Tilde: Teleoperation for Dexterous In-Hand Manipulation Learning with a DeltaHand

Zilin Si*, Kevin Zhang*, Zeynep Temel, and Oliver Kroemer
*Equal contribution
Robotics Institute, Carnegie Mellon University
{zsi,klz1,ztemel,okroemer}@andrew.cmu.edu
https://sites.google.com/view/tilde-

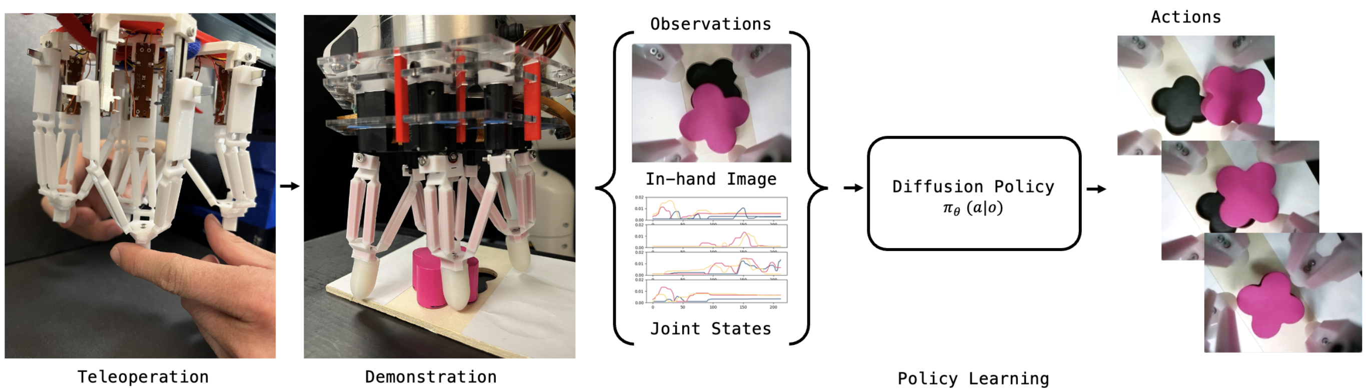

Fig. 1: We introduce an imitation learning-based in-hand manipulation system with a dexterous DeltaHand. We present a kinematic twin teleoperation interface, TeleHand, to collect demonstrations on various dexterous manipulation tasks, such as shape insertion shown above. By using vision-conditioned diffusion policies, the DeltaHand can autonomously complete the tasks.

***Abstract*—Dexterous robotic manipulation remains a challenging domain due to its strict demands for precision and robustness on both hardware and software. While dexterous robotic hands have demonstrated remarkable capabilities in complex tasks, efficiently learning adaptive control policies for hands still presents a significant hurdle given the high dimensionalities of hands and tasks. To bridge this gap, we propose *Tilde*, an imitation learning-based in-hand manipulation system on a dexterous DeltaHand. It leverages 1) a low-cost, configurable, simple-to-control, soft dexterous robotic hand, DeltaHand, 2) a user-friendly, precise, real-time teleoperation interface, TeleHand, and 3) an efficient and generalizable imitation learning approach with diffusion policies. Our proposed TeleHand has a kinematic twin design to the DeltaHand that enables precise one-to-one joint control of the DeltaHand during teleoperation. This facilitates efficient high-quality data collection of human demonstrations in the real world. To evaluate the effectiveness of our system, we demonstrate the fully autonomous closed-loop deployment of diffusion policies learned from demonstrations across nine dexterous manipulation tasks with an average 86% success rate.**

I. Introduction

Dexterous manipulation is essential for a wide range of real-world tasks such as inserting small components precisely for manufacturing, administering medicine in hospitals, and handling delicate ingredients while cooking. However, a significant skill gap exists between human and robotic proficiency due to the demands for precision, robustness, and rapid adaptation to unstructured environments on both the hardware and software. Thus, integrated systems are necessary to address the challenges of dexterous manipulation and advance the field.

Recent advances in imitation learning have shown great advantages in utilizing diffusion models [4, 13, 18] for efficient manipulation policy learning, as compared to deep reinforcement learning [1, 2] which is computationally expensive and data-hungry, or motion planning [3, 8, 11] which relies on accurate modeling. However, imitation learning methods require high-quality demonstrations, which are challenging to collect quickly and reliably for dexterous manipulations. To leverage imitation learning, we need highly precise and easy-to-use teleoperation interfaces for dexterous robotic hands that will allow us to collect diverse demonstrations.

