

Reactive Diffusion Policy: Slow-Fast Visual-Tactile Policy Learning for Contact-Rich Manipulation

Han Xue^{1*} Jieji Ren^{1*} Wendi Chen^{1*} Gu Zhang^{234†} Yuan Fang^{1†} Guoying Gu¹ Huazhe Xu^{234‡} Cewu Lu^{1‡}

¹Shanghai Jiao Tong University ²Tsinghua University, IIIS ³Shanghai Qi Zhi Institute ⁴Shanghai AI Lab

*Equal contribution †Equal contribution ‡Equal advising

reactive-diffusion-policy.github.io

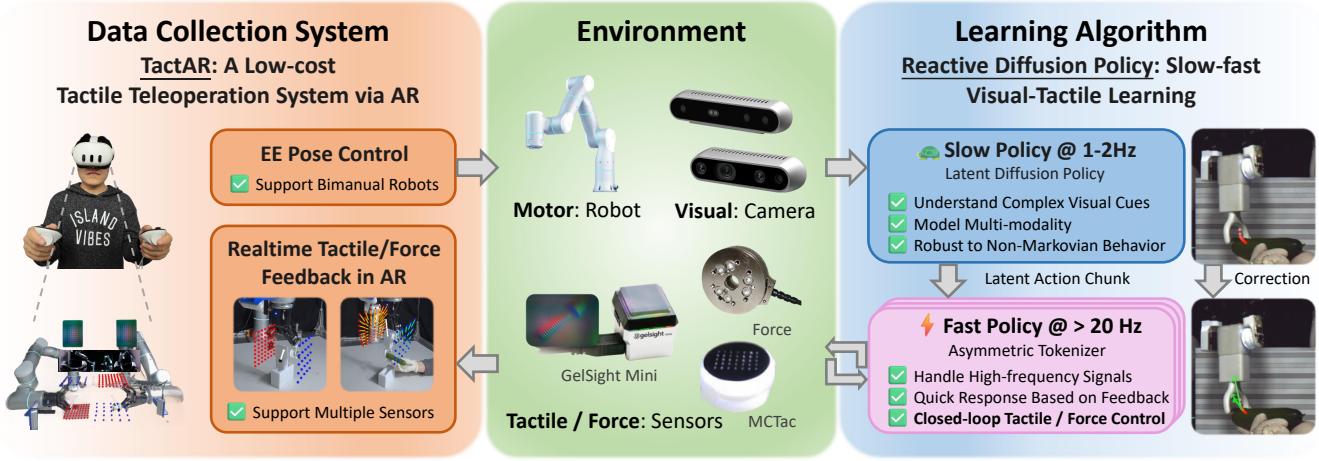


Fig. 1: **TactAR** is a low-cost and versatile teleoperation system which can provide real-time tactile / force feedback via Augmented Reality (AR). **Reactive Diffusion Policy (RDP)** is a slow-fast imitation learning algorithm that can model complex behaviors with a slow policy network and achieve closed-loop control based on **tactile** / force feedback with a fast policy network.

Abstract—Humans can accomplish complex contact-rich tasks using vision and touch, with highly reactive capabilities such as quick adjustments to environmental changes and adaptive control of contact forces; however, this remains challenging for robots. Existing visual imitation learning (IL) approaches rely on action chunking to model complex behaviors, which lacks the ability to respond instantly to real-time tactile feedback during the chunk execution. Furthermore, most teleoperation systems struggle to provide fine-grained tactile / force feedback, which limits the range of tasks that can be performed. To address these challenges, we introduce TactAR, a low-cost teleoperation system that provides real-time tactile feedback through Augmented Reality (AR), along with Reactive Diffusion Policy (RDP), a novel slow-fast visual-tactile imitation learning algorithm for learning contact-rich manipulation skills. RDP employs a two-level hierarchy: (1) a slow latent diffusion policy for predicting high-level action chunks in latent space at low frequency, (2) a fast asymmetric tokenizer for closed-loop tactile feedback control at high frequency. This design enables both complex trajectory modeling and quick reactive behavior within a unified framework. Through extensive evaluation across three challenging contact-rich tasks, RDP significantly improves performance compared to state-of-the-art visual IL baselines through rapid response to tactile / force feedback. Furthermore, experiments show that RDP is applicable across different tactile / force sensors. Code and videos are available on reactive-diffusion-policy.github.io.

I. INTRODUCTION

Humans are capable of performing numerous precise contact-rich tasks (*e.g.*, peeling vegetables) in daily life by employing both vision and touch. However, these contact-rich tasks that appear simple to humans can be quite challenging for robots. Some research work in neuroscience [32, 17, 31] has indicated that when humans engage in contact-rich tasks, the process can be divided into two components: **1) feedforward / predictive action** and **2) closed-loop fine-tuning** based on sensory feedback, such as tactile signals. Inspired by this, we aim to develop a robot learning system capable of emulating human control patterns when performing intricate contact-rich tasks.

In recent years, visual imitation learning (IL) methods [11, 71] have demonstrated strong performance in various real-world tasks. Representative works in this area, such as Diffusion Policy [11], ACT [71] and π_0 [7], have employed **action chunking** to mitigate compounding errors in long sequences and improve temporal consistency. Action chunking also effectively models non-Markovian behaviors commonly found in human demonstrations, such as pauses or oscillatory motion. However, these methods operate in an **open-loop** state during the execution of action chunks, which makes the

policy unable to respond instantly to environment changes in contact-rich tasks. Furthermore, the absence of tactile input significantly limits the capabilities of these methods. As a result, these IL methods are limited to low-precision tasks (*e.g.*, pick-and-place, push-pull), which do not require precise force control or fast response.

In order to compensate for the limitations of purely visual input, numerous approaches [42, 29, 47, 65, 33] have explored the integration of tactile input into imitation learning policies. However, most of these research works focus only on the observation level, using tactile input to provide additional information such as visual occlusion or the determination of contact state. At the action level, these methods still rely on conventional action chunking for action prediction, limiting their ability to respond quickly during the execution of the chunk. Furthermore, most approaches utilize traditional teleoperation to collect human demonstration data, making it challenging to obtain high-quality action data with fine-grained tactile feedback. This limitation also limits the types of tasks that these methods can effectively perform.

In this work, we propose two critical components to solve the above issues of visual-tactile imitation learning:

- A **teleoperation system** called **TactAR** which can provide fine-grained tactile feedback in real time through Augmented Reality (AR).
- An **imitation learning algorithm** called **Reactive Diffusion Policy (RDP)** that retains the advantages of action chunking while enabling high-frequency closed-loop control based on tactile signals during the execution of each chunk.

In our proposed TactAR teleoperation system, we use Meta Quest3 to provide real-time tactile / force feedback via Augmented Reality (AR). We use 3D deformation field as the unified representation for tactile / force feedback (Fig. 1 left) of different sensors. We render the 3D deformation field in AR and attach it to the robot end-effector in virtual space. Our system also supports camera streaming of tactile sensors and RGB cameras in AR. In this way, the user can get rich contact information during teleoperation including the tactile image, normal force, shear force, and torsional torques. Our TactAR system is designed with an emphasis on versatility and accessibility: (1) it supports multiple types of tactile / force sensors; (2) it can be easily deployed in different robot embodiments; (3) it is very cost-effective (\$500 for Meta Quest 3).

