathbf{x}_{\text{init}} \in \mathcal{X}_{\text{free}}$ is *caged* if \mathbf{x}_{init} lies in a bounded path component of $\mathcal{X}_{\text{free}}$.

A cage thus restricts the movement of an object from escaping arbitrarily far away from its initial configuration.

Definition 2: An object at an initial configuration $\mathbf{x}_{\text{init}} \in \mathcal{X}_{\text{free}}$ possesses potential energy $E(\mathbf{x}_{\text{init}})$. Its *escape energy* $\mathcal{E}(\mathbf{x}_{\text{init}})$ refers to the supremum of e such that the path component $\mathcal{PC}_{\mathbf{x}_{\text{init}}}(\mathcal{X}_e(\mathbf{x}_{\text{init}}))$ of a sublevel set $\mathcal{X}_e(\mathbf{x}_{\text{init}})$ containing \mathbf{x}_{init} is bounded:

$$\mathcal{E}(\mathbf{x}_{\text{init}}) = \sup \{e \geq 0 : \mathcal{PC}_{\mathbf{x}_{\text{init}}}(\mathcal{X}_e(\mathbf{x}_{\text{init}})) \text{ is bounded}\}, \quad (3)$$

where

$$\mathcal{X}_e(\mathbf{x}_{\text{init}}) = \{\mathbf{x} \in \mathcal{X}_{\text{free}} : E(\mathbf{x}) \leq E(\mathbf{x}_{\text{init}}) + e\}. \quad (4)$$

The object is deemed to be in a *soft fixture* if $\mathcal{E}(\mathbf{x}_{\text{init}}) > 0$. Note that $\mathcal{E}(\mathbf{x}_{\text{init}})$ is defined only if $\mathcal{PC}_{\mathbf{x}_{\text{init}}}(\mathcal{X}_0(\mathbf{x}_{\text{init}}))$ is bounded.

The notion of a soft fixture provides a generalized form of energy-bounded cages introduced for 2D polygonal objects [4]. A simple real-world soft fixture example is that of an object that lies inside a deep bowl under gravity (Fig. 2, left). Escape energy $\mathcal{E}(\mathbf{x}_{\text{init}})$ subject to potential energy then intuitively corresponds to determining the minimal lift of the object required to remove it fully from the bowl (along path σ_1). Note that the light blue region corresponds to the largest bounded path-component containing \mathbf{x}_{init} , confined by an energy sublevel set. The right of Fig. 2 further illustrates the concept with a synthetic 2D C-space and energy function. Consider an \mathbf{x}_{init} corresponding to the minimum of the displayed energy function. As the energy threshold increases, the sublevel set $\mathcal{X}_{e_1}(\mathbf{x}_{\text{init}})$ (denoted by solid lines) is still

bounded, but for $\mathcal{X}_{e_2}(\mathbf{x}_{\text{init}})$ (dashed lines), the sublevel set becomes unbounded along the x_2 axis direction. e_1 , the supremum over thresholds corresponding to bounded such path-components then corresponds to the escape energy in this scenario. With respect to our definition of the energy function above, also note that a soft fixture with infinite escape energy corresponds to the classical concept of a cage as in Def. 1.

Problem Statement The central challenge addressed in this paper is the following extension of energy-bounded caging of 2D rigid bodies to 3D rigid or deformable objects:

Given a 3D rigid or deformable object in a soft fixture configuration \mathbf{x}_{init} , which is modeled as above and subject to potential energy as in Eq. 4 or 2, determine the escape energy $\mathcal{E}(\mathbf{x}_{\text{init}})$ under fixed environmental constraints.

Approximating Escape Energy with Escape Paths

Definition 3: For any feasible path $\sigma : [0, 1] \rightarrow \mathcal{X}_{\text{free}}$, we denote the *energy cost* along σ by

$$C(\sigma) = -E(\sigma(0)) + \max_{t \in [0, 1]} E(\sigma(t)). \quad (5)$$

For a given \mathbf{x}_{init} , and feasible path σ leading arbitrarily far away from \mathbf{x}_{init} , $C(\sigma)$ establishes an upper bound of $\mathcal{E}(\mathbf{x}_{\text{init}})$. We are thus inspired to attempt to find “best escape paths” that lead infinitely far away from \mathbf{x}_{init} and encounter the lowest possible maximal potential energy values along the way, corresponding to tight upper bounds of escape energy. If no such escape path can be determined (for example due to classical caging by obstacles), no finite upper bound estimate can be established. Since we are working with bounded obstacles and due to the form of our energy function, we furthermore simplify the problem to consider only escape paths from \mathbf{x}_{init} to some bounded goal region $\mathcal{X}_{\text{goal}}$ that is sufficiently far away from obstacles and has lower energy values than \mathbf{x}_{init} . Once any path reaches this goal region, we can extend the path to lead infinitely far away from the initial configuration without further increasing the escape energy. This ensures that the supremum of energy values over the infinite extended path is the same as that over the initial bounded path from \mathbf{x}_{init} to $\mathcal{X}_{\text{goal}}$ and makes the problem more amenable to standard motion planning algorithms expecting a bounded goal region. The key question we address next is how to determine candidate escape paths that minimize $C(\sigma)$ to provide suitable estimates for $\mathcal{E}(\mathbf{x}_{\text{init}})$. The left of Fig. 2 provides an illustration in 2D under gravity with a sub-optimal escape path σ_2 and another escape path σ_1 that provides a tight upper bound estimate.

IV. ESCAPE ENERGY ESTIMATION

Iterative Search Our first set of algorithms for approximating soft fixture escape energy focuses on searching for feasible escape paths from \mathbf{x}_{init} to $\mathcal{X}_{\text{goal}}$ constrained within sublevel sets $\mathcal{X}_e(\mathbf{x}_{\text{init}})$ with increasingly lower threshold e . Specifically, we run an RRT planner constrained to $\mathcal{X}_e(\mathbf{x}_{\text{init}})$, initiating at \mathbf{x}_{init} , with the goal to reach $\mathcal{X}_{\text{goal}}$, positioned sufficiently distant from obstacles. In our experimental setup, we designate a cubic region

Algorithm 1: Conservative Search

```
1 Initialize,  $\hat{e} \leftarrow \infty, t \leftarrow 0$ 
2 Set stopping criteria,  $t_{\max}$ 
3 repeat
4    $\Delta t, \sigma \leftarrow \text{RRTSEARCH}(\mathcal{X}_{\hat{e}}(\mathbf{x}_{\text{init}}))$ 
5   if  $\sigma$  exists then
6      $\hat{e} \leftarrow C(\sigma)$ 
7    $t \leftarrow t + \Delta t$ 
8 until  $t \geq t_{\max}$  or  $\hat{e} \approx 0$ ;
9 Return  $\hat{e}, \sigma$ 
```

$\mathcal{H} \subset \mathbb{R}^3$, situated lower than the workspace obstacles in the direction of gravity. A path is considered to have adequately escaped from an initial configuration \mathbf{x}_{init} if it achieves a configuration $\mathbf{x}_{\text{goal}} = (\mathbf{r}_g, \mathbf{q}_g, \boldsymbol{\alpha}_g)$, where $\mathbf{r}_g \in \mathcal{H}$, \mathbf{q}_g is arbitrary, and $\boldsymbol{\alpha}_g = 0$ (in the general formulation). This approach ensures a goal space consisting of configurations with reduced total potential energy compared to \mathbf{x}_{init} .