Although anthropomorphic hands [5, 12, 14, 15] have already shown their ability to perform various manipulation tasks through teleoperation, these hands are designed to be general-purpose replacements for human hands which may be unnecessarily complex for certain domains. By contrast, non-anthropomorphic hands [9, 10, 16], with their lower control complexity and higher design flexibility, can be better tailored to tasks such as precise peg insertion or in-hand manipulation. However, these designs present additional challenges for imitation learning given the human-to-robot hand correspondence problem. DeltaHands [16], as shown in Fig. 1, are soft, compact, easy to customize, and possess high degrees-of-freedom (3-DoF per finger) that are simple

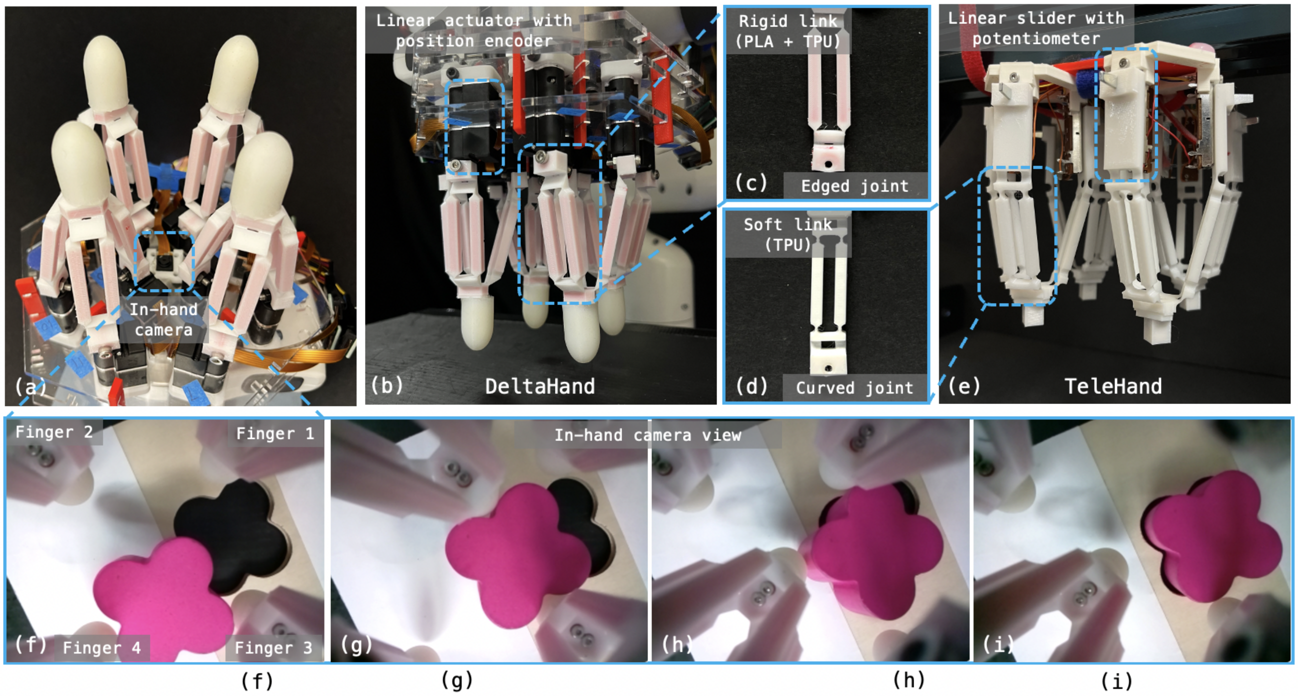

Fig. 3: (a) A DeltaHand with an in-hand RGB camera. A kinematic twin teleoperation interface including (b) a DeltaHand and (e) a TeleHand. The TeleHand uses linear sliders with potentiometers to record the joint states of each finger. The DeltaHand will reproduce the motions of a TeleHand by using the Telehand's potentiometer readings as desired joint positions for its linear actuators. (c) The DeltaHand's fingers have 3D-printed rigid-core embedded links and edged joints, which increase the stiffness of each finger and enable them to exert more force. (d) The TeleHand's fingers have 3D-printed soft links and curved joints, which induce more compliance in each finger. Therefore less force is required for users to teleoperate the robot, which makes teleoperation easier. (f)-(i) In-hand camera images that capture the object and the DeltaHand's fingers.

to control, which makes them a great fit for dexterous in-hand manipulation. However, we need an intuitive and precise teleoperation interface to enable efficient imitation learning for DeltaHands.