To leverage the high-quality visual tactile data collected by the TactAR system, we propose an imitation learning algorithm called Reactive Diffusion Policy (RDP) (Fig. 1 right). Inspired by human control strategies in contact-rich tasks, we adopt a “slow-fast” hierarchical control pipeline. We use a slow network (**latent diffusion policy, LDP**) to predict a high-level action chunk in latent space at low frequency (1-2 Hz), which is analogous to the **predictive action**. Then we use a fast network (**Asymmetric Tokenizer, AT**) to finetune the latent action chunk based on high-frequency tactile feedback (20-30 Hz), which is analogous to the **closed-loop finetuning**.

This hierarchical structure allows the slow network to preserve its ability to model complex and non-Markovian actions via the diffusion model [11] and action chunking, while the fast network achieves closed-loop control with real-time tactile feedback for precise force control and quick response.

We have conducted experiments on three challenging contact-rich tasks, which can evaluate the model performance on the following aspects: (1) precision, (2) precise and adaptive force control, (3) fast response under disturbances, and (4) bimanual coordination. Real-world experiments have shown that our Reactive Diffusion Policy algorithm can model complex actions while maintaining very fast reactive behavior, achieving a significant performance improvement (> 35%) in three tasks compared to state-of-the-art IL baselines. This also validates the data quality of our TactAR teleoperation system. Furthermore, we conduct cross-sensor experiments demonstrating that RDP is applicable across various tactile / force sensors (GelSight Mini[22], MCTac [50] and built-in joint torque sensors [20]). The code of the TactAR teleoperation system and Reactive Diffusion Policy algorithm are available at [reactive-diffusion-policy.github.io](https://github.com/reactive-diffusion-policy/reactive-diffusion-policy) .

II. RELATED WORK

A. Tactile / Force Sensing Hardware in Robotics

Many robotics tasks require physical interaction with the world. Tactile sensors and force sensors can provide richer information about the contact physics compared to RGB cameras. Each type of sensors have their advantages and limitations. Force sensors mounted on the robot end effector or joint torque sensors on the robot arm can directly obtain force / torque readings and are less prone to damage. Nevertheless, force and torque sensors are prone to signal noise, particularly during rapid movement, and they also tend to be relatively expensive.

The tactile sensors can be classified as electrical sensors and optical sensors. Electrical sensors use capacitative [24, 61, 63, 62], resistive [55, 4, 54], MEMS [59] or magnetic particles [5, 27, 57, 64, 6] to sense contact. Such sensors are usually more compact and thin, but have lower spatial resolution and are complex to manufacture. Only some electrical tactile sensors can directly output both normal force and shear force [6], but rely on a complex calibration process with force/torque sensors.

Optical tactile sensors [67, 56, 39, 50, 41, 40, 37] such as GelSight [67], GelSlim [56], MCTac[50] and 9DTact [41, 40] are another stream of works; they capture high-resolution images with cameras to track the deformation of gels; optical sensors can also be equipped with markers dots to better track the normal and shear deformation field of the gel surface. Compared to electrical sensors or force / torque sensors, optical tactile sensors are easier to fabricate, less expensive and have lower signal noise. Force and torque information can be implicitly represented on the shear field gel surface but they rely on complex calibration process [70] to indirectly calculate force/torque values. In this paper, we use two different optical tactile sensors (GelSight Mini [22] and MCTac[50]) and the

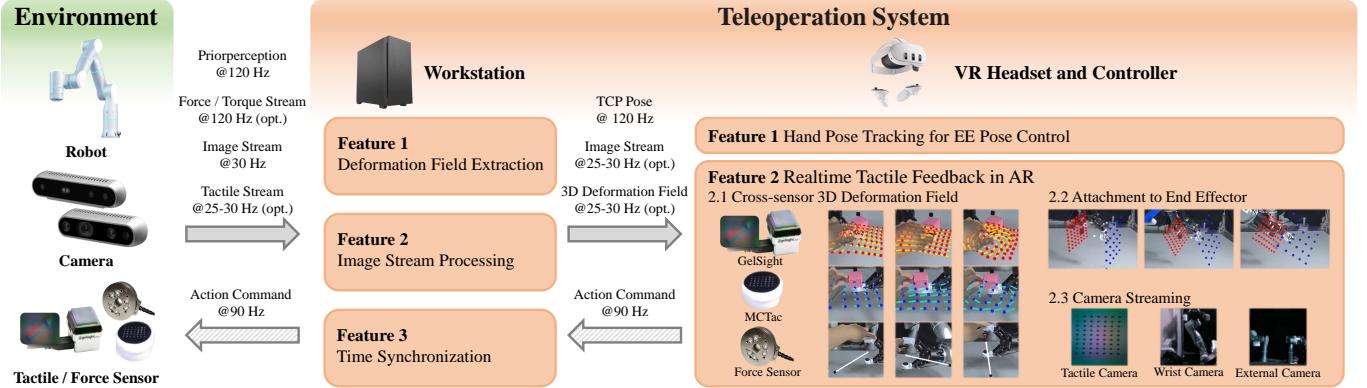


Fig. 2: Overview of **TactAR** teleoperation system. It can provide real-time tactile / force feedback via Augmented Reality (AR). The tactile feedback is represented as the 3D deformation field, which is a universal representation applicable to multiple different tactile / force sensors. The 3D deformation field is rendered and "attached" to the robot end-effector in AR, which makes the user perceive the rich contact information in 3D space. TactAR also support real-time streaming for multiple RGB cameras and optical tactile sensors. Please see the video in the supplementary file for more details.

joint torque sensors of the robot arm to further unleash the potential of tactile / force sensing.

B. Robot Data Collection System

Teleoperation is a common way to collect expert data with robots. Current teleoperation systems are mainly built upon VR controller [30], hand tracking [49] or direct joint mapping [60, 16, 71] with hardwares. Most teleoperation systems are solely based on visual feedback, which makes them hard to perform precise contact-rich tasks.

One stream of haptic teleoperation systems relies on isomorphic hardwares. Bi-ACT [9] and Kobayashi et al. [36] use bilateral control with ALOHA [71] to get force feedback. These teleoperation systems are difficult to deploy on different hardware platforms due to isomorphic hardware designs. Another stream of teleoperation systems use force/torque sensors for haptic feedback. FoAR [26] uses a haptic teleoperation device [52, 15] to obtain force feedback, but it also has a relatively high cost. ForceMimic [43] adds a force sensor on a hand-held device [12] to get force feedback, but suffers from the inaccuracy of pose estimation, and thus cannot directly train an end2end policy. ACP [28] uses a low-stiffness compliance kinesthetic teaching system to provide haptic feedback, but requires complex custom adapters for the end effectors.

Some research works use VR headset to get tactile feedback. Comp-ACT [34] and DIPCOM [1] use VR controller to adjust the stiffness of the low-level robot controller during teleoperation, but it cannot provide active force feedback and relies on robot arms with built-in torque sensors. Zhou et al. [73] and Bunny-VisionPro [14] use vibration to provide tactile / force feedback, but they can only provide very coarse information about contact. Our method combines the advantages of low-cost VR controller and tactile sensing, getting tactile feedback via Augmented Reality, while preserving the accuracy needed for precise contact-rich tasks. In addition, our teleoperation systems are flexible and easy to deploy on different tactile sensors and robot platforms.

C. Visual-tactile Manipulation

Tactile sensing is crucial in robotic manipulation, compensating for visual occlusions and providing force/torque feedback in complex tasks. While tactile perception has a long history [66, 68, 25, 2], many approaches rely on task-specific modeling [45], hand-designed primitives [66], or hand-crafted rewards [68], limiting their generalizability. For instance, TAVI [25] uses uSkin [57] sensors and online RL driven by visual rewards but is constrained to tasks with minimal wrist movement due to safety concerns. Similarly, RoboPack [2] employs SoftBubble [37] tactile sensors and MPC but struggles with slow planning speeds in contact-rich tasks.

