2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Generalize by Touching: Tactile Ensemble Skill Transfer for Robotic Furniture Assembly

Haohong Lin[1,2], Radu Corcodel[2] and Ding Zhao[1]

***Abstract*— Furniture assembly remains an unsolved problem in robotic manipulation due to its long task horizon and nongeneralizable operations plan. This paper presents the Tactile Ensemble Skill Transfer (TEST) framework, a pioneering offline reinforcement learning (RL) approach that incorporates tactile feedback in the control loop. TEST's core design is to learn a skill transition model for high-level planning, along with a set of adaptive intra-skill goal-reaching policies. Such design aims to solve the robotic furniture assembly problem in a more generalizable way, facilitating seamless chaining of skills for this long-horizon task. We first sample demonstration from a set of heuristic policies and trajectories consisting of a set of randomized sub-skill segments, enabling the acquisition of rich robot trajectories that capture skill stages, robot states, visual indicators, and crucially, tactile signals. Leveraging these trajectories, our offline RL method discerns skill termination conditions and coordinates skill transitions. Our evaluations highlight the proficiency of TEST on the in-distribution furniture assemblies, its adaptability to unseen furniture configurations, and its robustness against visual disturbances. Ablation studies further accentuate the pivotal role of two algorithmic components: the skill transition model and tactile ensemble policies. Results indicate that TEST can achieve a success rate of 90% and is over 4 times more efficient than the heuristic policy in both in-distribution and generalization settings, suggesting a scalable skill transfer approach for contact-rich manipulation.**

I. Introduction

Robotic furniture assembly is regarded as one of the most complex problems within the field of robotic manipulations given its contact-rich, long-horizon nature [1]–[5]. The contextual purpose of the objects and the associated sub-tasks that must be executed to succeed the overall task.

Figure 1 shows a typical real-world scenario where a robot is tasked to assemble two geometrically distinct by *functionally* identical objects: a three-legged and a four-legged table. The global tasks require the same set of robot skills: picking, insertion, and threading. A common way of assembling these skills in a working robotic platform is by Learning from Demonstration (LfD). LfD allows robots to learn a policy from humans or heuristic demonstrations. In real-world applications however, LfD is challenging due to its long task horizon and the multimodal nature of the observations, as shown in Figure 2(a).

The primary concern arises from the **multimodal inputs** that robots must rely on to observe their environment. With various sensor modalities, there's an inherent uncertainty in the provided data because not all modalities carry meaningful information at the same time during the task. The question then becomes: How can one ensemble multimodal inputs and make accurate predictions without the need for online interactions? The challenges do not end with sensor uncertainty. Robotic assembly tasks are implicitly **long-horizon** in nature. This means that robots need to plan, execute, and connect a series of relevant actions over an extended period of time to achieve the desired global outcome. Traditional LfD approaches, such as Behavioral Cloning (BC), often fall short in these scenarios. They lack the high-level skill awareness required for such complex tasks and struggle with generalization, especially when there is a change in certain sub-task modules in the deployment stage.

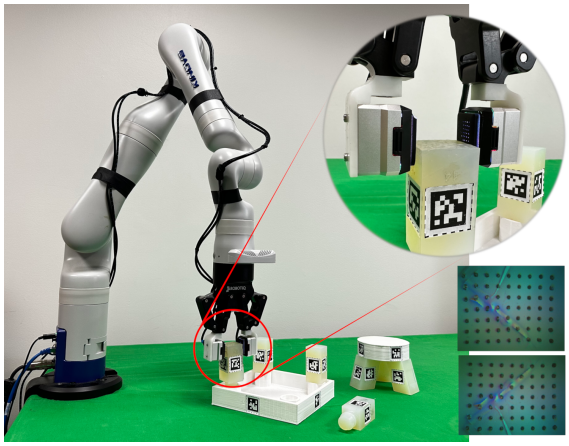


Fig. 1: Furniture assembly robot. A natural pipeline of such assembly tasks requires pick, reach, insert, adjust, and thread as the candidate skills to be learned.

