Automating Deformable Gasket Assembly

Simeon Adebola*[1], Tara Sadjadpour*[1], Karim El-Refai*[1], Will Panitch[1], Zehan Ma[1],
Roy Lin[1], Tianshuang Qiu[1], Shreya Ganti[1], Charlotte Le [1], Jaimyn Drake [1], and Ken Goldberg[1]

***Abstract*— In Gasket Assembly, a deformable gasket must be aligned and pressed into a narrow channel. This task is common for sealing surfaces in the manufacturing of automobiles, appliances, electronics, and other products. Gasket Assembly is a long-horizon, high-precision task and the gasket must align with the channel and be fully pressed in to achieve a secure fit. To compare approaches, we present 4 methods for Gasket Assembly: one policy from deep imitation learning and three procedural algorithms. We evaluate these methods with 100 physical trials. Results suggest that the Binary+ algorithm succeeds in 10/10 on the straight channel whereas the learned policy based on 250 human teleoperated demonstrations succeeds in 8/10 trials and is significantly slower. Code, CAD models, videos, and data can be found at https://berkeleyautomation.github.io/robot-gasket/.**

I. Introduction

Tasks such as clothes folding, thread untangling, and cable tracing have applications in manufacturing, logistics, and domestic applications, but present unique challenges for robots due to the complex physics and large configuration spaces of deformable objects. One such deformable manipulation task is the assembly of *gaskets*, deformable components that fill the space between two or more mating surfaces to provide a seal, generally to prevent leakage from or into the joined objects while under compression [1]. Gaskets compensate for small gaps or imperfections in mating surfaces and play critical roles in industries such as automotive and appliance manufacturing (where they are essential for sealing windows, engines, and fuel systems), plumbing, power generation, and construction. Almost all Gasket Assembly today is performed by humans.

We propose a robot gasket assembly task where a robot must pick and place the gasket, a 1D deformable object, into a channel of similar length and width, such that the gasket snugly and completely fills the entire channel. The setup can be easily and reliably replicated using common materials and a 3D printer with the provided CAD model. The task requires pick-and-place, press, and slide moves to be performed in succession. This is a high-precision, long-horizon task with a large state space, complicated dynamics, and low error tolerances[2].

Gasket Assembly qualifies as a dull/dreary repetitive task [3] and is thus a candidate for robotic automation with the goal of reducing worker fatigue and cost as a result of the automation. We propose four approaches to automating the gasket assembly task: one learned implementation and three procedural. The procedural approaches differ primarily in the strategy that each uses to choose and order the pick-and place and press motions on the gasket, while the learned end-to-end approach utilizes Diffusion Policy [4] to learn a gasket insertion policy from 250 human demonstrations.

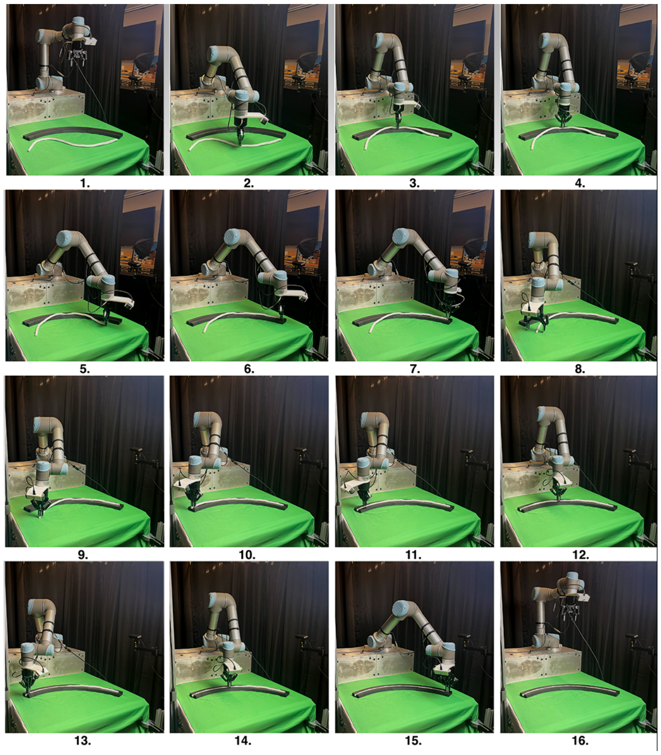

Fig. 1: **Gasket Assembly Example.** This is an example of the Binary+ procedural algorithm rollout on the curved channel. Initially, the gasket is separate from the channel (1). The robot picks the gasket's midpoint (2), places it into the channel's midpoint (3), and presses down to insert (4). Then an endpoint of the gasket is picked, placed, and pressed into the endpoint of the channel (5,6,7); the same is done with the second gasket endpoint (8,9,10,11). We omit the remaining pick-place-press steps at the quarter and eighth points, as well as the second press that occurs at each point for reinforcing insertion. Finally, the gripper returns to the middle of the channel (12) and slides from the middle to one end (13). It returns to the middle and slides to the other endpoint (14,15). The end result is a successfully assembled gasket (16).

This paper makes the following contributions:

1) A novel formulation of a gasket assembly task that is easy to replicate.
2) A deep imitation learning policy generated from 250 human-teleoperated demonstrations.
3) Three analytic/procedural algorithms for the same task.
4) Experimental results comparing the learned policy and the procedural algorithms based on 100 physical trials that suggests procedural algorithms can be superior in

*Equal Contribution

[1]The AUTOLab at UC Berkeley (automation.berkeley.edu) {simeon.adebola, goldberg} @berkeley.edu

terms of performance.