Recent advances in Visual Imitation Learning (e.g., [11, 71, 69]) have shown promise in learning complex tasks end-to-end. Several studies [42, 29, 8, 65] have attempted to integrate tactile sensing with imitation learning, but most focus solely on normal force readings, limiting their applicability to versatile tasks. For example, HATO [42] uses sparse tactile sensors for bimanual manipulation but lacks the resolution and shear force for more precise, contact-rich tasks. 3D-ViTac [29] adds tactile arrays to the ALOHA [71] gripper but only captures normal pressure, restricting its use to tasks with visual occlusion or in-hand force control.

Few works leverage shear force/torque data for imitation learning. MimicTouch [65] uses a GelSight [67]-based hand-held device for offline policy training but omits visual input due to the embodiment gap between human hands and robot grippers, limiting its task versatility. Similarly, [8] uses Gel-Slim [53]'s shear force field for policy transfer across robot embodiments but also excludes visual input, restricting their approach to simple tasks. In contrast, our method combines normal force, shear force, and visual RGB inputs into a unified visual-tactile policy, enabling deployment across a broader range of tasks. By integrating both tactile and visual modalities, our approach overcomes the limitations of prior works and achieves greater versatility in robotic manipulation.

III. TELEOPERATION SYSTEM: TACTAR

TactAR is an AR-based teleoperation system (see Fig. 2) that can provide real-time tactile / force feedback for complex contact-rich tasks. The main features of TactAR’s system design are:

- **Real-time tactile / force feedback via AR.** The 3D deformation / force field of the tactile / force sensors can be rendered in real time with Augmented Reality (AR). After a simple calibration step, the 3D deformation field will “attach” to the robot end-effector in AR, which can provide real-time tactile/force/torque feedbacks to the user.
- **Cross-sensor and cross-embodiment deployment.** The system can be deployed cross different types of tactile / force sensors and different robot arms (single-arm or bimanual).
- **Low-cost.** The system do not need specialized or isomorphic hardwares for haptic feedback. It only needs a consumer-level VR headeset (Meta Quest3) for teleoperation, which costs only \$500.

Next, we will introduce critical components and features of TactAR.

A. 3D Deformation Field Extraction

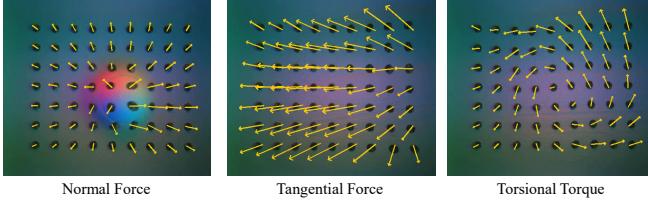


Fig. 3: Examples of marker deformation field in GelSight Mini [22] during different contact modes.

Previous works [34, 9, 14, 28] have demonstrated the importance of incorporating tactile/haptic feedback during teleoperation. For optical tactile sensors, one straightforward way to provide tactile feedback is to directly visualize the raw tactile images on the screen. However, we have found that directly presenting 2D tactile images does not enable operators to intuitively perceive force directions and torques in 3D space. Thus, we choose to render the 3D deformation field of the marker motion in Augmented Reality (AR) and attach it to the robot end-effector in virtual space. In this way, the operator can directly perceive the rich contact information of the tactile sensor (*e.g.*, the magnitude and direction of normal force, shear force and torque).

The marker array on the tactile optical sensors [67, 56, 50](see Fig. 3) can track the deformation of gel surface, which provides rich information about contact (*i.e.*, normal force , tangential force and torques). However, directly calculating the force/torque information from the 2D marker flow field requires complex calibration process with expensive force/torque sensors [70, 21]. In order to enhance the accessibility of our system, we choose to directly visualize the 3D marker

deformation field in AR. We denote the original image in frame t captured by the optical tactile sensor as I_t . We first extract the **normalized** 2D location of the marker arrays D_t with OpenCV [46] from the image I_t . Then we use a score-based tracking algorithm [23] to calculate the 2D optical flow from the undeformed frame D_0 to the current frame D_t as Eq 1:

$$F_t = [\mathbf{d}_x, \mathbf{d}_y] = \text{Flow}(D_0, D_t) \quad (1)$$

We can get 3D deformation field $V_t = [\mathbf{d}_x, \mathbf{d}_y, \mathbf{o}_z]$ used in AR, here \mathbf{o}_z denotes the z-axis offsets for 3D visualization. For force sensors, we can directly get $V_t = [\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z]$ for 3D visualization.

B. Real-time Tactile / Force Feedback Rendering in AR

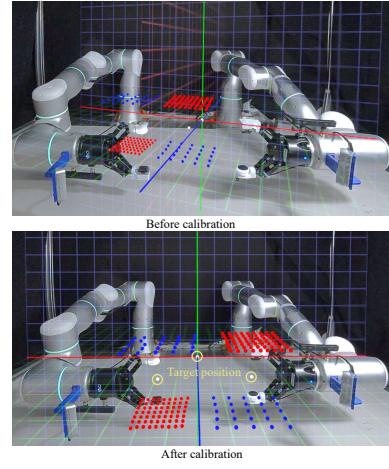


Fig. 4: Calibration process in AR. The user adjust the translation and rotation of the virtual coordinate system such that it can align with the pre-defined TCP position (the white sphere) and the origin of the world coordinate system.

In our TactAR system, we use the full-color pass-through mode of the Meta Quest3 VR headset to create an immersive AR world which is implemented by Unity. The Meta Quest 3 VR headset has a native refresh rate of 90Hz, and it has a bulit-in SLAM algorithm for pose estimation of the headset and controllers. Before teleoperation, we first align the AR coordinate system in Quest3 with the real-world robot coordinate system by a simple camera calibration process described in Fig. 4.

As shown in Fig. 2, our TactAR system consists of a work-station node, RGB camera nodes, tactile camera nodes, robot controller node and the VR headset node. We use ROS2 to perform time synchronization between different sensor topics. The robot controller node will publish the TCP pose (Tool Center Point) and the tactile camera nodes will publish the 3D deformation field V_t defined in Sec. III-A. We transform the 3D deformation field V_t based on robot TCP pose and render it in AR space to allow the 3D deformation field to “attach” to the TCP position at the robot end effector. In this way, the user can get rich contact information including the normal force,

tangential force, and the torsional torque in 3D space. Please see Appendix B and the video in the supplementary files for more details.

Our system can achieve low-latency feedback for tactile observation. Typically, the latency of the marker flow tracking algorithm is about 10ms. The latency for the force sensor is less than 1ms. The rendering latency in Quest 3 is about 10ms. And the network latency is about 1-6ms depending on the network condition. Optionally, our TactAR system also supports real-time streaming of multi-view RGB cameras (see in Fig. 2) and tactile cameras for more immersive teleoperation experience.