To approximate the upper bounds of escape energy, we first employ a simple conservative search (Algo. 1) as a baseline. In each iteration, we execute an RRT planner within a finite time budget to find a trajectory entirely within $\mathcal{X}_{\hat{e}}(\mathbf{x}_{\text{init}})$. This search initially sets $\hat{e} \gg 0$, begins at \mathbf{x}_{init} and connects to $\mathcal{X}_{\text{goal}}$ (Line 4). If a feasible path σ is found, it determines an upper bound $\hat{e} = C(\sigma)$ for escape energy, and the search recommences within the reduced sublevel set $\mathcal{X}_{\hat{e}}(\mathbf{x}_{\text{init}})$ (Lines 5-6). Each discovered path successively refines this upper bound in practice. If no path is located, the search repeats within the previous region. The process concludes when $\hat{e} \approx 0$, or upon reaching a preset maximum iteration or time limit for the motion planner, thus defining an upper bound \hat{e} for escape energy $\mathcal{E}(\mathbf{x}_{\text{init}})$.

We have also implemented a binary search approach (see Algo. 2), conceptually similar to the bisection method described in [34]. This binary search adheres to a divide-and-conquer principle to locate the escape energy, maintaining both its lower bound e_- and upper bounds e_+ , while progressively narrowing the range. In each iteration, we execute RRT within the sublevel set $\mathcal{X}_{\hat{e}}(\mathbf{x}_{\text{init}})$ with \hat{e} given by the average of the current upper and lower bounds (Lines 4-5). If a path is found, the upper bound for escape energy is updated (Lines 6-7) similar to Algo. 1. Otherwise, the lower bound is updated with \hat{e} (Lines 11-12). It is worth noting that in practice, e_- does not necessarily represent a strict lower bound for escape energy. Given the finite time budget allocated to RRT, failure to find a path may only imply a high probability that the range does not contain the target. Consequently, if energy cost $C(\sigma)$ lower than the lower bound e_- is discovered, we reset the lower bound to 0 (Lines 8-9).

Energy-biased Optimality Search Building upon the intuitive algorithms previously discussed, we introduce a configuration-cost function to prioritize states with lower potential energy within the framework of a standard RRT*

Algorithm 2: Binary Search

```
1 Initialize,  $\hat{e} \leftarrow \infty, t \leftarrow 0, e_- \leftarrow 0, e_+ \leftarrow \infty$ 
2 Set stopping criteria,  $t_{\max}, e_{\epsilon}$ 
3 repeat
4    $\hat{e} = (e_+ + e_-)/2$ 
5    $\Delta t, \sigma \leftarrow \text{RRTSEARCH}(\mathcal{X}_{\hat{e}}(\mathbf{x}_{\text{init}}))$ 
6   if  $\sigma$  exists then
7      $e_+ \leftarrow C(\sigma)$ 
8     if  $C(\sigma) < e_-$  then
9        $e_- \leftarrow 0$ 
10  else
11     $e_- \leftarrow \hat{e}$ 
12   $t \leftarrow t + \Delta t$ 
13 until  $t \geq t_{\max}$  or  $e_+ \approx 0$  or  $e_+ - e_- < e_{\epsilon}$ ;
14 Return  $e_+, \sigma$ 
```

algorithm³. Let $\mathcal{X}_v \subset \mathcal{X}_{\text{free}}$ denote the set of vertices in the RRT* graph. We define a cost-to-come function $c : \mathcal{X}_v \rightarrow \mathbb{R}_{\geq 0}$, mapping each vertex in \mathcal{X}_v to the cost of the unique path from the root of the tree to that vertex. This function is employed during vertex expansion and rewiring, following the standard sample-and-steer approach typical in RRT*-like methods. The relationship between a child vertex \mathbf{x}_{ch} and its parent vertex \mathbf{x}_{pa} is captured by the recurrence relation of the cost functions,

$$c(\mathbf{x}_{\text{ch}}) = \max\{c(\mathbf{x}_{\text{pa}}), E(\mathbf{x}_{\text{ch}}) - E(\mathbf{x}_{\text{init}})\}, \quad (6)$$

with $c(\mathbf{x}_{\text{init}}) = 0$. Taking vertex expansion as an example, we define the cost-to-come $c(\mathbf{x}_{\text{new}})$ of a new sample $\mathbf{x}_{\text{new}} \in \mathcal{X}_{\text{free}}$ as $c(\mathbf{x}_{\text{new}}) = \max\{c(\mathbf{x}_{\text{near}}), E(\mathbf{x}_{\text{new}}) - E(\mathbf{x}_{\text{init}})\}$, if it connects to a nearby vertex $\mathbf{x}_{\text{near}} \in \mathcal{X}_v$ already in the graph. This expression for the cost of \mathbf{x}_{new} represents the energy cost from \mathbf{x}_{init} to \mathbf{x}_{new} via \mathbf{x}_{near} . The cost function steers exploration to expand vertices and rewires edges based on their potential energy-informed cost-to-come. This approach aims to prune branches of the search tree that are unlikely to lead to an escape path with minimal energy. Other asymptotically optimal algorithms besides RRT*, such as BIT*, Informed RRT*, AIT*, etc., share the cost function formulation. We specifically employ BIT*, combining the efficiency characteristics of graph search techniques and anytime scalability advantages of sampling based motion planning. A cost-to-go heuristic is not defined for BIT* with this particular cost function.

V. EVALUATION

This section provides an evaluation of the algorithms proposed for quasi-static soft fixture analysis. The examination includes a variety of manipulation primitives depicted in Fig. 1 and illustrated in the supplementary video. Two scenarios are further illustrated in Fig. 4.

In these manually created simulation sequences, objects are subject to gravitational forces and their mobility is

³More details are provided at <https://sites.google.com/view/softfixture/>.

