Towards Generalizable and Adaptive Multi-Part Robotic Assembly

Yunsheng Tian[1,†], **Joshua Jacob**[1], **Edward Li Gu**[1], **Farhad Javid**[2], **Pingchuan Ma**[1], **Branden Romero**[1], **Sachin Chitta**[2], **Shinjiro Sueda**[3], **Hui Li**[2,†], **Wojciech Matusik**[1]
[1]MIT CSAIL, [2]Autodesk Research, [3]Texas A&M University

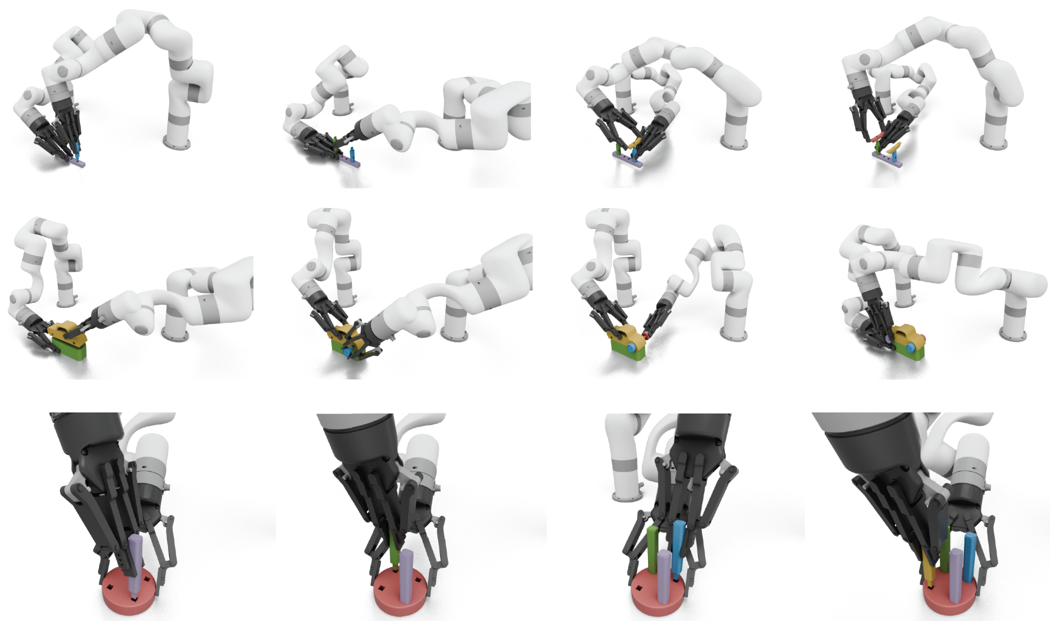


Figure 1: Our proposed dual-arm robotic system demonstrates adaptive manipulation and assembly of diverse multi-part objects. The system integrates task-oriented sequence, motion, and grasp planners for long-horizon assembly planning. For robust control, it leverages SE(3)-equivariance to learn assembly skills that generalize across various object geometries, assembly paths, and grasp poses, enabling it to achieve high success rates in zero-shot transfer on unseen assemblies.

Abstract: Multi-part assembly presents significant challenges for robotic automation due to the need for long-horizon planning, contact-rich manipulation, and broad generalization capabilities. In this work, we introduce a general dual-arm robotic system that end-to-end assembles multiple parts in simulation without any human effort. For learning contact-rich assembly skills, we propose a simple reinforcement learning framework that generalizes across object geometries, assembly paths, grasp poses, and robotic manipulators by leveraging SE(3)-equivariance. For effective long-horizon planning, our system integrates task-oriented sequence, motion, and grasp planners with a fixture generation method, facilitating multi-step assemblies using a standard dual-arm robotic setup. Following offline planning and training on a standard peg-in-hole benchmark, we perform zero-shot transfer experiments on six unseen multi-part assemblies from different categories. Our system achieves an average success rate of 90% per assembly step with random grasp poses, demonstrating robust performance and adaptability.

Keywords: Robotic Assembly, Assembly Planning, Contact-Rich Manipulation, Generalization

[†]Correspondence to: `yunsheng@csail.mit.edu` and `hui.xylo.li@autodesk.com`

1 Introduction

Real-world products often consist of complex multi-part assemblies. Currently in manufacturing, such assemblies are either still largely a manual process or require specially designed and fixed assembly lines. These fixed lines demand significant effort to design, program, set up, maintain, and modify, and they lack adaptability to uncertainty. There is a need for flexible, automated assembly lines.