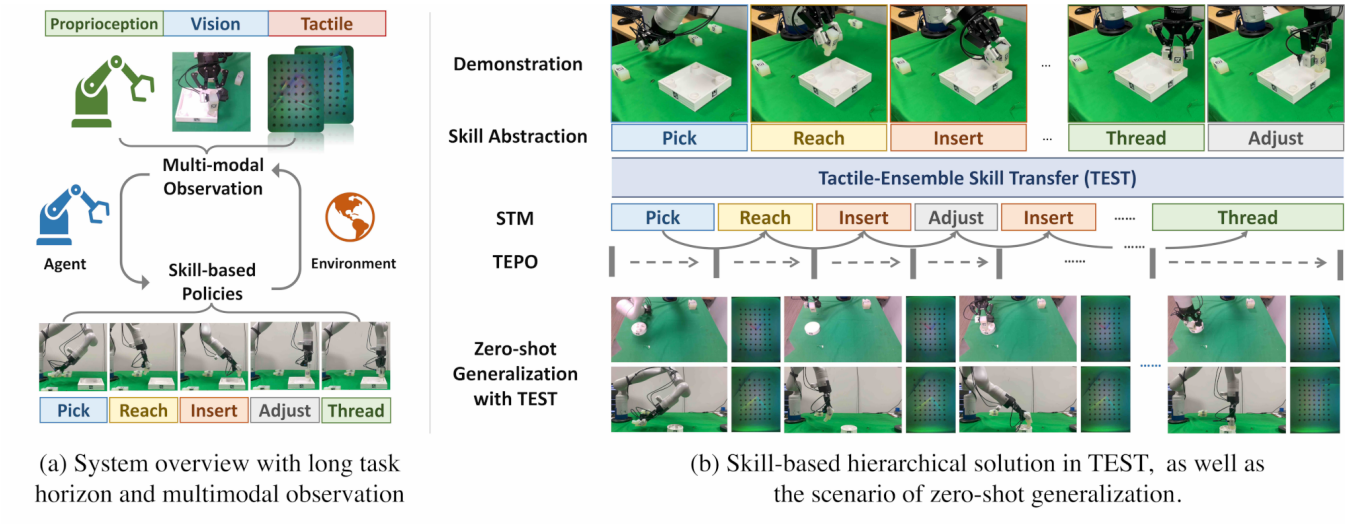
Given the aforementioned challenges, two primary solutions have been proposed. The first is to use ensemble policies that leverage self-supervised learning to counteract the uncertainties from multimodal sensors [6]–[8]. The second is to use hierarchical Reinforcement Learning (RL) methods to abstract and simplify long-horizon tasks [9]–[12].

In this paper, we introduce a new approach in Figure 2(b) that addresses both challenges simultaneously. We present the Tactile Ensemble Skill Transfer (TEST) framework, a unified solution to jointly model the human preferences (reward), multimodal observations, and actions of robotic assembly tasks. Our contributions to this work are listed below:

- We formulate robotic assembly as a skill-based RL problem over Goal-conditioned Partially Observable Markov Decision Process (GC-POMDP, in Section III) that describes

*This work was fully supported by Mitsubishi Electric Research Labs (MERL)

[1]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA, {haohongl, dingzhao}@cmu.edu
[2]Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139 USA, corcodel@merl.com



(a) System overview with long task horizon and multimodal observation

(b) Skill-based hierarchical solution in TEST, as well as the scenario of zero-shot generalization.

Fig. 2: Motivation and challenges in contact-rich robotic assembly problem.

the goal-reaching problem under multimodal sensor inputs instead of the fully observable settings.

- Building on this foundation, we introduce **T**actile **E**nsemble **S**kill **T**ransfer (TEST), a skill-based offline RL method that seamlessly integrates the strengths of ensemble learning with tactile feedback and skill-conditioned policy learning.
- We validate our approach with real-world experiments using the Furniture Bench platform [5], evaluating the accuracy and efficiency of learned policy. We also empirically study the generalizability of TEST towards unseen furniture assemblies, and consistency under visual disturbances, highlighting the significant improvements that characterized our framework.

II. Related Work

Contact-rich Robotic Manipulation Our furniture assembly problem originates in the field of contact-rich robotic manipulation. Computer Vision for robotic systems, while pivotal in parsing the semantic understanding of environments, cannot deliver robust information for contact-aware sensing needed to fully close the loop on intelligent robot assembly. This led to the integration of force/torque sensors and later, artificial tactile sensors [6], [13]–[15] which are crucial in robotic assembly tasks [11], [16], [17]. We highlight two particular methods of feedback control for robotic assembly. The first is an end-to-end deep reinforcement learning design, which integrates multiple sensor inputs into a unified framework, allowing for a more holistic understanding of the environment and the task at hand [18], [19]. The second is a hierarchical design with skill primitives plus task planning, where each task is broken down into a hierarchy of skills [10], [11], [16], [17] managed by specific scripted or learned controllers. This method gained more popularity in robotics research because it allowed for more generalizable and modular solutions.

Skill-based Reinforcement Learning The challenges of generalization and long-horizon tasks in robot learning led to hierarchical skill-based policies, often termed as *options* [12], [20], [21]. These policies, comprising a high-level skill planner and a low-level controller, aimed to reduce sample complexity and enhance interpretability. This framework proved especially beneficial for multi-modal decision-making in robotic manipulation [9]–[12]. Skill-based RL's primary challenges are skill discovery and skill chaining. Earlier works either utilized manually designed skill graphs [22] or employed unsupervised clustering-based methods [10], [23]. However, clustering often missed temporal information, and the learned policy's effectiveness depended heavily on cluster initialization and number. More recent works have explored parameterized skills using temporal abstraction. For instance, [23] evolved the policy gradient theorem, introducing a high-level gating policy and intra-option sub-policies. Subsequent works, such as [24], proposed end-to-end training methods, alternating between high-level *managers* and low-level *workers*. This approach was further refined by incorporating maximum-entropy RL [12], adversarial training [25], model-based RL [26], and meta RL [27]. Nevertheless, applying skill-based RL and planning frameworks onto the real robot systems is still an unsolved problem that has been marginally explored [7], [8], none of which addressed the contact-rich manipulation problem under tactile sensing.

