MimicTouch: Leveraging Multi-modal Human Tactile Demonstrations for Contact-rich Manipulation

Anonymous Author(s)
Affiliation
Address
email

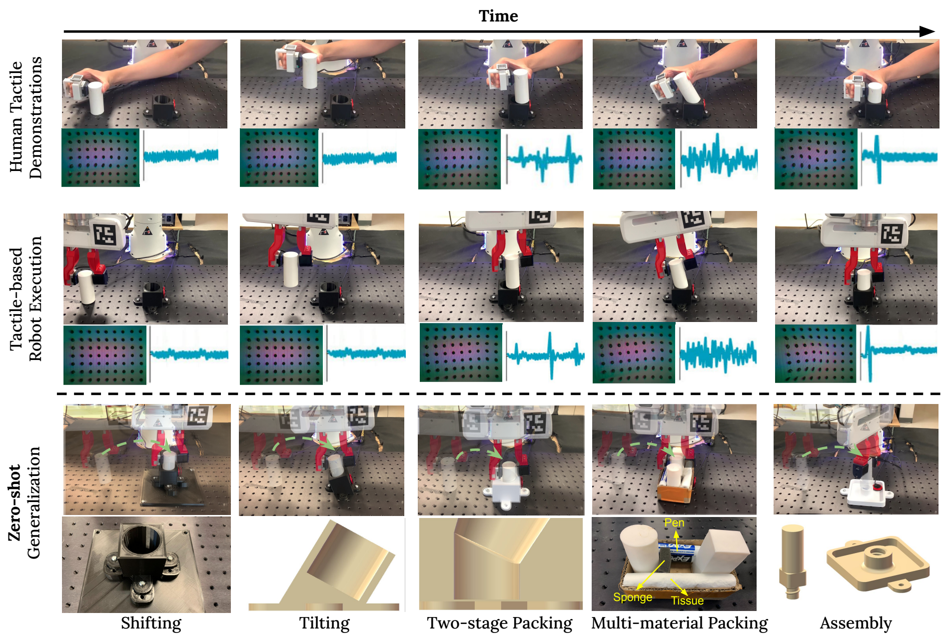

Figure 1: The first row shows the human tactile demonstrations, including the tactile and proprioception data. The second row shows the robot execution with tactile feedback. The third row below the dashed line describes the policy's **zero-shot** generalization capability in five different domains.

Abstract: Tactile sensing is critical to fine-grained, contact-rich manipulation tasks, such as insertion and assembly. Prior research has shown the possibility of learning tactile-guided policy from teleoperated demonstration data. However, to provide the demonstration, human users often rely on visual feedback to control the robot. This creates a gap between the sensing modality used for controlling the robot (visual) and the modality of interest (tactile). To bridge this gap, we introduce "MimicTouch", a novel framework for learning policies directly from demonstrations provided by human users with their hands. The key innovations are i) a human tactile data collection system which collects multi-modal tactile dataset for learning human's tactile-guided control strategy, ii) an imitation learning-based framework for learning human's tactile-guided control strategy through such data, and iii) an online residual RL framework to bridge the embodiment gap between the human hand and the robot gripper. Through comprehensive experiments, we highlight the efficacy of utilizing human's tactile-guided control strategy to resolve contact-rich manipulation tasks. The project website is at https://sites.google.com/view/MimicTouch.

Keywords: Tactile Sensing, Learning from Human, Imitation Learning

1 Introduction

Enabling robots to perform contact-rich tasks such as insertion remains a formidable challenge in robotics. The primary reason is the complex dynamic interaction between the robot and the object

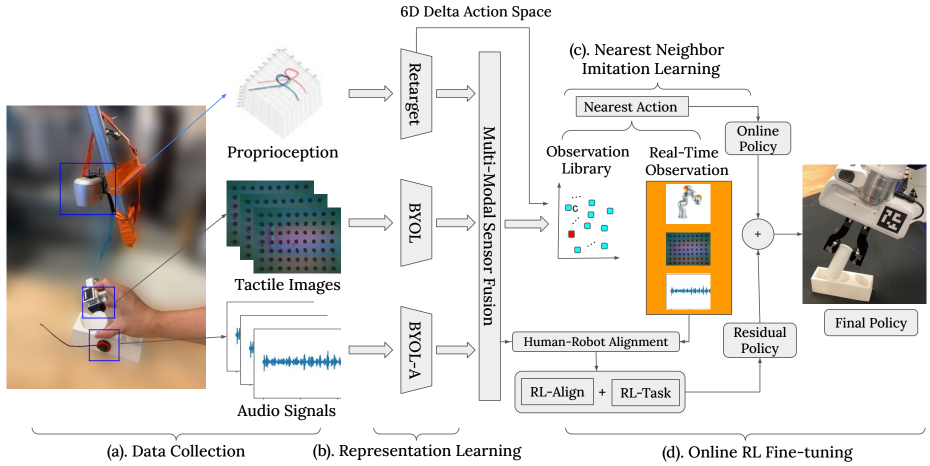

Figure 2: Illustration of the MimicTouch Framework. In part (a), we collect the multi-modal human tactile demonstrations. In part (b), we learn compact low-dimensional tactile representations. In part (c), we derive an offline policy through a non-parametric imitation learning method. In part (d), we refine the offline policy through online residual reinforcement learning on a physical robot.

which is influenced by various factors including intricate material properties and low tolerance for error. This necessitates an adaptive, data-centric insertion mechanism that utilizes real-time sensor feedback. Recent methods have heavily explored vision-based solutions to tackle this problem [1–4]. Notably, NVIDIA's sim-to-real transfer approach [1] achieves success rates of up to 99.2% in transferring assembly tasks using their customized "Factory" simulator [2]. However, vision-based approaches can fall short when visual feedback is compromised by cluttered occlusions or bad lighting conditions.

Humans exhibit fine-grained manipulation skills through tactile sensing, which allows for successful insertions by solely using tactile feedback to generate complex, continuous, and precise motions [5]. Motivated by this, recent studies collect demonstrations that combine various sensory inputs for policy learning [6–8]. However, these methods assume a limited action space, e.g., only 3D translations, to compensate for the complexity of collecting dynamic demonstrations. They also heavily rely on robot teleoperation systems [9–11], which inevitably creates a gap between the visual sensing used for data collection and the recorded tactile sensing for policy learning. As a result, these methods are unable to emulate human's tactile-guided control strategy for contact-rich tasks.

To tackle these challenges, we present "MimicTouch" (shown in Fig. 2), a novel framework that enables robots to learn human's tactile-guided control strategy. Specifically, MimicTouch first introduces a human tactile-guided data collection system to gather multi-modal tactile feedback (tactile + audio) directly from human demonstrators. Next, it incorporates a representation learning model to capture task-specific sensor input features. These compact representations enhance the performance of subsequent imitation learning by abstracting essential sensory information. Then, it employs a non-parametric imitation learning method [12] to derive an offline policy from the collected human tactile demonstrations. Finally, it leverages online residual reinforcement learning (RL) to fine-tune the offline policy on the physical robot, aiming to bridge the embodiment gap between the human hand and the robot gripper and enrich contact reasoning.

We conduct comprehensive experiments on contact-rich insertion tasks to evaluate the offline policy derived from demonstrations and the final policy fine-tuned via RL. We find that MimicTouch can collect tactile demonstrations more efficiently than teleoperation. More importantly, the MimicTouch policy can be effectively learned from such demonstrations and significantly outperforms the policy learned from teleoperation data. Additionally, we set up seven generalization tasks in five different domains and show the final policy exhibits superior **zero-shot** generalization capability.

