Hierarchical Hybrid Learning for Long-Horizon Contact-Rich Robotic Assembly

Jiankai Sun[1*], Aidan Curtis[2], Yang You[1], Yan Xu[3], Michael Koehle[4], Leonidas Guibas[1], Sachin Chitta[4], Mac Schwager[1], Hui Li[4]

***Abstract*—Generalizable long-horizon robotic assembly requires reasoning at multiple levels of abstraction. End-to-end imitation learning (IL) has been proven a promising approach, but it requires a large amount of demonstration data for training and often fails to meet the high-precision requirement of assembly tasks. Reinforcement Learning (RL) approaches have succeeded in high-precision assembly tasks, but suffer from sample inefficiency and hence, are less competent at long-horizon tasks. To address these challenges, we propose a hierarchical modular approach, named `ARCH` (Adaptive Robotic Compositional Hierarchy), which enables long-horizon high-precision assembly in contact-rich settings. `ARCH` employs a hierarchical planning framework, including a low-level primitive library of continuously parameterized skills and a high-level policy. The low-level primitive library includes essential skills for assembly tasks, such as grasping and inserting. These primitives consist of both RL and model-based controllers. The high-level policy, learned via imitation learning from a handful of demonstrations, selects the appropriate primitive skills and instantiates them with continuous input parameters. We extensively evaluate our approach on a real robot manipulation platform. We show that while trained on a single task, `ARCH` generalizes well to unseen tasks and outperforms baseline methods in terms of success rate and data efficiency. Videos and code can be found at https://long-horizon-assembly.github.io.**

I. Introduction

The manufacturing industry is increasingly turning to robotic systems for complex assembly tasks, driven by the need for greater precision, consistency, and efficiency [10, 8]. Nevertheless, long-horizon, contact-rich, and high-precision robotic assembly tasks continue to pose significant challenges in robotics and automation, as these tasks demand sophisticated learning and object interaction capabilities that go beyond traditional programming approaches [14]. Our research endeavors to ground the robotic assembly to real-world applications where the robots are asked to perform intricate and long-horizon assembly tasks widely demanded in industrial environments, e.g., parts insertion [24] and cable routing [23]. The acceleration of complex assembly tasks is a crucial step toward more flexible and adaptive industrial processes. By focusing on automating such long-horizon tasks, we seek to enhance productivity and efficiency across various sectors, including manufacturing, construction, and logistics.

Current industrial robotics applications to assembly and manufacturing are largely engineered for a specific task and struggle with adapting to varied assembly scenarios and component variations. Ideally, such systems are capable of handling high-precision contact-rich operations under uncertainty [16] and generalizing to new scenarios from limited demonstrations [38] using multimodal inputs such as visual and force-torque feedback [31, 43].

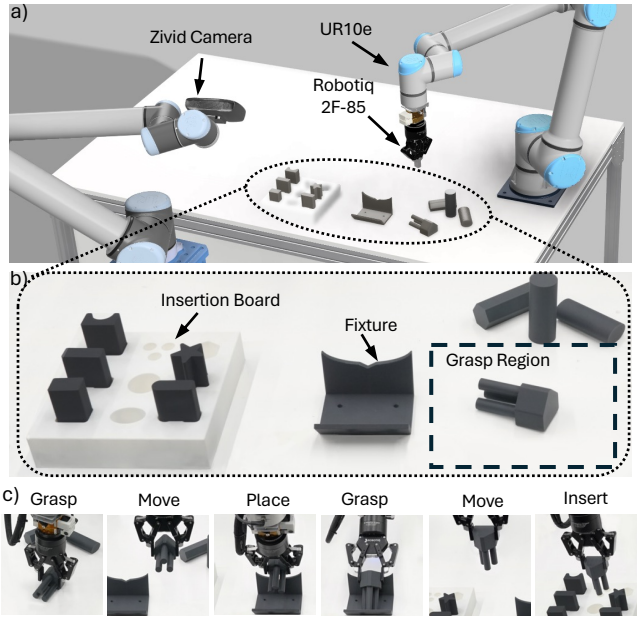

Fig. 1: a) A Zivid camera captures RGB-D images while a UR10e robot equipped with force-torque sensors is used for the manipulation tasks. b) Objects are randomly placed within a region on the table. The robot must insert them into the correct receptacle in the Insertion Board, adjusting their orientation with the Fixture if necessary. c) The library of RL and model-based action primitives.