C. Versatility and Accessibility

Our TactAR system is designed to be versatile and easily accessible with the following properties:

- **Cross-sensor.** Our TactAR system uses the 3D deformation field as the unified representation for tactile / force feedbacks, which can be applied on many different tactile / force sensors. In this paper, we the following sensors for experiments: (1) GelSight Mini [22] (Robotics Package) optical tactile sensor. (2) Our improved MCTac [50] optical tactile sensor. (3) Built-in joint torque sensors in Flexiv Rizon [20] robot arm.
Our TactAR system is highly modularized and can even support AR visualization with different tactile / force sensors at the same time (see Fig. 1). Theoretically, our teleoperation system is also compatible with electrically tactile sensors like AnySkin [6], whose data types can typically be represented as a sparse 3D force field. This format perfectly aligns with our 3D deformation field representation.
- **Cross-embodiment deployment.** Our TactAR system can easily be deployed in different robot arms and grippers. The tactile feedbacks in AR only need the robot TCP pose and the 3D deformation field, and the robot arm uses TCP control. We also support both single-arm and bimanual arm control. Thus, TactAR will not be limited by specific hardware configuration parameters and degrees of freedom. Compared to other haptic teleoperation systems based on isomorphic hardware[34, 9], our system only needs one Meta Quest3 VR headset, which greatly reduces the reproducibility difficulty.
- **Low cost.** Our TactAR system is built with low-cost hardwares. Our experiments use two different optical tactile sensors: (1) GelSight Mini Robotics Package [22] which is a commercialized product costs \$549. (2) Our customized MCTac [50] costs around \$50 in lab fabrication, and there is large potential for lower cost in industrial manufacturing in the future. Compared to force/torque sensors like ATI mini45 [3] which costs about \$3000, the optical tactile sensors offer a significant cost advantage. The Meta Quest3 VR heaset used for teleoperation and AR feedbacks costs \$499.

IV. LEARNING ALGORITHM: REACTIVE DIFFUSION POLICY

In this section, we will introduce Reactive Diffusion Policy (RDP), which is a slow-fast imitation learning algorithm that can respond instantly to tactile / force feedback with a fast network while simultaneously maintaining the powerful modeling capabilities of diffusion with a slow network.

A. Tactile / Force Representation

For an optical tactile sensor with n marker dots, we can calculate the 2D deformation field matrix F with shape $(n, 2)$ according to Eq. 1 for each frame. Previous works [8, 47, 10] has proven that low-dimensional tactile representation based on deformation field/force field are easier to learn for contact-rich tasks compared to the raw high-dimensional tactile image. Besides, the deformation field is more robust and is less affected by noise such as differences in lighting and texture when the gel on the sensor surface is replaced or slightly damaged during evaluation.

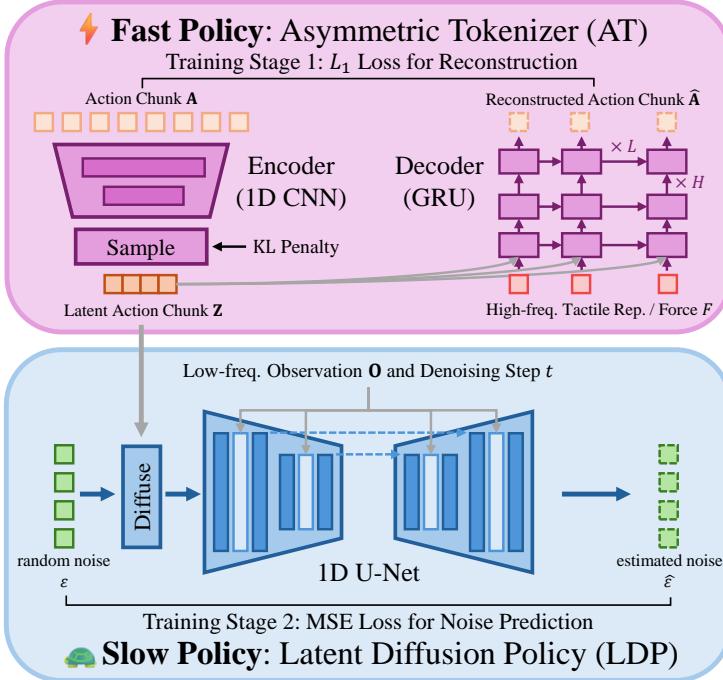
According to Zhang et al. [70], the deformation field of optical tactile sensors can be decomposed into several independent components that are highly explicable (*e.g.*, normal force, torsional force *etc*). Inspired by this, we use principal component decomposition (PCA) to reduce the dimension for the deformation field F . When encountering a new frame $F_{t'}$, we can get the d -dim tactile PCA feature with the PCA projection matrix. In practice, we find $d = 15$ is enough for reconstruction without losing much details. Please see the Appendix for more details of the PCA dataset (Appendix D) and PCA reconstruction results (Appendix C). We also visualize the first few main components of PCA and surprisingly find that these components are highly explicable (see Appendix C) and represent the force/torque of different directions. The use of PCA feature makes the model more robust to tracking errors and noise of marker deformation field. The interpretability also allows the policy network to utilize the tactile representation more easily.

For force representation, we simply concatenate the 6-D wrench (force + torque) into the observation vector.

B. Slow-Fast Policy Learning

Previous works [11, 72] have demonstrated that predicting action sequences or action chunks [38] effectively preserves temporal action consistency and handles non-Markovian or idle actions, which achieve superior performance in policy learning. However, when executing the action chunk, such approaches can be viewed as an open-loop policy, preventing it from achieving real-time feedback from high-frequency signals such as tactility. VISK [47] utilizes temporal ensembling to mitigate this issue. As shown in Fig. 5 (right), temporal ensembling finds a balance between closed-loop control and sequence consistency by aggregating the predictions of multiple iterations for the same timestep. A significant drawback of this solution is that it diminishes the policy’s ability to model multi-modal distributions and non-Markovian actions, making it prone to issues such as getting stuck. Moreover, we find that

(a) Training Pipeline of Reactive Diffusion Policy



(b) Inference Pipeline of Slow-Fast Policy

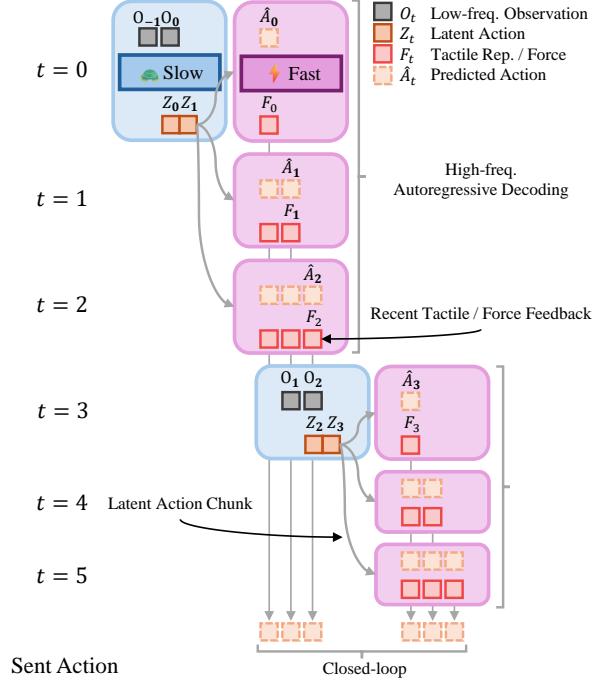


Fig. 5: Overview of **Reactive Diffusion Policy (RDP)** framework. (a) The training pipeline of RDP, comprising the first stage for training the fast policy (Asymmetric Tokenizer) and the second stage for training the slow policy (Latent Diffusion Policy). (b) The inference pipeline of RDP. The slow policy leverages low-frequency observations for modeling complex behaviors with diffusion and action chunking. The fast policy enables closed-loop control by using high-frequency tactile / force input and fine-tuning the latent action chunk predicted by the slow policy in an auto-regressive manner.

the policy performance is quite sensitive to the smoothing coefficient of temporal ensembling, which consequently reduces the policy’s applicability.