2 Related Works

Multi-modal tactile sensing. Vision-based tactile sensing is integral to robotic manipulation as it excels at estimating local contact geometry and frictional properties [13–16]. It further enables imitation learning-based methods through policy observations [6, 7, 17] or reinforcement learning methods through reward signals [18–20] for various complex manipulation tasks. Additionally, it is also widely used in shape reconstruction [21–23] and grasping [24–27]. On the other hand, audio-based tactile sensors, such as contact microphones have also been demonstrated effective in robotics applications such as manipulation [28], classification [29, 30], and dynamics modelling [31, 32]. These sensors can emulate nerve endings within human skin to better detect vibrations during tactile interactions. Therefore, incorporating both sensor modalities can yield tactile feedback more akin to human sensations, enabling the robot to learn a human-like tactile-guided control strategy.

Learning from human demonstrations. Learning from human demonstrations is a long-standing research topic. One group of methods learns the robot behaviors from human videos [33–35]. However, these methods adopt only visual sensing, which can be easily collected by low-cost cameras or accessed via online videos. As a result, they mostly focus on high-level scene reasoning rather than fine-grained, contact-rich tasks, which often require tactile feedback for reliable execution. To incorporate the tactile data into the demonstrations, recent works have turned to using robot teleoperation systems [6–8, 36–38], where a robot is equipped with all necessary sensors, including tactile sensors, and is directly controlled by a human operator during task execution. This multi-modal dataset will then be used to train robot policies. However, human operators must guide the robot using visual feedback, thereby creating a gap between the visual sensing used for data collection and the tactile sensing recorded for policy learning. Furthermore, collecting $6D$ dynamic motions for contact-rich manipulations via teleoperation is challenging. Therefore, in this work, we propose to collect human tactile demonstrations, in which the sensing gap is addressed and the demonstration motions are more versatile and dynamic.

Imitation learning. Offline imitation learning (IL) is an effective strategy to learn robot policies in the real world. We consider two classes of IL methods: parametric methods [39–41] and non-parametric methods [12, 42, 43]. Parametric methods typically train neural networks to map observations to expert actions. While general in principle, they are prone to covariant shift and compounding errors [44]. Our method instead adopts a non-parametric imitation learning method. These methods constrain robot behaviors to the demonstrated data via techniques such as nearest-neighbor lookup [12]. While they may be less general, they offer a safer alternative to their parametric counterparts, which is crucial for the real-world contact-rich manipulation tasks considered in this work.

3 MimicTouch Framework

We aim to enable the robot to resolve contact-rich manipulation tasks by learning control strategy from human tactile demonstrations. To achieve this, we propose a novel learning framework named "MimicTouch". It first introduces a human tactile-guided data collection system (Sec. 3.1) to collect a multi-modal tactile dataset from human demonstrators. Then, to emulate the human's tactile-guided control strategy for successful robot execution, MimicTouch has three distinct learning phases. Firstly, it learns lower dimensional tactile representations from the human tactile demonstrations in a self-supervised manner (Sec. 3.2). Next, it derives an offline policy with the learned representations using a non-parametric imitation learning method [12] (Sec. 3.3). Lastly, it refines the offline policy through online residual reinforcement learning (Sec. 3.4). Note that this refinement phase is efficient and reliable as the offline policy encodes human's tactile-guided control strategy. The overall MimicTouch framework is shown in Fig. 2.

3.1 Collecting Human Tactile Demonstrations

To collect tactile demonstrations, current teleoperation systems have three key limitations: i). limited scalability due to the need for a robot to collect demonstrations [10], ii). long training time and expertise to become proficient with the system for fine-grained manipulation, and iii). sensing gap between the visual sensing used for collection and recorded tactile sensing. To address these, our key innovation is a system that enables humans to provide tactile demonstrations with their hands. The system is elaborated in Fig. 6 (Appendix. A), and it collects the pose of human fingertips, tactile images, and audio signals when human demonstrators perform contact-rich insertion tasks.

The data collection system consists of the following components. We use the RealSense camera with Aruco Marker [45] for human fingertip pose tracking. The tracked fingertip poses are then treated as the robot end-effector's poses after calibration and filtering (Appendix. C.2). We also use the GelSight Mini [46], a compact vision-based tactile sensor that is conveniently mounted onto human fingertips using a custom fixture, to estimate the contacts between the object and the fingertip.

Notably, we only use one tactile sensor in our experiment setup instead of two. The Audio data, which is helpful for manipulation tasks due to its sensitivity to contact vibration signals [28, 47], is captured using the HOYUJI TD-11 piezo-electric contact microphone. Considering the potential discrepancies in the mechanical vibrations between the human and the robot, the microphone is placed at the base of the insertion hole to ensure signal consistency.

3.2 Learning Tactile Representation

The policy learning on high-dimensional sensor inputs struggles with real-world deployment due to computational burden and sensor noise. Additionally, in this work, variations may exist between sensor inputs from human tactile demonstrations and real-time robot sensor feedback due to discrepancies in finger-object contact force. Therefore, inspired by recent works that learn lower-dimensional embeddings for image-guided imitation learning [6, 12, 48] which can discover the appropriate features that are helpful for policy learning, we first learn the compact representation for both tactile and audio sensor data using self-supervised learning methods (part (b) in Fig. 2). Intuitively, it identifies a low-dimensional embedding space where differently augmented images, such as tactile images or audio spectrum, are projected to a similar embedding. As a result, these embeddings are more efficient for online processing and more robust to task-irrelevant sensor noise. This learning phase consists of the following two parts.

Data collection. We collect task-specific tactile-audio data from the human demonstrator. The dataset encompasses successful, failed, and sub-optimal demonstrations. For each, we segment the audio data at 2Hz. In total, we collect 7657 tactile images and 1,000 audio segments. More details are shown in Appendix. B

Self-supervised learning. We employ the Bootstrap Your Own Latent (BYOL) [49] for tactile images and BYOL for audio (BYOL-A) for audio segments [50], since they have demonstrated desired performance in computer vision [49], audio representation [50], and robotics [7, 28, 36] tasks. Details about BYOL and BYOL-A are included in the Appendix. B.

3.3 Learning Offline Policy from Human Tactile Demonstrations

Leveraging the learned representations, we then learn the robot policy from the human tactile demonstrations. Here, one unique challenge is that the human hand moves much faster than the robot, resulting in sparse temporal observation-action samples (i.e., large action values per observation). Also, human tactile demonstrations might partially be out-of-domain (OOD) demonstrations for robot policy due to the embodiment gap (e.g., different motion capability and finger-object contact forces). Therefore, executing a parametric policy might exhibit unreasonable robot behaviors as they are prone to covariant shift [44] (see Sec. 4.2.1 for experimental validation). As a result, we use a non-parametric imitation learning method [12] to ensure the execution efficacy of the policy learned from human tactile demonstrations (part (c) in Fig. 2). In addition to the learning algorithm, data pre-processing is necessary for synchronization and the details are included in Appendix. C.

Non-parametric imitation learning. We build our algorithm on the VINN framework [12] by extending it to tactile-audio representation. At the i-th time step, the observations and actions are denoted as $(\mathrm{o}_i^T, \mathrm{o}_i^A, \mathrm{o}^{EE}, \mathrm{a}_i)$, where o^T is the tactile representation, o^A is the audio representation, o^{EE} is the robot end-effector pose, and the action a is defined by the $6D$ delta pose of the robot end-effector, including a delta position and a delta Euler angle. Then, we extract tactile and audio features $(\mathrm{y}_i^T, \mathrm{y}_i^A)$ from $(\mathrm{o}_i^T, \mathrm{o}_i^A)$ using the pre-trained representation encoders, respectively. These tactile embeddings and the robot end-effector pose $(\mathrm{y}_i^T, \mathrm{y}_i^A, \mathrm{o}_i^{EE})$ are formulated as the key features of the demonstration library, with each associated with a corresponding action value a_i. Given the varying scales of these inputs, we normalize them such that the maximum distance for each input is unity in the library. In robot execution, for a given real-time observation $(\hat{\mathrm{o}}_i^T, \hat{\mathrm{o}}_i^A, \hat{\mathrm{o}}_i^{EE})$, we first obtain the query feature $(\hat{\mathrm{y}}_t^T, \hat{\mathrm{y}}_t^A, \hat{\mathrm{o}}_t^{EE})$, and then search the demonstration library for a nearest-neighbor-based action prediction.