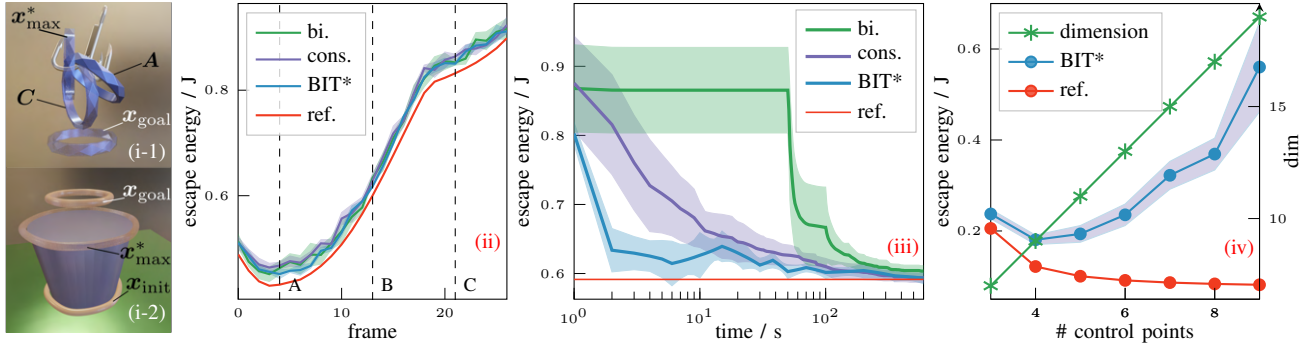


Fig. 3: Quasi-static analysis of escape energy in two scenarios with 27 and 1 frame, respectively. (i): a ring dangling on a hook and swinging back and forth (i-1, A, C), and an elastic band trapped by a conical frustum (i-2). (ii): estimated escape energy of the three algorithms w.r.t a reference as the ring swings downward (dashed lines A, C corresponding to configurations in (i-1)). (iii): convergence of the algorithms at frame B of (ii). (iv): the escape energy (blue line) of an elastic band in (i-2) with varying numbers of modeling control points and reference escape energy (red) corresponding to ring configuration \mathbf{x}_{\max}^* in (i-2).

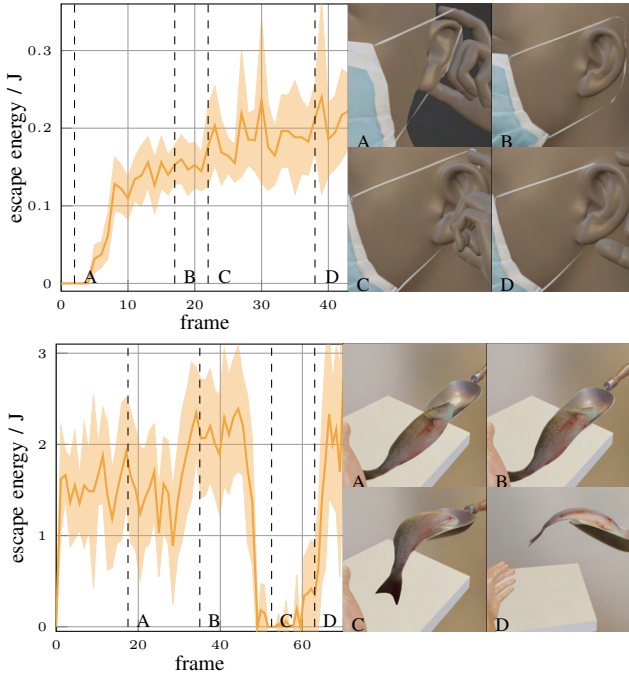


Fig. 4: Quasi-static soft fixture analysis of the mask-wearing and fish-scooping scenarios in 44 and 71 frames, respectively. We run 5-12 independent runs of a 5-minute execution of the BIT* planner for each frame, with 4 illustrated frames.

constrained by various obstacles, such as grippers, hands, in-hand tools, etc. The configurations of objects are recorded for each frame of the simulation during this process. Subsequently, we analyze the configuration of each frame using our proposed methods in a quasi-static manner and approximate escape energy per frame to evaluate our methods. Note that, while kinetic energy is disregarded in the per-frame soft fixture analysis here, it contributes to the simulation of the manipulation primitive sequence.

We utilized the Open Motion Planning Library (OMPL) [49] to implement the proposed motion planners. Pybullet [50] served as the platform for collision detection, while forward simulation of bodies under gravity was jointly

conducted in Pybullet and Blender. All experiments are performed on an Intel Core i9-12900H processor with 14 cores and speeds up to 5.0GHz.

Quantitative Analysis We conducted an analysis to verify both the accuracy and efficiency of our proposed approaches, with a focus on a rigid ring scenario (Fig. 3, i-1). Intuitively, we believe an optimal escape path of the ring attains maximal energy at configurations where it is just barely lifted above the fish hook to escape with the ring in a vertical plane, as \mathbf{x}_{\max}^* illustrates. We display the resulting escape energy of this (assumed optimal) reference escape path via \mathbf{x}_{\max}^* for each frame of the simulation as a reference curve in part (ii) of the figure. As can be seen in (ii), BIT* demonstrated a similar level of performance to the iterative methods across the frames of the simulation, all of which were closely aligned with the empirical reference escape energy. We observed that BIT* converged faster than the other two methods within a 10-min runtime (Fig. 3, iii).

Convergence in binary search is contingent on the location of e^* within the interval of the initial energy bounds $[e_-, e_+]$. When e^* is closer to e_- , the algorithm converges more rapidly, as the target resides within the search range in the initial iterations. A comparative analysis of different motion planning algorithms for frame B in the hook scenario in Fig. 3 under our energy cost function was conducted in Table I. The anytime planners, BIT* and AIT*, exhibited superior performance in this specific problem relative to RRT*-based planners. Moreover, we assessed the accuracy concerning state-space dimensionality within a simple scenario: an elastic hair tie escaping from a human wrist (modeled by a conical frustum) as depicted in Fig. 3 (i-2). We model the hair tie as closed elastic loop with an increasing number of equally spaced vertices connected by linear segments and an energy function as in Eq. 2. With the addition of more vertices, we observe an increasing deviation between estimated and expected reference escape energy (corresponding to \mathbf{x}_{\max}^*) in Fig. 3 iv). This illustrates the expected curse of dimensionality problem affecting sampling-based motion planners in the context of our escape energy estimation problem.

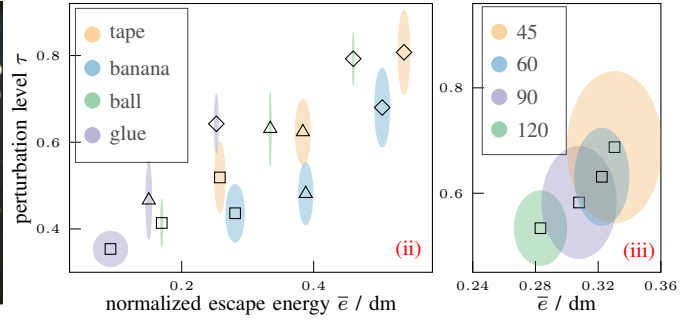
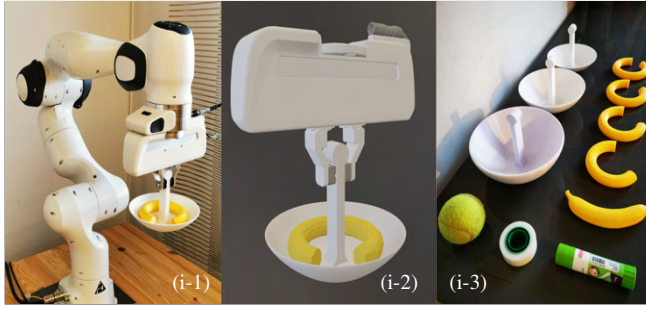