To break the above trade-off between sequence modeling and closed-loop control, we propose a slow-fast policy learning framework Reactive Diffusion Policy (RDP) as in Fig. 5. RDP is a slow-fast Latent Diffusion Model (LDM) [51], which explicitly processes signals of various frequencies at different stages. In particular, we first convert the original action chunks to the latent space by training an asymmetric tokenizer (AT). The AT decoder takes the instantaneous tactile representation apart from the latent action chuck as input. For policy learning, a slow Latent Diffusion Policy (LDP) is trained to predict the latent action chunk according to the observation in a way similar to Diffusion Policy [11]. During inference, we sample latent action chunks at a lower frequency, and within each chunk, actions are executed while the latest tactile representations are fed into the decoder of the AT to predict the real action for the next frame. This hierarchical design enables the slow network to maintain its capacity for modeling complex or non-Markovian actions by predicting temporally consistent latent action chunks and the fast network to achieve closed-loop control through real-time responsiveness. Next, we will discuss the design choices of each component within Reactive Diffusion Policy.

1) *Fast Policy*: The fast asymmetric tokenizer (AT) consists of a 1D-CNN encoder \mathcal{E} and a GRU [13] decoder \mathcal{D} . Given

an action chunk $\mathbf{A} \in \mathbb{R}^{T \times D}$ in the policy learning \mathcal{D}_{policy} , the encoder downsamples it to a latent one $\mathbf{Z} = \mathcal{E}(\mathbf{A}) \in \mathbb{R}^{T \times d}$. We choose to use a CNN-based encoder to preserve the spatial structure of the raw sequence, enabling the latent action chunk to be better processed by the latent diffusion policy, which takes sequences as input. After that, the decoder reconstructs the action via $\hat{\mathbf{A}} = \mathcal{D}(\text{concat}([\mathbf{Z}, \mathbf{F}^{reduced}]))$, where $\mathbf{F}^{reduced}$ is the corresponding tactile representation sequence proposed in Sec. IV-A. It is worth noting that we utilize tactile representation solely as input in the decoder. This deliberate asymmetry in structure is designed to ensure that the latent action chunk retains only high-level feedback strategies, while the precise locations are predicted by the decoder with the tactile information. The AT is trained using an L_1 reconstruction loss and a Kullback-Leibler (KL) penalty loss [35] as in Eq. 2.

$$L_{AT} = \mathbb{E}_{(\mathbf{A}, \mathbf{F}^{reduced}) \in \mathcal{D}_{policy}} \left[\|\mathbf{A} - \hat{\mathbf{A}}\|_1 + \lambda_{KL} L_{KL} \right]. \quad (2)$$

In practice, we keep the coefficient λ_{KL} small as in LDM [51] because we want to smooth the latent space of the AT rather than turning it into a generative model. As shown in Tab. I, our fast policy only takes less than 1ms for inference, which can even support higher-frequency inputs ($> 300Hz$) theoretically.

2) *Slow Policy*: We model the slow policy as a Diffusion Policy [11] operating on latent action chunks, which is called Latent Diffusion Policy (LDP). Diffusion Policy is a generative

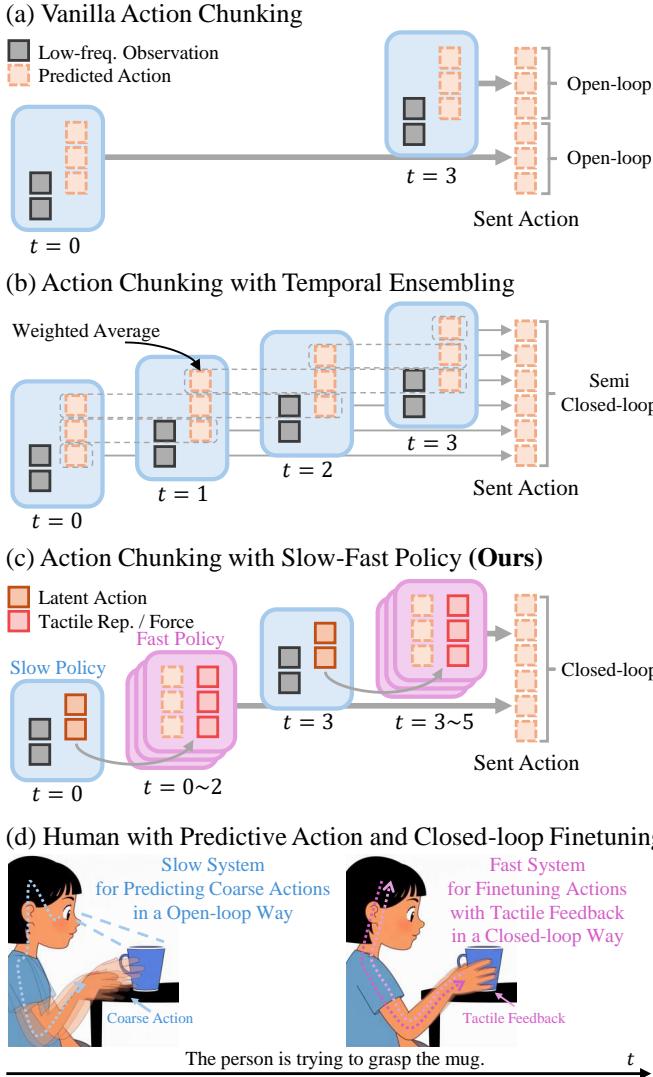


Fig. 6: Comparison among various pipelines. (a) Vanilla action chunking [11] with open-loop control during the chunk execution. (b) Action chunking enhanced with temporal ensembling [71, 42] for semi-closed-loop control. (c) Our slow-fast inference pipeline, showcasing closed-loop capabilities with fast responsive adjustments. (d) Human control patterns in contact-rich tasks.

model that iteratively denoises the noisy action \mathbf{A}^k to a clean one $\hat{\mathbf{A}}^0$ through Stochastic Langevin Dynamics [58] with the learned gradient field $\nabla E(\mathbf{A})$. To transform the model to latent space, we use the latent action chunk $\mathbf{Z}^0 = \mathcal{E}(\mathbf{A}^0)$. This modeling method offers several advantages. On the one hand, the downsampled latent representation reduces computational costs. More importantly, the asymmetric design in the AT allows challenging reactive behaviors to be excluded from latent action chunks, thereby reducing the learning difficulty of latent diffusion policy under low-frequency observation and enhancing its generalization capabilities. During training, given the observation \mathbf{O} (including image, tactility and proprioception), the gradient field is learned by ϵ_θ and the DDPM

training objective can be rewritten as

$$L_{LDP} = \mathbb{E}_{(\mathbf{O}, \mathbf{A}^0) \in \mathcal{D}_{policy}, k, \epsilon^k} \|\epsilon^k - \epsilon_\theta(\mathbf{O}, \mathbf{Z}^0 + \epsilon^k, k)\|_2, \quad (3)$$

where k is the iteration index and ϵ^k is a random noise with certain variance. We use CNN-base Diffusion Policy with FiLM-based [48] condition injection as the network architecture.

TABLE I: Inference Time of Different Modules

Diffusion Policy	Slow Policy (LDP)	Fast Policy (AE)
120ms	100ms	< 1ms

3) *Implementing Suggestions for Slow-Fast Policy:* Compared to the standard Diffusion Policy [11], our slow-fast control policy requires certain key design elements to achieve optimal performance.