3.4 Learning Residual Policy through Online Reinforcement Learning

The offline policy learned from human tactile demonstrations might not guarantee task success when deployed on the physical robot. This could be due to: i). morphological differences between the human hand and the robot gripper, ii). inaccurate fingertip tracking caused by fast movements, and iii). underexplored contact effects. Therefore, motivated by recent works using pure RL [17–20, 51] to learn tactile policies, we further leverage online reinforcement learning that allows in-domain robot interactions (part (d) of Fig. 2). It is noteworthy that the previous pure RL methods often generate quasi-static motions and utilize a limited action space [17–19, 51] because they learn from scratch without effective priors. On the contrary, we intend to leverage the best of both advantages by RL fine-tuning the offline policy learned from human tactile demonstrations.

Since it is infeasible to directly fine-tune the non-parametric policy, we instead learn a residual policy. Same as the offline policy, the input to the residual policy π_r is $(\hat{y}_t^T, \hat{y}_t^A, \hat{o}_t^{EE})$ and output is the residual action a_i^r. Considering we use $6D$ continuous action space and sparse observation-action pairs (around 70 actions per trajectory), we opt for SAC [52] to handle the continuous action space with entropy regularization and to generate a replay buffer to increase the size of training data. The pseudocode of the RL training is included in Appendix. D. Finally, the robot action is the sum of the action generated from offline policy π_i and the action from the residual policy π_r.

Another critical component of residual RL is the reward design, which must balance exploitation and exploration. To address this, we combine an expert-aligned reward with a task-specific reward. The expert-aligned reward encourages a policy distribution that mimics the demonstrations, whereas the task-specific reward drives exploration to optimize the in-domain robot policy. More details about pseudocode, policy design, and rewards are included in Appendix. D.

4 Experiments

In this section, we first describe the experiment setting and the data collection throughput to highlight that MimicTouch can efficiently collect useful demonstrations (Sec. 4.1). Then we introduce the Offline Policy Evaluation to validate the efficacy of the offline policy and highlight the benefits of using human tactile demonstrations (Sec. 4.2), and the Online Policy Improvement and Generalization Evaluation to demonstrate the efficiency of learning the residual policy through online RL and the superior zero-shot generalization capability (Sec. 4.3).

Hardware setting. All experiments are conducted on a Franka Emika Panda Arm. For each task, the learned policy generates the 6-DoF pose command and then maps it to 7-DoF joint torque actions using an inverse kinematics solver and a low-level built-in controller.

Teleoperation setting. We compare our human tactile-guided data collection system with Spacemouse-based teleoperation, a popular teleoperation interface for manipulation tasks [9–11]. To collect a similar number of observation-action pairs for each trajectory, we collect one robot state, one tactile image, and 0.5s audio segment at 5Hz. Since collecting teleoperation data necessitates considerable expertise, we allocate approximately 5 hours to practice with this system.

Tasks. We focus on two-piece insertion tasks that exemplify the challenge of many contact-rich manipulation settings. We 3D-print a cylinder and an insertion hole base and set up the same task environments for both data collection settings. An example of this task has been shown in Fig. 1

4.1 Human Tactile Demonstration Collection System

In this section, we demonstrate that Human Tactile Demonstrations can greatly improve data collection throughput for contact-rich manipulation tasks. We begin by determining the **usability** of a demonstration trajectory based on the following two metrics: i) the robot successfully inserts the object into the hole without any slipping or falling, and ii) the robot completes the task within 100 actions. Using these criteria, we record the time length of collecting 20 usable demonstration trajectories by using our customized system (3.1) and teleoperation system (4). Then, we evaluate the data collection throughput (see Table. 1) in two metrics: i). the number of usable demonstrations collected per hour, and ii). the success rate for collecting usable demonstrations.

The results in Table. 1 support our insights: i). human tactile demonstrations can be collected significantly more efficiently than teleoperation systems for contact-rich tasks, and ii). human tactile demonstrations can seamlessly integrate human's tactile feedback and motion capability, whereas teleoperation systems struggle to capture such dynamic tactile-guided motions. These factors together result in much lower data collection efficiency and success rates of the teleoperation system.

Methods	Frequency	Success Rate
Human Tactile Demonstrations	104 traj/hr	83.3% (20/24)
Teleoperation	19 traj/hr	38.5% (20/52)

Table 1: Data collection throughput for human tactile demonstrations and teleoperation.

4.2 Offline Policy Evaluation

4.2.1 MimicTouch effectively learns from human tactile demonstrations

In this subsection, we evaluate the offline policy learned from the human tactile demonstrations using both VINN and a parametric imitation learning method. Specifically, we aim to test whether the offline policies can be deployed in real-world environments within the desired error tolerance, which is crucial for physical robot execution. To evaluate it, we used the data collection system (Sec. 3.1) to collect 20 noise-free data sequences as the datasets to learn both offline policies. For testing, we gather another 5 data sequences with random noise to emulate the real-world environments (details are in Appendix. E). We then compute the mean square error loss (MSE Loss, defined in Appendix. F) to evaluate the policies on the testing set.

For the baseline parametric imitation learning method, we select the MULSA [6], which has been previously demonstrated effective in multisensory robot learning for insertion tasks. In our setting, we use the same sensor input $(\mathrm{y}_i^T, \mathrm{y}_i^A, \mathrm{o}_i^{EE})$ as in the VINN method to generate the continuous $6D$ delta action a_i. The same MSE loss is used for policy training and validation.

For both offline policies, we calculate the MSE losses between the generated action sequence and the ground truth action sequence in the testing set. We observed that the MSE loss from the VINN policy is **0.21**, which is significantly lower than that of MULSA policy (**1.53**). This suggests that the VINN policy can generate more accurate $6D$ continuous actions, indicating it is more suitable for subsequent online real-world RL fine-tuning. Notably, since the MSE Loss of MULSA is significantly higher, we do not use it for further real-world experiments.

Additionally, we conduct the ablation study on the choice of multi-modal sensor inputs in Appendix. G. The results suggest that multi-modal tactile feedback is crucial for the success of contact-rich insertion tasks, particularly during the insertion phase when most contact occurs.

4.2.2 Human tactile demonstrations trains better policies than teleoperated demonstrations

In this subsection, we compare the performance of the offline policies trained from human tactile demonstrations and teleoperation demonstrations. We collect 20 trajectories for each, and then we use the same VINN method to learn the offline policies for both sets of demonstrations. We evaluate the policies in two manners: i) the task success rate over 25 policy rollouts, and ii) the action serial numbers (right part of Fig. 3), i.e., the indexed numbers of the selected actions in the corresponding trajectory of the demonstration library, for each action of the rollout trajectories.

We first report that the task success rate of the offline policy rollouts from human tactile demonstration is **40%**, whereas that of the policy rollouts from teleoperation data is only **12%**. Then, in the right part of Fig. 3, we show the mean and variance of the action serial numbers of three successful rollout trajectories for each offline policy. For the human tactile demonstrations, we observe a linear relationship with minimal variance. On the contrary, the action serial numbers for the teleoperation policy exhibit a non-linear relationship with a significantly larger variance, particularly during the insertion phase. This performance discrepancy arises because the majority of contacts occur during this phase, and the teleoperation lack of human tactile feedback is not well-suited for capturing these contact-rich events. Therefore, the indexes of the selected key features tend to be more disordered. A similar conclusion can be drawn from the screenshots (left part of Fig. 3) for one successful rollout of both policies. The Human Tactile Demonstrations policy exhibits dynamic tilting for object insertion, which emulates the demonstrated human's tactile-guided control strategy (see Fig. 1). However, the teleoperation policy forcefully inserts the object with little orientation, indicating that teleoperation demonstrations capture less versatile motions that are necessary for contact-rich tasks.