II. Related Work

A. Deformable Linear Object Perception and Manipulation

Deformable objects can be categorized into three distinct types based on their dimensions: 1D objects (e.g., ropes, cables, strings, and electrical cords [2, 5–7]), 2D objects (e.g., fabrics and clothing [8]), and 3D objects (e.g., bags [9–11]). In this paper, we focus on the 1D class of deformable objects, also known as deformable linear objects (DLOs), since they best represent gaskets. We use the terms "DLO" and "gasket" interchangeably in this paper.

There have been recent advancements in robotics for a variety of manipulation tasks involving DLOs, such as knot tying [6], rope untangling [2], and the production of wire harnesses [7]. Nonetheless, DLO perception and manipulation remain open research areas. In this paper, we explore the assembly of gaskets, expanding upon the current scope of robotic manipulation for DLOs.

The typical approach for handling DLOs involves a three-step process: state estimation, path planning, and manipulation to achieve the desired outcome [12, 13]. Achieving the target state of a DLO can be complex, necessitating path planning strategies that can adapt to and recover from sub-optimal actions. A wide range of methods have been explored to solve different aspects of this problem, including Coherent Point Drift for DLO state estimation [14], learning from demonstrations for DLO manipulation [15–19], and self-supervised learning for understanding the state-space of DLOs [20]. In this work, we present 4 baselines that encompass a variety of methods seen in these prior works. The analytical algorithms use a computer vision algorithm to detect the channel and gasket, and primitives to help the robot recover from sub-optimal actions taken in prior steps. The learned baseline learns from human demonstrations using diffusion policy [4].

B. Cable Routing Tasks with DLOs

A common task category for 1D deformable manipulation is routing, in which a DLO is manipulated to match a given set of position and shape constraints.

Recent research has explored a range of techniques and approaches for "cable routing," a process in which a cable is guided along a path by a series of unconnected fixtures [21–25]. Some studies have focused specifically on the path planning aspect of routing, aiming to navigate the cable from its starting position to a designated endpoint [13]. Another rendition of this task involves guiding a string or rope through openings that are only slightly larger than the material itself, with a minimal clearance of 1.4mm [26]. In contrast with these scenarios that involve threading a rope through a series of openings or guiding cables along a path, this paper focuses on the precise challenge of snugly fitting a gasket into a specifically shaped channel. This task demands a greater level of accuracy in both perception and manipulation, highlighting the need for advanced techniques capable of handling the intricacies of secure gasket insertion.

C. Insertion Tasks

Robot manipulation has been applied to a number of insertion tasks involving both rigid and non-rigid objects. The most classic example is the peg-hole insertion task. Insertion tasks have been widely explored using learning from demonstrations [27–29], reinforcement learning [30–33], regression[34] and multimodal perception [35], where the peg and insertion hole are rigid. Another work on insertion [36] uses visual and tactile sensors and force-torque sensing together with self-supervised learning to achieve policies that allow a robot to insert a USB connector in an industrial task setting. [37] uses reinforcement learning techniques to learn how to insert a rigid peg into a deformable hole. Peg-hole insertion has recently been applied to medical settings in [38], which studies techniques to insert a flexible needle into a deformable foam. These iterations of the peg-hole insertion task differ from gasket assembly which requires the gasket to deform into the rigid channel.

Pirozzi and Natale [39] focuses on wire insertion, where a robot gripper informed by signals from its tactile sensor inserts a wire into a hole as part of a switchgear assembly.

D. Robot Policy Learning for Long-Horizon Tasks

In long-horizon sequential manipulation tasks, earlier actions affect the feasibility of subsequent actions [40, 41]. Recent approaches include the use of imitation and reinforcement learning [42–44], language-conditioned policy learning [45], self-supervised learning [46] and learning from human demonstrations[47]. Gasket Assembly is a long-horizon task with subtasks that include picking, placing, and insertion.

E. Deep Imitation Learning and Diffusion Policy

Deep imitation learning from human teleoperated demonstrations is an active area of research. Diffusion Policy [4], motivated by the powerful generative modeling capabilities of diffusion models [48, 49], was recently proposed to represent a robot's visuomotor policy as a conditional denoising diffusion process. It learns the gradient of the action-distribution score function during training and iteratively performs a series of stochastic Langevin dynamics steps during inference. Specifically, starting from $\mathbf{a}^K$ sampled from Gaussian noise, the Denoising Diffusion Probabilistic Model (DDPM) performs K iterations of denoising to produce intermediate vectors with decreasing levels of noise, $\mathbf{a}^K, \mathbf{a}^{K-1}, ..., \mathbf{a}^0$, until a desired level of noise is obtained. Mathematically, to learn the conditional distribution $p(\mathbf{a}_t|\mathbf{o}_t)$, where $\mathbf{o}_t$ is the observation of the current step and $\mathbf{a}_t$ is the desired action output, we use a conditional CNN $\epsilon_\theta(\mathbf{a}^k, k|\mathbf{o})$ to get

$$\mathbf{a}^{k-1} = \alpha(\mathbf{a}^k - \gamma\epsilon_\theta(\mathbf{a}^k, k, \mathbf{o}) + \mathcal{N}(0, \sigma^2 I)),$$

where α, γ, σ are noise schedule hyperparameters and functions of the iteration step k. During training, we minimize

$$\mathcal{L} = MSE(\epsilon^k, \epsilon_\theta(\mathbf{a}^k, k, \mathbf{o})),$$

where ϵ^k is a random noise with appropriate variance.