Recent advances in learning-based solutions have started to tackle some of these problems. End-to-end imitation learning (IL) from human demonstrations has made marked progress [6], but still fails to generalize outside a narrow training distribution, requires significant data from a trained expert demonstrator [15], and often struggles with high-precision tasks. Simply sequencing individual primitives can lead to compounding errors, diminishing overall system performance [36]. While reinforcement learning (RL) can drive specific assembly operations, it usually struggles with more complicated long-horizon tasks [35], where the training faces challenges of sparse rewards and an exploding sampling space

[1]Stanford University email: jksun@cs.stanford.edu. [2]MIT [3]University of Michigan [4]Autodesk Research
* Work done during internship at Autodesk.

due to larger long-horizon footprints.

To address these challenges, we propose a hierarchical approach for robotic assembly, especially targeted at long-horizon, contact-rich, high-precision settings. Our method employs a hierarchical strategy that involves developing a set of low-level primitive skills (e.g. grasp, insert) and a high-level policy to select and integrate these primitives. The framework combines classic motion planning algorithms with RL policies to develop low-level primitive skills, achieving both efficiency and adaptability. More specifically, we leverage motion planning algorithms [4] to efficiently guide the end-effector to target goals. For complex assembly operations in partially observed environments, we utilize RL policies that enable adaptive, contact-rich, and high-precision manipulation. The RL policies are trained fully in simulation with domain randomization such that they can directly transfer to the physical world without any fine-tuning. Based on these primitive skills, the high-level policy is learned from a few trajectories of human demonstrations to select and organize among these skills to achieve the final assembly goal. This hybrid strategy allows our system to leverage pre-existing knowledge for routine movements while maintaining the flexibility to learn and adapt to new, intricate assembly tasks in unfamiliar settings. Moreover, our high-level primitive selection policy has a small action space and is object-agnostic. Hence, it can be effectively trained via imitation learning, as it only requires a few human demonstrations.

In summary, our main contributions are as follows:

- We introduce ARCH, a hierarchical framework designed to tackle the challenging problem of *long-horizon robotic assembly*.
- Our hierarchical framework includes a low-level skill library and a high-level imitation-learned policy that selects and composes the primitive skills. The low-level skills are built upon efficient motion planning algorithms for end-effector movement and RL policies for high-precision, adaptive assembly in contact-rich scenarios.
- We conducted extensive hardware experiments on a robot manipulator in an assembly work cell. Despite being trained on a single task, our approach archives generalization to other novel tasks and outperforms baseline methods for long-horizon robotic assembly tasks.

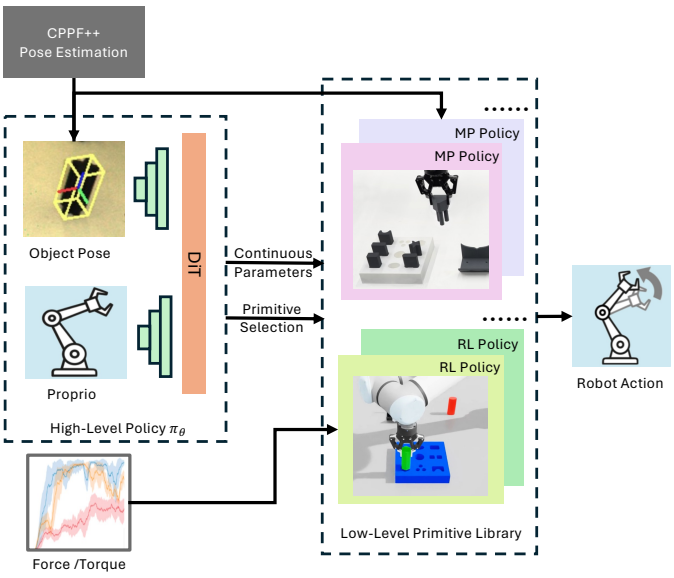

Fig. 2: We propose a hierarchical framework for long-horizon robotic assembly. The high-level policy π_θ, which takes as input object pose from pose estimation and robot proprioception, and outputs a categorical distribution to select the best low-level primitive as well as its continuous parameters, is obtained via imitation learning. The low-level policy executes the selected primitive using either an RL-based or a motion-planned (MP) policy. We train in simulation an RL policy for the contact-rich portion of the task, e.g. insertion, based on force-torque feedback. We use the MP policies for primitives that are in free space, e.g. move.

II. Related Work

A. Task and Motion Planning

Task and Motion Planning (TAMP) [17, 18, 13] is an effective approach for long-horizon manipulation problems as it can resolve temporally dependent constraints through hybrid symbolic-continuous reasoning. These plans may involve regrasping [1], clearing obstructing objects [11], or moving to gather information [12]. Despite these abilities, such methods typically require hand-designed symbolic transition functions and continuous parameter samplers. Additionally, TAMP methods are computationally expensive, which limits their ability to handle failures in dynamic tasks. We argue that the symbolic transition functions and samplers can be replaced by a learnable high-level policy to address the issues mentioned above.