- **Relative trajectory.** We use relative end-effector (EE) trajectory for action representation, proprioception, and intergripper proprioception, which has been proven to be effective even in complex tasks by UMI [12]. Specifically, instead of directly calculating the delta action between consecutive frames (may lead to large compounding errors), we convert an absolute pose trajectory to a relative one by calculating the relative transformation with respect to a base frame. In our setting, the base frame is the last observation frame in an action chunk.
- **Latency matching.** We calculate the latency caused by policy inference and action execution, and discard the first few action steps predicted by the model to send the accurately matched actions to the robot. This method has been mentioned in UMI [12] and is even more crucial for our slow-fast policy. It ensures smoother transitions between action chunks, preventing out-of-distribution tactile signals from causing the fast policy to predict abnormal actions.

V. EXPERIMENTS

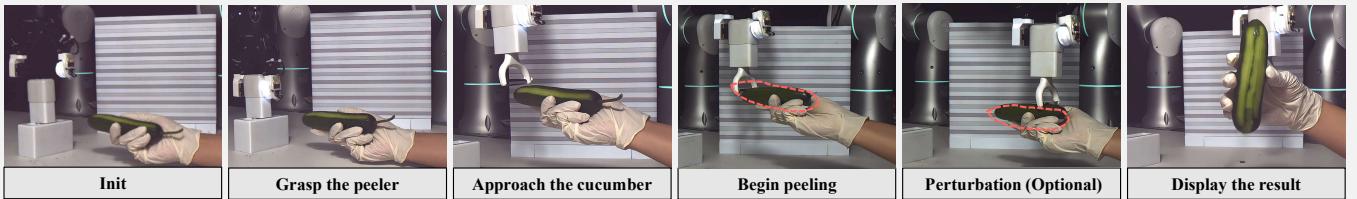
We design experiments to answer the following questions:

- **Q1:** Does tactile signals improve policy performance in contact-rich tasks?
- **Q2:** What are the capabilities of RDP?
- **Q3:** Are RDP applicable to different tactile / force sensors?
- **Q4:** Can RDP react immediately to external perturbations?
- **Q5:** Which design choices matter for RDP? Why can't we simply use small chunk size or temporal ensemble to increase closed-loop control frequency?
- **Q6:** How does tactile / force feedback in TactAR contribute to the data quality in human demonstration?

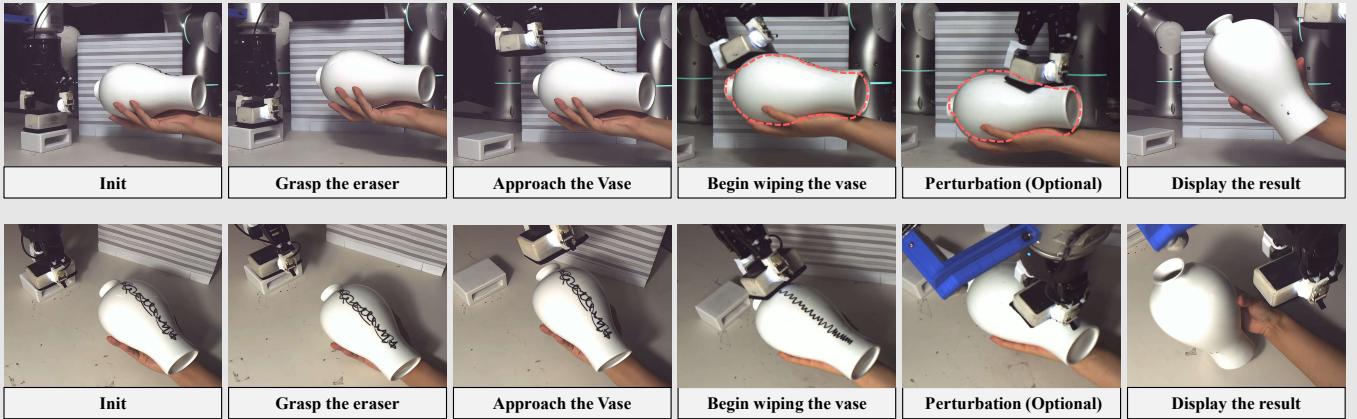
A. Setup

1) *Hardware:* The experimental platform consists of two Flexiv Rizon 4 [20] robotic arms with joint torque sensors and two Flexiv Grav [18] grippers. For single-arm tasks, we

Task1: Peeling



Task2: Wiping



Task3: Bimanual Lifting

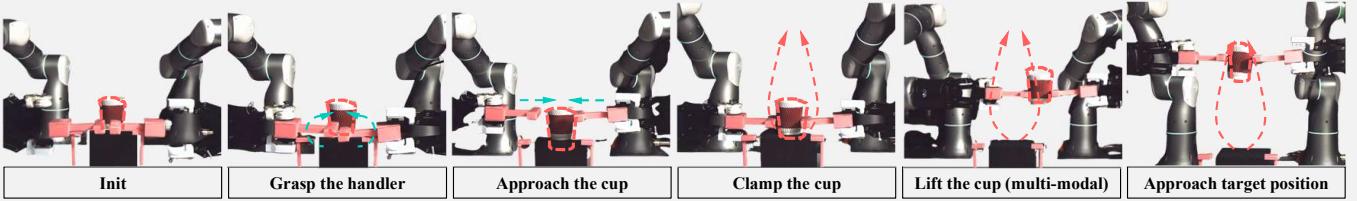


Fig. 7: Three experiment tasks including *Peeling*, *Wiping* and *Bimanual Lifting*.

only use one Realsense D435 camera on the robot arm for the wrist view. For the bimanual task, we use two Realsense D435 cameras for wrist views and a fixed Realsense D415 camera in front of the robot workspace for external view. We use three different tactile / force sensors for experiments:

- **GelSight Mini** [22] (Robotics Package) optical tactile sensor with 8MP resolution at 25 FPS, and it has a 7×9 marker dot array on the surface.
- **MCTac** [50] optical tactile sensor with 2MP resolution at 30 FPS, and it has a 5×7 marker dot array on the surface. We have improved the original design of MCTac [50], including increasing the size of the marker, reducing the density of the marker, and using white lightning for better tracking stability of the marker. Please see the Appendix A for the hardware details of the improved MCTac sensor.
- **Built-in joint torque sensors** in Flexiv Rizon 4 [20] robotic arm. We use the estimated TCP force/torque calculated by Flexiv RDK [19] for experiments. We stream the sensor data at 120Hz and downsample it to 24 FPS. Note that the estimated TCP force / torque signals have relatively larger noise compared to the force

sensor mounted on the robot end effector (*e.g.*, ATI mini 45[3]) due to inaccurate dynamics model, which further challenges the learning algorithm.

In order to evaluate policy performance under different tactile / force sensors, we attach MCTac and GelSight Mini to different fingertips of the same gripper. In this way, we can collect synchronized data from MCTac, GelSight Mini and force/torque sensors simultaneously. The TactAR teleoperation uses a Meta Quest 3 VR headset. All devices are connected to a workstation with an Intel Core i9-14900K CPU and an NVIDIA RTX 4090 GPU for both data collection and evaluation.

2) *Baselines*: We use the following baselines for comparison:

- **Diffusion Policy**: vanilla implementation of Diffusion Policy [11] with only visual input (RGB images) and open-loop action chunking.
- **Diffusion Policy (tactile image)**: Diffusion Policy with raw tactile images and visual input.
- **Diffusion Policy (tactile embedding)**: Diffusion Policy with tactile embeddings (PCA feature) and visual input.

- **Reactive Diffusion Policy (tactile embedding) (Ours):** our slow-fast policy with high-frequency tactile embedding (PCA feature) and visual input.
- **Reactive Diffusion Policy (force) (Ours):** our slow-fast policy with high-frequency wrench (force/torque) and visual input.

3) *Tasks:* As shown in Fig. 7, we evaluate Reactive Diffusion Policy with three challenging contact-rich tasks.