III. The Gasket Assembly Problem

We propose a problem where an automated system must reliably insert a deformable gasket into a rigid channel of predefined shape and length. Both the channel and the gasket are continuous. In this paper, we consider gaskets with circular cross-section. We propose two human evaluation metrics: (1) Alignment: How well does the gasket align with the target shape of the channel and (2) Insertion: How much of the gasket is contained within the channel.

As shown in Figure 2, given an RGB image of the workspace, a gasket of fixed length l_g and circular cross sectional diameter d, and a channel of width w, length l_c, and depth h which is equal to at least d, we attempt to insert the gasket into the channel such that the gasket is completely contained within the channel. We assume that $l_g \approx l_c$ and $w \leq d \leq w+\delta$, where δ is a deformation constant determined by the cross-sectional compressability of the gasket.

We denote the endpoints of the gasket as g_0 and g_1, and select two points in the channel, which we denote as c_0 and c_1. When the channel is open-ended (see Section V-B and Figure 2 A,B) g_0 and g_1 correspond to the two endpoints of the channel; however, when the channel is closed (Figure 2 C), g_0 and g_1 correspond instead to adjacent points in the channel such that the inserted gasket, taking the *longer* path between the two points, creates a closed loop.

We assume a planar work surface of known dimension, a six-axis robot arm with a parallel jaw gripper, and 3 RGB cameras with known intrinsic and extrinsic calibration matrices affixed at 1) the wrist of the robot, 2) above, and 3) to the side of the workspace. For the learned policy in addition to the 3 RGB cameras, we assume a human teleoperation input system such as a VR (Meta Quest2) controller or a 3D Mouse (SpaceMouse) or GELLO [50]. We additionally constrain the problem space by assuming access to a priori knowledge of the shape and dimension of given channels, a non-adversarial (no sharp corners, knots, or crossings) gasket starting configuration located within the reachable workspace of the robot and the camera's field of view, and that the gasket and the channel can be easily color segmented from the workspace.

IV. Four Baseline Methods

A. Learning Diffusion Policy

Deep imitation learning where an agent learns required skills using deep neural networks trained on human demonstrations are being considered for a range of manipulation tasks including grasping, pressing, and pick and place [51]. Various approaches have been used for deep imitation learning including Behavior Cloning, Generative Adversarial Imitation Learning (GAIL), Adversarial Reward-moment Imitation Learning (AdRIL) and recently, diffusion policies[4, 52, 53], used in this paper.

The learned policy uses two ZED 2 stereo depth cameras and a Logitech BRIO webcam. One ZED 2 is mounted above the workspace and the other ZED 2 is mounted to the left of the workspace, with the UR5 and workspace fully in view. The Logitech webcam is mounted on the wrist of the UR5. A human operator teleoperates the robot using a 3D mouse. We use the Gello codebase [50] for teleoperation.

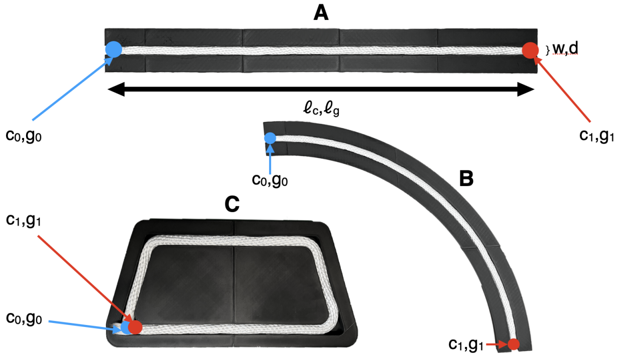

Fig. 2: **Channels and Gaskets in Goal Positions.** Each channel has a gasket fully inserted. The straight channel (A) and the curved channel (B) are both open-ended channels whereas the trapezoid channel (C) is closed. This means that for all channels, the gasket endpoints (g_0, g_1) and channel endpoints (c_0, c_1) lie nearly on top of each other, but in the trapezoid case, c_0 and c_1 also lie nearly on top of each other.

A team of co-authors patiently perform the task 250 times using teleoperation and record all images and joint angles for training. This required approximately 15 hours of human effort. Human demonstrations are collected as described on the website. The strategy used for the human demonstrations is closest to the Binary+ algorithm described in IV-B.3. As shown in Section VI, Binary+ is the best-performing of the three algorithms. We did this so we could properly compare the learned policy with the algorithms. Additionally, we focus on just the straight channel which is the simplest channel and collect all 250 demonstrations for that channel.

We use a CNN-based model architecture for Diffusion Policy with an action prediction horizon of 16 and observation history length of 2. $\epsilon_\theta(.)$ takes in the noisy action $\mathbf{a}^k_{t,t+1,\ldots,t+15}$ and uses 36 1D convolutional layers with FiLM conditioning [54] on the observation embeddings [4]. For training, we set $K = 100$ and use an initial learning rate of 0.0001 with a decay factor of 0.1 and we measure validation loss using Mean Squared Error(MSE). During execution, we use receding-horizon control with horizon 8. We use a Denoising Diffusion Implicit Model (DDIM) [55] with 10 inference steps. These hyperparameter choices are from the default values as provided in [56]. The result is a policy that takes as input the camera images and generates as output control actions and gripper position.