Fig. 5: Physical experiment setup and results. (i-1): the arm’s initial configuration. (i-2): the replication of (i-1) in simulation. (i-3): 3 bowl-like caging tools of various depths, objects (tennis ball, tape, glue stick, banana) for (ii) and toroidal objects with a section cut off, forming angles of $[45^\circ, 60^\circ, 90^\circ, 120^\circ]$ for (iii). The objects are respectively placed inside one of the bowls as in (i-1). (ii): observed perturbation level at which objects escape from bowls and estimated normalized escape energy obtained using BIT* with simple objects. Square, triangle and diamond markers denote small, medium and large bowls, respectively. Eight repetitions are performed in simulation and physical experiments for each data point. Ellipses showcase standard deviation. (iii): perturbation levels for the toroidal objects.

TABLE I: Deviation from reference escape energy

planners	BIT*	bi.	cons.	AIT*	inf. RRT*	RRT*
error mean	4.02	4.53	5.54	10.33	15.43	18.71
error std.	0.61	1.01	0.62	0.90	2.53	1.97

Qualitative Evaluation In the mask example (Fig. 4), we focus on a motion primitive where a person manipulates the mask strap by securing it around the ear with two fingers. The mask strap around the ear is represented as a spring-loaded open loop, consisting of 8 control points—two fixed at the filter’s boundary, with the remaining control points being freely movable (comprising 18 degrees of freedom). We consider elasticity but disregard the mass of the mask components, and assume the wearer is holding the central part of the mask with another hand until the elastic potential energy of the strap is zero. Consequently, we model a goal state for soft fixture analysis corresponding to a configuration in $\mathcal{X}_{\text{free}}$ with zero elastic potential energy from which the mask can be removed arbitrarily far from the face without increasing elastic potential energy. Note that the illustrated hand is not considered an obstacle during the soft fixture analysis, as we are primarily examining the interaction between the ear and the mask loop. Our initial estimated escape energy is zero, as the loop has not yet engaged the auricle (subfigure A) and a mere contraction offers a viable escape path. However, as the loop progresses behind the ear (B), begins to contract (C), and snugly fits around the back of the ear (D), an increased escape energy is estimated, indicating the stability of this configuration.

In a fish scooping scenario, we approximate an articulated fish model with $n_l = 10, n_j = 9$ (Fig. 4) and consider primarily its bending deformation, following the natural curvature of the spine. Initially, a hand, tabletop and shovel collectively act as obstacles, constraining the fish’s movement during scooping, and thus the fish needs to lift and increase the gravitational component of its potential energy slightly to escape (A, B). As it settles into the shovel, the shovel lifts from the desk (C). The fish’s elastic potential energy and the shovel’s mouth opening facilitate an effortless leftward and downward slide, implying escape energy close to 0 J. By rotating outward (D), the shovel incline angle

is changed, resulting in an increase in estimated escape energy corresponding to the intuition that the fish requires an addition of energy to escape this configuration.

Physical Experiment We conducted physical evaluation experiments using a Franka Emika Panda robot arm [51]. Bowl-like tools with three varying depths were attached to the end-effector of the arm (Fig. 5, i-1) and we positioned four different objects of various shapes within the bowls to create soft fixture candidates. Increasing levels of perturbation motions were applied to simulate increasing external disturbance to the initial configuration of the arm. The tool, influenced by this perturbation, rotates and translates randomly, eventually releasing the object (the object falls out of the bowl) upon reaching a perturbation level τ that was recorded. A positive correlation between perturbation level and mass-normalized escape energy $\bar{e} = \hat{e}/(mg)$, was observed. Here, m denotes the mass of the object. Normalized escape energy is computed by replicating the real-world geometry and poses in simulation (Fig. 5, i-2) and employing our upper bound escape energy estimation approach.

In a second experiment, we examined toroidal objects with cutouts (Fig. 5, i) to explore the empirical escape perturbation levels in relation to normalized escape energy, specifically with narrow C-space tunnels. Tori with a smaller cutout demonstrated resistance to higher perturbation levels and exhibited slower convergence in escape energy estimation, as is to be expected as this behavior stems from the necessity for denser sampling to find a feasible path through the narrow C-space tunnel (Fig. 5, iii). The observed exponential trend underscores that \bar{e} only partially characterizes the escape probability in complex scenarios exhibiting low-clearance escape paths since escape energy for all these cut-out objects intuitively is identical and corresponds to a small lift and sideways escape motion.

VI. CONCLUSION

We presented steps towards extending caging-based manipulation to novel scenarios involving both rigid and deformable objects. The energy-function based sublevel

set analysis approach investigated here can, we believe, provide a fruitful long-term research direction, with many interesting challenges, such as the creation of algorithms for soft fixture synthesis (rather than analysis) and the inclusion of dynamic rather than quasi-static settings in the analysis.