- **Peeling.** The robot needs to grasp the peeler, approach a cucumber held midair by a human hand, then begin peeling. This task requires the following capabilities: (1) **Precision.** The robot needs to finish the task under environment uncertainties (*e.g.*, different tool grasp locations, different cucumber pose) with high precision (millimeter-level). (2) **Fast response.** The robot needs to react instantly to human perturbations.
- **Wiping.** The robot needs to grasp the eraser, approach the vase held midair by a human hand, then begin wiping. This task requires the following capabilities: (1) **Adaptive force control with rotation.** The robot needs to adaptively track the curved vase surface with different environment uncertainties (*e.g.*, tool grasp locations, vase pose). (2) **Fast response.** The robot needs to react instantly to human perturbations.
- **Bimanual Lifting.** The two robot arms need to grasp the handlers, approach the paper cup, clamp the paper cup with the two handlers, carefully lift the cup along the trajectory of the curve without squeezing it. This task requires the following capabilities: (1) **Precise force control.** The two robots must apply precise force during the task execution. It is crucial to avoid exerting excessive force that could squeeze the cup while also ensuring that the force is sufficient to prevent the cup from slipping. (2) **Bimanual coordination.** (3) **Multi-modality.** As shown in Fig. 7, in the expert data, there are two upward lift trajectories, indicating the presence of multi-modality.

4) *Evaluation Protocols:* We use *similar initial states* across all methods for both the robots and the objects, by manually aligning the scene with the pre-defined images. There are three test-time variations for *Peeling* and *Wiping* tasks: (a) *No perturbation*. The object is fixed with a random 6D pose in the air. (b) *Perturbation before contact*. The human evaluator will move the object right before the tool makes contact. (c) *Perturbation after contact*. The human evaluator will move the object after the tool makes contact to break the contact state. There are two test-time variations for *Bimanual Lifting* task: (a) *soft paper cup*. (b) *hard paper cup*. We run 10 trials for each test-time variation. Please see Appendix E for the detailed definition of score and success rate calculation for each task.

5) *Implementation Details:* The Diffusion Policy and our slow policy (LDP) predict open-loop 12 FPS action sequences for each action chunk. The fast policy (AE) takes tactile / force observations at 24 FPS and output action predictions at 24 FPS. The final actions are interpolated and sent to robots

with a higher frequency ($>500\text{Hz}$). Note that we use 24 FPS because we are constrained by the frame rate limitation of GelSight [22], which is 25 FPS. Our RDP algorithm can also be applied to higher frequency tactile / force signals in theory. Please see Appendix F for more details on data collection and the inference process.

B. Results

Simply adding tactile signals into observation may NOT improve performance for complex contact-rich tasks (Q1). As shown in Tab. II, compared to low-dim tactile embedding (DP w. tactile emb.), simply using raw tactile images of GelSight (DP w. tactile img.) can significantly degrade the model performance of Diffusion Policy ($0.39 \rightarrow 0.03$). A possible explanation for this phenomenon is that high-dimensional image inputs are more susceptible to being out of distribution (OOD) when the dataset size is small.

We also found that Diffusion Policy (DP) incorporates tactile embedding (DP w. tactile emb.) performs similarly compared to DP with purely visual input in the three tasks (see Tab II, Tab III and Tab IV). However, despite similar performance, these two DP baselines exhibit different failure modes. We observe that DP with pure visual input frequently predicts inaccurate trajectories and results in large contact forces (*e.g.*, failure case 2 in Fig. 8 (b)), necessitating the human evaluators to move their hands to prevent sensor damage. In contrast, DP with tactile embedding rarely brings large contact forces. It may get stuck when making contact with the object (*e.g.*, failure case 2 in Fig. 8 (a) & failure case in Fig. 8 (c)). This may be because it learns some reactive behavior from the data (*e.g.*, slightly move up when making contact; move down when loose contact). However, since it executes action chunks in an open-loop manner, it lacks the capability for finer adjustments. Consequently, it might quickly transition to a different contact state and repeatedly switch between various contact states. In conclusion, we need a better way to incorporate tactile signals into the policy.

TABLE II: Policy Performance for Peeling Task

	No Perturb.	Perturb. before Contact	Perturb. after Contact	All
DP	0.56	0.58	0.19	0.44
DP w. tactile img.	0.00	0.08	0.00	0.03
DP w. tactile emb.	0.48	0.55	0.15	0.39
RDP (GelSight)	0.98	0.93	0.80	0.90
RDP (MCTac)	1.00	0.84	0.79	0.88
RDP (Force)	0.99	0.98	0.88	0.95

TABLE III: Policy Performance for Wiping Task

	No Perturb.	Perturb. before Contact	Perturb. after Contact	All
DP	0.75	0.70	0.25	0.57
DP w. tactile emb.	0.60	0.75	0.15	0.50
RDP (GelSight)	0.85	0.95	0.50	0.77
RDP (Force)	0.95	0.85	0.80	0.87

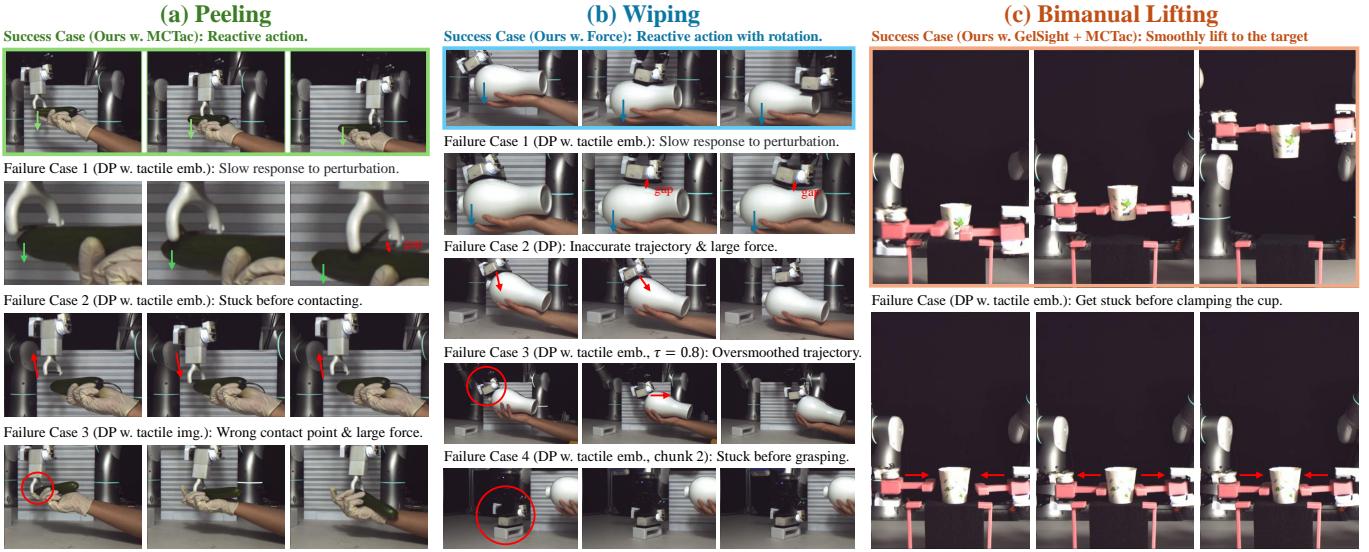


Fig. 8: Evaluation results and failure cases of baselines. Please see the website for more details.