Robotic automation of multi-part assemblies is a challenging task due to several factors. First, long-horizon sequence planning: generating physically feasible assembly sequences that also satisfy robot kinematic and grasping constraints is non-trivial. For example, a naively planned sequence may not be collision free or reachable by the robot. Moreover, the horizon increases with the number of parts and the number of potential assembly sequences grows exponentially with the number of parts. Second, the assembly process involves contact-rich manipulation and requires the robot to interact with its environment by applying appropriate amount of force as well as adapt to any misalignment and uncertainty. Finally, broad generalization capabilities are necessary, as the robot must adapt to a wide range of parts and assembly scenarios, often with minimal prior knowledge or specific programming for each unique task. Together, these factors make multi-part assembly a particularly complex and demanding application for robotic automation.

In this work, we tackle these challenges by building a general system for flexible multi-part assemblies with a dual-arm setup in simulation. More specifically, to guarantee physical feasibility and reachability of the assembly sequence, we propose an offline planning system that integrates sequence, grasp and motion planning. To achieve generalizability in contact-rich manipulation, we utilize equivariant representations and learn generalist reinforcement learning (RL) policies in combination with compliance control.

In summary, our work makes the following contributions:

1. We build a virtual dual-arm robotic system for automated multi-part assembly. The system end-to-end assembles individual parts to complete assemblies, generalizes to assemblies from different categories, and is robust to realistic uncertainties.
2. We propose a simple RL framework to learn contact-rich assembly policies generalizable to diverse object geometries, assembly paths, grasp poses, and robotic manipulators through SE(3)-equivariant representations and guidance from planned assembly paths.
3. We propose a comprehensive offline assembly planner that integrates sequence, grasp, and motion planning for dual robots arms. Our planner ensures feasibility for subsequent online execution over long horizons of complex multi-part assembly.
4. We propose an automated fixture design pipeline for accurate initial part placing and facilitating task-oriented grasps satisfying geometry, motion, and stability constraints.
5. We design and compile a benchmark suite of diverse assemblies from multiple categories assemblable by a dual-arm robotic system with parallel grippers. Our system shows end-to-end, generalizable, and robust robotic assembly of multiple parts.

2 Related Work

2.1 Multi-Part Robotic Assembly Systems

Existing literature on multi-part assembly focuses on automated planning of assembly sequences and paths, starting from the non-directional blocking graph [1] for geometric reasoning over simple geometries to motion planning methods through randomized tree search [2, 3, 4] for complex-shaped objects. Recently, physics-based motion planning [5] has shown success in assembling many complex parts with tight clearance. Beyond motion, realistic constraints have been considered during planning for execution on real-world robot setups, such as Tian et al. [6] and Rodrıguez et al. [7]. However, they are hard to deal with uncertainties in autonomous control of assembly skills beyond accurate planning. On the other hand, complete real-world robotic assembly systems have been demonstrated on IKEA chairs [8], bar structure assemblies [9], timber joints [10]. However, they are domain-specific systems and do not generalize to assemblies in any categories. In our work, we propose an integrated robotic assembly system with planner and controller that both generalize to diverse assemblies and completes end-to-end multi-part assembly.

2.2 Learning Contact-Rich Assembly Skills

Even with planned assembly sequences and paths, physically controlling the robot to assemble each part is still highly challenging due to its rich contact, millimeter-level clearance, uncertainties and errors in real robotic systems, and the need for generalization. RL has shown success in tackling this problem, for example, Thomas et al. [11] first combines the motion plans generated from CAD models and a RL policy for tracking the planned motion. Fan et al. [12] similarly warm-starts the policy with supervised trajectory optimization. High-precision insertion with sim-to-real has been also demonstrated [13, 14]. However, they primarily demonstrate assembly with top-down insertion directions and fixed grasps, which is unrealistic and inflexible for multi-part assembly where side-way insertion or tilted grasps are necessary for collision-free motion. The robotics community has been exploring spatial-equivariant techniques for improving generalization on various tasks such as pick-and-place [15] or object rearrangement [16], but is underexplored in robotic assembly tasks. As the closest work, Seo et al. [17] proposes geometric impedance control for learning SE(3)-equivariant gain scheduling policy, but does not consider varying grasps and diverse assembly geometries. Several benchmarks on robotic assembly have been proposed, but either focus on two-part insertion [18, 19], part alignment [20], or learning from demonstration [21, 22], where success rates and generalization capabilities are limited. For the first time, in our work, assembly policies are learned over diverse geometries, assembly paths, grasp poses, and robot manipulators, combined with planning to complete multi-part assembly tasks with high success rates.