B. Learning for Long-Horizon Manipulation

There is a large body of research in learning for long-horizon manipulation tasks. Diffusion policies [6, 7, 42] have shown to be a powerful tool for addressing complex manipulation tasks by leveraging their generative and multi-modality capabilities. However, they require a large amount of training data, i.e. human demonstrations, to learn effective end-to-end policies for complex, long-horizon tasks. Therefore, a significant amount of research has adopted the divide-and-conquer mindset and tackled the task by decomposing it into easier and reusable subtasks, instead of learning an entire task with a single policy.

Skill decomposition and chaining [20, 19, 2, 21, 22, 9] is a promising way to synthesize long-horizon and complex behaviors by sequentially chaining previously learned simpler skills through transition functions. Mishra et al. [29] trains individual skill diffusion models as action primitives and combine them at test time, when learned distributions of the skills are linearly chained to solve for a long-horizon goal during evaluation. Mao et al. [27] learns primitive skills from human demonstrations based on haptic feedback and

segments a long-horizon task into a sequence of skills for dexterous manipulation. Chen et al. [5] introduces a bi-directional optimization framework that chains RL-trained sub-policies together using a transition feasibility function for long-horizon dexterous manipulation.

C. Hierarchical Modeling in Robotics

Hierarchical approaches typically offer policies at varying abstraction levels. Xian et al. [39] and Ma et al. [25] propose to learn a high-level policy from demonstrations to predict end-effector key poses and a low-level diffusion-based trajectory generator for connecting these key poses to achieve the final goal. MimicPlay [38] learns a high-level plan from human videos of manipulating objects and low-level visuomotor controls from teleoperated demonstrations on the real robot. MAPLE [32] enhances standard RL algorithms by incorporating a predefined library of behavioral primitives. The most relevant work to our approach uses hierarchical policy decomposition for a multi-stage cable routing task [23] . They define scripted and imitation learned low-level primitives and learn a high-level policy to select which primitive to execute from human demonstration, all in real. We combine model-based and model-free primitives on the low level and learn a high-level policy for parameterized primitive selection and instantiation. We also train a robust RL policy for the contact-rich primitive - insertion - in simulation and perform zero-shot sim-to-real transfer. Additionally, we train all primitive modules in advance, and allow expert demonstrators to make use of these primitives as necessary.

Recent advancements in robotic learning have also highlighted the need for benchmarks that effectively balance generalization and the complexity of manipulation tasks. The Functional Manipulation Benchmark (FMB) [24] proposed by Luo et al. addresses this gap by defining functional manipulation as a series of relevant behaviors, such as grasping, repositioning, and physical interaction with objects. This benchmark emphasizes the importance of both contact dynamics and object generalization, making it a valuable resource for researchers aiming to develop robots capable of performing intricate assembly tasks. For these reasons, we focus on the multi-peg assembly benchmark, introduced in FMB [24], to showcase the generalizability of our approach.

III. Background

In this work, we model the long-horizon robot assembly task as a parameterized-action Markov decision process (PAMDP) [28]. A PAMDP problem consists of the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ where $\mathcal{S}$ is the continuous state space, $\mathcal{A}$ is a set of discrete action primitives $\{a_1, a_2, ..., a_k\}$, each with m_a continuous parameters $X_a \subseteq \mathbb{R}^{m_a}$, $P(s, a, s')$ is the probability of transitioning to state s' when taking action primitive a in state s, $R(s, a)$ is the reward from taking action primitive a in state s, and γ is the discount factor. Our goal in a PAMDP is to find a policy $\pi(a, x|s)$ that minimizes the discrepancy between the learned policy and the expert demonstrations that maximize reward under the PAMDP.

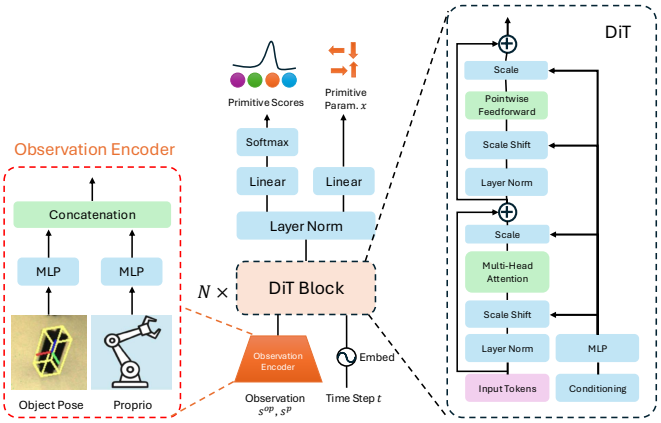

Fig. 3: High-Level Policy Architecture.