Offline Reinforcement Learning Offline RL, also known as batch RL, focuses on learning policies from previously collected data without further interactions with the environment. This approach is crucial for scenarios where online data collection is expensive or risky. Several algorithms have been proposed in this domain, such as Conservative Q-Learning (CQL) [28] which aims to address overestimation in value functions. TD3-BC [29] combines the strengths of TD3, a popular actor-critic method, with Behavioral Cloning. The Decision Transformer [30], or DT, rethinks the RL paradigm by treating it as a sequence modeling problem, leveraging transformers to predict future rewards. Notably, DT tokenizes the reward, observation, and action from offline trajectories into different tokens, enabling the potential to incorporate

multimodal inputs from the observation space.

III. Problem Formulation

Task Objective The objective of TEST is to improve the quality of Learning from Imperfect Demonstration (LfID) for long-horizon robotic assembly tasks. Assume we have N skill primitives and a skill set denoted as $\{z^{(i)}\}_{i=1}^{N}$. We are given a skill-labeled offline dataset by some heuristic behavior policy $\pi_0^{(i)}$, where (i) refers to the skill index of z. In general, the objective of the furniture assembly task includes two parts: *accuracy* and *efficiency*. For the accuracy of assembly, we evaluate the accuracy via the Average Success Rate (ASR), i.e. ASR $= \frac{\text{\# tasks succeeded}}{\text{\# all tasks}}$, which indicates success in different assembly tasks or sub-tasks. For the efficiency of assembly, we evaluate the Average Steps (AS), where AS $= \frac{\text{\# timesteps}}{\text{\# skill phase}}$. To better evaluate the quality of the goal-reaching quality in the learned policy, we will also consider the Average Reward (AR) as one of the metrics.

Framework Formulation We formulate our problem in the Goal-conditioned Partially Observable Markov Decision Process (GC-POMDP) following the formulation of GC-MDP [31] and POMDP [32]. A GC-POMDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{G}, \Omega)$, where $\mathcal{S}$ is the state space, here we define states as the 6D pose of the objects of interest. $\mathcal{A}$ is the action space that indicates the target pose and movement of the end-effector. $\mathcal{O}$ is a finite set of observations, and our robotic assembly system, in fact, gives us multimodal observations $\boldsymbol{c} = [o^p, o^v, o^c]$, where o^p is the proprioceptive observation of the manipulator, o^v represents the vision observation from an external camera, and o^c refers to the contact-aware observation given by the tactile sensors. $\mathcal{P}$ is the state transition probability function. $\mathcal{G}$ is the goal space in the 6D pose of the objects to be assembled together, $G \subset S$. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, in practice our reward function is induced by the target goal $g \in G$. $\Omega : S \times A \to O$ is the observation function, which maps a state-action pair to an observation. It captures the probability of observing o after taking action a and ending up in state s', i.e., $\Omega(o|s', a)$. The objective in GC-POMDP is to find a policy that maximizes the expected cumulative reward $\mathbb{E}_\pi\left[\sum_{t=0}^{T} \gamma^t r_t | O_t\right]$ over time.

Additional Assumptions We further model our robotic assembly task by adopting the skill learning formulation [23] in the above GC-POMDP. We represent the skill-based RL problem as a tuple (I_z, π_z, β_z) associated with certain skill z. I_z is the initial set of states of skill z, $\pi_z = \pi(\cdot|o, z)$ is a goal-conditioned skill-conditioned policy, and $\beta_z : \mathcal{S} \to [0, 1]$ is a termination function of the skill z.

Firstly and most importantly, we assume the *invariance* of skill primitives across different assembly tasks. The skill primitives required to finish the assembly tasks during testing is the superset of skills demonstrated in the training environments, i.e. $z_{\text{train}} \subseteq z_{\text{test}}$. Secondly, we assume whenever the end-effector reaches the goal of skill z, the manipulator always has *smooth transition* to the next candidate skill in the assembly tasks, i.e. $\exists\ z'$, $\forall\ G_z = \{s|\beta_z(s) = 1\}$, $G_z \subset I_{z'}$.

IV. Methodology

In this section, we introduce our proposed framework TEST. We use a Skill Transition Model (STM), which learns the higher-level transition model $p(z'|z, \boldsymbol{c})$. Then for each sub-skill, we learn the intra-skill goal-reaching policies $\pi(\cdot|o, z)$ via Tactile Ensemble Policy Optimization (TEPO), which transforms offline RL into a sequential modeling problem with hindsight relabeling as data augmentation. We implement both STM and TEPO in an end-to-end Tactile Ensemble Skill Transformer, which is visualized in Figure 3. Lastly, we introduce the hierarchical skill transfer pipeline of TEST during the online deployment stage that aims the maximize the zero-shot performance in the target domain and improve robustness against sensor noise, *i.e.* image corruption.