B. 3 Procedural Algorithms

1) Perception: The perception system aims to detect, separate, and localize the channel and gasket given an RGB image from an overhead workspace camera. Since the channel shape is fixed, we use a template matching algorithm for channel localization and identification. To enable this, we generate a ground truth binary mask and aspect ratio for each channel by using the CAD source files for each part. These ground truth masks are passed to the perception pipeline and used in the classification and planning steps. We detail the approach:

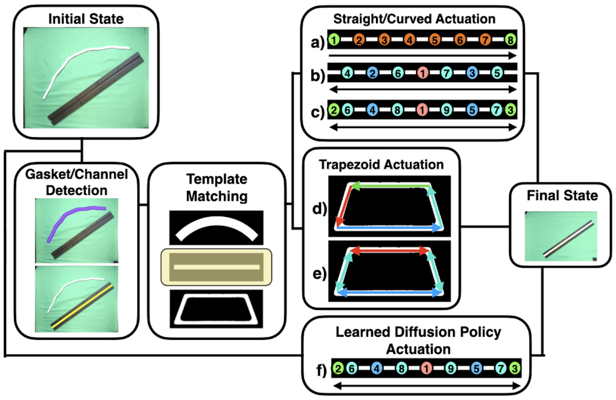

Fig. 3: **Methods.** The Gasket/Channel Detection box shows gasket segmentation (above) and channel segmentation (below). The Template Matching box shows the three templates for the curved, straight and trapezoid channel. The Straight/Curved Actuation box shows selection and actuation strategies for the straight and curved channels: (a) is Unidirectional insertion, (b) is Binary search insertion, and (c) is Binary+ insertion. The colors on the channels represent the locations the robot attempts to place and press the gasket into while the numbers represent the order they are placed and pressed. Endpoints are green, midpoints are pink, half-points are blue and the quartile-points are cyan. The arrows indicate the direction(s) of the slide(s). For the trapezoid channel, we treat each segment of the trapezoid as an instance of the straight channel. In the unidirectional approach (d) we process each segment in a counterclockwise manner, starting at the blue segment. For hybrid and binary (e), we evaluate the blue segment, then the cyan segments, and finally the red segment. The learned policy proceeds directly from the initial state to actuation (f). The Final State box shows the final assembled gasket.

(i) **Segmentation and Classification:** Given an RGB image of the scene, we apply a Gaussian smoothing operator with a 5×5 kernel to reduce high-frequency sensor noise. Then, we threshold the image to have pixel values within $[100, 255]$ to minimize the impact of the background color and apply a Canny edge detector. With the mask of edges in the scene, we use the Ramer–Douglas–Peucker algorithm to get contours for both the gasket and the mold containing the channel. Given these contours, we extract the smallest rectangle that encloses the input 2D point set, which can be treated as a detection bounding box defined by the box's center, dimensions, and rotation angle with respect to the x-axis. We find the aspect ratio of each detected bounding box and compare it to those of the given templates to classify the channel.

(ii) **Localization via Alignment:** Given the detection bounding box information and inferred channel, the template is rescaled to match the dimensions of the observed bounding box, translated to align its center to that of the bounding box, and rotated to match the bounding box's rotation with respect to the x-axis. Once the template mask and channel are aligned, we can precisely localize the points in pixel space where the insertion channel exists in the physical channel.

(iii) **Skeletonization:** We skeletonize the localized channel to obtain a one-pixel-wide skeleton along the center of the channel. We then perform depth-first-search starting from an arbitrary pixel along this skeleton to get the pixels corresponding to the endpoints of the channel and order all points in the channel relative to these endpoints. This ordered list is necessary to relate pixels to their positions on the channel.

(iv) **Waypoint Selection:** Using the ordered gasket skeleton, we sample a set of points between the two endpoints according to one of the insertion strategies (enumerated in section IV-B.3). These points are used for the pick-place-press actions.

This perception pipeline runs after every step to provide an up-to-date estimate of the workspace state.

2) Robot Primitives: We define a small set of versatile primitives to enable efficient environment interaction.

(i) Pick and Place: The robot moves just above a point with grippers open, then descends to the height of the work surface and closes the grippers. The robot then rises, moves to a target point and opens the grippers.

(ii) Shift and Place: In this variant of pick and pace, the robot slides along the surface until it reaches the target point to avoid lifting any adjacent gasket insertions.

(iii) Press: The robot moves downward from above a target pose until it experiences an upward force above a given threshold (in our case, 40 N as measured by the robot's torque sensors).

(iv) Slide: The robot moves sideways toward a target point across the work surface, while also pressing downward with a small amount of force (20 N).

(v) Home: The robot returns to a position above the workspace to afford an unobstructed view of the surface.

3) 3 Insertion Algorithms: **Unidirectional** insertion, **binary search** insertion, and **Binary+**. These methods differ in the points selected for alignment, as well as the order in which the robot actuates to those points. (See Figure 3)

(i) Unidirectional insertion (Fig. 3a): The robot picks, places, and inserts the gasket into the channel, starting at one end of the gasket and progressing toward the other end. Then, the robot presses each selected point of the gasket into the channel for a second time to reinforce the insertion. Finally, the robot slides its gripper along the entire length of the channel to seal the gasket.

(ii) Binary search insertion (Fig. 3b): The robot begins by picking and placing the midpoint of the gasket, followed by the points located at $\frac{1}{4}$ and $\frac{3}{4}$ of the gasket length, followed by those at the eighths, and so on until the algorithm reaches a termination limit (we set $\frac{1}{8}$ for the straight and curved channels, and $\frac{1}{4}$ for the trapezoid channel). Then, the robot presses each selected point of the gasket into the channel in the same order as they were picked and placed to reinforce the insertion. Lastly, the robot performs a "binary slide" by starting at the midpoint and sliding once toward each end.