IV. Method

Solving a PAMDP involves not only selecting the best discrete primitive but also optimizing over the continuous parameter space for each primitive at each state. In this work, we fix a set of parameterized primitives that are building blocks of manipulation for assembly tasks (See Section IV-A) and attempt to learn the high-level policy $\pi_\theta(a, x|s)$ as a neural network with parameters θ. This results in a hierarchical framework, shown in Fig. 2. In this section, we describe our low-level primitive library, high-level policy architecture, data collection pipeline, and the pose estimation module.

A. Low-level Primitive Library: Modeling Basic Skills with Motion Planning and Reinforcement Learning

Our focus is on providing agents with a library of flexible primitives that act as foundational components for high-precision, contact-rich robotic assembly tasks. The hierarchical decision-making system operates independently of the specific implementations of these primitives, which may include either closed-loop, learning-based skills or analytical motion planners. Regardless of how they function internally, it is essential that the primitives are adaptable to varying behaviors, which is why we introduce parameters x to customize each primitive. In our learning framework, we treat these primitives as functional APIs, where input parameters x define the actions to be executed. These input parameters typically have clear semantics, such as specifying a 6-DoF end-effector pose for a grasping primitive. While these primitives offer flexibility, we acknowledge that they may not cover all possible scenarios, and relying solely on them could restrict the range of behaviors an agent can achieve.

We design our low-level primitive library according to the building blocks of manipulation for assembly tasks. These primitives are not exhaustive and easily extendable.

Grasp: The robot moves its end-effector to a pre-grasp pose $p_g \in \mathbf{SE}(3)$, specified by the input parameter, and closes its gripper. A motion planner is used for its execution.

Place: The robot moves its end-effector to a pre-place pose $p_p \in \mathbf{SE}(3)$, specified by the input parameter, and opens its gripper. A motion planner is used for its execution.
Move: The robot moves its end-effector to a pose $p \in \mathbf{SE}(3)$, specified by the input parameter. A motion planner is used for its execution.
Insert: The robot has the object in grasp and inserts it into the correct hole on the insertion board. An RL policy $\pi_L(\phi)$ is used because the insertion task involves rich contact between the object and its receptacle. We train an RL policy in simulation [30] using PPO [34]. Its observation space includes end-effector (EE) force-torque (FT) and EE pose relative to the target pose; its action is EE velocity u_t; and its reward is the negative distance between the current pose s_t^p and the target pose g.

$$\tilde{r}(s_t^p, u_t, g) = -||s_t^p - g||_2, \tag{1}$$

The objective of goal-conditioned Insertion primitive is to reach goal pose g that maximize the expectation of the cumulative return as Eq. 2 shows:

$$\mathcal{J}(\phi) = \mathbb{E}\left[\sum_t \gamma^t \tilde{r}(s_t^p, u_t, g)\right]. \tag{2}$$

B. High-level Policy: Composing Primitives via Imitation

The high-level policy $\pi_\theta(a, x|s)$ takes object pose obtained from pose estimation (see Section IV-B2) and robot proprioception as inputs s, to select the appropriate primitive a from the low-level primitive library and predict its continuous parameter $x \in \mathbb{R}^{d_{\text{control}}}$ for low-level control. We collect human demonstration data (see Section IV-B1) and train the high-level policy via imitation learning. As shown in Fig. 3, we modify the Diffusion Transformer (DiT) [33] architecture as the backbone of our high-level policy, considering its strong sequential data handling capability is suitable for handling with a history of previous states and actions. The DiT outputs softmax scores for each primitive and the continuous action parameter for the primitive with the highest score. These two functions are supervised using cross-entropy loss and MSE loss, respectively. We employ DiT blocks with adaptive LayerNorm-Zero conditioning. We use the Robomimic observation encoder [26] to extract features from object pose (s^{op}) and proprioceptive pose (s^p). The goal of imitation learning is to find a parameterized policy π_θ that can maximize the likelihood function based on the currently collected demonstration data $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, \mathbf{x})\}$:

$$\theta = \arg\max_\theta \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{x}\sim\mathcal{D}} \pi_\theta(\mathbf{a}, \mathbf{x}|\mathbf{s}). \tag{3}$$