Algorithm 1: TEST Training and Inference

Data: Number of Skills N, number of trajectories M, number of iterations K, maximum timesteps T, offline behavior policy $\{\pi_0^{(i)}\}_{i=1}^{N}$, step-wise penalty c, pose distance measure d_z, initial State Set I_z, terminal state condition β_z

Result: Optimized STM $\hat{p}_\theta$, skill policies $\{\hat{\pi}_\phi^{(k)}\}_{k=1}^{N}$

`/* TEST Hierarchical Training */`

Offline data collection: $\mathcal{D} = \{\tau\}_{m=1}^{M} \leftarrow \{\pi_0^{(i)}\}_{i=1}^{N}$;

for $k = 1, \cdots, K$ **do**

  Reward modeling:
  $r(s, g; z) = -r_{\text{penalty}} - d(s, g; z) + \alpha\mathbb{I}(s = g)$;

  Initialize $s_0 \sim I_z$;

  `/* Hindsight Relabeling */`

  Sample goal: $g \sim p_s(\tau)$;

  Relabel: $\tau' = \text{Relabel}(\tau)$ with (5);

  Data Augmentation: $\mathcal{D} \cup \{\tau'\}$;

  for $m = 1, \cdots, M$ **do**

    Sample context $\boldsymbol{c}_t' \sim \mathcal{B}_k$ with a horizon H:
    $\boldsymbol{c}_{t'} \leftarrow \{R_t, o_t^p, o_t^v, o_t^c, s_t, a_t\}_{t=t'-H+1}^{t'} \backslash \{a_{t'}\}$;

    `/* STM */`

    Next skill sampling: $z_{t'} \sim \hat{p}_\theta(\cdot|z_{t'-1}, \boldsymbol{c}_{t'})$;

    Update STM $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{STM}}$ with (4);

    `/* TEPO */`

    Skill-conditioned policy:
    $a_{t'+1} \sim \pi_\phi(a|\boldsymbol{c}_{t'}, z_{t'})$;

    Update TEPO $\phi \leftarrow \phi - \eta\nabla_\phi\mathcal{L}_{\text{TEPO}}$ with (6);

`/* TEST Hierarchical Inference */`

$z \leftarrow z_0$;

Initialize $s_0 \sim I_z$;

for $t = 1, \cdots, T$ **do**

  `/* Skill-based policy rollout */`

  while $!\beta_z(s_i)$ **do**

    $a_i \leftarrow \arg\max_a \pi_\phi(a|\boldsymbol{c}_i, z)$;

    $o_i, r_i, \beta_z(s_i) \leftarrow \text{env.step}(a_i; z)$;

    $\hat{R}_i \leftarrow \max(\hat{R}_{i-1} - r_i, 0)$;

    $\boldsymbol{c}_i$.update($\{\hat{R}_i, o_i, a_i\}$);

    $i \leftarrow i + 1$;

  `/* Switch skill at termination */`

  STM Prediction: $z \leftarrow \arg\max_{z'} \hat{p}_\theta(z'|z, \boldsymbol{c}_i)$;

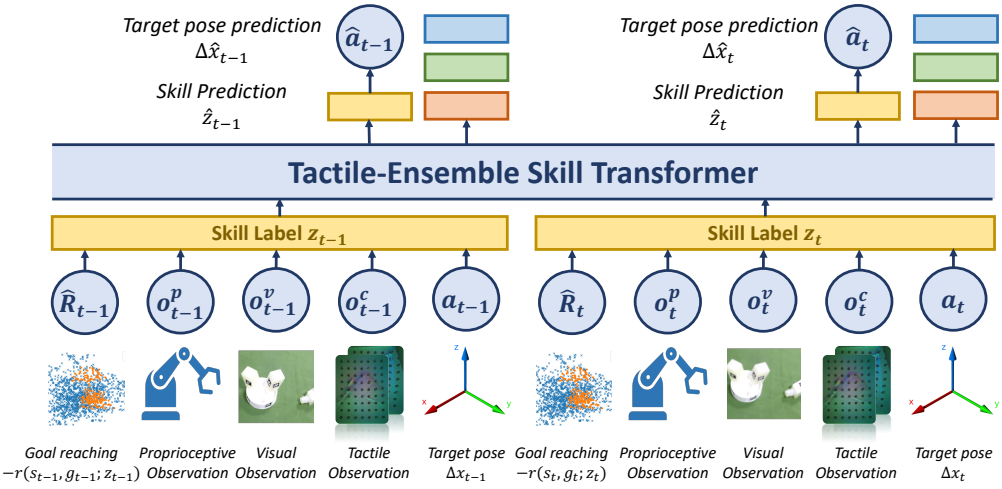


Fig. 3: Diagram of TEST. We tokenize the input sequence with reward-to-go, proprioceptive observation, visual observation, followed by a tactile observations. The input bundle will predict the target pose as the action for the current timestep. At each step, the inputs are aggregated to predict the state at the current timestep.