REFERENCES

- [1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.
- [2] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [3] J. Mahler, F. T. Pokorny, Z. McCarthy, A. F. van der Stappen, and K. Goldberg, "Energy-bounded caging: Formal definition and 2-d energy lower bound algorithm based on weighted alpha shapes," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 508–515, 2016.
- [4] J. Mahler, F. T. Pokorny, S. Niyaz, and K. Goldberg, "Synthesis of energy-bounded planar caging grasps using persistent homology," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 908–918, 2018.
- [5] Y. Dong and F. T. Pokorny, "Soft fixtures: Towards practical caging-based manipulation of rigid and deformable objects," <https://deformable-workshop.github.io/icra2023/spotlight/24-Dong-spotlight.pdf>, 2023 ICRA Workshop on Representing and Manipulating Deformable Objects.
- [6] anastasiaremezova, "Milk delivery background mesh," <https://sketchfab.com/3d-models/milk-delivery-b41e30c422d943409291050b77364568>, CC BY 4.0.
- [7] JennyB, "Tileable stylized wooden floor texture," <https://sketchfab.com/3d-models/tileable-stylized-wooden-floor-texture-167a3fd14d65456dacfacd9ee793e6f>, CC BY 4.0.
- [8] kevinruiz, "Left hand model," <https://sketchfab.com/hand-a43c9c9059b24e2aa4af08d1a76f0916>, CC BY 4.0.
- [9] jimbogies, "Kitchen cabinets," <https://sketchfab.com/3d-models/kitchen-cabinets-5926fa64fcaa498c818682990db3df27>, CC BY 4.0.
- [10] D. Tudor, "a stylized wok kitchen," <https://sketchfab.com/3d-models/3d-tudor-creates-a-stylized-wok-kitchen-dc4b86cd1049496ab32b9c6c5c1ceea5>, CC BY 4.0.
- [11] L. . Leoskateman, "Disposable medical masks," <https://sketchfab.com/3d-models/disposable-medical-masks-657b3a5d3e184a01b1708a3a24c27b44>, CC BY 4.0.
- [12] Yimit, "Fish model," <https://sketchfab.com/3d-models/fish-c4328d6c1ae445198abc1a3d4c1e2570>, CC BY 4.0.
- [13] mkultraviolence, "Realistic korean girl's head model," <https://sketchfab.com/3d-models/realistic-korean-girl-head-lowpoly-basemesh-c0cd098fe6bb47fa9117cd8b48151703>, CC BY 4.0.
- [14] Stan, "Antique kitchen model," <https://sketchfab.com/3d-models/antique-kitchen-017dbf073761426896e715053cf2d883>, CC BY 4.0.
- [15] kevjumba, "Right hand model," <https://sketchfab.com/3d-models/kevin-hand-865a1a6cddb44afeba5ce8d738fdd300>, CC BY 4.0.
- [16] L. Moffitt, "Cozy blood bag," <https://sketchfab.com/3d-models/cozy-blood-bag-0302621453a54ae7a31ec56f22d7eb44>, CC BY 4.0.
- [17] shuvalov.di, "Fish shovel model," <https://sketchfab.com/3d-models/grain-scoop-9316aaafac1d40679b309f953fabb53f>, CC BY-NC-SA 4.0.
- [18] M. I. of Art, "Mina'i ware bowl, 1200-1299 ce," <https://sketchfab.com/3d-models/minai-ware-bowl-1200-1299-ce-304aa606135a47ec9291b464bba6b357>, CC BY-NC-SA 4.0.
- [19] ChrisRE, "Charité university hospital - operating room," <https://sketchfab.com/3d-models/charite-university-hospital-operating-room-9ec46c4d615a4581a235eebf162f574>, CC BY-NC 4.0.
- [20] Drakery, "Starfish model," <https://sketchfab.com/3d-models/starfish-df589166a4b44602a24b2226161a63df>, sketchfab license.
- [21] HQ3DMOD, "Fishing hook model," <https://sketchfab.com/3d-models/fishing-hook-triple-b9b0ccaaa49e4115a39a285c526985fc>, sketchfab license.
- [22] 3dlowpoly, "Radish model," <https://www.cgtrader.com/3d-models/food/vegetable/radish-39058f4e-7d61-4f92-a5f4-fc231e12fa45>, cgtrader license.
- [23] "Robotiq - 3-finger adaptive robot gripper," <https://robotiq.com/products/3-finger-adaptive-robot-gripper>, accessed on July 17, 2023.
- [24] W. Kuperberg, "Problems on polytopes and convex sets," in *DIMACS Workshop on polytopes*, 1990, pp. 584–589.
- [25] E. Rimon and A. Blake, "Caging 2d bodies by 1-parameter two-fingered gripping systems," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1996, pp. 1458–1464.
- [26] —, "Caging planar bodies by one-parameter two-fingered gripping systems," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 299–318, 1999.
- [27] T. F. Allen, J. W. Burdick, and E. Rimon, "Two-finger caging of polygonal objects using contact space search," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1164–1179, 2015.
- [28] T. F. Allen, E. Rimon, and J. W. Burdick, "Two-finger caging of 3d polyhedra using contact space search," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2005–2012.
- [29] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 886–900, 2012.
- [30] F. T. Pokorny, J. A. Stork, and D. Kragic, "Grasping objects with holes: A topological approach," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 1100–1107.
- [31] A. Varava, D. Kragic, and F. T. Pokorny, "Caging grasps of rigid and partially deformable 3-d objects with double fork and neck features," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1479–1497, 2016.
- [32] T. Makapunyo, T. Phoka, P. Pipattanasomporn, N. Niparnan, and A. Sudsang, "Measurement framework of partial cage quality," in *2012 IEEE international conference on robotics and biomimetics (ROBIO)*. IEEE, 2012, pp. 1812–1816.
- [33] S. Makita and K. Nagata, "Evaluation of finger configuration for partial caging," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4897–4903.
- [34] M. C. Welle, A. Varava, J. Mahler, K. Goldberg, D. Kragic, and F. T. Pokorny, "Partial caging: a clearance-based definition, datasets, and deep learning," *Autonomous Robots*, vol. 45, pp. 647–664, 2021.
- [35] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1021–1043, 2012.
- [36] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [37] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [38] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [39] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3067–3074.
- [40] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Batch informed trees (bit*): Informed asymptotically optimal anytime search," *The International Journal of Robotics Research*, vol. 39, no. 5, pp. 543–567, 2020.
- [41] K. Solovey and D. Halperin, "Efficient sampling-based bottleneck pathfinding over cost maps," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2003–2009.
- [42] J. D. Gammell and M. P. Strub, "Asymptotically optimal sampling-based motion planning methods," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 295–318, 2021.
- [43] P. S. Schmitt, W. Neubauer, W. Feiten, K. M. Wurm, G. V. Wichert, and W. Burgard, "Optimal, sampling-based manipulation planning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3426–3432.
- [44] R. Shome and K. E. Bekris, "Improving the scalability of asymptotically optimal motion planning for humanoid dual-arm manipulators," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 271–277.
- [45] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 625–632.
- [46] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 729–746, 2004.

- [47] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2d," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6520–6526.
- [48] H. D. Young, R. A. Freedman, and L. A. Ford, *University physics with modern physics*, 2020.
- [49] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.
- [50] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.
- [51] S. Haddadin, S. Parusel, L. Johannismeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.

VII. APPENDIX

A. Energy-biased Optimality Search

We display the algorithm of RRT* with a hybrid cost function in Algo. 3, as is discussed in Section 14.