In this work, we present *Tilde*, a dexterous manipulation learning system (Fig. 1) from three aspects: a dexterous robotic hand adapted from DeltaHands, a kinematic twin teleoperation interface, and imitation learning with diffusion policies. We demonstrate key features of our system including 1) the high dexterity and precision of the DeltaHand, 2) the low latency, ease-of-use, and precision of the teleoperation interface, TeleHand, and 3) the efficiency and generalizability of the policy learning for dexterous manipulation.

II. Methodology

A. Dexterous Hand Design

a) Finger Design: DeltaHands [16] is a configurable, highly dexterous, low-cost robotic hand framework based on Delta robots. The original DeltaHand's fingers were configured with 3D-printed soft TPU links (Fig. 3(d)) for compliant and safe interactions. However, to benefit from the DeltaHand's compliance and extend its capabilities to manipulation tasks that require more forces such as pushing the plunger of a syringe, we modify the design of the Delta finger's links and joints as shown in Fig. 3(c). To improve the force profile of the Delta finger, we 3D-print hybrid TPU and PLA links where we embed rigid PLA material (red) inside a thin outer shell of TPU (white). This strengthens the whole finger structure and enables fingers to apply more force while still preserving enough compliance to safely handle collisions. To improve kinematic precision, we use edged joints with smaller joint lengths instead of the original curved joints to reduce undesired buckling and deformations during finger motions. These modifications can be easily incorporated by just swapping out the Delta fingers from the linear actuators, showcasing the flexibility of the DeltaHands framework.

b) In-hand Camera: Sensors are key components for closed-loop control by providing real-time feedback. In particular, local visual sensing is crucial for dexterous manipulation by capturing detailed geometric features [17]. Therefore, we integrate a mini Arducam camera module[1] into the hand for in-hand visual sensing as shown in Fig. 3(a). The camera is located at the center of the hand and on the same level as the Delta finger bases without taking extra space. It has symmetric observations to provide useful inductive bias [7]. The DeltaHand's kinematics permit a mostly unobstructed view of the fingertips which benefits visual servoing.

c) Fingertip Design: To increase the contact friction and enable soft contact for more secure grasps, we first 3D print the "bone" of the fingertip with TPU material, and then cast an

[1] https://www.arducam.com/product/arducam-raspberry-pi-5mp-spy-camera-b0066/

additional layer of silicon rubber using Ecoflex 00-20 FAST[2].

d) Hand Configurations: An overview picture of the DeltaHand can be seen in Fig. 3(a). We arrange four 3-DoF Delta fingers in a circular layout with a 40 mm radius from the hand center to each Delta finger center. Each finger has a 40 mm link length and 20 mm base radius, and is individually actuated by three linear motors with 20 mm stroke length. This gives a total of 12 DoF and a 110 mm × 110 mm × 30 mm workspace for the DeltaHand.

B. Teleoperation Interface

We develop a kinematic twin teleoperation system for the DeltaHand to get precise and high-quality demonstrations. The system includes a TeleHand (Fig. 3(e)) manipulated by a human teleoperator, and a DeltaHand (Fig. 3(b)) to reproduce the TeleHand's finger motions in real-time.

The TeleHand has the same configurations including the hand size, finger arrangement, and finger size, as the DeltaHand to enable direct one-to-one joint position mapping. The DeltaHand's fingers are actuated by linear motors with 20 mm stroke length and each finger's link bases move prismatically. Similarly, the TeleHand consists of linear sliders with a 20 mm motion range to create the same linear mobility for each finger as the DeltaHand except they move passively. Teleoperators can easily drag and move the finger end-effectors of the TeleHand which will lead to joint position changes in the sliders. The joint positions of TeleHand's fingers will be recorded by each sliders' potentiometers and then directly mapped to the DeltaHand as the linear motors' desired positions. For the TeleHand, we use the original Delta finger design (Fig. 3(d)) which is more compliant and easier for humans to manipulate.