A. Learning the Skill Transition Model from Offline Data

Our goal is to learn the State Transition Model as an inter-skill transition model that operates at a high level, focusing on how different skills or sub-tasks can be chained together to achieve a complex, long-horizon task.

We have an input trajectory $\tau = \{\tau_i\}_{i=1}^{T}$ with a skill horizon T. For each $i \in [1, T]$, we have

$$\tau_i = \{o_0, a_0, r_0, s_1, \cdots, o_{T-1}, a_{T-1}, r_{T-1}; g\} \tag{1}$$

The step reward in our method is goal-conditioned, labeled by the sequential information from demonstrated trajectories:

$$r(s_t, g_t; z) = \underbrace{-r_{\text{penalty}}}_{\text{Time Penalty}} \underbrace{-d(s_t, g_t; z)}_{\text{Distance to Goal}} + \underbrace{\alpha \mathbb{I}(s_t = g_t)}_{\text{Arrival Bonus}}, \tag{2}$$

where $g_t = \{s_{t'} | \max_{t'} t' < t,\ s.t.\ \beta_z(s_{t'}) = 1\}$, which is the last demonstration that satisfies the termination condition β. Following the autoregressive structure in [30], every future z will depend on a context $\boldsymbol{c}$ of trajectory history,

$$\boldsymbol{c}_t = \{R_{t-H+1}, o_{t-H+1}, a_{t-H+1}, \cdots R_t, o_t, a_t\}, \tag{3}$$

where $R_t = \sum_{t' \geq t} r_{t'}$ is the summation of the future reward till the end of the episode, denoted as reward-to-go. The inter-skill transition determines the sequence in which different skills should be executed, ensuring smooth execution between consecutive trajectories of the skills. The STM follows a categorical distribution: $p_\theta(z'|z, \boldsymbol{c}) = \text{Categorical}(\ell_\theta(z, \boldsymbol{c}))$, where $\ell_\theta(\cdot, \cdot)$ is the output logits of decoder output followed by the Skill Transformer's encoder, as shown in the skill prediction block of Figure 3. It also considers potential dependencies between skills, ensuring that prerequisite tasks are completed before dependent ones.

In the demonstration collection phase, we randomly sample from the heuristic policy with a Finite State Machine (FSM). We then fit the skill transfer based on the trajectory observation $\boldsymbol{c}$ and current skill z by minimizing the negative log-likelihood loss:

$$\mathcal{L}_{\text{STM}} = \mathbb{E}_{\tau \sim \pi_0} \mathbb{E}_{z \sim \tau} \Big[-\log p_\theta(z'|z, \boldsymbol{c}) \Big], \tag{4}$$

By leveraging tactile feedback and ensemble learning, the inter-skill policy can make real-time decisions about what would be the most likely next skills to perform, enabling diverse way of skill composition in online deployment.

B. Skill-conditioned Goal-reaching Policy Optimization

The Tactile Ensemble Policy Optimization (TEPO) module in the TEST framework is designed to learn a skill-conditioned goal-reaching policy $\pi(a|\boldsymbol{c}, g, z)$, where the goal is implicitly induced by $g = \{s|\beta_z(s) = 1\}$. Without loss of generality, we still denote $\pi(a|\boldsymbol{c}, g, z) \triangleq \pi(a|\boldsymbol{c}, z)$. We parameterize our action distribution by the output logits as follows a Gaussian distribution: $\pi_\theta(a|\boldsymbol{c}, z) = \mathcal{N}\Big(\mu_\theta(\boldsymbol{c}, z), \Sigma_\theta(\boldsymbol{c}, z)\Big)$.

Intuitively, TEPO learns a goal-reaching policy at the sub-skill level. Although the horizon is significantly shortened compared to directly learning over the entire horizon of tasks, the rewards could still be sparse, being provided only when the exact goal is achieved. This sparsity can adversely affect learning, especially in our offline settings where the robot cannot interact with the environment to gather more data. Therefore, we conduct an additional goal relabeling strategy for TEPO training. For the input sub-skill trajectory τ_k corresponding to z_k introduced in (1), the original $g \in \{s|\beta_{z_k}(s) = 1\}$. We then resample the goal states from those in trajectories τ_k,

$$\begin{aligned} &\text{Goal Relabeling: } g' \sim p_s(\tau_k) \\ &\text{Reward Relabeling: } r'_t = r(s, g'; z_k), \end{aligned} \tag{5}$$

where $p_s(\cdot)$ is the empirical marginal state distribution of the input trajectories. After the hindsight relabeling, we can generate multiple relabeled trajectories $\tau'_k = \{o_0, a_0, r'_0, s_1, \cdots, o_{T-1}, a_{T-1}, r'_{T-1}; g'\}$, which diversifies the step reward, and corresponding reward-to-go predictions