1) Data Collection: As our low-level primitives are executed with either RL policies or MP policies, no data collection is needed. In this section, we focus on the data collection process[1] for the high-level policy. After training the individual action primitives, a human expert sequentially selects them to complete the multi-stage peg insertion task, using a keyboard. The continuous input parameter of each action primitive is the pose obtained from pose estimation. For example, to initiate the *grasp* primitive, the demonstrator selects the primitive index, and the object pose obtained from the pose estimation module is passed to the motion planner to execute the primitive. To enhance robustness, we augment the dataset by introducing noise into the robot's state. The dataset denoted as $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, \mathbf{u})\}$, consists of sensor observations $\mathbf{s}$, primitive indexes $\mathbf{a}$, and corresponding continuous primitive parameters $\mathbf{x}$.

The dataset consists of two types of demonstrations: "successful" and "recovery" trials. In approximately 20 trials, the demonstrator successfully inserts the object, while in the remaining 20 trials, failures occur, followed by recovery actions demonstrated by the operator. These recovery trials are crucial to making the learned policy more robust to errors.

In summary, we allow the demonstrator to select the discrete primitive to execute while the continuous input parameter comes from pose estimation. This is a novel way to collect demonstration data as it is often difficult and time-consuming to collect teleoperated demonstrations for high-precision tasks.

2) Pose Estimation: As shown in Fig. 2, object pose is needed by the high-level policy and also serves as the input parameter to the low-level primitive skills, such as grasp. We integrate the state-of-the-art pose estimation method CPPF++ [40, 41], which demonstrates strong generalization from simulation to real-world scenarios. Specifically, given a 2D RGB-D image $\mathcal{I}$, the model produces an accurate 6D pose $\xi \in \mathfrak{se}(3)$, where the first three components represent rotation and the last three correspond to translation. While the original CPPF++ is designed for category-level pose estimation, we adapt it for instance-level pose estimation by using distinct industrial CAD models as different "categories". To enhance precision, inspired by the Iterative Closest Point (ICP) [3] al-

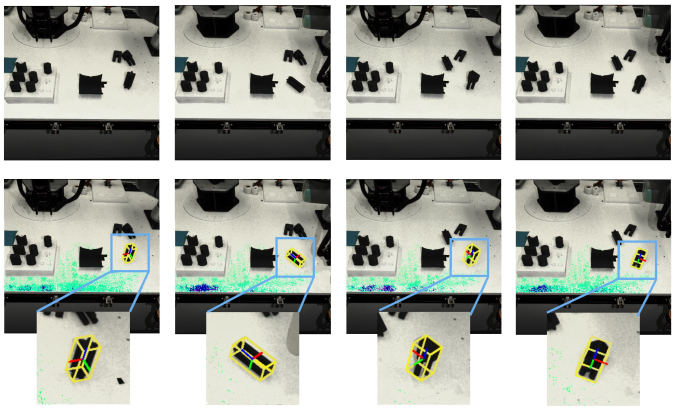

Fig. 4: **Pose Estimation Examples**. Our method gives accurate pose estimation of differently shaped parts. The top row shows the input images and the bottom row shows the pose estimation. Red, green, blue colors indicate the xyz axes in the canonical space, respectively.

[1]Although datasets like FMB [24] exist, our setup and hardware differ. We use a UR10e instead of the Panda robot. Additionally, we found that images captured by the RealSense camera were not accurate enough for pose estimation, so we opted for the Zivid camera. As a result, we need to collect our own data to suit our setup.

gorithm, we propose a post-optimization procedure conducted in the tangent space of the Lie group.

Specifically, we compute the one-way Chamfer distance from the object-masked point cloud to the full CAD model, transformed by the current pose ξ:

$$L_{CD}(\xi) = \sum_i \min_j \|T_\xi(p_i) - p_j^*\|_2, \tag{4}$$

where p_j^* is the j-th point on the back-projected point cloud obtained from the predicted masks, p_i is the i-th point on the object mesh, and T_ξ is the rigid transformation that aligns the canonical object point with the scene, defined as $T_\xi(p_i) = R \cdot p_i + t$. We utilize the one-way Chamfer distance due to self-occlusion in the observation and optimize ξ iteratively using the Lietorch [37] library to obtain a refined pose.

Fig. 4 presents some qualitative pose detection results. The high accuracy of the pose estimator enables us to have a high success rate in terms of both the low-level MP policies and the high-level policy.