(iii) Binary+ insertion (Fig. 3c): This method attempts to combine the advantages of the unidirectional and binary search approaches. The robot begins by picking, placing, and inserting the midpoint of the gasket, followed by the endpoints. The robot then places the quarter and eighth points, as in the binary method. Then, the second reinforcing presses follow the unidirectional method.

Finally, the binary slide is performed.

V. Physical Experiments

A. Workspace Setup

We utilize a Universal Robots UR5 to conduct experiments. The UR5 is a 6-degree of freedom manipulator which can be operated in either position- or force-control modes to enable pressing and sliding motions. The work surface is aligned with the XY-plane and covered with a green tablecloth to make it perceptually uniform. The workspace is surrounded by a series of black drapes to control lighting and eliminate distractors. We found that the drapes were invaluable to the template matching. Additionally, we mount a front camera level with the workspace on the far side of the table from the robot. This camera is only used for evaluating insertion. A red backdrop is also used to increase contrast between the gasket and channel when viewed from the front camera for evaluation, and is not present during execution.

The workspace is observed by three cameras during training and execution. The overhead camera is placed 97.1 cm above the center of the workspace and pointed downward so that it is able to observe the entire workspace. This is the only camera used during the execution of the procedural trials. The "side" camera is placed 40.3 cm outside the workspace on the left side and 32.7 cm above it, positioned such that the camera is able to see the entire workspace along with the robot. The "wrist" camera is mounted 16.0 cm above the end-effector on a fixed "handle" such that it observes the movement of the gripper along with the section of the workspace near the robot end-effector.

B. Channels and Gasket

We consider 3 channels in increasing order of difficulty, as shown in Fig. 2. Fig. 2A: The first channel is an open straight channel with dimensions 26.5" x 2.68" x 0.56". Fig. 2B: The second channel is an open curved strut channel covering a 90° arc of a circle, with inner diameter 32.4", outer diameter 35.1", and height 0.75". This results in a channel with an arc length of 26.5", a width of 2.68", and a height of 0.75", which is analogous to the dimensions of the straight channel. Fig. 2C: The third channel is a closed trapezoidal channel with a long side of 10", a short side of 7.5", and two 4.5"-long non-parallel sides. All channels have an inner channel width of 0.5". All channels are 3D printed from Black PLA using a Bambu Lab P1S FDM 3D printer. We use a white 0.5" braided nylon rope as our gasket analogue as it is deformable enough to meet the gasket-channel constraints as discussed in Section III. The rope is cut to a length of 26.5" to precisely match the length of the testbed channels. Figure 2 shows the channels and gasket in the goal condition.

C. 100 Gasket Assembly Trials

We only trained the demonstrations on the straight channel in a fixed position with various gasket initializations both above and below the channel. Thus, during testing, we only had the channel in the same fixed position. For Trials 1-10, i.e., for the learned policy, the channel is at a fixed pose of 0°. For Trials 11-100, the position and orientation of the channel are randomized at the beginning of each trial to any location completely within the reachable workspace and any angle within $\theta = \pm 45^\circ$ of the horizontal, respectively.

At the beginning of all 100 trials, the starting position of the gasket is randomized. We perform this randomization by lifting the gasket with one fist and dropping it over either the top or bottom half of the workspace. The ends of the gasket are then moved outwards until the configuration of the gasket conforms to all of the constraints specified in Section III. A trial ends after the robot successfully completes the task, the program terminates, or ten minutes has elapsed, whichever comes first. For the unidirectional insertion, in the physical experiments, we always pick the left endpoint as a starting point but the algorithm can start from either endpoint.

Lightproof curtains on all sides of the robot are kept closed during experiments, with the exception of the curtains directly in front of the robot, which are kept open so as not to obstruct the view of the "front" workspace camera and to facilitate the placement of a workspace key light. During data collection for the learned policy, we additionally open a small section of drapes behind the "side" camera, from which the human demonstrator can view the workspace for teleoperation. We maintain this during evaluation to ensure that the environment for running trials was as similar as possible to the environment in which the human demonstrations were collected for best test time results.

D. Experimental Evaluation Metrics

Since automated evaluation methods were prone to error, we developed two manually-evaluated performance metrics.

After the robot execution has terminated, a human judge visually rates performance into one of four alignments categories and one of four insertion categories.

We note that given the specifications of Gasket Assembly there is a relationship between alignment and insertion. In order for the gasket to be properly inserted, it must first have been well aligned. The quality of alignment affects the resultant quality of insertion.

VI. Results

A. Alignment and Insertion

We perform 100 physical trials: 10 for the learned diffusion policy on the straight channel in fixed 0° pose and 90 across all procedural approaches and all channel types with varying channel pose positions and orientations as noted in Section V-C. See results in Table I. We again note that the learned policy was only evaluated for a fixed channel pose which matched the pose used during data collection whereas for the procedural algorithms, the channel pose varied significantly.