We use Robot Operating System (ROS) for real-time communication. Both the TeleHand and the DeltaHand use Arduino microcontrollers to directly publish and receive ROS topics via a control PC. Our teleoperation system including the DeltaHand and the TeleHand can be manufactured in a day with off-the-shelf materials, 3D printing, and laser cutting, and costs around $1000.

C. Learning with Diffusion Policies

We adapt the CNN-based Diffusion Policy [4] to our system for dexterous in-hand manipulation policy learning with a DeltaHand. We condition the diffusion policies on visual observations from the in-hand camera and joint states of the DeltaHand and predict action sequences. Both the joint states and actions are represented as the 12-dimensional absolute actuator joint positions.

We found that using various data augmentation techniques on observations greatly improved task performance. We leveraged 1) random image cropping and rotation to improve the rotational and translational invariance of fingers' visual servoing to the objects, and 2) Gaussian noise to joint state observations to guide the policy in learning funneling behaviors that can make the policy more robust when encountering unseen joint states. Specifically, we randomly cropped the images from their original size of $(240, 320)$ to $(216, 288)$ and rotated the images within 30 degrees. We added Gaussian noise with a standard deviation of 3.16 mm to each joint state.

III. Experimental Evaluation

We evaluate the system on five dexterous manipulation tasks as shown in Fig. 4: **Grasp**, **Block Slide**, **Block Lift**, **Ball Roll**, **Cap Twist**, and four challenging tasks: **Shape Insert**, **Syringe Push**, and **Finger Gait in the Air** (Ball Roll and Block Slide). **Grasp** is a fundamental skill for most manipulation tasks. The second to fifth tasks focus on different in-hand object repositioning skills: **Block Slide** corresponds to horizontal XY translations, **Block Lift** corresponds to vertical Z translations, **Ball Roll** corresponds to rotations around the X and Y axes, and **Cap Twist** corresponds to rotations around the Z axis. The above tasks mostly require repeated motions. The **Syringe Push** requires fingers to precisely align the syringe and forcefully push the plunger. The **Shape Insert** consists of multi-modal action sequences which require the fingers to translate, rotate, and transport the object to the final goal pose. The **Finger Gait in the Air** has non-recoverable failure risks which requires more robust policies.

A. Data Collection

We mount a DeltaHand on a Franka robot arm. For most tasks, we keep the robot arm static while the DeltaHand uses its in-hand capabilities to manipulate the objects. An external RGB camera[3] is placed in front of the experiment workspace. For each task, we manually preset the height and the location of the arm to approximately align the DeltaHand's workspace with the object. To collect demonstrations, we first define the goal for each task which can be verified from the visual observations. If we reach the goal, we end the demonstration, or we run until we reach 5000 time steps which roughly equates to 250 seconds (data collection runs at 20fps speed).

B. Experimental results

The number of demonstration data for all tasks can be seen in the first rows of Table I. They vary depending on the task difficulty and the number of objects we use. After the initial policy training, we run 10 test trials on both the train and additional test objects. The success rates are shown in the second rows of Table. I. We use joint states and in-hand images as observations for all tasks. We observe that for most tasks, with less than 50 demonstrations, we can achieve a success rate of over 60%, and even a 100% success rate on the **Grasp**, **Block Slide**, **Cap Twist**, and **Ball Roll** tasks. On average, we can achieve a 86% success rate over all tasks.

Most failures during the **Shape Insert** task are due to unseen object poses, which we believe can be resolved with more data. Most failure cases for **Block Slide in the Air** resulted when one pair of fingers pinched too tightly and the other