Algorithm 3: RRT* with a Hybrid Cost Function

```

1  $\mathcal{X}_v \leftarrow \{\mathbf{x}_{\text{init}}\}; c(\mathbf{x}_{\text{init}}) \leftarrow 0; \mathcal{W} \leftarrow \emptyset;$ 
2 for  $i = 1$  to  $n$  do
3    $\mathbf{x}_{\text{rand}} \leftarrow \text{SAMPLEFREE}_i$ 
4    $\mathbf{x}_{\text{nearest}} \leftarrow \text{NEAREST}(G = (\mathcal{X}_v, \mathcal{W}), \mathbf{x}_{\text{rand}})$ 
5    $\mathbf{x}_{\text{new}} \leftarrow \text{STEER}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}})$ 
6   if  $\text{OBSTACLEFREE}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{new}})$  then
7      $\mathcal{X}_{\text{near}} \leftarrow \text{NEAR}(G =$ 
        $(\mathcal{X}_v, \mathcal{W}), \mathbf{x}_{\text{new}}, r(\text{CARD}(V)))$ 
8      $\mathcal{X}_v \leftarrow \mathcal{X}_v \cup \{\mathbf{x}_{\text{new}}\}$ 
9      $\mathbf{x}_{\text{min}} \leftarrow \mathbf{x}_{\text{nearest}}$ 
10     $c_{e,\text{min}} \leftarrow \max\{c_e(\mathbf{x}_{\text{nearest}}), E(\mathbf{x}_{\text{new}}) - E(\mathbf{x}_{\text{init}})\}$ 
11     $c_{l,\text{min}} \leftarrow c_l(\mathbf{x}_{\text{nearest}}) + \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{nearest}}\|$ 
12     $c_{\text{min}} \leftarrow c_{e,\text{min}} + \gamma c_{l,\text{min}}$ 
13    foreach  $\mathbf{x}_{\text{near}} \in \mathcal{X}_{\text{near}}$  do
14       $c_{e,\text{que}} \leftarrow$ 
         $\max\{c_e(\mathbf{x}_{\text{near}}), E(\mathbf{x}_{\text{new}}) - E(\mathbf{x}_{\text{init}})\}$ 
15       $c_{l,\text{que}} \leftarrow c_l(\mathbf{x}_{\text{near}}) + \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{near}}\|$ 
16       $c_{\text{que}} = c_{e,\text{que}} + \gamma c_{l,\text{que}}$ 
17      if  $\text{COLLISIONFREE}(\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}}) \wedge c_{\text{que}} <$ 
         $c_{\text{min}}$  then
18         $\mathbf{x}_{\text{min}} \leftarrow \mathbf{x}_{\text{near}}$ 
19         $c_{\text{min}} \leftarrow c_{\text{que}}$ 
20    foreach  $\mathbf{x}_{\text{near}} \in \mathcal{X}_{\text{near}}$  do
21       $c_{e,\text{que}} \leftarrow$ 
         $\max\{c_e(\mathbf{x}_{\text{near}}), E(\mathbf{x}_{\text{new}}) - E(\mathbf{x}_{\text{init}})\}$ 
22       $c_{l,\text{que}} \leftarrow c_l(\mathbf{x}_{\text{near}}) + \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{near}}\|$ 
23       $c_{\text{que}} = c_{e,\text{que}} + \gamma c_{l,\text{que}}$ 
24      if  $\text{COLLISIONFREE}(\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}}) \wedge c_{\text{que}} <$ 
         $c(\mathbf{x}_{\text{near}})$  then
25         $\mathbf{x}_{\text{pa}} \leftarrow \text{PARENT}(\mathbf{x}_{\text{near}})$ 
26         $\mathcal{W} \leftarrow$ 
         $(\mathcal{W} \setminus \{(\mathbf{x}_{\text{pa}}, \mathbf{x}_{\text{near}})\}) \cup \{(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{near}})\}$ 
27 Return  $\mathcal{G} = (\mathcal{X}_v, \mathcal{W})$ 

```

The algorithm follows the framework defined in [38]. It first samples a new point \mathbf{x}_{new} , finds its nearest neighbor $\mathbf{x}_{\text{nearest}}$ in the tree, and steers towards it (Lines 3-5) as in RRT. Instead of directly connecting the two, RRT* expands and rewires nodes to create a minimum-cost path. It establishes a set of nodes $\mathcal{X}_{\text{near}}$ nearby \mathbf{x}_{new} within a radius

$$r(\text{CARD}(V))) = \min \left\{ \gamma_{\text{RRT}^*} \left(\frac{\log(\text{card}(\mathcal{X}_v))}{\text{card}(V)} \right)^{\frac{1}{d}}, \eta \right\}, \quad (7)$$

where $\text{CARD}(\cdot)$ denotes the cardinality of a set, i.e. the number of nodes in the set. γ_{RRT^*} and η are tuning hyperparameters so as to guarantee the radius decreases in a reasonable way as the tree grows. d denotes the dimension of the state space. We introduce a hybrid cost function, mixing

path energy and path length cost (Lines 10-12), rather than an additive cost function only. Starting from $\mathbf{x}_{\text{nearest}}$, the algorithm exhausts $\mathcal{X}_{\text{near}}$ and creates an edge from the vertex in $\mathcal{X}_{\text{near}}$ to \mathbf{x}_{new} along a path with minimum cost (Lines 13-19). Subsequently, new edges are created from \mathbf{x}_{new} to $\mathcal{X}_{\text{near}}$, if the current path through \mathbf{x}_{pa} costs higher than a path through \mathbf{x}_{new} ; the suboptimal edge connecting \mathbf{x}_{pa} and \mathbf{x}_{near} is removed from the tree in this case, since it is not a part of the minimum-cost path from the root of the tree (Lines 20-26).

Furthermore, we place the definitions of probabilistic completeness and asymptotic optimality in [38] here to help readers better understand the proof in Section 14.

Definition 4: An algorithm is *probabilistically complete*, if, for any robustly feasible path planning problem $(\mathcal{X}_{\text{free}}, \mathbf{x}_{\text{init}}, \mathcal{X}_{\text{goal}})$,

$$\liminf_{n \rightarrow \infty} \mathbb{P}(\{ \exists \mathbf{x}_{\text{goal}} \in \mathcal{X}_v^n \cap \mathcal{X}_{\text{goal}} | \mathbf{x}_{\text{init}} \text{ is connected to } \mathbf{x}_{\text{goal}} \text{ in } \mathcal{G}^n \}) = 1,$$

where $\mathcal{G}^n = (\mathcal{X}_v^n, \mathcal{W}^n)$ denotes the tree at step n with a node set \mathcal{X}_v^n and an edge set \mathcal{W}^n .

Definition 5: An algorithm is *asymptotically optimal* if, for any path planning problem $(\mathcal{X}_{\text{free}}, \mathbf{x}_{\text{init}}, \mathcal{X}_{\text{goal}})$ and cost function $c : \Sigma \rightarrow \mathbb{R}_{\geq 0}$ that admit a robustly optimal solution with finite cost c^* ,

$$\mathbb{P} \left(\left\{ \limsup_{n \rightarrow \infty} C(\sigma_n^*) = c^* \right\} \right) = 1,$$

where σ_n^* denotes the optimal path found by the planner at iteration n . Since an asymptotically optimal algorithm is able to find an optimal path given infinite runtime, it clearly satisfies probabilistic completeness as well.