For the straight channel, the binary search and Binary+ approaches achieve 75-100% in all trials. For the curved channel, the Binary+ approach separates itself from the binary search approach, attaining the highest alignment and insertion performance. Finally, for the most difficult channel,

Trial No.	Channel	Method	Alignment Performance				Insertion Performance			
			0-25%	25-50%	50-75%	75-100%	0-25%	25-50%	50-75%	75-100%
1-10	Straight	Diffusion Policy	2	0	0	8	2	0	1	7
11-20	Straight	Unidirectional	6	0	1	3	6	1	1	2
21-30	Straight	**Binary Search**	0	0	0	**10**	0	0	0	**10**
31-40	Straight	**Binary+**	0	0	0	**10**	0	0	0	**10**
41-50	Curved	Unidirectional	5	1	0	4	7	0	1	2
51-60	Curved	Binary Search	0	0	4	6	1	3	6	0
61-70	Curved	**Binary+**	0	0	1	**9**	0	0	1	**9**
71-80	Trapezoid	Unidirectional	10	0	0	0	10	0	0	0
81-90	Trapezoid	Binary Search	9	1	0	0	9	**1**	0	0
91-100	Trapezoid	Binary+	9	0	**1**	0	9	**1**	0	0

TABLE I: **Alignment and Insertion Performance of 100 Physical Trials.**

the trapezoid, the Binary+ approach attains the best alignment performance, while having the same outcomes as the binary search for the insertion performance.

For the diffusion policy (Trials 1-10), there were three failed trials across the two metrics. These failures occurred because the diffusion policy failed to fully insert the gasket within the 10-minute threshold. In all three instances, the failure stemmed from incorrect execution of the pick-place-press sequence. Specifically, the robot either did not pick up the gasket, picked it but failed to place it correctly in the channel, or pressed in the wrong position, disrupting the subsequent pick-place-press sequences. Additionally, the robot could not recover from these failures, often spending several minutes attempting to improve the press without success or failing to detect the error.

Similarly, for the procedural methods on the straight and curved channels (Trials 11-70), the failures stemmed from failing to press the gasket into the channel at some points along the channel, leaving raised portions. When executing slide primitive in these scenarios, the raised portions caught on the gripper, causing large portions of the gasket to be unseated from the channel. For the trapezoid channel (Trials 71-100), both the binary search and Binary+ methods were initially able to insert the gasket into the long side. However, when the robot would try to round the corners to go to the other sides of the trapezoid, it inadvertently unseated the already inserted parts of the gasket. These errors compounded across all four sides of the trapezoid channel.

B. Completion Time Analysis

We observe a difference in the time required to complete the task when comparing the learned and procedural methods. The learned diffusion policy ran until the task was completed or it was terminated when the maximum time horizon of 10 minutes was reached. Across all 10 trials, the average completion time is approximately 5 minutes 34 seconds. This includes the three trials that terminated at the maximum time horizon (10 minutes) without task completion; excluding these trials reduces the average completion time to 3 minutes 40 seconds. The procedural algorithms ran for approximately 3 minutes 30 seconds for the straight and curved channels and 7 minutes for the trapezoid.

C. Discussion

For the learned diffusion policy, four humans provided 250 demonstrations.

With the unidirectional algorithm, we observe that after the first pick and place of an endpoint, most of the gasket still remains on the table. Consequently, the first press into the channel often fails to seat the gasket properly. This misalignment makes the next part of the gasket difficult to pick since it is positioned right alongside the channel. Moreover, even if the robot does grab the next point, since the previous pick-place-press does not properly seat the gasket, moving the next point along the gasket for pick and place can cause that previous segment to be completely unaligned from the channel. Since the unidirectional policy never revisits this prior point for pick and place, the push will very likely fail as well. Alternatively, with binary search, since the robot picks the gasket first at the middle, a larger amount of the gasket is now on the channel—approximately double compared to the unidirectional policy's initial pick. This increases the likelihood that the midpoint will be properly inserted when pressed. This proper seating at the midpoint is crucial for successful assembly, particularly in straight and curved channels. Compared to binary search, the Binary+ policy has more pick-place-pushes at points close to but not exactly at points that the binary search already reaches. This corrects any errors that the binary search might leave. Additionally, the Binary+ policy intentionally carries out pick-place-press at the endpoints, which ensures that when the robot executes the final slide, the grippers do not get caught at an unseated endpoint and drag the entire gasket out of the channel.

VII. Limitations + Future Work

In this paper, we present a new Gasket Assembly problem and provide results from 4 methods, a learned diffusion policy and 3 procedural algorithms. In the future, we will perform more experiments varying channel pose for the diffusion policy and learning diffusion policies for the curved and trapezoidal channels. We also plan to better handle recovery from poorly executed primitives by exploring additional approaches including hierarchical imitation learning [25] and self-supervised learning. We will also explore more complex channel shapes and sizes, work on a perception system that is more robust against different lighting conditions and distractors and examine if various gripper types have any significant effect on the task of Gasket Assembly.

Acknowledgements

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. We thank our colleagues who provided helpful feedback, help and suggestions, in particular Lawrence Chen, Philip Wu, Kaushik Shivakumar, Justin Kerr and Chung Min Kim.

References

[1] N. Aibada, R. Manickam, K. K. Gupta, and P. Raichurkar, "Review on various gaskets based on the materials, their characteristics and applications," *Int. J. Text. Eng. Process*, vol. 3, no. 1, pp. 12–18, 2017.

[2] V. Viswanath, K. Shivakumar, J. Kerr, B. Thananjeyan, E. Novoseller, J. Ichnowski, A. Escontrela, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Autonomously untangling long cables," *RSS*, 2022.

[3] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff, "Industrial robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Cham: Springer International Publishing, 2016, pp. 1385–1422.

[4] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *RSS*, 2023.