V. Experiments

A. Task Description

We focus on the multi-stage assembly tasks [24]. As shown in Fig. 1, 9 objects of different shapes must be inserted into the board. The robot must grasp the object, may need to reorient and regrasp it using the fixture depending on its grasp pose, and then insert it into the board. We assume CAD models of all the objects are available and the location of the fixture and the board is known. The objects range from simple or symmetrical shapes, such as rectangle, circle, and oval, to more complex ones, including star, 3-prong, square-circle, and arch. We evaluate methods across subgoal and long-horizon task categories and examine generalization using unseen objects.

B. Setup and Implementation

The workcell setup includes a UR10e robotic arm paired with a Robotiq 2F-85 gripper and a third-person view Zivid 2+ M60 camera (Fig. 1). The simulation platform for RL training is built on the IsaacLab engine [30], while motion-level manipulation plans are computed using the lazyPRM approach [4]. Model training is performed on a machine equipped with an Intel 3.80GHz i7-10700k CPU and a GeForce RTX 4090 Ti GPU, running on a Ubuntu system.

For pose estimation, we use 512×512 RGB-D image as input. For high-level policy π_θ, We adopt the Diffusion Transformer model [33] for predicting action feasibility score. Our RL primitive is trained using 1,000 parallel environments in IsaacLab. The maximum task horizon is 25,000 action steps, equivalent to 200 seconds of robot execution at a 125Hz control frequency.

C. Baselines

a) End-to-end Learning: We evaluate two end-to-end methods: 1) **End-to-end IL**: Diffusion Policy (DP) [6], a goal-conditioned imitation learning framework that leverages a diffusion model for generating diverse action trajectories. Teleoperated demonstration data is collected in the real world. 2) **End-to-end RL**: PPO [34], a widely-used RL method that optimizes policy performance through proximal updates, balancing exploration and exploitation while ensuring stable training. Training is conducted in the IsaacLab simulation environment.

b) Hierarchical Methods: We evaluate two hierarchical methods: 1) **MimicPlay** [38] learns high-level latent plans from human play data to guide low-level visuomotor control. Teleoperated demonstration data is collected in the real world. 2) **Luo et al.** [23] propose a Multi-Stage Cable Routing approach through hierarchical imitation learning. Teleoperated demonstration data for both low-level and high-level policies is collected in the real world. We adapt the above methods to our task.

TABLE I: Success rate of the multi-stage task of Hexagon assembly, comparing our method with baseline methods after training on 40 demonstrations on the Hexagon object.

Method	SR (%) ↑	SPL ↑
E2E RL [34]	0	0.00 ± 0.00
E2E IL [6]	0	0.00 ± 0.00
MimicPlay [38]	40	0.78 ± 0.05
Luo et al. [23]	50	0.83 ± 0.13
ARCH (Ours)	55	0.93 ± 0.11
Human Oracle [24]	65	0.95 ± 0.08

TABLE II: Success rate for multi-stage assembly task and single-stage tasks by object. Our system demonstrates the ability to generalize to unseen objects.

	Object	SR (%)	% Grasped	% Inserted
Seen	Hexagon	55	80	75
Unseen	Star	50	85	60
	SquareCircle	35	75	50
	3Prong	80	90	90
	Circle	75	95	80
	Oval	55	85	70
	Arch	40	65	65
	DoubleSquare	40	70	60
	Rectangle	65	90	75

D. Evaluation Metrics

For each trial, the selected object is placed within the grasp randomization region, centered at one of four random positions, and oriented in one of five possible angles. A total of 20 trials are conducted. For the grasping primitive, the object must be securely gripped by the gripper to prevent it from falling after being lifted. Finally, the object's bottom surface must be fully inserted into the corresponding hole.

The primary metric is the task success rate (%). We found that failures primarily stem from two challenging primitives: grasping and insertion. To better understand the sources of error, we introduce two specific metrics: "% Grasped" and "% Inserted," which measure the success rates of these two primitives.

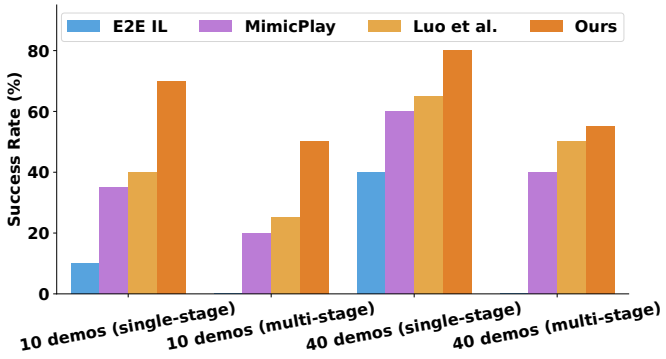

Fig. 5: Data efficiency comparison across imitation learning methods. Trained with just 10 demonstrations, our system achieves a higher success rate compared to baselines, which require significantly more data to perform well.