3 Method

We tackle the problem of multi-part assembly automation in simulation. The requirement is that the system should be generalizable to a variety of part geometries and assembly scenarios, and robust to misalignment and uncertainties. Specifically, given a CAD model of an assembly in the assembled state and two robotic arms on the same work table with a parallel gripper on each arm, our system outputs an assembly sequence that is collision free and reachable by the robotic arms, along with suitable grasp poses, motion plan, and control policy, for completing the assembly.

3.1 System Overview

Our goal is to develop an autonomous planning and control system for flexible and generalizable multi-part assembly. The system is composed of three main components: an offline planner, a fixture generator, and an online controller. These components work together to achieve efficient and accurate robotic assembly, leveraging the strength of accuracy from offline planning and robustness from online policy execution.

Input The system receives as input the assembly meshes in their final assembled states provided by the user, and specifications of the robotic setup, mainly the geometry and kinematics of the robot arms and grippers.

Output The system produces the complete information needed for executing assembly of the given object on the provided robotic setup starting from part pick-up to the final complete assembly, including the feasible assembly sequences, desired robot arm motion, grasping poses, fixture designs for holding the part initially, and the closed-loop learned control policy for tracking planned motion and adapting to realistic uncertainties.

The system details of each component are expanded in the following sections.

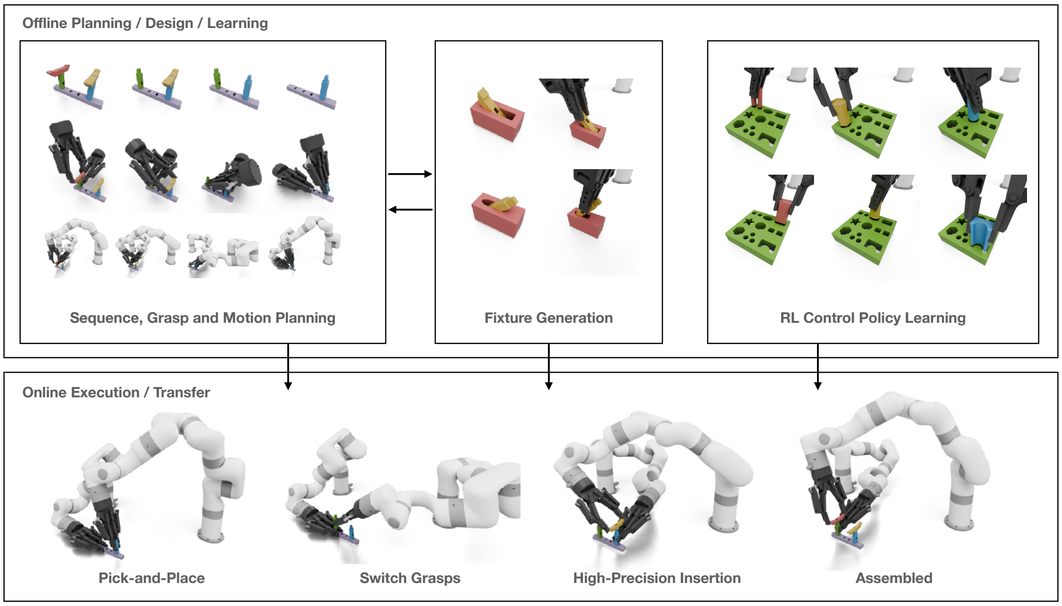


Figure 2: Overview of our robotic assembly system. Our system operates in two stages: offline and online. During the offline stage, we perform planning for the assembly sequence, grasp, and motion, generate necessary fixtures, and train the RL control policy using FMB pegs. In the online stage, we execute the generated plan by tracking it closely and employ the pretrained RL policy to mitigate real-world noise and uncertainties.