[5] V. Viswanath, K. Shivakumar, M. Parulekar, J. Ajmera Jainil Kerr, J. Ichnowski, R. Cheng, T. Kollar, and K. Goldberg, "Handloom: Heterogeneous autoregressive learned deformable linear object observation and manipulation," in *CoRL*, PMLR, 2023.

[6] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data," in *ICRA*, 2020.

[7] F. Süberkrüb, R. Laezza, and Y. Karayiannidis, "Feel the tension: Manipulation of deformable linear objects in environments with fixtures using force information," in *IROS*, 2022.

[8] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "SpeedFolding: Learning efficient bimanual folding of garments," in *IROS*, 2022.

[9] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *ICRA*, 2021.

[10] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, and K. Goldberg, "Autobag: Learning to open plastic bags and insert objects," in *ICRA*, IEEE, 2023.

[11] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," in *Robotics Research*, A. Billard, T. Asfour, and O. Khatib, Eds., ser. Springer Proceedings in Advanced Robotics, Cham: Springer Nature Switzerland, 2023, pp. 36–51.

[12] I. Kabul, R. Gayle, and M. C. Lin, "Cable route planning in complex environments using constrained sampling," in *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, 2007, pp. 395–402.

[13] A. Keipour, M. Bandari, and S. Schaal, "Efficient Spatial Representation and Routing of Deformable One-Dimensional Objects for Manipulation," in *IROS*, Oct. 2022.

[14] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE RA-L*, vol. 3, no. 4, pp. 3426–3433, 2018.

[15] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *ISRR*, Springer, 2016, pp. 339–354.

[16] A. X. Lee, S. H. Huang, D. Hadfield-Menell, E. Tzeng, and P. Abbeel, "Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects," in *IROS*, 2014.

[17] T. Tang, C. Liu, W. Chen, and M. Tomizuka, "Robotic manipulation of deformable objects by tangent space mapping and non-rigid registration," in *IROS*, 2016.

[18] L. Chen, S. Jayanthi, R. R. Paleja, D. Martin, V. Zakharov, and M. Gombolay, "Fast lifelong adaptive inverse reinforcement learning from demonstrations," in *CoRL*, PMLR, 2023, pp. 2083–2094.

[19] M. L. Schrum, E. Hedlund-Botti, and M. Gombolay, "Reciprocal mind meld: Improving learning from demonstration via personalized, reciprocal teaching," in *CoRL*, PMLR, 2023, pp. 956–966.

[20] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE RA-L*, vol. 5, no. 2, pp. 2372–2379, Apr. 2020, IEEE RA-L.

[21] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.

[22] G. A. Waltersson, R. Laezza, and Y. Karayiannidis, "Planning and control for cable-routing with dual-arm robot," in *ICRA*, 2022, pp. 1046–1052.

[23] A. Monguzzi, M. Pelosi, A. M. Zanchettin, and P. Rocco, "Tactile based robotic skills for cable routing operations," in *ICRA*, 2023.

[24] A. Wilson, H. Jiang, W. Lian, and W. Yuan, "Cable routing and assembly using tactile-driven motion primitives," in *ICRA*, 2023.

[25] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, "Multi-stage cable routing through hierarchical imitation learning," *IEEE Transactions on Robotics*, 2024.

[26] W. Wang, D. Berenson, and D. Balkcom, "An online method for tight-tolerance insertion tasks for string and rope," in *ICRA*, 2015.

[27] K. Wang, Y. Zhao, and I. Sakuma, "Learning robotic insertion tasks from human demonstration," *IEEE RA-L*, vol. 8, no. 9, pp. 5815–5822, 2023.

[28] T. Tang, H.-C. Lin, and M. Tomizuka, "A learning-based framework for robot peg-hole-insertion," in *Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, vol. 57250, 2015, V002T27A002.

[29] T. Tang, H.-C. Lin, Y. Zhao, Y. Fan, W. Chen, and M. Tomizuka, "Teach industrial robots peg-hole-insertion by human demonstration," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2016, pp. 488–494.

[30] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-reinforcement learning for robotic industrial insertion tasks," in *IROS*, 2020.

[31] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," in *ICRA*, 2022.

[32] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *ICRA*, 2022.

[33] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. Aparicio Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," in *IROS*, 2020.

[34] O. Spector and D. D. Castro, "Insertionnet - a scalable solution for insertion," *IEEE RA-L*, vol. 6, no. 3, pp. 5509–5516, 2021.

[35] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[36] L. Fu, H. Huang, L. Berscheid, H. Li, K. Goldberg, and S. Chitta, "Safe self-supervised learning in real of visuo-tactile feedback policies for industrial insertion," in *ICRA*, 2023.

[37] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *IROS*, IEEE, 2018.

[38] Y. Adagolodjo, L. Goffin, M. De Mathelin, and H. Courtecuisse, "Robotic insertion of flexible needle in deformable structures using inverse finite-element simulation," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 697–708, 2019.

[39] S. Pirozzi and C. Natale, "Tactile-based manipulation of wires for switchgear assembly," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2650–2661, 2018.

[40] S. Pirk, K. Hausman, A. Toshev, and M. Khansari, "Modeling long-horizon tasks as sequential interaction landscapes," *CoRL*, 2020.

[41] D. Driess, J.-S. Ha, R. Tedrake, and M. Toussaint, "Learning geometric reasoning and control for long-horizon tasks from visual input," in *ICRA*, 2021.

[42] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning," *CoRL*, 2019.