[2]https://www.smooth-on.com/products/ecoflex-00-20-fast/

[3]https://www.amazon.com/gp/product/B0C289GYVZ/

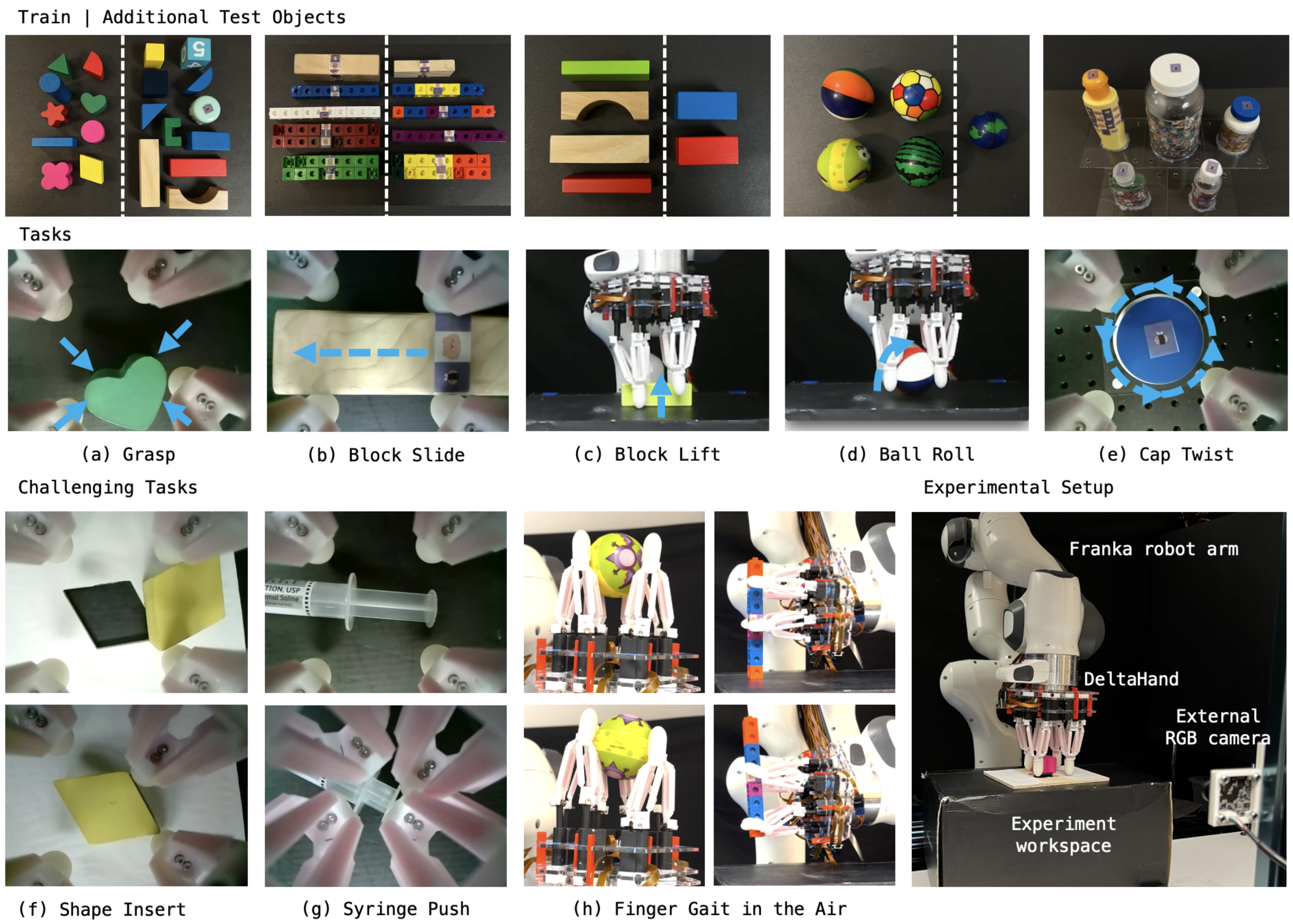

Fig. 4: We evaluate our system on five dexterous manipulation tasks: (a) **Grasp** (b) **Block Slide** (c) **Block Lift** (d) **Ball Roll** (e) **Cap Twist**, and three challenging tasks: (f) **Shape Insert** (g) **Syringe Push** (h) **Finger Gait in the Air**. The goals of tasks are indicated by blue arrows. For tasks (a)-(d), we separate the training and additional unseen testing objects with white dashed lines. For experimental setup, we mount a DeltaHand on a Franka robot arm. We pre-set the height and location of the Franka arm on top of the experiment workspace. An external RGB camera is mounted in front of the experiment workspace.