a) Success Rate (SR): the percentage of successfully completed multi-stage tasks (instead of each primitive) out of the total multi-stage tasks attempted.

b) SPL: Success weighted by Path Length, as Eq. 5.

$$\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{l_i}{\max(l_i, l_i^{opt})}, \tag{5}$$

where N is the total number of evaluation trials, S_i is 1 if the i-th trial is successful and 0 otherwise, l_i^{opt} is the shortest path (number of primitives) to the goal, and l_i is the actual path length taken by the agent.

c) % Grasped: the percentage of target objects that are successfully grasped as a single-stage task.

d) % Inserted: the percentage of target objects that are successfully inserted as a single-stage task.

E. Experiment Analysis

As shown in Tab. I, long-horizon tasks often involve sparse rewards, which complicates RL training. Consequently, **end-to-end (E2E) RL** [34] fail at such long-horizon assembly tasks. With diffusion models, **E2E IL** [6] falls short in handling high-precision tasks and requires a large number of expert demonstrations. The hierarchical baselines, **MimicPlay** [38] and **Luo et al.** [23], achieve better success rate than the E2E methods, but struggle with grasp failures and consequently have lower overall success rate than our method. Our ARCH demonstrates higher success rates and SPL compared to the baselines due to our hierarchical hybrid design. By combining model-based and adaptive learning-based components, our approach achieves both high precision and flexibility. Although our RL insertion primitive is trained solely in simulation on the Hexagon object for the insertion primitive, it transfers well to real-world scenarios and to other objects. Finally, **Human Oracle** is not a baseline but serves as the upper bound for high-level policy. The human oracle selects primitives with near-flawless accuracy and can self-correct mistakes. Failures with this method are primarily due to pose estimation or grasp errors.

a) Generalization to Unseen Objects: Our system has been trained exclusively with the Hexagon object, both for the RL primitive and for the high-level policy. Tab. II shows that our system generalizes well to unseen objects without any fine-tuning. Objects with a narrow edge, such as the Square-Circle and DoubleSquare, present more challenges during the grasping stage. Additionally, certain object shapes, such as the Star, SquareCircle, and Arch, present more challenges to pose estimation.

b) Data Efficiency: As Fig. 5 shows, ARCH demonstrates high data efficiency. Collecting teleoperated demonstration data is both costly and time-consuming, making it crucial to maximize the utility of each sample. Our approach outperforms previous methods in terms of sample efficiency, enabling the system to achieve effective learning with fewer training examples. Additionally, our system excels in multi-stage task success rates, indicating that it can maintain performance over extended sequences of actions. This capability not only reduces the overall data collection burden but also enhances the practical applicability of our method in real-world scenarios, where data acquisition can be a limiting factor.

c) Robustness: Our high-level policy exhibits robustness to human disturbances and facilitates failure recovery, allowing the robot to recover from unexpected situations absent in the demonstration data. For example, we observed that when the grasp primitive fails, the policy automatically triggers another grasp attempt using the new pose estimation.

d) Failure Cases: Although our hierarchical system performs well, long-horizon assembly remains challenging due to several factors. We have identified that failures are primarily caused by errors in grasp pose estimation and RL insertion primitive.

The sources of error include:

- Sensor: Particularly the depth camera, which impacts object/grasp pose estimation.
- Grasp Pose Estimation: Errors in estimating the precise position and orientation of the object and grasp can lead to failure initially and impact the overall assembly process.
- RL: Inherent uncertainties in the RL policy.

VI. Conclusion, Limitations, and Future Work

In this paper, we introduce a hierarchical hybrid learning system for long-horizon contact-rich robotic assembly. It includes a low-level parameterized skill library and an imitation-learned high-level policy for selecting and composing primitive skills. The hierarchical structure and the pre-trained primitive skills enable our system to be data efficient for long-horizon tasks, while satisfying the high-precision requirement of assembly.

Although our system demonstrates excellent performance and generalizability, there are limitations. For example, we manually identify the object beforehand. Incorporating an object detection method would be beneficial in the future. We would also like to extend to other types of assembly tasks and incorporate a more comprehensive primitive skill library.

References

[1] Alphonsus Adu-Bredu, Nikhil Devraj, and Odest Chadwicke Jenkins. Optimal constrained task planning as mixed integer programming, 2022. URL https://arxiv.org/abs/2211.09632.

