

# Scale-Invariant Sampling in Multi-Arm Bandit Motion Planning for Object Extraction

Servet B. Bayraktar<sup>1</sup>, Andreas Orthey<sup>1</sup>, and Marc Toussaint<sup>1</sup>

Technical University of Berlin

**Abstract.** Object extraction tasks often occur in disassembly problems, where bolts, screws, or pins have to be removed from tight, narrow spaces. In such problems, the distance to the environment is often on the millimeter scale. Sampling-based planners can solve such problems and provide completeness guarantees. However, sampling becomes a bottleneck, since almost all motions will result in collisions with the environment. To overcome this problem, we propose a novel scale-invariant sampling strategy which explores the configuration space using a grow-shrink search to find useful, high-entropy sampling scales. Once a useful sampling scale has been found, our framework exploits this scale by using a principal components analysis (PCA) to find useful directions for object extraction. We embed this sampler into a multi-arm bandit rapidly-exploring random tree (MAB-RRT) planner and test it on eight challenging 3D object extraction scenarios, involving bolts, gears, rods, pins, and sockets. To evaluate our framework, we compare it with classical sampling strategies like uniform sampling, obstacle-based sampling, and narrow-passage sampling, and with modern strategies like mate vectors, physics-based planning, and disassembly breadth first search. Our experiments show that scale-invariant sampling improves success rate by one order of magnitude on 7 out of 8 scenarios. This demonstrates that scale-invariant sampling is an important concept for general purpose object extraction in disassembly tasks.

**Keywords:** Motion Planning · Scale-Invariant Sampling · Object Extraction

## 1 Introduction

Removing an object out of a tight, narrow passage is a fundamental skill for robot disassembly tasks in recycling [3], repair [42], and remanufacturing [33,15]. Sampling-based methods [41] can tackle such problems and provide completeness guarantees [55,1,17,52]. However, due to narrow passages, such problems often become intractable to solve [17]. This is often due to the large scale on which methods like rapidly exploring random tree (RRT) [31] operate, where motions out of narrow passages are almost always invalid.

To tackle this problem, we propose a novel scale-invariant sampling strategy. This strategy is based on the observation that scale is an important consideration for motion planning of object extraction tasks. For example, if a point robot

is inside a narrow passage, and you sample points inside a large radius around it, then almost all of the points will be unreachable (the robot just bumps into the wall). In contrast, if we sample in an infinitesimal small radius, then almost all samples will be reachable. Those two scales have, however, a low information entropy [27] and do not help a planner to make good decisions. What this implies, however, is that there is a scale, at which the information entropy is high, meaning samples are maximally useful for planning. Such a scale corresponds to situations where roughly half of the samples become reachable [27]. This scale would not only have a high information entropy, but it would be an exceptional guidance to the planner.

To implement this insight, we devise two sampling strategies. One is a scale sampler, which explores different scales to find a high information entropy scale. The other is a principal component analysis (PCA) sampler, which exploits the principal components in the samples at the previously computed high information entropy scale. Both of those samplers are integrated into a multi-arm bandit RRT (MAB-RRT) [19]. An overview about this framework is shown in Fig. 1. We found scale-invariant sampling in MAB-RRT to be an effective planning strategy for disassembly tasks where bolts, pins, or gears have to be removed from a narrow passage.

In summary, our contributions are:

1. We propose a novel scale sampler which can autonomously find high information entropy scales for efficient sampling densities.
2. We develop a PCA sampler to exploit a given high information entropy scale by biasing samples along a positive or negative principal component direction.
3. Embedding both samplers into the MAB-RRT planner to combine it with other sampling strategies like uniform sampling.
4. We provide an implementation of samplers and MAB-RRT as an extension of the Open Motion Planning Library (OMPL) and provide a set of demos scenarios.

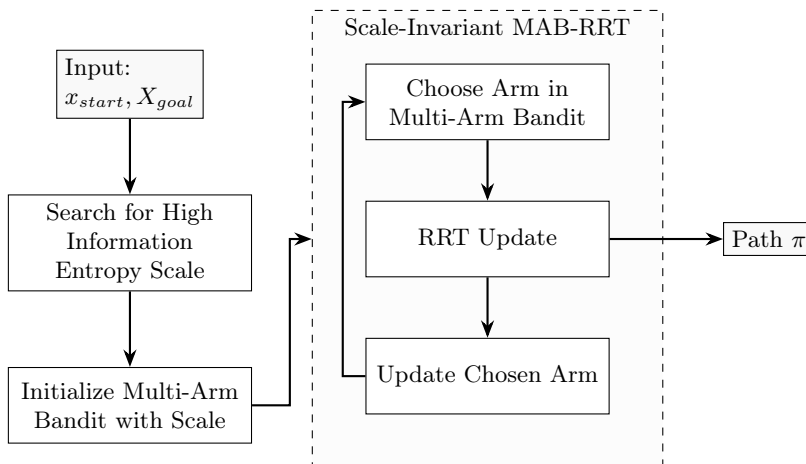
Eventually, this sampling strategy is tested against other classical sampling methods on eight challenging 3D scenarios involving realistic object extraction tasks.

## 2 Related Work

This work is closely related to object extraction, biased sampling in motion planning, and PCA-based sampling. We review those topics and discuss how our work is connected.

### 2.1 Object Extraction

Object extraction is the task of separating two objects until the distance between them reaches a predefined threshold. Our interest is in tasks where the objects



**Fig. 1:** Overview about scale-invariant sampling in MAB-RRT. As input, we use a start configuration  $x_{start}$  and a goal region  $X_{goal}$ . We then search for a high-information entropy scale and initialize the multi-arm bandit with this scale. We then start a MAB-RRT loop, where an arm is chose, the chosen sampler is used for one RRT update step, and the arm reward is updated. This loop continues until a termination condition is true or a path  $\pi$  has been found which solves the problem.

have to move along tight, narrow passages to achieve separation, a task which is often encountered in disassembly problems [52].

A main focus of research in this area is on fastener removal. For example, detecting screw types and using the correct tool for screw extraction [54] or learning policies for single object extraction tasks with a robot manipulator [47]. Specialized systems exists for specific screw types like hexagonal screws [36]. Our work is complementary in that we tackle arbitrary objects, including bolts, nuts, or gears, for which a removal plan has first to be computed to achieve a separation.

Apart from fastener removal, research has also focused on cluttered extraction. This includes extracting boxes from piles without other boxes collapsing [43], or carefully extracting objects from a pile without disturbing the other objects [39].

For object extraction in disassemblies, specialized planners have been developed. For example, the Targetless-RRT [1] explores the space without a specified goal region, but instead tries to minimize its distance to a predefined exterior space which represents a disassembled state. This idea has later been combined with mating vectors in the Mating Vector and Targetless-RRT (MateVec-TRRT) [17,52]. MateVec-TRRT uses the concept of mating vectors, which are vectors which point into the direction of a separation between two objects. This is particularly advantageous when two objects are close by and just have to be pulled apart [17]. Another approach is to use physics-based separation vectors [52,55]. This idea is based on the observation that nuts and bolts can be

moved by an approximate force which has a component into the motion direction [52]. This avoids costly computation of the exact motion direction. This can then be implemented in a planner like the Behavioral Kinodynamic Rapidly-Exploring Random Trees (BK-RRT) [55], where the random physical forces are used as forward propagations to reach new states. A more dedicated planner is the disassembly breadth first search (BFS) [52], which uses physics-based force directions in combination with a breadth first search and state similarity checks.