3.2 Planning Multi-Step Dual-Arm Assembly

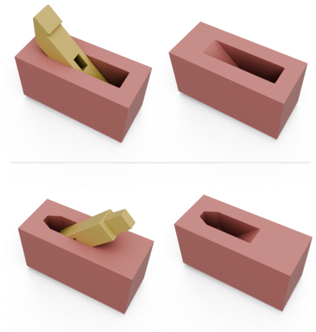
To guarantee feasible multi-step assembly execution, it is essential to develop an offline planner that outputs the desired sequence, grasp, and motion plans. We build upon the ASAP [6] framework for sequence planning, which utilizes disassembly tree search from a completely assembled state to fully disassembled parts, and extend it to support dual-arm grasp and motion planning.

Specifically, in each step, we perform antipodal grasp sampling on the surfaces of both the moving and holding objects to generate grasp proposals. We then select pairs of grasps that ensure collision-free interactions between objects, grippers, and arms, and also reachability by the arms through inverse kinematics (IK). Additionally, in each assembly step, after a part is inserted by one robot arm, the supporting robot arm needs to switch its grasp to support the next part. To facilitate this, we use RRT-connect [23] to plan collision-free robot motions between assembly steps for feasible grasp switching. Finally, the same motion planner is used to plan motions between the pick-up area and the assembly area.

This integrated approach ensures that all planned motions are feasible and that the assembly process is both efficient and reliable, capable of handling the complexities of multi-part assembly with dual robotic arms.

3.3 Generating Grasp-Oriented Assembly Fixtures

To perform assembly tasks using a dual-arm robotic setup, one arm must insert the part while the other arm holds the base part firmly. Therefore, the arm responsible for insertion should grasp the part in the correct orientation for insertion, ensuring that the gripper does not touch any section of the part that will be inserted later. If this condition is not met, both arms would need to reorient and re-grasp the part by passing it between them, therefore release the grasps of partially assembled objects. To address this challenge, we design a fixture that stably positions and orients each part for grasping. The parts are manually placed inside the fixtures before the assembly process starts. Knowing the exact location and orientation of the parts, the robotic arm can easily reach and grasp them in the proper orientation for insertion. The fixtures are designed such that the grasping section of each part extends out, allowing the arm to grasp the part effectively.



To generate the molds, an acceptable orientation for each part is first determined based on its geometry. This desired orientation is not unique. The depth of the fixture's hole for each part is calculated so that the part's center of mass aligns with the hole's opening after placement, ensuring gravitational stability when the part is placed inside the fixture. Although the parts are oriented obliquely in the mold, they are removed with a straight upward motion after being grasped by the arm. Therefore, the fixture's holes cannot be created simply by subtracting the part's geometry from a cubic block; the fixture's hole geometry must be convex. To achieve this, we project the part's 3D geometry onto a 2D plane perpendicular to the direction of the removal motion. The generated 2D polygon is then cut-extruded to the calculated depth from the fixture block. Consequently, the mold can be generated in a completely automated way for arbitrary shapes of objects and any given grasps generated by the planner, as shown in the inset figure above.

3.4 Learning General Single-Step Assembly Policy

Once the sequence, grasp, and motion are planned with feasibility guarantees, the next challenge is to develop a robust controller that can accurately track the desired plan. Multi-part robotic assembly involves manipulating various objects with different grasps and motions, which are hard constraints inherent to the task. To effectively address the multi-part assembly problem, the controller policy must generalize across multiple dimensions: object, grasp, assembly motion, and ideally, the type of robotic hardware to achieve broader and more transferable generalization.

Although this may seem challenging, we argue that with the appropriate design choices in the RL setup, a highly capable generalist policy can be learned to achieve such generalization. With the goal of broad generalization in mind, we propose a set of design choices that are simple to implement and extend while being highly effective for achieving robust and adaptable assembly policies.