for identical historical sequences, improving goal scenario generalization. After the data augmentation with hindsight relabeling, we get the augmented trajectories s. Given the offline demonstration, TEPO aims to minimize the following negative log-likelihood loss with an entropy regularizer [33]:

$$\mathcal{L}_{\text{TEPO}} = \mathbb{E}_{\tau_z \sim \pi_0^z}\Big[-\log \pi_\phi(a|\boldsymbol{c}, z) - \lambda H[\pi_\phi(\cdot|\boldsymbol{c}, z)]\Big]. \quad (6)$$

where λ is the weight of the regularizer. The learned low-level policy $\pi(a|\boldsymbol{c}, z)$ takes into account both the context $\boldsymbol{c}$ consisting of multimodal observations, goal preference, and skill representation z. Through a combination of tactile feedback and ensemble learning, the intra-skill policy optimizes the trajectory in real-time, ensuring that the robot can adapt to changes and uncertainties in the environment. This adaptability is crucial for tasks like *insertion* that require fine motor skills, such as aligning parts with tight tolerances. The training pipeline is summarized as a pseudocode in Algorithm 1. After we train the TEST model with STM and TEPO in an alternative optimization, we apply hierarchical inference at the online deployment stage to further improve the performance of TEST. As illustrated in Algorithm 1, TEST conducts hierarchical inference between the skill-conditioned goal-reaching policies and skill transition predictions. TEST follows the transformer structure of GPT-2 [34].

V. Experiments

A. Environment Design

Tasks Design: We design our furniture assembly platform based on the FurnitureBench set [5]. Specifically, we select the one-leg assembly of the square table, the most widely studied environment as our in-distribution setting. For the generalization setting, we selected a different furniture stool with a different leg geometry and different angles of threading. Similar to the in-distribution setting, here we only consider the one-leg assembly task for real-robot evaluation.

Hardware Setup: As is illustrated in Figure 4, to support our hierarchical decision-making systems with multimodal sensory inputs, we use a Kinova Gen3 collaborative robot arm as the manipulator to which we instrumented its gripper fingers with tactile sensing devices. We chose the Gelsight Mini [35], a type of optical-based tactile sensor with excellent optical resolving power. We are also using the optional tracker gel pads which give us additional feedback about the slipping state of the grasped parts. The sampling frequency is 15 Hz on the external camera and 30 Hz on the tactile sensors on both fingers. In the low-level control, we use position control in the end-effector's Cartesian space at 100 Hz which allows us to generate smooth interpolated trajectories.

Real-world Offline Data Collection: In the real-world experiments, we use the heuristic policies $\pi_0^{(i)}$, where $z^{(i)} \in \{\text{pick, reach, insert, screw, adjust}\}$. The skill is parameterized by the starting pose and goal 6D pose of the end effector. During the data collection phase, we use AprilTags to represent the objects' state, then use the estimated state to design goal-reaching policies with some randomness. To guarantee the safety of the real robot, we actively detect the violation of safe contact constraints by the movement of the tactile sensor's markers. Specifically, we use the optical flow to detect such violations. We collected a total of 2,000 trajectories for all the skills and this heuristic policy is fully operated in the real world.

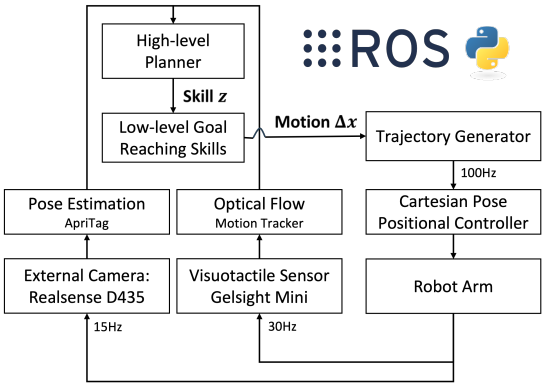


Fig. 4: Hardware system setup for the experiments.

Method	Square Table / Stool		
	AR (↑)	ASR (↑)	AS (↓)
BC+GSA	0.59 / 0.14	0.3 / 0.0	36.8 / 80.0
LSTM+GMM	1.02 / 1.05	0.8 / 0.5	22.4 / 24.4
TEST w/o STM	0.33 / 0.16	0.4 / 0.2	49.0 / 76.4
TEST w/o TEPO	0.85 / 1.12	0.7 / 0.4	62.0 / 66.4
TEST (Ours)	**1.64 / 1.52**	**0.9 / 0.9**	**16.4 / 14.0**
Heuristic Policy	1.00 / 1.00	0.7 / 0.7	67.1 / 64.0

TABLE I: Results on online assembly of square table, and generalization setting for the stool. The evaluation results average over 10 episodes on our furniture assembly robot. The **Bold** means the best results among all.

Noise Level (cm)	**Evaluation Metrics**		
	Reward (↑)	ASR (↑)	AS (↓)
0.1	1.58	0.9	28.8
0.2	1.34	0.9	41.2
0.5	1.09	0.8	64.4
1.0	0.68	0.5	77.2
0.0	**1.64**	**0.9**	**16.4**

TABLE II: Experiment results on TEST's robustness under disturbances. We add Gaussian noise to the state prediction from vision inputs. The noise level indicates the standard deviation of the target position (on the x, y, and z-axis).