Our work extends approaches like MateVec-TRRT [17], BK-RRT [55], and BFS [52] in that we focus on general removal tasks in disassembly scenarios, where we add the concept of scale-invariant sampling to ensure that we progress along long elongated narrow passages. Additionally, instead of using a possibly costly physical simulation [52], we use purely geometrical arguments to find the direction of motion.

## 2.2 Biased Sampling

To circumnavigate narrow passages, most research has focused on biasing samples. Common approaches include Obstacle-based sampling [8], where samples are biased towards obstacle boundaries, Bridge-based sampling [24] where samples are biased towards narrow passages, Utility-based sampling [10], where samples are drawn based upon their utility to improve the roadmap, and Dynamic-domain RRT [53], where the Voronoi bias of samples is adjusted based on their performance. Sample biases can also be learned from similar situations [26,12] to improve sampling quality. Once paths are found, biasing can be used to improve the path quality by using informed sampling [21], where an admissible ellipsoid heuristic is used in RRT\* [29] to bias samples towards paths with lower cost. More recent approaches focus on selective densification [25], where sparse regions are identified, and their resolution is changed so that more samples are produced in narrow passages. Our approach differs by concentrating on object extraction tasks while carefully finding trade-offs between uniform and scale-invariant sampling.

This trade-off is similar to the exploration/exploitation trade-off [45], where an exploration step finds locally free configuration space regions, while an exploitation step uses this region to quickly find valid paths. We follow a similar approach, but see exploration as a search for high information entropy samples, while exploitation as the biasing operation along the high-information entropy samples.

## 2.3 PCA-based sampling

A widely used sampling analysis tool in motion planning is Principal component analysis (PCA) [44,28]. PCA can extract a principal component from a set of samples, thereby showing a direction in which the samples are distributed. This is particularly useful to find directions to sample inside narrow passages.

One way to leverage PCA is to detect if samples are currently inside a narrow passage. For example, ADD-RRT [11] uses PCA to identify nodes which are

likely in a narrow passages, and changes the sampling strategy accordingly in the immediate neighborhood. If regions of interests are available, like in manipulation tasks, PCA can be used to ensure sampling is better distributed in those regions [46]. Another application of PCA in this context is to reduce the conformal states for protein folding [50] thereby reducing the effective dimensions of the problem.

PCA can also be used globally to shape the sampling distributions around an existing configuration. The pioneering work by Dalibard et al. [13,14] uses PCA to locally adjust the sampling radius around a node proportional to the Eigenvalues along each principal component. To make this more efficient, later work [35] used a bridge test [24] to identify narrow passages before running PCA on the samples. This reduces the calls to PCA to relevant regions of the configuration space. For dynamical systems, PCA can also be used to learn the bias stemming from the dynamics [37] to make sampling more efficient.

Our work differs in that we do not consider PCA as a global tool for efficient sampling [14], but instead leverage PCA as a local object extraction tool to exploit samples at high information entropy scales.

## 2.4 Multi-Arm Bandit

While novel sampling methods are important to better explore the space, it is also important to decide when to use which sampling method. A convenient framework to make this decision is the multi-arm bandit (MAB) [4,9,48]. MABs have been used in motion planning for different purposes, for example to select the best path from a set of candidate paths [30], to select from different grasp strategies [18], and to run multiple trees with different sampling strategies, then select the best performing tree [32].

A major work in this area is the MAB-RRT [19,20], which is an RRT [31] with a MAB replacing the uniform sampling strategy. In the original MAB-RRT [19], each arm is used to select a particular region of the state space to sample. However, MAB-RRT is a general purpose planner which can take as input any set of sampling strategies if the MAB is clearly defined. Our work is complementary in that we treat MABs, similar Faroni et al. [19], as our global decision framework. However, while we leverage MAB-RRT as our framework, we differ by tackling a different application (object extraction) while using different sampling strategies and reward functions in the MAB.

## 3 Scale-Invariant Sampling in Multi-Arm Bandit Motion Planning

We propose a new sampling scheme which consists of three interconnected methods: Scale sampling to find high information entropy scales, directional sampling to exploit those scales, and a multi-arm bandit planner to integrate those samplers with classical sampling strategies. Each method is further detailed below.

---

**Algorithm 1: Finding High Information Entropy Scale**

---

**Input** : start configuration  $q_0$   
**Parameters:** initial radius  $r_0$ ; optimal validity rate interval  $[\alpha_{\min}, \alpha_{\max}]$ ;  
shrink factor  $s$ ; growth factor  $g$ ; min radius  $r_{\min}$ ; batch size  $b$ ,  
max steps  $S$   
**Output** : final radius  $r^*$ , valid samples  $V$

```

1  $r \leftarrow r_0$ ;  $V \leftarrow \emptyset$ 
2 for  $i = 1$  to  $S$  do
3    $\mathcal{Q} \leftarrow \text{SAMPLESPHERE}(q_0, r, b)$ 
4    $V_{\text{new}} \leftarrow \{q \in \mathcal{Q} \mid \text{CHECKMOTION}(q_0, q)\}$ 
5    $\alpha \leftarrow \frac{|V_{\text{new}}|}{|\mathcal{Q}|}$  // validity rate
6   if  $\alpha \in [\alpha_{\min}, \alpha_{\max}]$  then
7      $r^* \leftarrow r$ ;  $V \leftarrow V \cup V_{\text{new}}$ ; break
8   else if  $\alpha < \alpha_{\min}$  then
9      $r \leftarrow r/s$ ;  $V \leftarrow V \cup V_{\text{new}}$  // shrink radius
10  else
11     $r \leftarrow r \cdot g$  // grow radius
12  end
13 end
14 if  $r \leq r_{\min}$  then
15    $r^* \leftarrow r_{\min}$ 
16 return  $(r^*, V)$ 

```

---

### 3.1 Exploration: Finding High Information Entropy Scales

The scale sampler is an adaptive sampling strategy designed to identify radii at which sampling yields a high information content. In particular, it aims to find a sampling scale at which the fraction of valid samples lies within a given target interval. This interval specifies a desired range of validity rates that characterizes informative sampling scales.

The algorithm is depicted in Alg. 1. At each iteration, the scale sampler draws a fixed-size batch of quasi-random samples from a sphere (see below) of the current radius and evaluates their validity (Line 3,4). Based on the observed validity rate, the radius is increased or decreased (Line 6–12). This process continues until a radius is found whose validity rate falls within the target interval (Line 6–7), or until a predefined computational budget of  $S$  steps is exhausted. In that case, we check if the radius is smaller than the minimal radius, in which case we clip it (Line 14–15). Afterwards, the best radius so far is returned (Line 16).