SE(3)-equivariant observation and action Humans have the intuition to assemble diverse objects using the same skills, regardless of the object's pose or the assembly motion. We emulate this capability by leveraging SE(3)-equivariant transformations, which convert all possible straight-line assembly motions into top-down insertions, allowing the RL agent to perceive them in a standardized manner. By aligning the assembly path to the top-down direction, these transformations become computationally feasible with the guidance from our assembly path planner. The agent's observation consists of the SE(3)-equivariant pose and velocity of the object being assembled, derived from the global pose and velocity through the equivariant transformation. Since we transform the motion to be top-down insertion, there is no need to specify a goal in the observation. Similarly, the agent's actions are defined as equivariant delta poses. To execute a target action, we compute the kinematic transformation from the end effector to the object following the grasp pose planned by the planner. This design ensures that the observation and action spaces are minimal yet essential, facilitating easy information acquisition and enhancing generalizability, and is fully transferrable to different robot arm and end effectors. Please see the supplementary material for implementation details.

Sparse reward with path-guided reset Instead of designing a complex reward function with multiple terms and weighting factors, we keep the reward as simple as possible: task success, defined as a sparse but highly indicative measure. We have observed that the most crucial and challenging aspect of assembly occurs during the initial alignment between the peg and the hole, which requires millimeter-level precision. Once this alignment is achieved, completing the rest of the insertion motion becomes straightforward. Therefore, we define task success as the completion of the initial alignment. Quantitatively, this means determining whether the agent can insert the peg slightly into the hole, approximately 1 cm, without getting stuck above the hole.

Such a sparse reward can make exploration challenging for the agent due to the difficulty of achieving initial alignment in the presence of noise. To address this, we draw inspiration from the concept of reverse curriculum generation [24] and implement a reset method that effectively leverages the planned path guidance. Instead of always starting the agent from a position directly above the hole, which necessitates extensive exploration, we reset the agent randomly along the planned assembly path. This means the agent can be reset to positions already close to the goal, making exploration easier. After experiencing success from these close-to-goal positions, the agent gains better intuition for exploration when reset farther away from the hole. To further enhance the agent's capability for alignment without extensive guidance, we gradually increase the lower bound of the random initial positions, moving incrementally higher above the hole. We find that this combination of a sparse yet directed reward and a path-guided reset mechanism successfully facilitates the learning of a high-performance policy.

Hybrid RL and motion tracking control Separating the stages of high-accuracy alignment and low-accuracy motion tracking offers the added advantage of generalizing to assembly paths of arbitrary lengths and varying numbers of required alignments. The key intuition is to have the RL agent focus on the most challenging alignment task while leaving the remaining motion to a simple motion tracking controller that does not need to be learned. In our implementation, control begins with a position-based motion tracking controller. Whenever the system detects that the controller is stuck, we switch to the RL policy to complete the alignment. Once the RL policy successfully aligns the part, control is handed back to the motion tracking controller. This hybrid control scheme leverages the strengths of both learning-based and classical controllers, ensuring efficient and accurate assembly.

Note that for both RL and motion tracking, we leverage a low-level admittance controller to adjust the robot's compliance to external forces, allowing for smooth and controlled movements during the assembly process. It ensures that the robotic arms can handle minor misalignments and variations in force, further enhancing the robustness and precision of both the RL policy and the motion tracking controller.

Network architecture We use a standard MLP architecture for both the policy and critic network, which has 4 hidden layers with 64 neurons in each layer. The standard Tanh activation function is used. We use this naive network design for simplicity and ease of implementation. Despite its simplicity, this architecture is capable of learning effective policies for complex assembly tasks.

4 Experiments

To achieve high performance in multi-part assembly, it is essential to develop assembly policies that can reliably assemble each component. In this section, we address several critical questions:

1. Can the reinforcement learning (RL) policy learn effective assembly skills that generalize across diverse geometries and various grasping poses?
2. Does the SE(3)-equivariant representation of the observation and action space enhance policy performance?
3. Can the RL policy effectively leverage guidance from the planner to improve assembly outcomes?
4. How well does the learned policy transfer to unseen scenarios, including new objects, grasping configurations, and assembly paths?

Figure 3: Benchmark assemblies.

Table 1: Success rates of same-domain policy evaluation on FMB peg assemblies.