B. Baselines

We compare the performance of TEST with the following baselines and the variants of TEST. **BC+GSA** [10], or Behavior Cloning with Generalized State Abstraction applies unsupervised clustering for state abstraction and hierarchical policy learning based on the input offline data in the entire task horizon. **LSTM+GMM** [36], uses Long Short-term Memory and Gaussian Mixture Model, aiming to capture both the sequential nature of the demonstrations and variability in

the actions across the states. It's particularly designed for long-horizon LfD problems with multimodal observations. **TEST w/o STM**, is an ablation variant on TEST by removing the skill transition model. **TEST w/o TEPO** is an ablation variant on TEST by removing the tactile ensemble in policy optimization. **Heuristic Policy**: a set of heuristic policies $\{\pi_0^{(i)}\}_{i=1}^N$ that are used to collect data. Safe yet conservative, the heuristic policies have access to privileged information on the skill label during the evaluation stage.

C. Evaluation Protocol

As mentioned in Section III, we compare the following metrics: AR, ASR, and AS. The original definition of the reward is given by (2). Based on our furniture assembly robot, as visualized in Figure 1, we evaluate all the experiments with an average of 10 episodes. The maximum number of skills per episode is 80, and we count each skill as a step in the AS metric. For the AR, we normalize the average return with respect to the heuristic policy π_0.

D. Results and Analysis

In this part, we answer the following research questions:

- **RQ1**: What is the in-distribution performance of TEST compared to the heuristic policies and other baselines?
- **RQ2**: What is TEST's generalization performance towards unseen furniture, compared to other baselines?
- **RQ3**: How is the robustness of TEST under noise disturbances in the observation space, e.g. image corruption?

We illustrate our key findings on the above three research questions in Table I, II and Figure 5, 6.

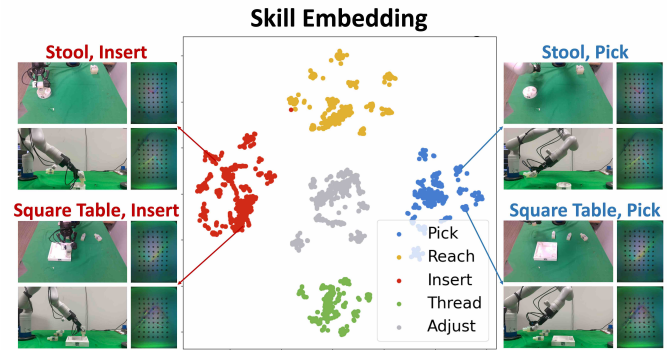


Fig. 5: Visualization of the embedding of TEST's learned skills t-SNE. We see a clear cluster in each sub-skill, and rollout trajectories in the stool and square table align well.

For **RQ1**, TEST outperforms BC+GSA and LSTM+GMM with higher accuracy and efficiency in the long-horizon assembly tasks in the square table. For **RQ2**, TEST still manages to generalize and outperform the Heuristic Policy even if it does not access the direct skill labels in the unseen furniture stool. Compared to other baselines, TEST has the lowest performance drop and the highest success rate compared to the other baselines. For **RQ3**, we manually add the Gaussian noise from the prior pose estimation from vision observation o^v and evaluate TEST's performance in

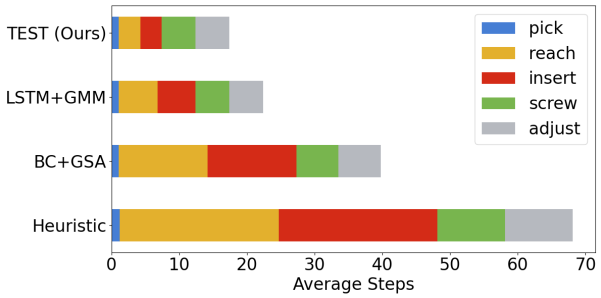


Fig. 6: Bar plot for comparison on the Average Steps (↓) of each sub-skill between four methods. We can see that TEST significantly outperforms other baselines and the heuristic behavior policy π_0.

the square table environment. Though the efficiency of TEST drops significantly, the accuracy is still maintained to some extent. Attributed to the multi-modal design, TEST is actually robust under mild disturbances in the visual observation.

We also provide ablation studies by removing key modules in TEST. TEST w/o STM only conducts the tactile ensemble in policy optimization for the long task horizon, which indicates the agent can hardly generalize even with contact awareness, if the skill transition is not properly represented. TEST w/o TEPO keeps the hierarchical skill-based learning structure while removing the tactile signals in policy optimization. This reinforces the idea that the addition of tactile sensors improves the performance of contact-rich manipulation.