Importantly, the scale sampler does not attempt to compute an optimal radius. Instead, it performs a feasibility-driven search whose goal is to find any radius that yields informative samples under noisy evaluation with finite samples. Our method is thereby robust to stochastic variability in validity estimates and avoids the need for strong assumptions such as deterministic monotonicity of validity with respect to scale.

---

**Algorithm 2:** Sample from PCA-Aligned Cylinder

---

**Input** : Normalized PCA axis  $a_0$ , direction  $d \in \{0, 1\}$ , cylinder height interval  $[h_{\min}, h_{\max}]$ , extension  $\delta$ , radius  $R$

**Output:** sample configuration  $q$

// (1) Sample random height vector

- 1  $h \sim \mathcal{U}(h_{\min}, h_{\max})$
- 2  $a \leftarrow h \cdot a_0$

// (2) Sample random direction in (N-1)-dimensional ball

- 3  $u \sim \mathcal{U}(0, 1)$
- 4  $t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{N-1})$
- 5  $p \leftarrow R \cdot u^{1/(N-1)}$  // Account for volume density
- 6  $b \leftarrow p \cdot t / \|t\|$

// (3) Assemble vector to get final sample

- 7  $Q \leftarrow \text{ORTHONORMALBASIS}(a)$
- 8 **return**  $a + Q \cdot b$

---

*Sphere Sampling* An important aspect of our algorithm is the sphere sampling scheme. When using uniform sampling, we often face clustering of points on the sphere, which negatively impacts the validity rate, which requires a more uniform coverage of samples. To achieve this, we employ a dual sampling scheme, where alternate between uniform sampling and a jittered Fibonacci lattice sampling scheme. The Fibonacci lattice provides low-discrepancy samples [40,34] which guarantee a better coverage that avoids clustering. Without jitter, the lattice would place samples at identical positions across radii, making coverage entirely dependent on the uniform component. This jitter is controlled by the parameter  $p_{\text{fibonacci-jitter}}$ .

### 3.2 Exploitation: Principal Components Sampling

Once a high-information entropy radius has been found, our planner should exploit it. A common way to better understand the valid sample distribution at the chosen scale is to run a principal component analysis (PCA). A PCA is a linear transformation of the data onto a new coordinate system, such that the (first) principal component (or principal axis) captures the largest variation in the data [7]. PCA is a staple of many scientific software packages, and can be implemented with tools like Eigen [23].

However, to exploit the principal component for biased sampling requires a dedicated sampler. One possibility is to create a hyper-cylinder from the principal component and sample around it. This is depicted in Alg. 2. Our algorithm proceeds in three stages. First, we sample a random height from the axis  $a_0$  to obtain a vector  $a$  (Line 1–2). Second, we create an  $(N - 1)$ -dimensional ball around  $a$ , which is orthogonal to  $a$  itself. To achieve this, we first sample directly in an  $(N - 1)$ -dimensional ball by getting a random distance variable  $u$  (Line 3) and a random direction  $t$  (Line 4). We then compute the radius  $p$  by accounting for the volume density of the ball in  $N - 1$  dimensions (there is more density

**Algorithm 3: Scale-Invariant MAB-RRT**


---

**Input:** Start state  $x_{\text{start}}$ , goal region  $X_{\text{goal}}$   
**Output:** Path  $\pi$  or failure

- 1  $(r^*, V) \leftarrow \text{FINDHIGHINFOENTROPYSCALE}(x_{\text{start}})$
- 2  $T \leftarrow \{x_{\text{start}}, V\}$
- 3  $\mathbf{a} \leftarrow \text{COMPUTEPCA}(V)$  // Principal escape direction
- 4  $S \leftarrow \{\text{UNIFORM}, \text{PC-POSITIVE}(\mathbf{a}, r^*), \text{PC-NEGATIVE}(\mathbf{a}, r^*)\}$
- 5 **while** *Not*  $\text{TERMINATE}()$  **do**
- 6  $h_{\text{ext}} \leftarrow \delta \cdot r^*$  // Cylinder extension from  $r^*$
- 7  $s \leftarrow \text{SELECTARMMAB}(S)$  // UCB arm selection
- 8  $x_{\text{sample}} \leftarrow \text{SAMPLE}(s, h_{\text{ext}})$
- 9  $x_{\text{near}} \leftarrow \text{NEAREST}(T, x_{\text{sample}})$
- 10  $x_{\text{new}} \leftarrow \text{STEER}(x_{\text{near}}, x_{\text{sample}})$
- 11  $\text{valid} \leftarrow \text{COLLISIONFREE}(x_{\text{near}}, x_{\text{new}})$
- 12 **if**  $\text{valid}$  **then**
- 13  $T \leftarrow T \cup \{x_{\text{near}}, x_{\text{new}}\}$
- 14 **if**  $s \neq \text{UNIFORM}$  **then**
- 15  $V \leftarrow V \cup \{x_{\text{new}}\}$  // Accumulate valid samples
- 16  $\mathbf{a} \leftarrow \text{RECOMPUTEPCA}(V, \mathbf{a})$  // Online recalibration
- 17  $r^* \leftarrow \max(r^*, r_{\text{sample}})$  // Expand reach
- 18 **end**
- 19 **if**  $x_{\text{new}} \in X_{\text{goal}}$  **then**
- 20 **return**  $\text{EXTRACTPATH}(T, x_{\text{new}})$
- 21 **end**
- 22 **end**
- 23  $\text{UPDATEREWARD}(S, s, \text{valid})$
- 24 **end**
- 25 **return** failure

---

further out). We then take  $p$  and use it to compute the final direction  $b$  in the  $N - 1$  ball. Third, we assemble the final sample by projecting  $b$  into the null-space of  $a$ . This is achieved by getting an orthonormal basis  $Q$  [5] and projecting  $b$  into it. The final result  $a + Q \cdot b$  (Line 8) gives a uniform distributed sample inside the cylinder of radius  $R$  around the principal component as desired.

### 3.3 Integration of Samplers into Multi-Arm Bandit RRT

To decide when to use which sampler and to integrate them with other sampling strategies, we utilize a geometric version of the multi-arm bandit RRT (MAB-RRT) [19]. The algorithm itself is depicted in Alg. 3. This planner is similar to RRT [31], but uses in each iteration a possibly different sampling function as decided by the multi-arm bandit framework. This algorithm gets as input a start state  $x_{\text{start}}$ , a goal region  $X_{\text{goal}}$ , and outputs a path  $\pi$  or a failure. We start by searching for a useful (high-information entropy) scale (Line 1) using the grow-shrink algorithm (Alg. 1). The outcome is a radius  $r^*$  and valid nodes and edges  $V$ . The nodes and edges are added to the initial tree at  $x_{\text{start}}$  (Line 2).

PCA is then computed on  $V$  to obtain the principal escape direction (Line 3). Afterwards, we define the set of samplers (Line 4), which includes a uniform sampler and the two principal component samplers, one for the positive and one for the negative direction.