B. Hybrid Cost Function

In this section, we empirically clarify that $\gamma = 0$ is a practical and valid simplification of the original hybrid cost function in Algo. 3. We define normalized total cost $\bar{C}(\hat{\sigma}^*)$ along $\hat{\sigma}^*$ as follows,

$$\bar{C}(\hat{\sigma}^*) = \frac{C_e(\hat{\sigma}^*) + \gamma C_l(\hat{\sigma}^*)}{c^*}, \quad (8)$$

with a path length cost

$$C_l(\hat{\sigma}^*) = \sup_{\{n \in \mathbb{N}, 0 < \tau_0 < \tau_1 < \dots < \tau_n = 1\}} \sum_{i=1}^n \|\hat{\sigma}^*(\tau_i) - \hat{\sigma}^*(\tau_{i-1})\|, \quad (9)$$

which is in line with the definition of total variation in [38]. $\hat{\sigma}^*$ denotes the estimated optimal path given by Algo. 3. For example, in a simple 2D 1×1 workspace, a point-mass object with $mg = 1$ and gravitational potential energy traverses from $(0, 0)$ to $(1, 0)$ while avoiding obstacles (Fig. 8, 6). Note that the negative direction of the y axis aligns with the direction of gravity.

⁴1. w.s. dim.: workspace dimension. 2. # DoF: degrees of freedom of a fixtured object. 3. rigid: a rigid fixtured object, deformable or articulated otherwise. 4. E_g : gravitational potential energy is considered. 5. E_e : elastic potential energy is considered. 6. planner: asymptotically optimal algorithm used. 7. runtime: per iteration per run. 8. # runs: number of runs per iteration. 9. s.p.b.: samples per batch of BIT*.

TABLE II: Technical details of all soft fixture scenarios.⁴

scenarios	radish	mask	hook	starfish	shovel	rope	cube	lock
w.s. dim.	3	3	3	3	3	3	3	2
# DoF	18	18	6	16	15	18	6	4
rigid			✓				✓	
E_g			✓	✓	✓	✓	✓	
E_e	✓	✓			✓			✓
planner	BIT*	BIT*	BIT*	BIT*	BIT*	RRT*	BIT*	BIT*
runtime /s	300	300	300	200	300	200	200	200
# runs	-	12	5	6	12	16	10	-
s.p.b	1000	1000	2000	2000	5000	-	5000	1000

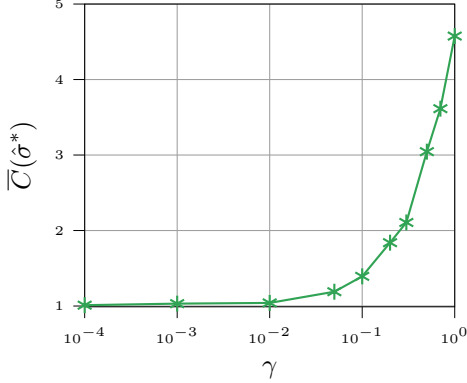
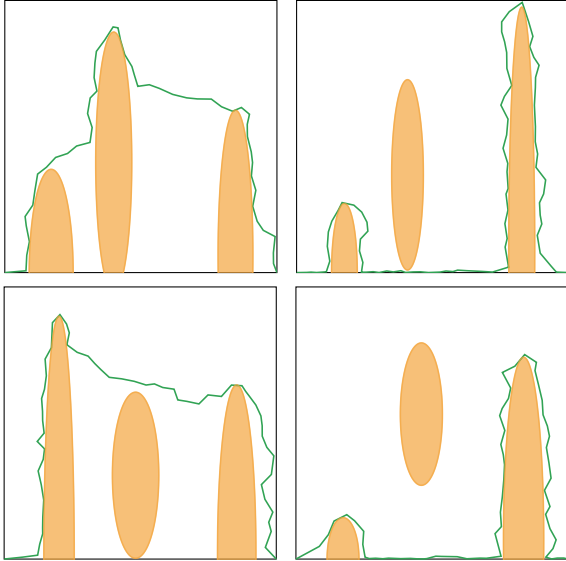
Fig. 7: Normalized total cost with respect to the mixture parameter γ . Each data point denotes an average of results over 8 10-sec runs with varying obstacle space composition (Fig. 6). BIT* rewiring factor is set to 0.5, with 2000 samples per batch.

TABLE III: Normalized total cost w.r.t the mixture parameter.

γ	0.0	10^{-3}	10^{-2}	10^{-1}
$\bar{C}(\hat{\sigma}^*)$ mean	1.021	1.032	1.041	1.393
$\bar{C}(\hat{\sigma}^*)$ std.	0.012	0.016	0.009	0.065

Fig. 6: 4 examples of randomly generated obstacle spaces. Estimated optimal paths are displayed with $\gamma = 0$, i.e. the planner tends to minimize the energy cost only. $\bar{C}(\hat{\sigma}^*)$ of the 4 scenarios above lies within 1.01 and 1.02.

Depending on the mixture parameter γ , the planner takes different strategies to balance the path length cost $C_l(\hat{\sigma}^*)$ and energy cost $C_e(\hat{\sigma}^*)$, as is illustrated in Fig. 8. When

$\gamma = 0$, the path goes along the lower border of the free space to minimize $C_e(\hat{\sigma}^*)$. A higher weight on the length cost ($\gamma = 10.0$, Fig. 8 right) encourages a path taking a shortcut around the top of the middle ellipsoid at the cost of higher $C_e(\hat{\sigma}^*)$. A compromise is made when $\gamma = 1$, taking both length and energy costs into consideration.

To evaluate a degenerated form of the hybrid cost function, i.e. $\gamma = 0$, we consider the environment in Fig. 6 with a limited level of randomness of obstacle space compositions. A normalized cost $\bar{C}(\hat{\sigma}^*)$ is considered, instead of $C(\hat{\sigma}^*)$, to create a unified measure under various scenarios. Presented in Fig. 7 and Tab. III, $\bar{C}(\hat{\sigma}^*)$ tends to 1.0 as γ decreases to 0.0. $\bar{C}(\hat{\sigma}^*)$ slightly larger than 1.0 indicates that the planner provides a convincing upper bound to the escape energy. Combining with the paths in Fig. 6, we take $\gamma = 0$ as an empirically valid simplification of the hybrid cost function, although it slightly violates the strict positivity in the definition of cost function in [38].

C. Qualitative Evaluation of Other Scenarios

4 additional scenarios of soft fixtures are presented in Fig. 9. In the first example, a starfish, modeled by kinematic links, falls into a bowl from a tilting human palm. The other hand, holding the rim of the bowl, wobbles the bowl back and forth to make the starfish slide down to the bottom of the bowl. In the meantime, the escape energy estimated by BIT* goes up and down. In a second scenario (Fig. 9, top right), a frozen fish, modeled as a rigid object, lies initially on the table. It is pierced by a fish hook through its gill and lifted. A rope loop with blood bags, as presented in the bottom left, is fixtured by a Robotiq gripper. The loop is modeled by kinematic links and only the gravitational potential energy of the blood bags is considered. To firmly grasp the loop in hand, the wrist rotates and the palm turns upward, as the fingers further close. The last scenario emulates the process that a Rubik cube player examines the cube by holding it in hand. The other gripper covers it from the top and forms a cage (infinite escape energy in the middle of the plot). The two grippers rotate, centering at the cube, ending up being aligned horizontally. As the two fingers covering the cube open up, the cage is broken up (the last few iterations in the plot). Please refer to the complementary video for more details.