Therefore, combined with the results in Sec. 4.1, MimicTouch not only efficiently collects human tactile demonstrations, but also enables effective policy learning using these demonstrations.

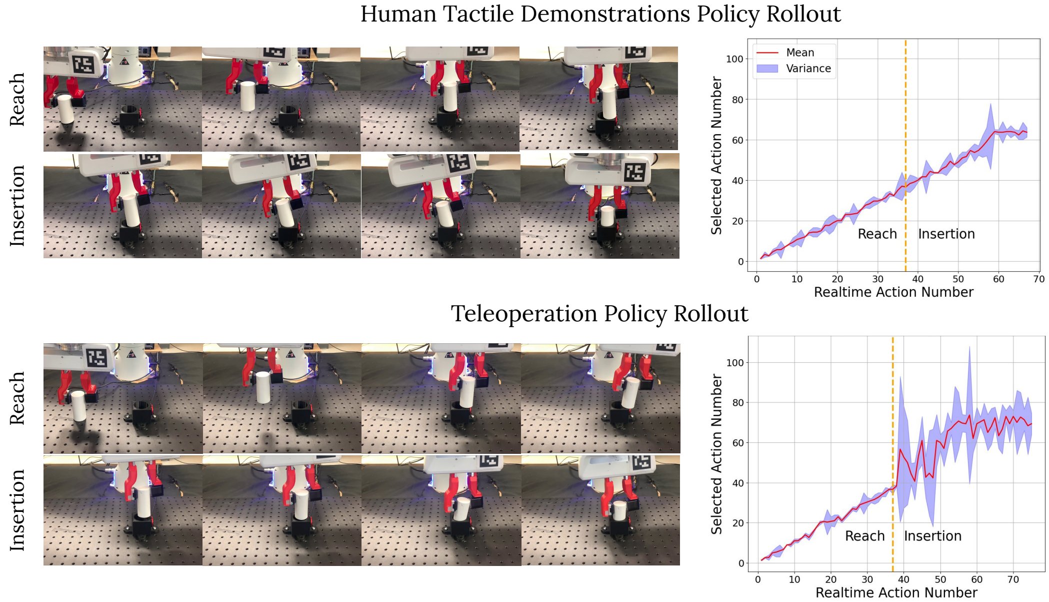

Figure 3: **Left:** Qualitative results for Human Tactile Demonstrations policy and Teleoperation policy. **Right:** Visualization of the action serial numbers for three trajectories generated by both policies. Solid red lines indicate mean trends and shaded areas show $\pm$ standard deviations. The left side of the dashed orange line is the Reach phase, and the right side is the Insertion phase.

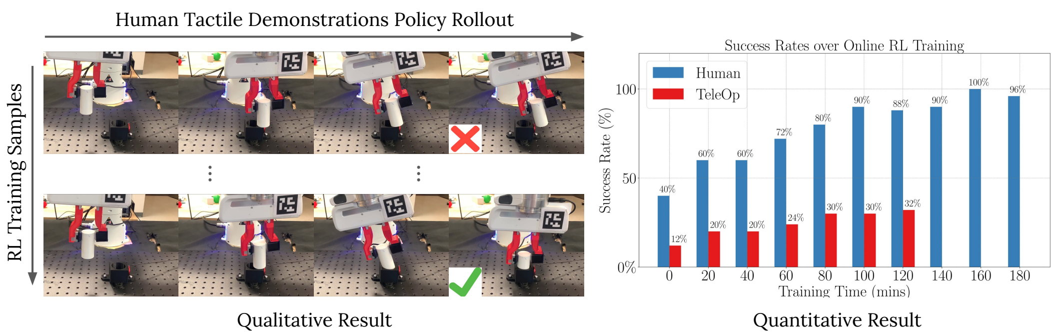

Figure 4: **Left:** Demonstrations of the online RL fine-tuning process, which further improves the task performance. **Right:** Quantitative task evaluations for offline policies learned from human tactile demonstrations (Human) and teleoperation (TeleOp) during online RL fine-tuning show that Human significantly outperforms TeleOp in terms of task success rate and RL training efficiency.

4.3 Online Policy Improvement and Generalization to New Settings

In this section, we evaluate the final policy trained through online reinforcement learning (RL). To ensure the robustness of the RL policy, we perform domain randomization on the robot's starting position so that the initial object-hole contact is located differently. For the policy update, at each iteration, we use five newly collected trajectories along with another five randomly selected trajectories from the replay buffer.

Online RL fine-tuning significantly and efficiently improves task performance. We evaluate the trained policy every 20 minutes, approximately after every 13 RL epochs. After each hour, we compute the task success rates over 25 policy rollouts, in alignment with the offline policy evaluation. For other time instances, we only compute the task success rates over 10 policy rollouts to minimize sensor wear. The evaluation results are shown in Fig. 4, and we can observe that the policy can reach **96%** task success rate in **3 hours**. Additionally, the offline policy can reach 88% task success rate after 2-hour RL fine-tuning, which is significantly more training efficient than the policy learned from teleoperation, which could only achieve 32% task success rate at that time. This result supports the effectiveness of online RL fine-tuning, as it allows the robot to further interact

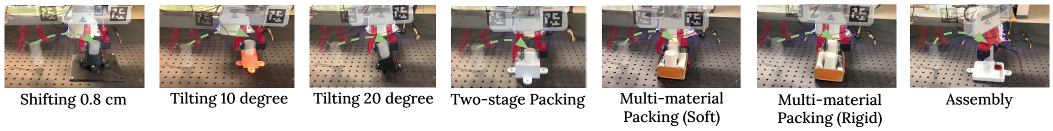

Figure 5: Setup of zero-shot generalization tasks.

with the task environment. Moreover, it once again highlights the importance of using human tactile demonstrations since the offline policy learned from teleoperation demonstration exhibit significant training inefficiency in the subsequent online RL fine-tuning.

MimicTouch policy exhibits superior zero-shot generalization capability. We evaluate the zero-shot generalizability of the MimicTouch final policy. We consider the following generalization settings: i). *Shifting Positions*: an insertion task with the hole positions shifted for 0.8cm in either $\pm x$ or $\pm y$ horizontal directions, ii). *Tilting Angles*: an insertion task with the hole angle tilted for 10 degrees or 20 degrees, iii). *Two-stage Dense Packing*: a two-stage dense insertion task, which requires the robot to perform two consecutive insertion alignments, iv). *Multi-material Dense Packing*: an insertion task where the hole contains multiple objects, such as a pen, tissues, or a sponge, and v) *Furniture Assembly* [53, 54]: an insertion task, which requires the robot to insert and adjust the leg into a small hole in a table for screwing. Each setting is depicted in Fig. 5. Additionally, to demonstrate the complexity of these tasks, we introduce another baseline: *Openloop Policy*, where we collect five successful insertion trajectories in the initial setting and execute each of these trajectories for those generalization tasks five times for each of the tasks. See Appendix. H for the details of each task and the policy evaluation process.

Policy	Shift	10°	20°	Two-stage	Rigid	Soft	Assem (I)	Assem (A)	Assem
Openloop	60%	56%	40%	52%	52%	36%	32%	37.5%	12%
MimicTouch	**92.5%**	**92%**	**80%**	**88%**	**80%**	**64%**	**76%**	**68.4%**	**52%**

Table 2: Task success rate for each generalization task. Specifically, Shift is *Shifting Position*, 10°and 20°are different tilted angles for *Tilting Angles*, Two-stage is *two-stage Dense Packing*, Rigid means the object contacts the pen in *Multi-material Dense Packing*, Soft means the object contacts the tissue and sponge in *Multi-material Dense Packing*, Assem (I), Assem (A), and Assem are the insertion, adjustment, and overall results for *Furniture Assembly* respectively.