After those initialization steps, we start the inner loop while a terminate condition is not met. At each iteration, a cylinder extension height  $h_{\text{ext}}$  is computed from the current best radius  $r^*$ , allowing the principal component samplers to sample beyond the initial scale (see below). This inner loop starts with the selection of one sampler using the sliding window UCB policy denoted below. The remainder of the loop continues as in the original RRT by sampling a state, computing the nearest tree node, steering from this node to the sampled configuration, and checking if the connection is valid. If the connection is valid, we add it to the tree. For principal component samples, three additional updates occur: the valid sample is accumulated, the principal escape direction is recomputed online from all accumulated valid samples (see Online Recalibration below), and  $r^*$  is updated if the sample was drawn at a radius exceeding the current best, where  $r_{\text{sample}}$  denotes the radius at which the cylinder sampler generated the point (see Cylinder Extension below). If both the connection is valid and the last added configuration belongs to the goal region, we extract the path from the tree through backward search and return it. If no solution has been found yet, we update the bandit rewards and restart the loop. If no solution has been found when the terminate condition becomes true, we end the loop and return a failure.

**Multi-Arm Bandit Dynamics** The multi-arm bandit algorithm selects samplers upon the Upper Confidence Bound (UCB) algorithm while each arm is updated via a reward function computed from the validity of a sample.

*Arm Selection.* At each iteration, arm selection follows a sliding-window UCB policy [22]:

$$b^* = \arg \max_{b \in B} \hat{\mu}_b + \beta \sqrt{\frac{\ln(\sum_c n_c + 1)}{n_b + 1}},$$

where  $b \in B$  are the bandit arms,  $\hat{\mu}_b$  is the empirical mean rewards,  $n_b$  is the recent count of arm  $b$ ,  $\beta$  is the exploration coefficient balancing exploitation against exploration, and  $n_c$  is the count of the last  $N_{\text{sliding-window}}$  arm iterations. The window ensures that arm selection reflects recent performance, which is crucial in an object extraction task where the geometry changes as object parts separate. In practice, we scale uniform and sphere-based rewards by constants  $c_u$  and  $c_s$ , which tune the trade-off between outward tree growth and narrow-passage exploration.

*Reward function.* Rewards are computed with respect to the last pulled arm, and the resulting value is propagated to all bandits along the decision path. Uniform samples are rewarded proportionally to their distance from the origin, encouraging outward tree expansion. Scale invariant samples (Positive and negative principal component) are rewarded inversely to their distance, prioritizing

informative moves in narrow passages and down-weighting redundant samples once free space is reached. Invalid samples yield zero reward. Since the sliding-window UCB policy guarantees that each arm is selected infinitely often, and the uniform arm is itself probabilistically complete, the overall planner retains probabilistic completeness.

*Online Recalibration.* During the planning loop, the principal component axis is not fixed to the initial estimate from the scale search. Each time a principal component sampler produces a valid sample, the principal component is recomputed. To prevent axis flipping due to the sign ambiguity of PCA, the new axis is checked for consistency with the previous axis via their dot product. This online recalibration allows the escape direction to adapt as the planner discovers more of the local geometry beyond the initial radius.

*Cylinder Extension and Radius Growth.* The principal component samplers update their sampling region through an integrated growth mechanism. Instead of setting the cylinder height to  $r^*$ , an extension height of  $h_{\text{ext}} = \delta \cdot r^*$  with  $\delta \geq 0$  being the extension factor. With  $\delta = 0$ , no extension occurs and samples are drawn at exactly  $r^*$  along the axis. Each principal component sampler samples along its respective direction within the range  $[r^*, r^* + h_{\text{ext}}]$  from the origin along the axis. Importantly,  $r^*$  is not fixed after the scale search: whenever a valid principal component sample is generated at a radius exceeding the current  $r^*$ , the radius is updated. Since  $h_{\text{ext}}$  is recomputed from  $r^*$ , the escape direction is further increased for subsequent iterations.

### 3.4 Open Source Software Implementation

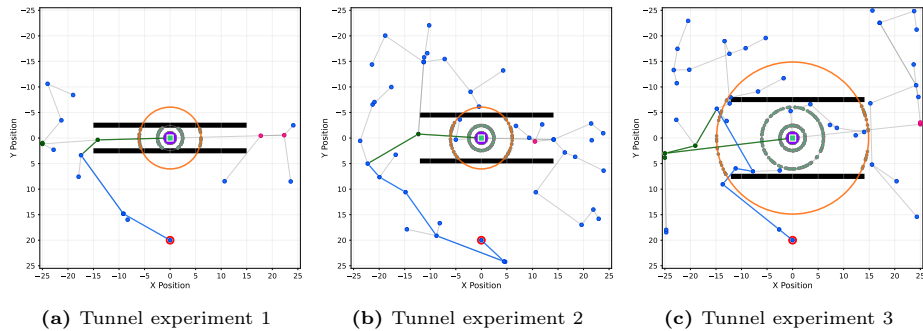
The samplers and the multi-arm bandit RRT have been implemented as an extension of the open motion planning library (OMPL) [49]. This code is open source and available on Github<sup>1</sup>. We added a MAB-RRT planner demo, which can run on arbitrary occupancy maps in 2D.

### 3.5 Example: Tunnel Environment

To showcase scale-invariant sampling with MAB-RRT, we created three toy scenarios as shown in Fig. 2. Those three scenarios are tunnel environments, where a point robot has to move from the origin (green point) to a goal configuration (red point). The scenarios differ by the size of a tunnel obstacle (black bars), where two bars restrict the start configuration from above and below. The distance between the bars differ for each scenario (5 units, 10 units, and 15 units).

For each scenario, we visualize different properties of scale-invariant sampling. First, we show the final burn-in radius (high information entropy), at which roughly half of the samples are valid and half invalid. Furthermore, we

<sup>1</sup> Link: <https://github.com/serboba/ompl/tree/MAB-RRT>



**Fig. 2:** Visualization of scale-invariant sampling in tunnel passages of different sizes (5 units, 10 units, and 15 units). Final burn-in radius is shown as —, and path are shown depending on the type of active sampler: — uniform, — positive principal component, and — negative principal component. Start is marked with ■, goal with ■. Valid samples are shown as ■ uniform, ■ positive principal component, ■ negative principal component, and burn-in valid samples as ■.