[43] F. Schmalstieg, D. Honerkamp, T. Welschehold, and A. Valada, "Learning long-horizon robot exploration strategies for multi-object search in continuous action spaces," in *ISRR*, Springer, 2022, pp. 52–66.

[44] G. Wang, M. Xin, W. Wu, Z. Liu, and H. Wang, "Learning of long-horizon sparse-reward robotic manipulator tasks with base controllers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 4072–4081, 2024.

[45] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, “Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks,” *IEEE RA-L*, vol. 7, no. 3, pp. 7327–7334, 2022.

[46] S. Nair and C. Finn, “Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation,” *ICLR*, 2020.

[47] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, “Learning to generalize across long-horizon tasks from human demonstrations,” *RSS*, 2020.

[48] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *ICML*, ser. ICML’15, Lille, France: JMLR.org, 2015, pp. 2256–2265.

[49] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *NeurIPS*, ser. NIPS’20, Vancouver, BC, Canada: Curran Associates Inc., 2020.

[50] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, *Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators*, 2023.

[51] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, “Deep imitation learning for complex manipulation tasks from virtual reality teleoperation,” in *ICRA*, 2018.

[52] K. Arulkumaran and D. Ogawa Lillrank, “A pragmatic look at deep imitation learning,” in *ACML*, 2023.

[53] A. Correia and L. A. Alexandre, “A survey of demonstration learning,” *arXiv preprint arXiv:2303.11191*, 2023.

[54] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *AAAI*, vol. 32, 2018.

[55] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *ICLR*, 2021.

[56] *Robomimic/robomimic/config/diffusion_policy_config.py at r2d2 · ARISE-Initiative/robomimic*.

VIII. Appendix

A. Human Demonstrations

Human demonstrations are collected as follows:

1) The channel is fixed in place horizontally across the workspace, separating the workspace into a lower and upper section.
2) The gasket is randomly dropped in either the lower or upper section so that it does not overlap itself and does not touch the channel.
3) The midpoint of the gasket is grasped and placed on top of the midpoint of the channel. The gripper then presses the gasket down into the channel.
4) One endpoint of the channel is chosen arbitrarily. The gasket is placed on top of the selected endpoint of the channel and pressed down into the channel.
5) The remaining endpoint of the gasket is then placed on top of the other endpoint of the channel and pressed to insert it into the channel.
6) The gripper is moved to the quartile points (the order of the quartile points the gripper goes to is chosen arbitrarily) and pressed down on the gasket such that at those points the gasket is inserted into the channel.
7) The gripper goes to the ’eighth’ points (again the order of the points the gripper goes to is chosen arbitrarily) and presses down on the gasket such that at those points the gasket is inserted into the channel.
8) The gripper goes to the midpoint of the gasket, moves down slowly to the channel surface such that the gripper touches the channel surface, and moves horizontally with no vertical movement towards one of the endpoints of the channel (chosen arbitrarily). The gripper returns to the midpoint of the channel and repeats this motion towards the other endpoint of the channel. This 8-step procedure is repeated for each human demonstration.

B. Experimental Evaluation Metrics Breakdown

After the robot execution has terminated, a human judge visually rates performance into one of four alignments categories, as follows:

1) **0% - 25%:** A **major alignment failure**, in which the robot has successfully aligned **less than 25%** of the gasket with the channel.
2) **25% - 50%:** A **partial alignment failure**, in which **between 25% and 50%** of the gasket has been successfully aligned.
3) **50% - 75%:** A **partial alignment success**, in which **between 50% and 75%** of the gasket has been properly aligned.
4) **75% - 100%:** A **full alignment success**, in which the robot has properly aligned **at least 75%** of the gasket length with the channel.

Similarly, a human judge visually rates performance into one of four insertion categories, as follows::

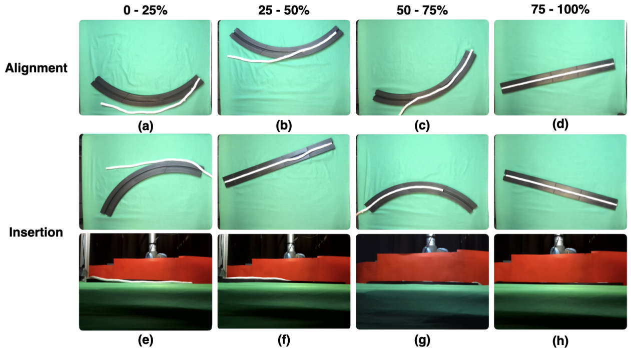

Fig. 4: **Evaluation Metric Examples.** We provide examples for all four categories of the alignment and insertion evaluation metrics discussed in Section V-D. We show the final gasket and channel states after the robot attempts gasket assembly. For alignment we only consider the view from the overhead camera to determine alignment between the gasket and channel. To determine the snug fit of the insertion, we consult both the overhead view (top row) and the front view (bottom row), because (f), for example, shows how a gasket that is aligned with the channel can have poor insertion.

1) **0% - 25%:** A **major insertion failure**, in which **less than 25%** of the gasket is inserted into the channel.
2) **25% - 50%:** A **partial insertion failure**, in which **between 25% and 50%** of the gasket is inserted.
3) **50% - 75%:** A **partial insertion success**, in which **between 50% and 75%** of the gasket is inserted.
4) **75% - 100%:** A **full insertion success**, in which **at least 75%** of the gasket length is inserted.

Figure 4 shows qualitative results from the trials of the three analytical algorithms in increasing order of success.