We report the task success rate of both policies in Table. 2. Based on these results, we can observe that the MimicTouch policy can significantly outperform all the openloop policies in all those generalization tasks. Also, since the *Openloop* policy has lower success rates in all the generalization tasks, it indicates the robustness of the MimicTouch policy in different challenging generalization domains. We also include the qualitative results and detailed analysis in Appendix. I.

5 Limitation and Future Work

MimicTouch pioneers the pathway to learning human tactile-guided control strategies from human tactile demonstrations. However, it still has several limitations for future improvements. Firstly, MimicTouch still requires several hours to RL fine-tune the policy to address the embodiment gap and enrich the contact reasoning. We will explore a better representation learning method to further reduce the gap between humans and robots. Secondly, MimicTouch is task-specific and can not directly generalize human's tactile-guided control strategy to other contact-rich manipulation tasks. One potential solution is to learn a generalizable tactile-based dynamic model, rather than a task-specific control policy. Finally, the method of learning to perform contact-rich manipulation tasks from human tactile demonstrations could be extended to other robot tasks, such as dexterous manipulation, bimanual manipulation, and deformable object manipulation.

6 Conclusion

We presented MimicTouch, a multi-modal imitation learning framework that: i) enables humans to perform tactile demonstrations with their hands without a robot in the loop ii) learns from such demonstrations and safety transfer to robot with non-parametric imitation learning, and iii) improves the policy performance with residual-based online RL to bridge the human-robot embodiment gap. We show that MimicTouch enables high-throughput data collection and achieves high success rate and generalization across a wide range of two-piece insertion and assembly tasks.

References

[1] 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. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[2] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu, A. Handa, and D. Fox. Factory: Fast contact for robotic assembly. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[3] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. Aparicio Ojea, E. Solowjow, and S. Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555, 2020. doi:10.1109/IROS45743.2020.9341714.

[4] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow. Meta-reinforcement learning for robotic industrial insertion tasks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9728–9735, 2020. doi:10.1109/IROS45743.2020.9340848.

[5] G. Lengyel, G. Žalalytė, A. Pantelides, J. N. Ingram, J. Fiser, M. Lengyel, and D. M. Wolpert. Unimodal statistical learning produces multimodal object-like representations. *eLife*, 8:e43942, may 2019. ISSN 2050-084X. doi:10.7554/eLife.43942. URL `https://doi.org/10.7554/eLife.43942`.

[6] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In *CoRL*, 2022.

[7] I. Guzey, B. Evans, S. Chintala, and L. Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=EXQ0eXtX3OW`.

[8] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 879–893. PMLR, 29–31 Oct 2018. URL `https://proceedings.mlr.press/v87/mandlekar18a.html`.

[9] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[10] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[11] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *6th Annual Conference on Robot Learning (CoRL)*, 2022.

[12] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The surprising effectiveness of representation learning for visual imitation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[13] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017. ISSN 1424-8220. doi:10.3390/s17122762. URL `https://www.mdpi.com/1424-8220/17/12/2762`.

[14] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934, 2018. doi:10.1109/IROS.2018.8593661.

[15] D. Ma, E. Donlon, S. Dong, and A. Rodriguez. Dense tactile force estimation using gelslim and inverse fem. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424, 2019. doi:10.1109/ICRA.2019.8794113.

[16] S. Dong and A. Rodriguez. Tactile-based insertion for dense box-packing. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7953–7960, 2019. doi:10.1109/IROS40897.2019.8968204.

[17] 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 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. URL `https://arxiv.org/abs/1810.10191`.

[18] 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)*, pages 6437–6443, 2021. doi:10.1109/ICRA48506.2021.9561646.

[19] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *ICRA*, 2022.

[20] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang. Rotating without seeing: Towards in-hand dexterity through touch. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[21] S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, and M. Kaess. Shapemap 3-d: Efficient shape mapping through dense touch and vision. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7073–7080, 2022. doi:10.1109/ICRA46639.2022.9812040.

[22] E. J. Smith, R. Calandra, A. Romero, G. Gkioxari, D. Meger, J. Malik, and M. Drozdzal. 3d shape reconstruction from vision and touch. *arXiv preprint arXiv:2007.03778*, 2020.

[23] E. Smith, D. Meger, L. Pineda, R. Calandra, J. Malik, A. Romero Soriano, and M. Drozdzal. Active 3d shape reconstruction from vision and touch. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16064–16078. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/8635b5fd6bc675033fb72e8a3ccc10a0-Paper.pdf`.

[24] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, oct 2018. doi:10.1109/lra.2018.2852779. URL `https://doi.org/10.1109%2Flra.2018.2852779`.

[25] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta. The curious robot: Learning visual representations via physical interactions. In *European Conference on Computer Vision*, pages 3–18. Springer, 2016.

[26] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 314–323. PMLR, 13–15 Nov 2017. URL `https://proceedings.mlr.press/v78/calandra17a.html`.

[27] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao. Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer. *arXiv preprint arXiv:2112.06374*, 2023.

[28] A. Thankaraj and L. Pinto. That sounds right: Auditory self-supervision for dynamic robot manipulation. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=sLhk0keeiseH`.

[29] J. Sinapov, M. Wiemer, and A. Stoytchev. Interactive learning of the acoustic properties of household objects. In *2009 IEEE International Conference on Robotics and Automation*, pages 2518–2524, 2009. doi:10.1109/ROBOT.2009.5152802.

[30] J. Christie and N. Kottege. Acoustics based terrain classification for legged robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3596–3603, 2016. doi:10.1109/ICRA.2016.7487543.

[31] C. Matl, Y. Narang, D. Fox, R. Bajcsy, and F. Ramos. Stressd: Sim-to-real from sound for stochastic dynamics. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 935–958. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/matl21a.html.

[32] D. Gandhi, A. Gupta, and L. Pinto. Swoosh! rattle! thump! – actions that sound. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[33] P. Sharma, D. Pathak, and A. Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[35] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE, 2021.

[36] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969, 2023. doi:10.1109/ICRA48891.2023.10160547.

[37] 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. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=hRZ1YjDZmTo.

[38] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, 2023.

[39] J. Ho and S. Ermon. Generative adversarial imitation learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf.

[40] M. Xie, A. Handa, S. Tyree, D. Fox, H. Ravichandar, N. D. Ratliff, and K. V. Wyk. Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1355–1367. PMLR, 14–18 Dec 2023.

[41] Y. Han, M. Xie, Y. Zhao, and H. Ravichandar. On the utility of koopman operator theory in learning dexterous manipulation skills. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=pw-OTIYrGa.

[42] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell. Non-parametric imitation learning of robot motor skills. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5266–5272, 2019. doi:10.1109/ICRA.2019.8794267.

[43] M. Vaandrager, R. Babuška, L. Buşoniu, and G. A. Lopes. Imitation learning with non-parametric regression. In *Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 91–96, 2012. doi:10.1109/AQTR.2012.6237681.

[44] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3: 297–330, 2020.

[45] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.*, 47:2280–2292, 2014. URL https://api.semanticscholar.org/CorpusID:12519167.

[46] Gelsight mini. https://www.gelsight.com/gelsightmini/.

[47] M. Du, O. Y. Lee, S. Nair, and C. Finn. Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[48] S. Haldar, J. Pari, A. Rai, and L. Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[49] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[50] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. doi:10.1109/IJCNN52387.2021.9534474.

[51] J. Ding, C. Wang, and C. Lu. Transferable force-torque dynamics model for peg-in-hole task, 2019.

[52] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[53] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[54] H. Lin, R. Corcodel, and D. Zhao. Generalize by touching: Tactile ensemble skill transfer for robotic furniture assembly, 2024.

[55] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.

Appendices

A Data Collection

In this section, we show the novel data collection system in Fig. 6.