Method		Success Rate (%)							
		Rectangle	Round	Oval	Hexagon	Arch	Square + Circle	Double Square	3 Prong
Motion Tracking		0.72	0.62	0.72	0.54	0.7	0.56	0.72	0.54
Generalist Policy	Global	0.74	0.86	0.82	0.54	0.7	0.72	0.74	0.58
	Equiv.	0.72	0.86	0.88	0.6	0.74	**0.92**	**0.94**	0.58
	Equiv. + Guided	**0.8**	**0.92**	**1.0**	**0.9**	**0.92**	**0.92**	0.92	**0.76**

4.1 Benchmark Suite

To evaluate our multi-part robotic assembly system, we developed a diverse benchmark suite spanning furniture, toys, and industrial equipment categories. The suite includes a beam structure, stool, plumber's block (5 parts each), a gamepad, toy car (6 parts each), and a drone (9 parts). These assemblies cover various geometries and connection types found in real-world applications, with some featuring both top-down and side insertions, unlike other datasets. The gamepad assembly is challenging, requiring the alignment of 2 parts with 4 differently shaped pegs in a single step. The drone involves inserting a cantilevered arm needing precise alignment and admittance control. Certain assemblies necessitate specific sequences, testing the system's planning abilities.

All benchmarks contain rigid parts assemblable by a dual-arm robot with parallel grippers. CAD models of the final assembled configuration serve as input to our system. Evaluating on this diverse set demonstrates the generalizability and robustness of our system to different geometries, scales, orientations, and sequences, showcasing its potential for real-world multi-part assembly automation.

To simulate realistic types of noise and fairly compare all methods, we inject a 4 mm translational noise and a 4 degree rotational noise to the pose of the object to be inserted for all experiments.

4.2 Same-Domain Policy Evaluation

We first evaluate the performance of our RL assembly policy by training on 8 different peg insertion from the functional manipulation benchmark (FMB) [22] and testing on the same pegs in Table 1. Note that we train generalist policies instead of specialist policies, which means that we randomize the environments during training across all pegs and also randomize the grasp poses within 30 degrees. We compare multiple variants of the generalist policy with the baseline motion tracking controller. **Global** means using global observation and action space instead of our equivariant spaces. **Equiv.** means policies learned from our equivariant observation and action spaces. **Equiv. + Guided** means policies learned using the equivariant representation as well as the path-guided reset mechanism. The results suggest that **Equiv. + Guided** variant is the most effective one for learning the generalist policy due to the proper representation of the task spaces and effective guidance from the path planner.

4.3 Cross-Domain Policy Transfer

To test how well the pretrained generalist policy work on other different scenarios, we perform zero-shot policy transfer evaluations on our benchmark assemblies beyond FMB pegs, and the results can be found in Table 2. We surprisingly find that the policy pretrained on the PMB pegs can effectively transfer to new scenarios with different objects, assembly paths and random grasps. While

Table 2: Success rates of zero-shot cross-domain policy transfer on unseen multi-part assemblies after training on FMB peg assemblies.

Method		Success Rate (%) Beams (5 Parts)	Stool (5 Parts)	Plumber Block (5 Parts)	Gamepad (6 Parts)	Car (6 Parts)	Drone (9 Parts)
Motion Tracking		0.73 ± 0.08	0.68 ± 0.04	0.77 ± 0.12	0.84 ± 0.16	0.68 ± 0.07	0.75 ± 0.14
Pretrained Generalist Policy	Global	0.64 ± 0.11	0.5 ± 0.03	0.45 ± 0.13	0.78 ± 0.17	0.42 ± 0.08	0.81 ± 0.13
	Equiv.	0.75 ± 0.08	0.78 ± 0.04	0.78 ± 0.1	0.89 ± 0.13	0.73 ± 0.08	**0.87 ± 0.15**
	Equiv. + Guided	**0.93 ± 0.05**	**0.99 ± 0.01**	**0.89 ± 0.07**	**0.92 ± 0.08**	**0.8 ± 0.02**	**0.86 ± 0.12**

the baseline motion tracking controller can achieve roughly 70 percent success rates thanks to the complicance control, our pretrained policy shows a clear advantage compared to it, which matters even more in the multi-part assembly setup where the error accumulates from multiple steps. We believe further finetuning the pretrained policy using a little amount of data from the test scenario will further strengthen the performance. Moreover, Table 2 also demonstrates superior performance of learning generalist policy with equivariant representations and path-guided reset. We hope this training scheme of the RL assembly policy serves as a simple but strong baseline for future research.