[2] C. Agiaa, T. Migimatsu, J. Wu, and J. Bohg. Taps: Task-agnostic policy sequencing. *arXiv preprint arXiv:2210.12250*, 2022.

[3] K. S. Arun, T. K. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9 (5):698–700, 1987.

[4] R. Bohlin and L.E. Kavraki. Path planning using lazy prm. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 521–528 vol.1, 2000. doi: 10.1109/ROBOT.2000.844107.

[5] Yuanpei Chen, Chen Wang, Li Fei-Fei, and C. Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. *CoRL*, 2023.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[7] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[8] Henrik Christensen, Nancy Amato, Holly Yanco, Maja Mataric, Howie Choset, Ann Drobnis, Ken Goldberg, Jessy Grizzle, Gregory Hager, John Hollerbach, et al. A roadmap for us robotics–from internet to robotics 2020 edition. *Foundations and Trends® in Robotics*, 8(4):307–424, 2021.

[9] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics*, 2018.

[10] Yuval Cohen, Hussein Naseraldin, Atanu Chaudhuri, and Francesco Pilati. Assembly systems in industry 4.0 era: a road map to understand assembly 4.0. *The International Journal of Advanced Manufacturing Technology*, 105: 4037–4054, 2019.

[11] Aidan Curtis, Xiaolin Fang, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Caelan Reed Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. *CoRR*, abs/2108.04145, 2021. URL https://arxiv.org/abs/2108.04145.

[12] Aidan Curtis, George Matheos, Nishad Gothoskar, Vikash Mansinghka, Joshua Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Partially observable task and motion planning with uncertainty and risk awareness, 2024. URL https://arxiv.org/abs/2403.10454.

[13] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021.

[14] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning and acting*. Cambridge University Press, 2016.

[15] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[16] Tadanobu Inoue, Giovanni De Magistris, Asim Munawar, Tsuyoshi Yokoya, and Ryuki Tachibana. Deep reinforcement learning for high precision assembly tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 819–825, 2017. doi: 10.1109/IROS.2017.8202244.

[17] L. P. Kaelbling and T. Lozano-Perez. Hierarchical task and motion planning in the now. *ICRA*, 2011.

[18] L. P. Kaelbling and T. Lozano-Perez. Pre-image backchaining in belief space for mobile manipulation. *Robotics Research*, 2017.

[19] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 2012.

[20] George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *In Advances in Neural Information Processing Systems*, 2009.

[21] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim. Composing complex skills by learning transition policies. *In International Conference on Learning Representations*, 2019.

[22] Youngwoon Lee, Joseph J. Lim, Anima Anandkumar, and Yuke Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization. *CoRL*, 2021.

[23] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *IEEE Transactions on Robotics*, 2024.

[24] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: A functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

[25] Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. *CVPR*, 2024.

[26] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demon-

strations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

[27] Xiaofeng Mao, Gabriele Giudici, Claudio Coppola, Kaspar Althoefer, Ildar Farkhatdinov, Zhibin Li, and Lorenzo Jamone. Dexskills: Skill segmentation using haptic data for learning autonomous long-horizon robotic manipulation tasks. *arXiv preprint arXiv:2405.03476*, 2024.

[28] Warwick Masson and George Dimitri Konidaris. Reinforcement learning with parameterized actions. *CoRR*, abs/1509.01644, 2015. URL http://arxiv.org/abs/1509.01644.

[29] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. *CoRL*, 2023.

[30] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

[31] Andrew S Morgan, Bowen Wen, Junchi Liang, Abdeslam Boularias, Aaron M Dollar, and Kostas Bekris. Vision-driven compliant manipulation for reliable, high-precision assembly tasks. *arXiv preprint arXiv:2106.14070*, 2021.

[32] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[35] Jiankai Sun, Lantao Yu, Pinqian Dong, Bo Lu, and Bolei Zhou. Adversarial inverse reinforcement learning with self-attention dynamics model. *IEEE Robotics and Automation Letters*, 6(2):1880–1886, 2021.

[36] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, 2022.

[37] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10338–10347, 2021.

[38] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[39] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. *CoRL*, 2023.

[40] Yang You, Ruoxi Shi, Weiming Wang, and Cewu Lu. Cppf: Towards robust category-level 9d pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6866–6875, 2022.

[41] Yang You, Wenhao He, Jin Liu, Hongkai Xiong, Weiming Wang, and Cewu Lu. Cppf++: Uncertainty-aware sim2real object pose estimation by vote aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[42] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[43] Xiang Zhang, Masayoshi Tomizuka, and Hui Li. Bridging the sim-to-real gap with dynamic compliance tuning for industrial insertion. *ICRA*, 2024.