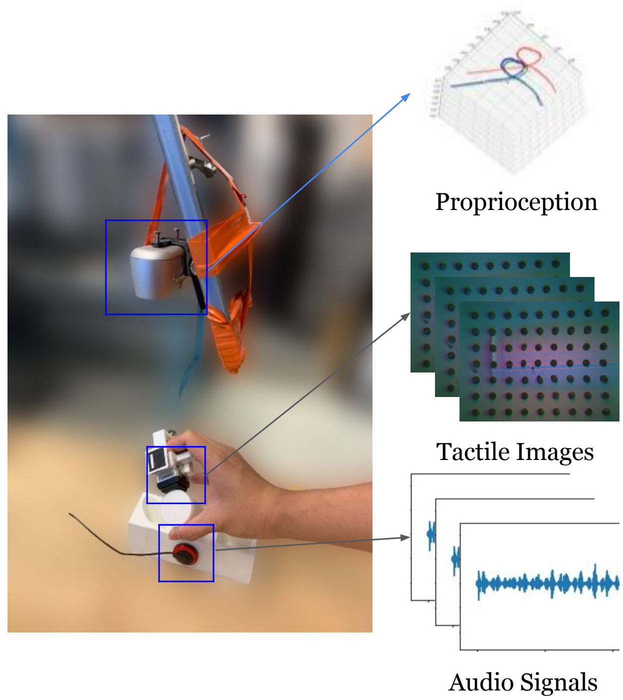

Figure 6: The human tactile demonstrations collection system.

B Representation Learning

Data Collection For Representation Learning, we collect a large task-specific dataset that contains 7657 tactile images and 1000 audio sequences. We collect 100 trajectories, each of them approximately five seconds long and containing around 70 tactile images and 10 audio sequences. These trajectories contain various data qualities, which include successful cases, failure cases, and sub-optimal cases. In detail, successful cases refer to the cases that human finishes the task with smooth trajectories; failure cases mean that human did not insert the object successfully or used more than five seconds to finish this task; sub-optimal cases mean that human used unnecessary motions to finish the insertion task.

BYOL BYOL [49] generates two augmented views, $v \triangleq t(x)$ and $v' \triangleq t'(x)$, from a given x by applying image augmentations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$ respectively, where $\mathcal{T}$ and $\mathcal{T}'$ represent distinct augmentation distributions. The architecture of BYOL comprises a primary encoder f_θ and a target encoder f_ξ, where the latter being an exponential moving average of the former. Given the augmented views v and v', they are processed to yield representations y and y'. These representations are subsequently transformed by projectors g_θ and g_ξ to produce higher-dimensional vectors z and z'. The primary encoder and its associated projector are designed to predict the output from the target projector, resulting in $q_\theta(z_\theta)$ and $sg(z'_\xi)$. The model's output consists of l_2-normalized versions of these predictions, which are trained using a similarity loss function. Post-training, the encoder f_θ is utilized for feature extraction from observations.

To utilize BYOL in tactile images, we scale the tactile image up to 256x256 to work with standard image encoders. We use the ResNet [55] architecture, also starting with pre-trained weights. Unlike SSL (self-supervised learning) techniques used in visual images, we only apply the Gaussian blur and small center-resized crop augmentations, since other augmentations such as color jitter and grayscale would violate the assumption that augmentations do not change the tactile signal significantly. For each input, the trained model will generate a 1×2048 representation vector.

Audio Representation Learning BYOL-A [50] is an extended version of BYOL to audio representation learning, processing log-scaled mel-spectrograms through a specialized augmentation module. To utilize BYOL-A in our audio data, we down-sampled signals from 44.1kHz to 16kHz, with a window size of 64 ms, a hop size of 10 ms, and mel-spaced frequency bins F = 64 in the range 60–7,800 Hz. Then, the Pre-Normalization step stabilizes the input audio for subsequent augmentations. Once normalized, the Mixup step introduces contrasts in the audio's background, defined by the log-mixup-exp formula:

$$\tilde{\mathrm{x}}_{\mathrm{i}} = \log((1-\lambda)\exp(\mathrm{x}_{\mathrm{i}}) + \lambda \exp(\mathrm{x}_{\mathrm{k}}))$$

where x_{k} is a mixing counterpart and λ is a ratio from a uniform distribution. The next one is the RRC block, an augmentation technique, that captures content details and emulates pitch shifts and time stretches. For each input, the trained model will generate a 1×2048 representation vector.

C Data Pre-processing

C.1 Sensor Data Alignment

Each sensor operates at different frequency: i) RealSense operates at 60 Hz with a resolution of 320x240 pixels, ii) GelSight Mini streams tactile images at 15 Hz with 400x300 pixel resolution, and iii) HOYUJI TD-11 piezo-electric contact microphone has a 44.1kHz sampling rate, and the audio data is segmented at 2Hz.

Therefore, to ensure synchronization across our sensors, we first address the disparate sampling rates of the fingertip poses, tactile images, and audio sequences, which are 60Hz, 15Hz, and 2Hz, respectively. Specifically, we downsample the fingertip poses to 15 Hz. For the audio data, instead of collecting entirely new 0.5-second segments, we record the extended audio signals at intervals of every 0.07 seconds. As a result, the new 0.5s segment has a new-collected 0.07s interval and an old overlapped 0.43s segment in the past, which results in an overlap of 0.43 seconds between consecutive audio segments. Therefore, all sensor inputs are sampled at 15Hz.

C.2 Calibration and Filtering of Fingertip Poses

The 6D human fingertip poses extracted from the AruCo marker include 3D positions along with rotation vectors. To use these fingertip poses as the end-effector's poses for robot policy learning, we need to address two problems: i). developing a calibration method to align the data collection system with the robot execution system, and ii). implementing a filtering method to generate smooth trajectories.

Calibration Given that data collection and robot experiments occur in disparate scenarios, it is crucial to align our human-centric data collection system with the physical robot system. Initially, we record the distance between the object (starting point) and the base (ending point) within the data collection system and replicate this setup in the robot environment. Following this, six equidistant positions between the starting and ending points are identified within both systems. The object is gripped at these predetermined positions using both hands and the robot's end-effector so that we can capture the corresponding poses. In this calibration process, the hand poses, denoted as the "Eye" in the calibration function, are referenced to the camera frame, while the end-effector poses, represented as the "Hand" in the calibration system, are referenced to the robot frame. Conclusively, we employ the calibrateHandEye function from OpenCV, using the six captured poses, to calibrate these two frames (camera frame and robot frame).

Filtering Given the inherent noise and occasional outliers in the poses obtained from the RealSense and AruCo markers, it is imperative to implement post-processing techniques to ensure the quality and smoothness of the trajectories. For each pose sequence, outliers are detected by sorting the values of each delta transformation (i.e., the delta translations and the delta Euler angles). The Interquartile Range (IQR) method is employed to establish the upper and lower bounds, which are then used to identify outliers. The IQR is defined as: IQR $= Q_3 - Q_1$ where Q_3 and Q_1 are the third and first quartiles, respectively. Outliers are replaced using a median filter with a window size

Algorithm 1 Online Residual Reinforcement Learning

Input: offline policy π_i, randomly initialized residual policy π_r
Input: step size sequences $\{\beta_t\}$, number of iterations K, Replay Buffer D
Initialize replay buffer D with pre-collected data
for $k = 1$ to K **do**
    Sample mini-batch D_k from Replay Buffer D
    Obtain current trajectory C_k by executing $\pi_i + \pi_r$
    To collect more data in D: $D \leftarrow D \cup C_k$
    Combine D_k and C_k to form batch B_k for update
    for all $(s, a_i, r, s') \in B_k$ **do**
        $a_i \leftarrow \pi_i(s)$ ▷ Obtain latent action
        $a_r \leftarrow \pi_r(s)$ ▷ Obtain residual action
        $\hat{a} \leftarrow a_i + a_r$ ▷ Combine latent and residual actions
        $Q_r \leftarrow Q_r(s, a_i) + r + \gamma Q_r(s', \pi_i(s'))$
        $\pi_r \leftarrow \pi_r - \alpha \nabla_{\pi_r} L(\pi_r)$ ▷ Update residual policy with gradient step
    end for
end for
Return: Trained residual policy π_r

of 3. To enhance the temporal consistency of the estimated hand and object pose, a digital low-pass filter is applied to eliminate high-frequency noise. Specifically, the filter has a sampling frequency of 5Hz and a cutoff frequency of 2Hz. The low-pass filter can be represented as:$H(f) = \frac{1}{1+\left(\frac{f}{f_c}\right)^2}$ where f is the sampling frequency and f_c is the cutoff frequency.