Tasks	**Grasp**	**Block Slide**	**Block Lift**	**Cap Twist**	**Ball Roll**	**Syringe Push**	**Shape Insert**	**Finger Gait in the Air**	
								Ball Roll	Block Slide
# demos	55	40	30	40	25	40	40	25	25
# Success / # tests	20/20	10/10	8/10	10/10	10/10	8/10	5/10	9/10	6/10

TABLE I: Experimental results on nine tasks. We show that with less than 60 demos, we can achieve success rates over 80% on first five tasks. For challenging tasks, we can still achieve over 50% success rates on all of them.

pair could not move the block any further. This demonstrates the importance of incorporating tactile sensors in the fingers which we plan to explore in future work.

IV. Conclusions

We present *Tilde*, an imitation learning-based in-hand manipulation system with a dexterous DeltaHand. We introduce a kinematic twin teleoperation interface for low-cost data collection of high-quality human demonstrations and efficient end-to-end real-world policy learning by using diffusion policies. We show that with our system, we can perform a variety of dexterous manipulations and achieve an average success rate of 86% across our evaluation tasks. These tasks include grasping, in-hand object re-positioning and re-orientation, and finger gaiting. Our experiments show the capability of the system to learn robust vision-based dexterous manipulation policies from demonstrations that were acquired with our easy-to-use and precise teleoperation interface.

In the future, we would like to improve the generalizability of our system for broader dexterous manipulation tasks in unstructured environments. Therefore, we plan to augment sensing modalities with tactile sensing by incorporating fingertip tactile sensors for more delicate tasks, integrate arm motions into our teleoperation system to achieve intrinsic and extrinsic dexterity [6, 19], and explore object-centric approaches to improve the policies' robustness to unseen scenarios.

References

[1] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pages 1300–1313. PMLR, 2020.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[3] Xianyi Cheng, Sarvesh Patil, Zeynep Temel, Oliver Kroemer, and Matthew T Mason. Enhancing dexterity in robotic manipulation via hierarchical contact exploration. *IEEE Robotics and Automation Letters*, 9(1):390–397, 2023.

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

[5] Shadow Robot Company. Shadow dexterous hand. https://www.shadowrobot.com/dexterous-hand-series/, 2023.

[6] Arnav Gupta, Yuemin Mao, Ankit Bhatia, Xianyi Cheng, Jonathan King, Yifan Hou, and Matthew T Mason. Extrinsic dexterous manipulation with a direct-drive hand: A case study. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4660–4667. IEEE, 2022.

[7] Kyle Hsu, Moo Jin Kim, Rafael Rafailov, Jiajun Wu, and Chelsea Finn. Vision-based manipulators need to also see from their hands. *arXiv preprint arXiv:2203.12677*, 2022.

[8] Jacky Liang, Xianyi Cheng, and Oliver Kroemer. Learning preconditions of hybrid force-velocity controllers for contact-rich manipulation. In *Conference on Robot Learning*, pages 679–689. PMLR, 2023.

[9] Raymond Ma and Aaron Dollar. Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Robotics & Automation Magazine*, 24(1):32–40, 2017.

[10] Connor McCann, Vatsal Patel, and Aaron Dollar. The stewart hand: A highly dexterous, six-degrees-of-freedom manipulator based on the stewart-gough platform. *IEEE Robotics & Automation Magazine*, 28(2):23–36, 2021.

[11] Andrew S Morgan, Kaiyu Hang, Bowen Wen, Kostas Bekris, and Aaron M Dollar. Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics and Automation Letters*, 7(2):4821–4828, 2022.

[12] Steffen Puhlmann, Jason Harris, and Oliver Brock. Rbo hand 3: A platform for soft dexterous manipulation. *IEEE Transactions on Robotics*, 38(6):3434–3449, 2022.

[13] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[14] Wonik Robotics. Allegro hand. http://wiki.wonikrobotics.com/AllegroHandWiki/index.php/Allegro_Hand_v4.0, 2023.

[15] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[16] Zilin Si, Kevin Zhang, Oliver Kroemer, and F Zeynep Temel. Deltahands: A synergistic dexterous hand framework based on delta robots. *IEEE Robotics and Automation Letters*, 2024.

[17] Oren Spector and Dotan Di Castro. Insertionnet-a scalable solution for insertion. *IEEE Robotics and Automation Letters*, 6(3):5509–5516, 2021.

[18] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[19] Wenxuan Zhou and David Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning*, pages 150–160. PMLR, 2023.