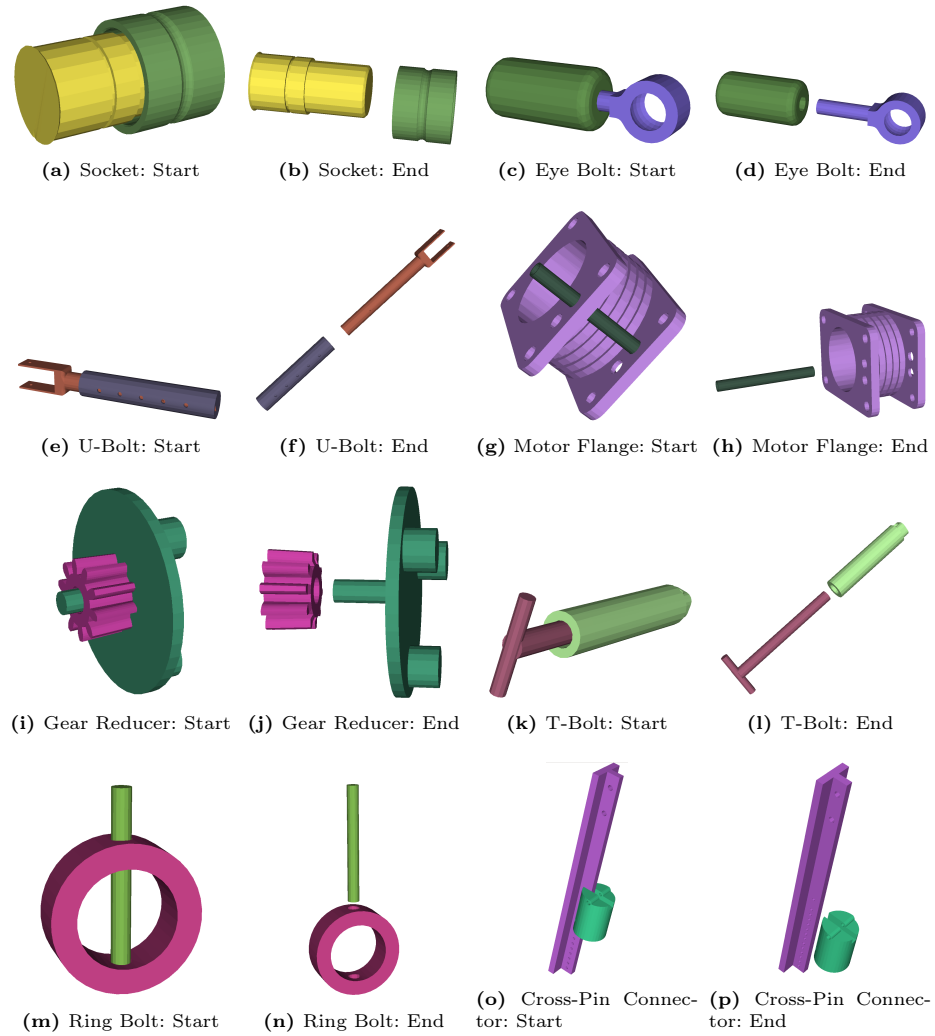
show the search trees and the way the samples have been obtained. This includes uniform sampling (blue), positive principal component (green), and negative principal component (magenta). It can be seen that the high information entropy radius gives a good insight into where to move in each scenario, which leads to an efficient exploration of the configuration space.

## 4 Results

We evaluate our scale-invariant sampler on 8 environments involving multiple tight narrow passage scenarios inspired by realistic disassembly problems. The eight environments are shown in Fig. 3 and represent a diverse set of disassembly tasks adopted from an existing dataset [52].

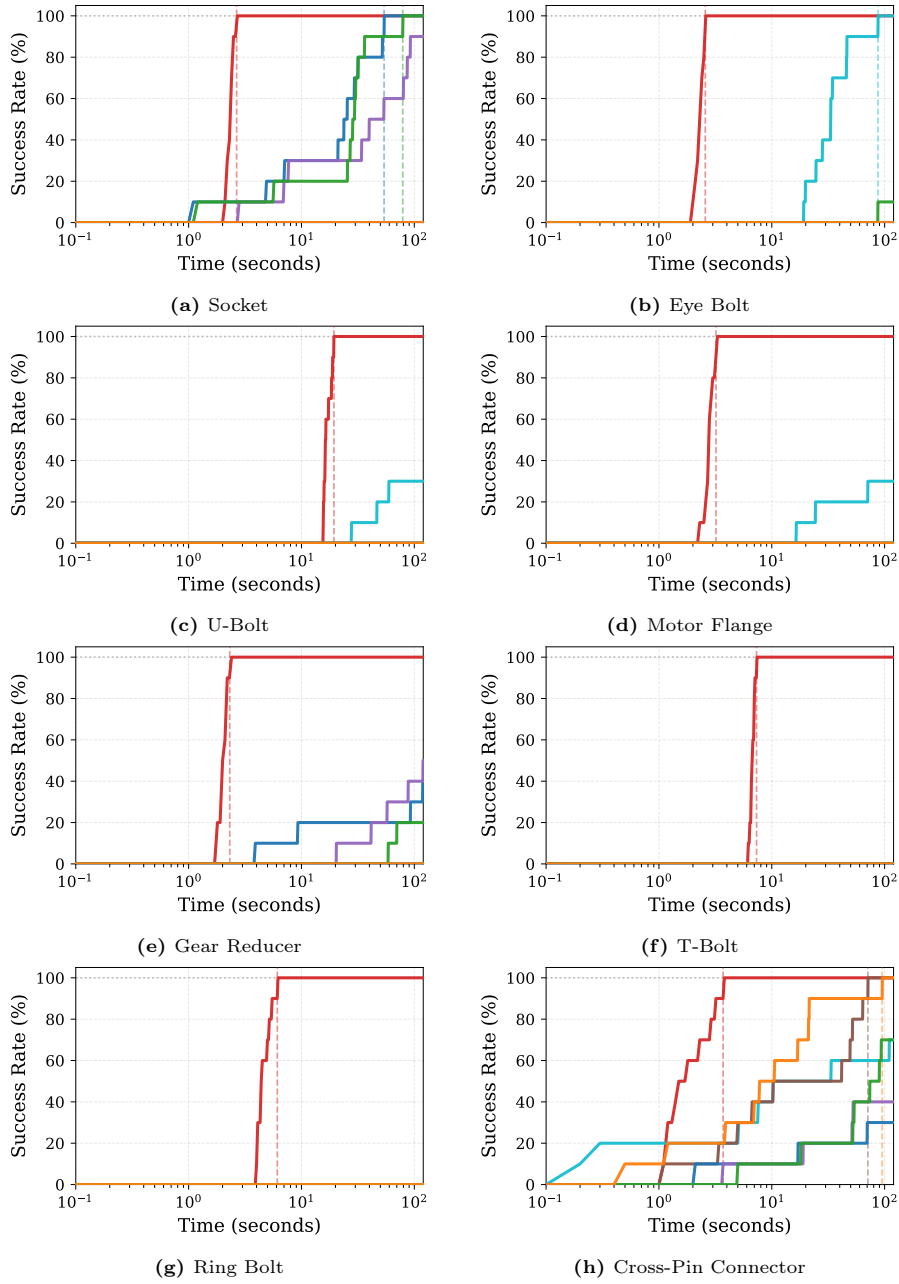
*Experimental Setup* All experiments were conducted on a laptop running Ubuntu 20.04.6 LTS with an Intel Core i7-5820K CPU (3.30GHz, 12 cores) and 32GB RAM. In each scenario, we compare MAB-RRT using scale-invariant sampling with six alternative sampling strategies. This includes two sets of planners. First a set of classical sampling strategies integrated into RRT [31], namely RRT + bridge sampling [24], RRT + Gaussian sampling [8], and RRT + obstacle-based sampling [2]. Second, a set of modern strategies, including mating vectors (MateVec-TRRT) [17], physical simulation using behavioral kinodynamic RRT (BK-RRT) [55,52], and physical simulation using similarity checks and disassembly breadth first search (BFS) [52]. For each scenario, we run each planner for a total of 10 runs with a timeout of 100 seconds.