In addition, to understand the effectiveness of TEST's design, we scatter the embedding of skill representation in Figure 5 to verify the invariance of skills between different furniture configurations. We also analyze the efficiency in each sub-skill in Figure 6. The results show that *pick* is the easiest skill, while *insertion* is the hardest one where TEST outperforms baselines with the clearest margin.

VI. Conclusion

In this work, we introduced TEST, a hierarchical, skill-based offline reinforcement learning framework tailored for robotic assembly tasks. TEST emphasizes the integration of tactile feedback, enhancing the contact-aware decision-making of robotic agents. At its core, TEST employs a Skill Transition Model, parameterized by a trajectory-level transformer for inter-skill transitions, and leverages Hindsight goal relabeling for intra-skill policy learning. Comprehensive evaluations on a furniture assembly robot underscore TEST's superiority over existing LfID baselines.

The limitation of TEST is that it still assumes access to accurate skill labels in offline data, which may not always be available in more general teleoperation cases. The assumption that skill primitives seamlessly integrate is another simplification, overlooking potential mismatches between consecutive skill states. Additionally, as furniture part geometries become complicated, the challenge of effectively fusing tactile imprints with marker movements for decision-making emerges as a promising direction for future research.

REFERENCES

[1] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[2] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[3] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[4] 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)*. IEEE, 2021, pp. 6343–6349.

[5] 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.

[6] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8943–8950.

[7] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," *arXiv preprint arXiv:2302.12422*, 2023.

[8] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," *arXiv preprint arXiv:2307.09955*, 2023.

[9] C. Daniel, G. Neumann, and J. Peters, "Hierarchical relative entropy policy search," in *Artificial Intelligence and Statistics*. PMLR, 2012, pp. 273–281.

[10] R. Akrour, F. Veiga, J. Peters, and G. Neumann, "Regularizing reinforcement learning with state abstraction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 534–539.

[11] Z. Hou, J. Fei, Y. Deng, and J. Xu, "Data-efficient hierarchical reinforcement learning for robotic assembly control applications," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 11, pp. 11 565–11 575, 2020.

[12] K. Pertsch, Y. Lee, and J. Lim, "Accelerating reinforcement learning with learned skill priors," in *Conference on robot learning*. PMLR, 2021, pp. 188–204.

[13] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[14] S. Luo, N. F. Lepora, U. Martinez-Hernandez, J. Bimbo, and H. Liu, "Vitac: Integrating vision and touch for multimodal and cross-modal perception," *Frontiers in Robotics and AI*, p. 134, 2021.

[15] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6437–6443.

[16] K. Nottensteiner, A. Sachtler, and A. Albu-Schäffer, "Towards autonomous robotic assembly: Using combined visual and tactile sensing for adaptive task execution," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, p. 49, 2021.

[17] B. Belousov, B. Wibranek, J. Schneider, T. Schneider, G. Chalvatzaki, J. Peters, and O. Tessmann, "Robotic architectural assembly with tactile skills: Simulation and optimization," *Automation in Construction*, vol. 133, p. 104006, 2022.

[18] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[19] Y. Chen, A. Sipos, M. Van der Merwe, and N. Fazeli, "Visuo-tactile transformers for manipulation," *arXiv preprint arXiv:2210.00121*, 2022.

[20] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[21] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *Advances in neural information processing systems*, vol. 22, 2009.

[22] ——, "Skill discovery in continuous reinforcement learning domains using skill chaining," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22. Curran Associates, Inc., 2009. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2009/file/e0cf1f47118daebc5b16269099ad7347-Paper.pdf

[23] A. Srinivas, R. Krishnamurthy, P. Kumar, and B. Ravindran, "Option discovery in hierarchical reinforcement learning using spatio-temporal clustering," *arXiv preprint arXiv:1605.05359*, 2016.

[24] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "FeUdal networks for hierarchical reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3540–3549. [Online]. Available: https://proceedings.mlr.press/v70/vezhnevets17a.html

[25] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, "Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization," *arXiv preprint arXiv:2111.07999*, 2021.

[26] L. X. Shi, J. J. Lim, and Y. Lee, "Skill-based model-based reinforcement learning," *arXiv preprint arXiv:2207.07560*, 2022.

[27] T. Nam, S.-H. Sun, K. Pertsch, S. J. Hwang, and J. J. Lim, "Skill-based meta-reinforcement learning," *arXiv preprint arXiv:2204.11828*, 2022.

[28] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[29] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20 132–20 145, 2021.

[30] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[31] P. Hansen-Estruch, A. Zhang, A. Nair, P. Yin, and S. Levine, "Bisimulation makes analogies in goal-conditioned reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8407–8426.

[32] W. S. Lovejoy, "A survey of algorithmic methods for partially observed markov decision processes," *Annals of Operations Research*, vol. 28, no. 1, pp. 47–65, 1991.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[35] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[36] F. Pastor, J. García-González, J. M. Gandarias, D. Medina, P. Closas, A. J. García-Cerezo, and J. M. Gómez-de Gabriel, "Bayesian and neural inference on lstm-based object recognition from tactile and kinesthetic information," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 231–238, 2020.