We observe from above that real-life fixturing motions tend to increase the escape energy of objects so that they won't readily escape, even in the existence of external disturbances. This indicates the capability of escape energy

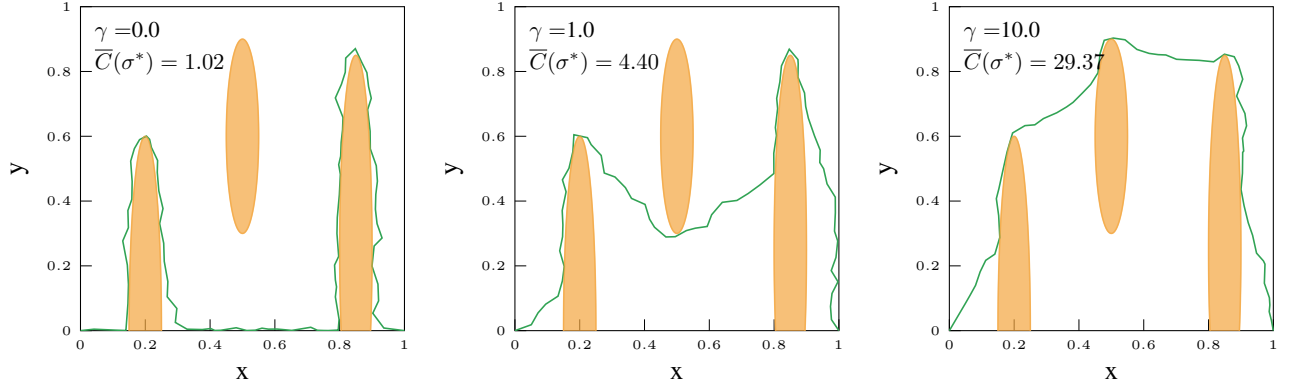


Fig. 8: Estimated optimal paths found by an asymptotically optimal algorithm (BIT*) with a hybrid cost function. Varying values of $\gamma \in \{0.0, 1.0, 10.0\}$ are used in 3 scenarios with fixed obstacles. Paths are denoted in green and ellipsoidal obstacles in orange. The runtime and BIT* setup stay the same as Fig. 7.

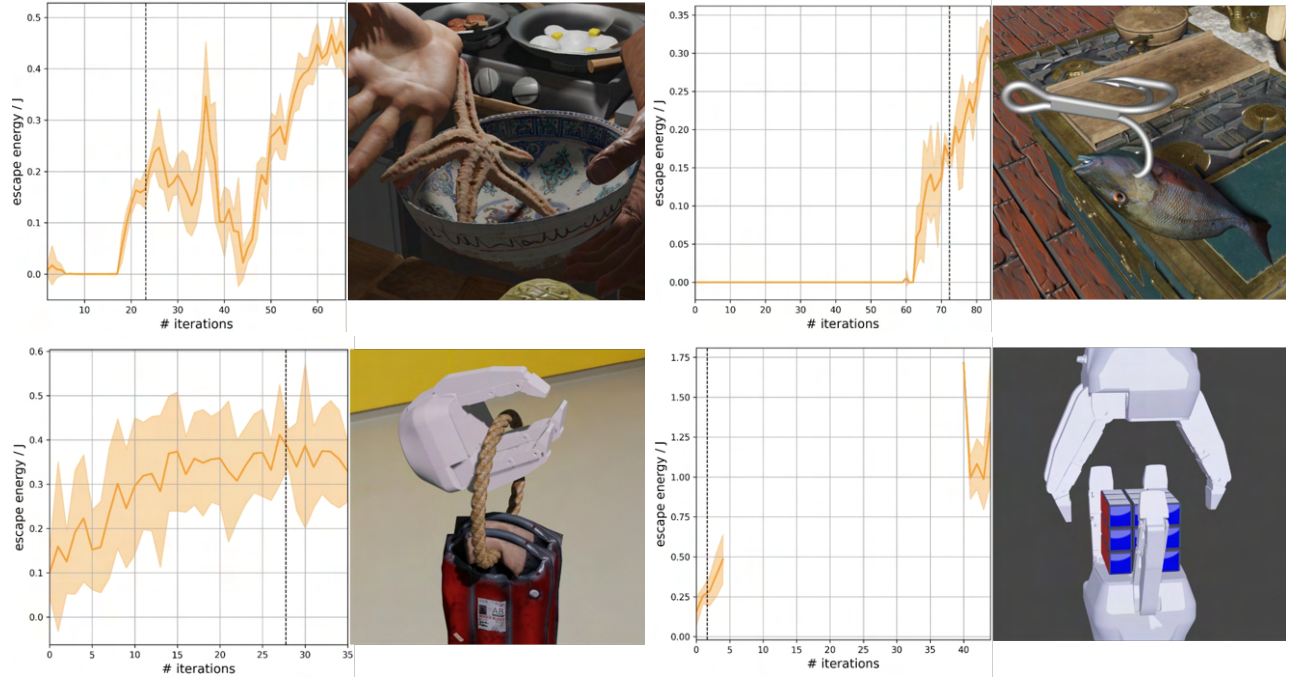


Fig. 9: From top to bottom: (1) catching a starfish with a bowl, (2) hooking a frozen fish with a fishing hook, (3) grasping a rope loop of blood bags, (4) manipulating a Rubik cube with two grippers. Escape energy is plotted over iterations. One of the iterations is illustrated to the right of each plot, corresponding to the dashed vertical lines in the plots.

in measuring the stability of caging-based grasps. It also inspires us to explore the correlation between escape energy and the probability of escape in real-world experiments. Technical details of all 8 soft fixtures scenarios in Fig. 1 are listed in Tab. II.

D. Details of Real World Experiments

The initial configuration of the 7-axis robot arm with joint angles in Fig. 5 (i-1) could be expressed as $\beta_{\text{init}} = [0, -\pi/4, 0, -3\pi/4, 0, \pi/2, \pi/4]$. For a perturbation level τ , the arm is commanded to reach a perturbed configuration β_p with joint $\beta_p^k, k \in \{2, 3, 5, 6, 7\}$ uniformly sampled from

$[\beta_{\text{init}}^k - \pi\tau/4, \beta_{\text{init}}^k + \pi\tau/4]$, which repeats 10 times. Therefore, the arm reaches 10 perturbed configurations consecutively under τ . Joint $k \in \{1, 4\}$ stay fixed since their aggressive and random movement might cause a collision. The perturbation level τ increases from 0 with a fixed resolution, for instance, $\Delta\tau = 0.03$ until objects escape from the caging tool. We placed an object to the bottom of the caging tool, marking its configuration w.r.t. the tool within the inner side of the bowl-like tool. This is a valid way since most of the objects are regular in shape and 3D printed, a perception system with visual markers (i.e. April tag) attached to the objects is better but too expensive for our demands.