*Hardware and Parameters* For MAB-RRT, we use the following parameter values. For scale-invariant sampling, we use a Fibonacci jitter  $p_{\text{fibonacci-jitter}} = \frac{\pi}{8}$ , initial



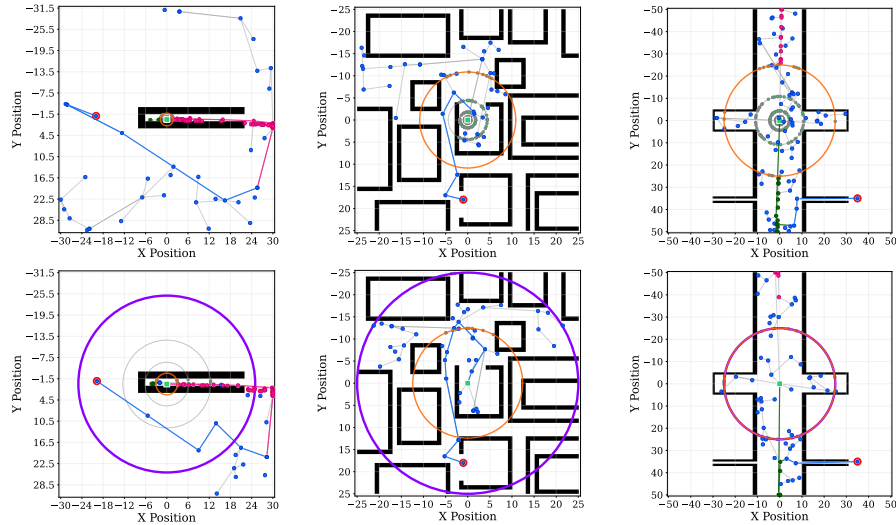
**Fig. 3:** Environment Comparison: Start vs. End states for selected environments. Each pair shows the initial (left) and final (right) configuration.

radius  $r_0 = 1.0$ , min radius  $r_{\min} = 1e-6$ , max radius  $r_{\max} = 25.0$ , batch size  $b = 64$ , growth factor  $g = \exp(-0.7)$ , shrink factor  $s = \exp(0.9)$ , max steps  $S = 50$ , and validity rates  $\alpha_{\min} = 0.1$  and  $\alpha_{\max} = 0.5$ . For the Multi-arm bandit, we use a sliding window of  $N_{\text{sliding-window}} = 256$ , an UCB exploration coefficient  $\beta = \sqrt{2}$ , and constants for the uniform rewards of  $c_u = 1e8$  and PCA samplers as  $c_s = 5.0$ . For the classical sampling strategies, we use the default param-



**Fig. 4:** Comparison of success rate using planners ■ MAB-RRT (ours), ■ RRT + Bridge Sampling [24], ■ RRT + Gaussian Sampling [8], ■ RRT + Obstacle Sampling [2], ■ MateVec-TRRT [17], ■ BK-RRT [55,52], ■ BFS [52],

ters as specified in OMPL [49,38]. For the modern strategies, we use the default parameters as specified by the Assemble-Them-All [52] software package <sup>2</sup>.



**Fig. 5:** Final sampling results across environments (radius set to  $1e-6$  vs.  $25.0$ ). Final radius from grow-shrink shown as —, initial radius as —. Tree edges shown as —. Valid samples: ■ uniform, ■ positive principal component, ■ negative principal component, ■ scale-invariant sampler. Start marked with ■, goal with ■.

To show the results, we plot the success rate from zero to one hundred percent over time in log scale, as shown in Fig. 4. It can be seen that MAB-RRT solves all eight scenarios with 100% success rate, meaning that MAB-RRT found a solution in every single run. In the Socket environment, MAB-RRT reaches 100 percent with over one order of magnitude (OoM) better runtime compared to the next best planners (RRT + Obstacle Sampling and RRT + Bridge Sampling). In the Eye Bolt scenario, MAB-RRT and BFS reach 100 percent success rate, with MAB-RRT outperforming BFS by over 1 OoM. In the U-Bolt environment, only MAB-RRT reaches 100 percent, while BFS only reaches 30 percent success rate until the timeout. In Motor Flange, the situation is similar with MAB-RRT reaching 100 percent and BFS reaching 30 percent. In Gear Reducer, only MAB-RRT reaches 100 percent, while planners RRT + Gaussian Sampling reaches 50 percent, RRT + Bridge Sampling reaches 40 percent, and RRT + Obstacle Sampling reaches 20 percent. In the T-Bolt and Ring Bolt scenarios, MAB-RRT is the only planner reaching 100 percent, each time with a runtime below  $10^1$ , which is 1 OoM below timeout. Finally, in Cross-Pin Connector, MAB-RRT, MateVec-TRRT, and BK-RRT reach 100, while BFS and RRT + Obstacle Sampling reach

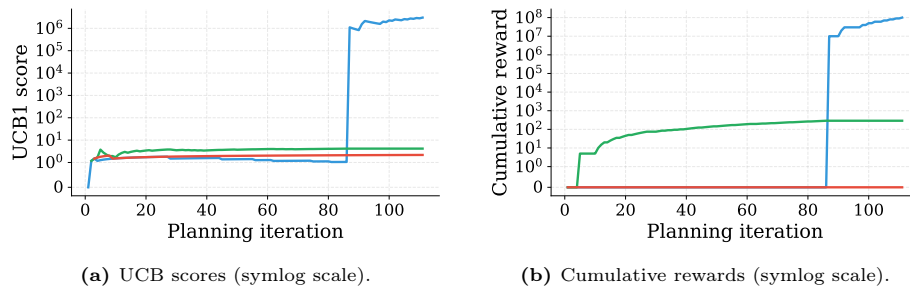
<sup>2</sup> Link: <https://github.com/yunshengtian/Assemble-Them-All>

70, RRT + Gaussian Sampling reaches 40, and RRT + Bridge Sampling reaches 30 percent. Again MAB-RRT outperforms the next best planner (BK-RRT) by 1 OoM. In terms of runtime, MAB-RRT outperforms the next best planner or the timeout by at least 1 OoM on seven out of eight scenarios, with the exception of the U-Bolt scenario.