D Details of RL training

Training pipeline The overall pseudo-code for RL Training is given in Alg. 1.

RL policy details For the residual policy π_r within our framework, we employ the Soft Actor-Critic (SAC) algorithm with an MLP architecture. The training strategy aims to effectively combine reinforcement learning principles with residual corrections, thus enhancing the overall performance of the system. The following formula represents the objective for training the residual policy:

$$\pi_r = \operatorname*{argmax}_{\pi} \left\{ \mathbb{E}_{(s,a)\sim D} \left[Q(s, a + \pi(s)) - \alpha \log \pi(a|s) \right] \right\}$$

- $Q(s, a + \pi(s))$: The Q-value function, which estimates the value of executing the residual action $\pi(s)$ in addition to the base action a in the state s. This represents the total action influenced by both the offline policy and the residual corrections suggested by π_r.
- $\mathbb{E}_{(s,a)\sim D}$: The expectation over state-action pairs sampled from the replay buffer D, which contains data from both past experiences and current new explorations.
- $\alpha \log \pi(a|s)$: The entropy regularization term for the policy π, which encourages exploration by penalizing the certainty of the policy's action selection. This term is crucial in SAC to ensure sufficient exploration and avoid premature convergence to suboptimal policies.
- $\pi(a|s)$: The policy network (MLP) outputs the probability distribution over actions given the state s, from which the action a is sampled.

This formula ensures that the residual policy π_r learns to adjust the actions generated by the offline policy by optimizing the SAC objective. It balances the maximization of expected returns (via Q-values) and the maintenance of behavioral diversity (via entropy regularization), allowing π_r to adapt and refine actions based on real-time environmental feedback and historical data from the replay buffer.

RL Reward Design We will give a detailed explanation for each component in our reward design.

Distance Reward:

$$d = 1 - tanh(10.0 * ||distance||_2)$$

where the *distance* is between the current position of the gripper center and the target gripper center.

Orientation Reward:

$$o = 1 - tanh(7.5 * ||diff\_ori||_2)$$

where the $diff\_ori$ stands as the quaternion difference between the current gripper orientation and the target gripper orientation.

Penalty for blocking

$$c = \begin{cases} 0.2, & \text{successfully complete this action} \\ 0, & \text{cannot complete this action in 0.5s} \end{cases} \tag{1}$$

Penalty for Slippery

$$s = \begin{cases} 0.5, & ||y_t^i - y_t^{i-1}|| \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The y_t^i and y_t^{i-1} stands for the embeddings of the tactile images in the current step i and the last step $i-1$.

Overall Rewards

$$R = \begin{cases} 1, & \text{if success} \\ \alpha D_{KL}(P\|Q) + \beta d + \gamma \cdot o - c - s, & \text{otherwise} \end{cases} \tag{3}$$

In Eqn. 3, P is the executed trajectory, Q is the expert trajectory, $D_{KL}(P\|Q)$ is the KL Divergence between the executed trajectory and the expert trajectory, d, o, c, s are defined above. The setup of each weight: $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$.

E Emulate the Physical Environment for Policy Evaluation

To emulate the physical robot environment, we introduce random noise to those 10 unseen data sequences. The robot state space input undergoes a random position noise within the range $[-3\text{cm}, +3\text{cm}]$ for each axis. Gaussian noise, denoted as $\mathcal{N}(0, \sigma)$, is added to both the tactile image and audio signal. In this notation, $\mathcal{N}(0, \sigma)$ signifies a Gaussian distribution with a mean of 0 and a standard deviation of σ. For tactile images, the noise affects pixel values in the range $[0, 255]$, while for audio data, it impacts signal values in the range $[0, 1]$. Given their distinct ranges, we apply Gaussian noise with standard deviations of $\sigma = 100$ for tactile images and $\sigma = 0.4$ for audio data.

F MSE Loss

For calculating the MSE Loss between two action sequences, we need to normalize the actions' translation vectors and rotation vectors since they have different scales. Specifically, we use min-max normalization on both the translation vectors and rotation vectors, where the max vector and min vector are selected from the training set. As a result, translation vectors and rotation vectors will have the same scale for calculating the MSE Loss. The formula is shown as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2$$

Where: y_i represents the ground truth normalized action, $\hat{\mathrm{y}}_i$ represents the generated normalized action, and n is the number of all action steps.

G Ablation Study: Do Multi-Modal Tactile Feedback Improve the Task Performance?

In this section, we evaluate the performance of our multi-modal tactile embeddings. We consider the following baselines: i). *MimicTouch w/o T & A*: MimicTouch without tactile or audio embeddings, ii). *MimicTouch (T)*: MimicTouch incorporating only tactile embeddings, iii). *MimicTouch*

Models	*MimicTouch w/o T & A*	*MimicTouch (T)*	*MimicTouch (A)*	*MimicTouch (T + A)*
MSE Loss	0.62	0.38	0.48	**0.21**
Success Rate	4%	24%	16%	**40%**

Table 3: MSE Loss over test sets and Task success rates of 25 policy rollouts.

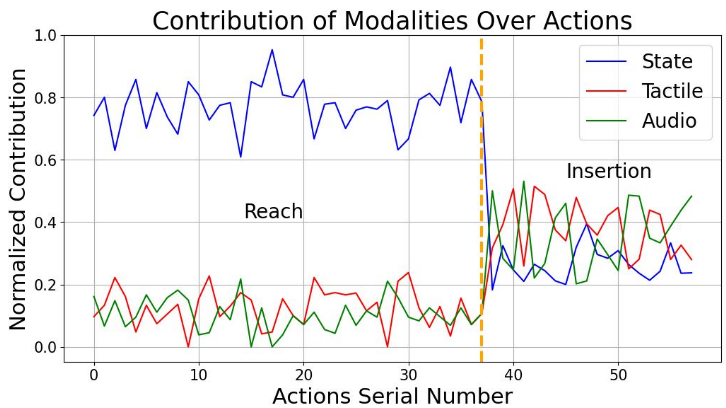

Figure 7: Visualization of the impact of each sensor modality during policy execution. The left side of the dashed orange line is Reach Phase, and the right side is Insertion Phase.

(A): MimicTouch incorporating only audio embeddings., and iv). *MimicTouch (T + A, Ours)*: MimicTouch incorporating both tactile and audio embeddings. We evaluate the policy performance in terms of the MSE losses over the test sets (see Sec. 4.2.1) and task success rates over 25 policy rollouts. In addition, we visualize the impact of each sensor modality during policy execution. Specifically, we plot the normalized distance between the query feature and the selected key feature for each sensor input. A larger distance means that the corresponding sensor modality contributes more in selecting the key feature from the demonstration library.

From Table. 3, we observe that without using both tactile images and audio signals, the MSE loss (task success rate) is 0.62 (4%), which is significantly higher (lower) than the others. By incorporating both tactile and audio feedback, the MSE loss can be as low as 0.21, and more importantly, the task success rate can reach 40%.