5 Limitations and Future Work

While our system shows promising results for generalizable multi-part assembly, several limitations remain. Currently, we assume stable grasps, light part weights, and focus primarily on insertion skills. This leaves areas such as handling heavier parts, managing grasp slippage, and performing other operations like screwing unaddressed. Incorporating these capabilities would significantly improve the robustness and applicability of our system in more complex and diverse assembly tasks.

Moreover, the current setup does not involve multi-part reorientation, which is often necessary in real-world scenarios. Developing more dexterous manipulators capable of such reorientations is an important avenue for future work. Additionally, integrating vision systems for alignment feedback could greatly enhance the accuracy and adaptability of the assembly process. Training policies to handle zero-shot insertion of arbitrary objects with arbitrary grasps would further improve the system's flexibility.

Sim-to-real transfer is a natural next step for our system, allowing it to be deployed in real-world settings. However, this transition poses significant challenges, including issues related to perception, slippage handling, and bin-picking. Effectively addressing these challenges will require substantial research and development efforts. By overcoming these limitations, we can expand the real-world capabilities of our system and provide a solid foundation for future research in adaptive assembly automation.

References

[1] D. Halperin, J.-C. Latombe, and R. H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26(3):577–601, 2000.

[2] S. Sundaram, I. Remmler, and N. M. Amato. Disassembly sequencing using a motion planning approach. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1475–1480. IEEE, 2001.

[3] D. T. Le, J. Cortés, and T. Siméon. A path planning approach to (dis) assembly sequencing. In *2009 IEEE International Conference on Automation Science and Engineering*, pages 286–291. IEEE, 2009.

[4] X. Zhang, R. Belfer, P. G. Kry, and E. Vouga. C-space tunnel discovery for puzzle path planning. *ACM Transactions on Graphics (TOG)*, 39(4):104–1, 2020.

[5] Y. Tian, J. Xu, Y. Li, J. Luo, S. Sueda, H. Li, K. D. Willis, and W. Matusik. Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Transactions on Graphics (TOG)*, 41(6):1–11, 2022.

[6] Y. Tian, K. D. Willis, B. A. Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda, et al. Asap: Automated sequence planning for complex robotic assembly with physical feasibility. *arXiv preprint arXiv:2309.16909*, 2023.

[7] I. Rodrıguez, K. Nottensteiner, D. Leidner, M. Kaßecker, F. Stulp, and A. Albu-Schäffer. Iteratively refined feasibility checks in robotic assembly sequence planning. *IEEE Robotics and Automation Letters*, 4(2):1416–1423, 2019.

[8] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham. Can robots assemble an ikea chair? *Science Robotics*, 3(17):eaat6385, 2018.

[9] Y. Huang, C. R. Garrett, I. Ting, S. Parascho, and C. T. Mueller. Robotic additive construction of bar structures: Unified sequence and motion planning. *Construction Robotics*, 5:115–130, 2021.

[10] A. A. Apolinarska, M. Pacher, H. Li, N. Cote, R. Pastrana, F. Gramazio, and M. Kohler. Robotic assembly of timber joints using reinforcement learning. *Automation in Construction*, 125:103569, 2021.

[11] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel. Learning robotic assembly from cad. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3524–3531. IEEE, 2018.

[12] Y. Fan, J. Luo, and M. Tomizuka. A learning framework for high precision industrial assembly. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 811–817. IEEE, 2019.

[13] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. *arXiv preprint arXiv:2305.17110*, 2023.

[14] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR, 2023.

[15] H. Huang, D. Wang, A. Tangri, R. Walters, and R. Platt. Leveraging symmetries in pick and place. *The International Journal of Robotics Research*, page 02783649231225775, 2024.

[16] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[17] J. Seo, N. P. Prakash, X. Zhang, C. Wang, J. Choi, M. Tomizuka, and R. Horowitz. Contact-rich se (3)-equivariant robot manipulation task learning via geometric impedance control. *IEEE Robotics and Automation Letters*, 2023.

[18] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal. Benchmarking off-the-shelf solutions to robotic assembly tasks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1046–1053. IEEE, 2021.

[19] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on robot learning*, pages 767–782. PMLR, 2018.

[20] Y. Lee, E. S. Hu, and J. J. Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks. In *2021 ieee international conference on robotics and automation (icra)*, pages 6343–6349. IEEE, 2021.

[21] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *arXiv preprint arXiv:2305.12821*, 2023.

[22] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

[23] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[24] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.