#### 4.1 Multi-Arm Bandit Dynamics

To verify that the multi-arm bandit correctly changes between arms over time, we run it on the T-Bolt experiment from Fig. 3. We plot both the UCB scores over time and the cumulative rewards per arm. This is depicted in Fig. 6. We can see that the cumulative rewards stays zero for uniform sampling, capturing the fact that uniform samples lead to direct collision. The reward for PCA positive is growing over time which reflects the fact that the robot can escape into only one direction from the tunnel. Once the robot has escaped (around iteration 85), there is a sharp uptick in reward for the uniform sampler, which reflects the growing of the tree into the open, free space.

The UCB scores closely follow this trend. In the beginning, there is a sharp decrease in UCB score for uniform sampling, reflecting the invalidity of the samples. Over time, the UCB scores for uniform and PCA negative tend towards similar values ( $\sim 1.0$ – $2.0$ ), so that uniform samples are still occasionally tried to verify that the tree has not yet reached the open space. Finally, at iteration 85, the UCB score has a sharp increase reflecting the switch to the uniform sampling scheme.



**Fig. 6:** T-Bolt Experiment: MAB dynamics during planning. (a) UCB scores adjust dynamically as the planner gathers reward information. (b) Rapid reward accumulation for cylinder arms indicates their effectiveness in generating valid samples within the narrow passage, while the uniform arm’s reward surges once free space is reached. ■ Uniform, ■ PCA positive, ■ PCA negative.

#### 4.2 Robustness of Scale-Invariant Sampling

Another important aspect of our framework is the grow-shrink algorithm to find a useful high-information entropy scale. In this section, we like to show that

this algorithm is robust against different initial radii. For this, we created three environments as depicted in Fig. 5. Each row shows one environment, whereby the left image shows a starting radius of  $1e-6$  and the right image shows an initial radius of 25. For each scenario, we showcase the initial radius (violet circle) and the final radius after applying grow-shrink (orange circle). It can be seen that both initial radii faithfully converge to a similar radius at which roughly half of the samples are reachable. We also showcase the resulting trees with start (light green) and goal (red), where we show valid samples from uniform sampling (blue), samples from the principal component sampler with positive (green) and negative (magenta) direction, and initial samples from grow-shrink (light green).

## 5 Discussion and Conclusion

We presented scale-invariant sampling, a novel sampling strategy for object extraction from tight, narrow passages. Our sampling strategy is based upon two methods. First, we explore the local space around an object’s start position to find a scale where the information entropy is largest. This scale is maximally useful to find valuable samples. By using a grow-shrink method, we can robustly and quickly find such a scale. Second, we exploit the computed scale by leveraging the principal direction in the data, thereby finding a sampling bias which is helpful to robustly extract objects by finding the right directions inside a narrow passage. Finally, the samplers are integrated into a multi-arm bandit RRT (MAB-RRT) [19] which switches between different sampling strategies depending on the rewards obtained.

The results show that scale-invariant MAB-RRT can successfully solve complex disassembly tasks. This involves sockets, gears, multiple bolt types, and connector pins. In our experiments, scale-invariant MAB-RRT outperforms similar biased sampling approaches and is the only planner which can reach a 100% success rate on all scenarios.

While the results are promising, there are still some open questions. First, while PCA could be used for arbitrary spaces and objects, we have not yet demonstrated this. It would be important to verify that this works also on challenging scenarios where objects have to be rotated (e.g. screws, keys), where longer narrow passages are present (e.g. pipes, rods, axles), where objects are deformable (e.g. pulling out cables), or where narrow passages are non-linear (e.g. buzz wire game [16]). Second, we believe it would be important to analyze the convergence properties of the high information entropy scale search. While the planners work robustly, there is currently no guarantee that it will converge in every situation. Third, it is important to integrate our algorithm into a larger system for disassembly tasks [51], so that we can show that this can be used in combination with a real robot for complex object extraction tasks.

Despite limitations, scale-invariant MAB-RRT is a probabilistically complete and efficient planner which works reliably for complex object extraction tasks. We believe it is therefore an important component of a larger task and motion planning system [6] for automatic object disassembly.

## References

1. Aguinaga, I., Borro, D., Matey, L.: Parallel rrt-based path planning for selective disassembly planning. *The International Journal of Advanced Manufacturing Technology* **36**(11), 1221–1233 (2008)
2. Amato, N.M., Bayazit, O.B., Dale, L.K., Jones, C., Vallejo, D.: OBPRM: An obstacle-based PRM for 3D workspaces. In: Agarwal, P.K., Kavraki, L.E., Mason, M.T. (eds.) *Robotics: The Algorithmic Perspective*, pp. 155–168. CRC Press (1998)
3. Asif, M.E., Rastegarpanah, A., Stolkin, R.: Robotic disassembly for end-of-life products focusing on task and motion planning: A comprehensive survey. *Journal of Manufacturing Systems* **77**, 483–524 (2024)
4. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. *Machine learning* **47**(2), 235–256 (2002)
5. Axler, S.: *Linear algebra done right*. Springer Nature (2024)
6. Bayraktar, S.B., Orthey, A., Kingston, Z., Toussaint, M., Kavraki, L.E.: Solving rearrangement puzzles using path defragmentation in factored state spaces. *IEEE Robotics and Automation Letters* **8**(8), 4529–4536 (2023)
7. Bishop, C.M., Nasrabadi, N.M.: *Pattern recognition and machine learning*, vol. 4. Springer (2006)
8. Boor, V., Overmars, M.H., Van Der Stappen, A.F.: The gaussian sampling strategy for probabilistic roadmap planners. In: *IEEE International Conference on Robotics and Automation*. vol. 2, pp. 1018–1023 (1999)
9. Bubeck, S., Cesa-Bianchi, N., et al.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* **5**(1), 1–122 (2012)
10. Burns, B., Brock, O.: Toward optimal configuration space sampling. In: *Robotics: Science and Systems*. vol. 1, pp. 105–112 (2005)
11. Cai, P., Yue, X., Zhang, H.: Add-rrv for motion planning in complex environments. *Robotica* **40**(1), 136–153 (2022)
12. Chamzas, C., Kingston, Z., Quintero-Peña, C., Shrivastava, A., Kavraki, L.E.: Learning sampling distributions using local 3d workspace decompositions for motion planning in high dimensions. In: *IEEE International Conference on Robotics and Automation*. pp. 1283–1289 (2021)
13. Dalibard, S., Laumond, J.P.: Control of probabilistic diffusion in motion planning. In: *Algorithmic Foundation of Robotics VIII*. pp. 467–481. Springer (2009)
14. Dalibard, S., Laumond, J.P.: Linear dimensionality reduction in random motion planning. *The International Journal of Robotics Research* **30**(12), 1461–1476 (2011)
15. Das, A.R., Koskinopoulou, M.: Towards sustainable manufacturing: A review on innovations in robotic assembly and disassembly. *IEEE Access* (2025)
16. Dorussen, R.: Learning how to solve the buzz-wire game with a robot arm. Ph.D. thesis, Msc Thesis Report, Eindhoven University of Technology (2021)
17. Ebinger, T., Kaden, S., Thomas, S., Andre, R., Amato, N., Thomas, U.: A general and flexible search framework for disassembly planning. In: *IEEE International Conference on Robotics and Automation*. pp. 3548–3555 (Sep 2018)
18. Eppner, C., Brock, O.: Visual detection of opportunities to exploit contact in grasping using contextual multi-armed bandits. In: *IEEE International Conference on Intelligent Robots and Systems*. pp. 273–278 (2017)