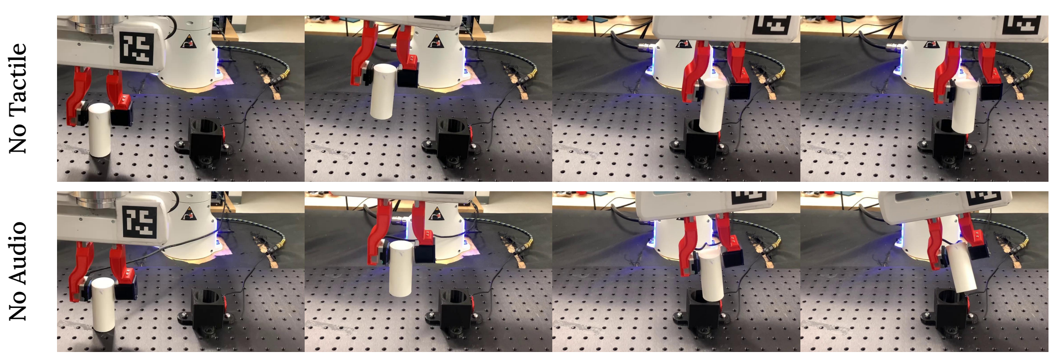

Figure 8: Policy rollouts of some failure examples with only tactile feedback or only audio feedback.

As shown in Fig. 7, tactile and audio inputs start to play important components during the Insertion phase, when most contacts occur. We also have qualitative results shown in the Fig. 8. According to those results, we can find that a lack of tactile feedback easily leads to incorrect motion when contact appears, whereas the lack of audio feedback easily leads to an inability to detect external collisions.

Therefore, we can conclude that the multi-modal tactile feedback is crucial for the success of contact-rich insertion tasks.

H Generalization Setting

In this section, we describe the setting of each generalization task.

Shifting Positions: In this generalization task, we shift the hole positions for 0.8 cm in either $\pm x$ or $\pm y$ to test the generalization ability for finishing the insertion task with under varied alignment conditions.

Tilting Angles: In this generalization task, we tilted the hole angle for 10 degrees or 20 degrees to test the generalization ability for finishing the insertion task with different contact positions.

Two-stage Dense Packing: We introduce the two-stage dense packing task, which requires the robot to perform two consecutive alignment adjustments to complete the dense packing process. Each hole will challenge the robot's ability to adjust its alignment according to tactile feedback efficiently.

Multi-material Dense Packing: In this generalization task, the robot is required to insert the cylinder into the hole which contains multiple objects: pen, tissue, and sponge. This setting has rigid objects (pen) and deformable soft objects (tissue and sponge) with different materials and shapes, which challenge the robot's ability to accomplish the task with different tactile feedback from different materials.

Furniture Assembly: In this generalization task, the robot is required to insert the cylinder into a small hole in a table for screwing. This task will test two aspects of our policy: whether the robot can insert the object into a smaller hole, and whether the robot can adjust it to a position, that is deep enough to be skewed by a human-defined simple script (to rotate the end-effector for $120°$), based on the tactile feedback from the threads in the hole.

Policy Evaluation Process: We evaluate the policy performance in those five different generalization settings for both MimicTouch final policy and the *Openloop* Policy. For *Shifting Positions*, we rollout the policies 10 times in each direction of $\pm x$ or $\pm y$, resulting in a total of 40 evaluations. For *Tilting Angles*, we rollout the policies 25 times for both tilting directions in $\pm x$, resulting in a total of 50 evaluations for the $10°$ tilting and $20°$ tilting, respectively. For *Two-stage Dense Packing*, we rollout the policies 25 times. For *Multi-material Dense Packing*, we rollout the policies 25 times on both rigid object "pen" (Rigid in the table) and deformable soft objects "tissue and sponge" (Soft in the table). For *Furniture Assembly* (Assem in the table), we rollout the policies 25 times. This task has two sub-evaluation metrics: insertion (Assem (I) in the table) and adjustment (Assem (A) in the table). Notably, the success rate for Assem (I) is the success rate for the number of attempts (25), and the success rate for Assem (A) is the success rate when the insertion is successful.

I Generalization Results

In this section, we summarize the zero-shot generalization results based on the quantitative results shown in Table. 2, and the qualitative results shown in Fig. 9.

- MimicTouch policy can zero-shot transferring to insertion tasks with different contact positions, tilted angles, and even different sizes of holes (Aseembly (Insertion)).
- For the Two-stage Dense Packing, MimicTouch policy displays its robustness to adjust according to multiple stages of contact information according to the quantitative result and the video. This shows that our model can make continuous and correct adjustments based on the continuously varied contact information.
- MimicTouch policy also shows its power in the multi-material task domains. Due to different materials in the environment, the sensor feedback will be different from the training environment. In this case, our policy still has great performance on other rigid objects (pen). For the deformable soft object (tissue and sponge), The success rate is a bit lower because of two challenging issues: i). it's hard to get audio feedback for contact with soft objects, ii). sponge sometimes is too soft to get tactile feedback. With those issues, our policy still gets 64% success rate. Moreover, the qualitative result from the video shows impressive performance in adjusting continuously according to the deformation of the tissue.

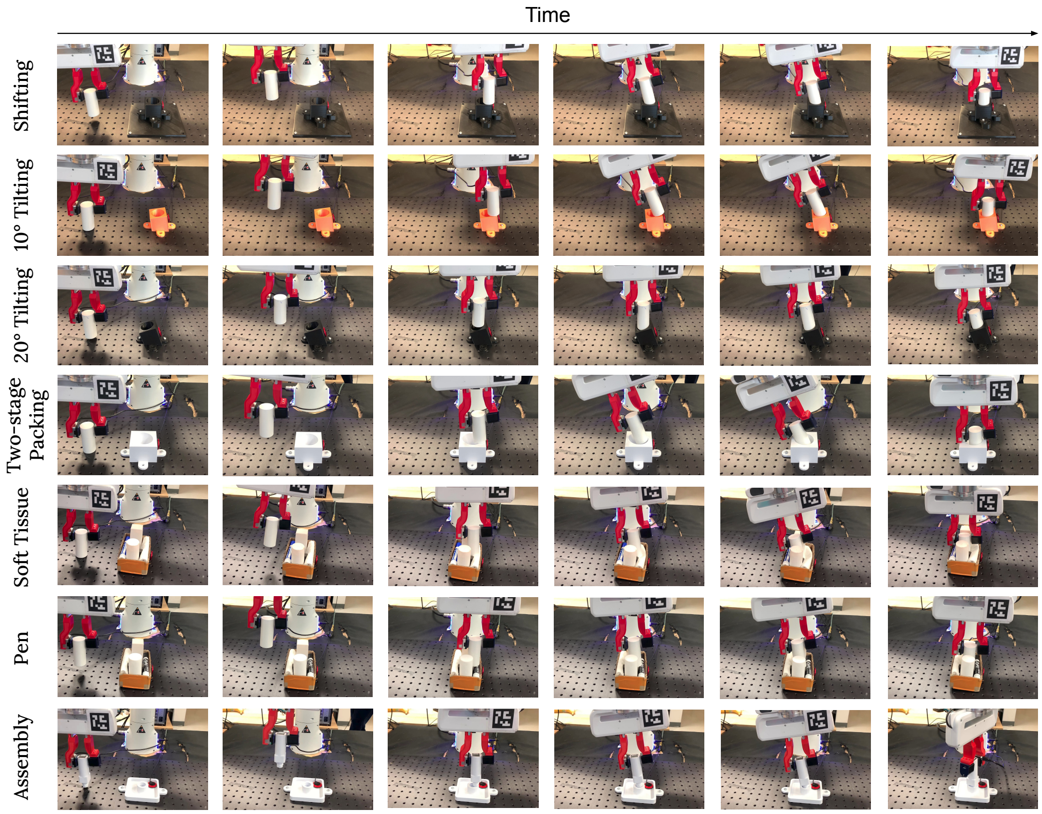

Figure 9: Task setup and qualitative results for zero-shot generalization tasks

- In the assembly task, MimicTouch policy not only can insert the object into a small hole but also can adjust the object to a correct pose and insert it to a deep-enough position according to the tactile feedback from the contact between screw threads. This allows us to use a very simple script (to rotate the end-effector for $120°$) to solve the assembly task.