19. Faroni, M., Berenson, D.: Motion planning as online learning: A multi-armed bandit approach to kinodynamic sampling-based planning. *IEEE Robotics and Automation Letters* **8**(10), 6651–6658 (2023)
20. Faroni, M., Berenson, D.: Online adaptation of sampling-based motion planning with inaccurate models. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2382–2388. IEEE (2024)
21. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Informed rrt\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In: *IEEE International Conference on Intelligent Robots and Systems*. pp. 2997–3004 (2014)
22. Garivier, A., Moulines, É.: On upper-confidence bound policies for non-stationary bandit problems. *International Conference on Algorithmic Learning Theory* (May 2008)
23. Guennebaud, G., Jacob, B., et al.: Eigen. <https://libeigen.gitlab.io> (2010)
24. Hsu, D., Jiang, T., Reif, J., Sun, Z.: The bridge test for sampling narrow passages with probabilistic roadmap planners. In: *IEEE International Conference on Robotics and Automation*. vol. 3, pp. 4420–4426 (2003)
25. Huang, L., Meng, L., Jing, X., Wang, J.: Selective densification for rapid motion planning in high dimensions with narrow passages. *IEEE Transactions on Automation Science and Engineering* (2025)
26. Ichter, B., Harrison, J., Pavone, M.: Learning sampling distributions for robot motion planning. In: *IEEE International Conference on Robotics and Automation*. pp. 7087–7094. IEEE (2018)
27. Jaynes, E.T.: *Probability theory: The logic of science*. Cambridge university press (2003)
28. Jolliffe, I.: Principal component analysis. In: *International encyclopedia of statistical science*, pp. 1094–1096. Springer (2011)
29. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. *The international journal of robotics research* **30**(7), 846–894 (2011)
30. Koval, M.C., King, J.E., Pollard, N.S., Srinivasa, S.S.: Robust trajectory selection for rearrangement planning as a multi-armed bandit problem. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. pp. 2678–2685 (2015)
31. Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: *IEEE International Conference on Robotics and Automation*. vol. 2, pp. 995–1001 (2000)
32. Lai, T., Ramos, F.: Rapidly-exploring random forest: Adaptively exploits local structure with generalised multi-trees motion planning. *IEEE Robotics and Automation Letters* **7**(1), 323–330 (2022)
33. Laili, Y., Wang, Y., Fang, Y., Pham, D.T.: *Optimisation of robotic disassembly for remanufacturing*. Springer (2022)
34. LaValle, S.M.: *Planning algorithms*. Cambridge university press (2006)
35. Lee, J., Kwon, O., Zhang, L., Yoon, S.e.: Sr-rrt: Selective retraction-based rrt planner. In: *IEEE International Conference on Robotics and Automation*. pp. 2543–2550 (2012)
36. Li, R., Pham, D.T., Huang, J., Tan, Y., Qu, M., Wang, Y., Kerin, M., Jiang, K., Su, S., Ji, C., et al.: Unfastening of hexagonal headed screws by a collaborative robot. *IEEE Transactions on Automation Science and Engineering* **17**(3), 1455–1468 (2020)
37. Li, Y., Bekris, K.E.: Balancing state-space coverage in planning with dynamics. In: *IEEE International Conference on Robotics and Automation*. pp. 3246–3253 (2010)

38. Moll, M., Şucan, I.A., Kavraki, L.E.: Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization. *IEEE Robotics & Automation Magazine* **22**(3), 96–102 (September 2015)
39. Motoda, T., Petit, D., Nishi, T., Nagata, K., Wan, W., Harada, K.: Multi-step object extraction planning from clutter based on support relations. *IEEE Access* **11**, 45129–45139 (2023)
40. Niederreiter, H.: *Random Number Generation and Quasi-Monte-Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia (1992)
41. Orthey, A., Chamzas, C., Kavraki, L.E.: Sampling-based motion planning: A comparative review. *Annual Review of Control, Robotics, and Autonomous Systems* **7**(1), 285–310 (2024)
42. Parker, L.E., Draper, J.V.: Robotics applications in maintenance and repair. *Handbook of industrial robotics* **2**, 1023–1036 (1998)
43. Pathak, A., Muthusamy, R.: Collapse and collision aware grasping for cluttered shelf picking. *arXiv preprint arXiv:2503.22427* (2025)
44. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* **2**(11), 559–572 (1901)
45. Rickert, M., Sieverling, A., Brock, O.: Balancing exploration and exploitation in sampling-based motion planning. *IEEE Transactions on Robotics* **30**(6), 1305–1317 (2014)
46. Rosell, J., Suárez, R., Pérez, A.: Path planning for grasping operations using an adaptive pca-based sampling method. *Autonomous Robots* **35**(1), 27–36 (2013)
47. Serrano-Munoz, A., Arana-Arexolaleiba, N., Chrysostomou, D., Bøgh, S.: Learning and generalising object extraction skill for contact-rich disassembly tasks: an introductory study. *The International Journal of Advanced Manufacturing Technology* **124**(9), 3171–3183 (2023)
48. Slivkins, A., et al.: Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning* **12**(1-2), 1–286 (2019)
49. Şucan, I.A., Moll, M., Kavraki, L.E.: The Open Motion Planning Library. *IEEE Robotics & Automation Magazine* **19**(4), 72–82 (December 2012)
50. Teodoro, M.L., Phillips Jr, G.N., Kavraki, L.E.: Understanding protein flexibility through dimensionality reduction. *Journal of Computational Biology* **10**(3-4), 617–634 (2003)
51. Tian, Y., Jacob, J., Huang, Y., Zhao, J., Gu, E.L., Ma, P., Zhang, A., Javid, F., Romero, B., Chitta, S., Sueda, S., Li, H., Matusik, W.: Fabrica: Dual-arm assembly of general multi-part objects via integrated planning and learning. In: *Conference on Robot Learning* (2025)
52. Tian, Y., Xu, J., Li, Y., Luo, J., Sueda, S., Li, H., Willis, K.D., Matusik, W.: Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Trans. Graph.* **41**(6) (2022)
53. Yershova, A., Jaillet, L., Siméon, T., LaValle, S.M.: Dynamic-domain rrtts: Efficient exploration by controlling the sampling domain. In: *IEEE International Conference on Intelligent Robots and Systems*. pp. 3856–3861 (2005)
54. Zhang, X., Eltouny, K., Liang, X., Behdad, S.: Automatic screw detection and tool recommendation system for robotic disassembly. *Journal of Manufacturing Science and Engineering* **145**(3), 031008 (2023)
55. Zickler, S., Veloso, M.M.: Efficient physics-based planning: sampling search via non-deterministic tactics and skills. In: *AAMAS* (1). pp. 27–33 (2009)