TABLE IV: Policy Performance for Bimodal Lifting Task

	Soft Paper Cup			Hard Paper Cup			All
	Clamp	Lift	Score	Clamp	Lift	Score	Score
DP	0%	0%	0.00	0%	0%	0.00	0.00
DP w. tactile emb.	10%	10%	0.10	20%	10%	0.05	0.08
RDP (GelSight + MCTac)	100%	100%	0.55	90%	80%	0.40	0.48
RDP (Force)	100%	90%	0.80	90%	90%	0.60	0.70

RDP can perform challenging contact-rich tasks that require precision, adaptive and precise force control, or bimodal coordination (Q2). As shown in Tab. II, Tab. III and Tab. IV, RDP improves the overall score by a large margin ($> 35\%$) compared to various Diffusion Policy baselines in all three tasks. These tasks require different capabilities, including precision (*Peeling*), adaptive force control with rotation (*Wiping*) and precise force control with bimodal coordination (*Bimodal Lifting*). We believe that these capabilities are highly related to closed-loop adjustments with high-frequency tactile / force feedback.

We have performed some case studies and visualization to analyze how RDP works during these contact-rich tasks in Fig. 9. We have observed that RDP indeed learns reactive behaviors similar to those of humans. For instance, in Case Study 1, when a peeler suddenly comes into contact with a cucumber, the contact force increases abruptly. The fast policy then applies an upward adjustment to the action chunk predicted by the slow policy to reduce the contact force. In Case Study 2, as the peeler approaches the end of the cucumber, the fast policy predicts a downward corrective action, allowing the peeler to closely follow the cucumber’s surface and remove more skin. In Case Study 3, when two handlers attempt to clamp the paper cup, the fast policy swiftly predicts an outward corrective action once contact is made, preventing the cup from being squeezed. The magnitude of these reactive behavior adjustments is at the sub-millimeter level, which might appear small but significantly impacts the

overall performance of these contact-rich tasks. These minor reactive actions are hard to learn accurately for a policy with open-loop action chunking, but much easier to learn for our fast policy with high-frequency tactile input.

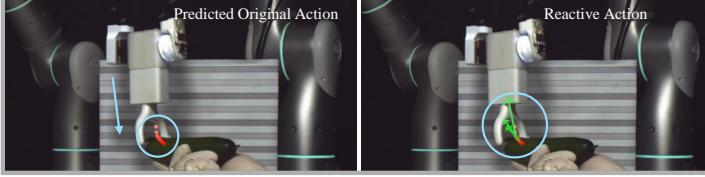
Particularly, in *Bimodal Lifting* task, we have observed the multi-modal behavior of the RDP model with force input, which indicates that our slow policy (LDP) remains the powerful modeling capabilities of diffusion model.

RDP are applicable to different tactile / force sensors (Q3). As shown in Tab. II, the performance of RDP with the two optical tactile sensors (GelSight Mini [22] and MCTac [50]) are very close (0.9 vs 0.88) on *Peeling* task. Furthermore, we surprisingly find that RDP can perform well using different tactile sensors (MCTac on the left gripper and GelSight on the right gripper) at the same time for the *Bimodal Lifting* task (see Tab. IV). Note that GelSight Mini and MCTac have different LED colors (RGB vs white), different gel material (hard vs soft), different marker arrays (7×9 vs 5×7), different resolutions (8MP vs 2MP) and different frame rates (25 FPS and 30 FPS). This validates the effectiveness of our tactile representation and the versatility of the RDP algorithm.

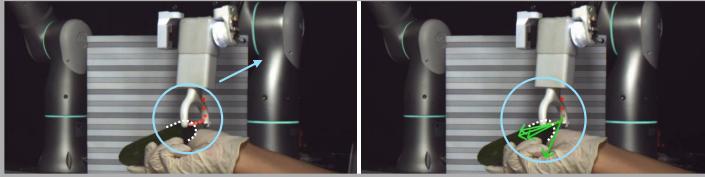
RDP is also applicable for force / torque sensors. Tab. II, Tab. III and Tab. IV have shown that RDP with simple force input without any additional design achieves the best results in all three tasks. Despite the significant noise associated with the force sensor during rapid robot movements (as evidenced in the supplementary video’s content for TactAR), the RDP algorithm successfully identifies useful patterns from the noisy data. We hypothesize that the force sensor may yield better results due to its lower latency and reduced dimensionality compared to optical tactile sensors, which potentially facilitates easier learning by the network.

RDP can react immediately to external disturbances (Q4). We find that the slow-fast design of RDP significantly improved the model’s response speed to external disturbances. For example, in Tab. II, RDP (GelSight) achieves a score

Case Study 1: Correct minor positional errors during contact using tactile feedback.



Case Study 2: Adaptive capability for tracking complex surfaces.



Case Study 3: Precise force control during clamping the cup.

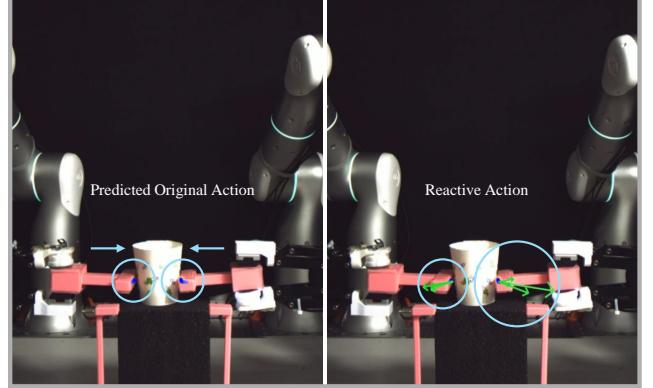


Fig. 9: Visualization of the RDP inference process. The red (left) and blue (right) dots can be seen as the predicted action chunk of the slow policy. The green arrow represents the correction direction and magnitude (scaled up for better visibility) of the reactive action predicted by the fast policy during inference. Please see the videos on the website for more details.

of 0.79 on the hardest setting (Perturbation after Contact) in *Peeling* task, while the DP baseline with tactile embedding only achieves a score of 0.15. In the *Wiping* task (see Tab. III), we find that RDP also achieves a higher score compared to DP baselines under human perturbations. Fig. 8 has shown some typical failure cases for DP baselines under human perturbations (*e.g.*, failure case 1 in *Peeling* (a) task and *Wiping* (b) task). When the robot loses contact with perturbations (*e.g.*, moving down), the DP baselines will execute the remaining trajectory in the action chunk in an open-loop manner, resulting in broken cucumber peels and residual handwriting on the vase that were not completely wiped off. In contrast, the RDP algorithm can immediately change the predicted trajectory with the fast policy in a closed-loop manner, which leads to better results.

Slow-fast hierarchy, relative trajectory and latency matching are essential for RDP performance (Q5). As shown in Fig. 5 (right), there are two ways to increase the closed-loop control frequency without our slow-fast hierarchy: (1) reducing action chunk size. (2) using temporal ensemble. However, experiments in Tab. V have proved that these two options both have significant side effects. We can see from Tab. V that when the action chunk size is reduced from 8 to 2, the DP baseline tends to get stuck before grasping (failure case 4 in Fig. 8 (b)), which makes the grasp success rate drop from 100% to 20%. Policy with small chunk size is very sensitive to non-markovian behaviors (*e.g.*, pauses in the air) commonly found in human demonstration data, so we can not simply reduce chunk size.

Temporal ensemble [71, 42] can perform semi-closed-loop control by averaging predictions from multiple timesteps. We have also tried different temporal ensemble factors τ in HATO [42], and the experiments in Tab. V have shown that the model performance is very sensitive to τ . When $\tau = 0.2$, the average weight will focus more on the newest predictions, which makes the model behavior similar to the policy with small chunk size and causes low grasp rate (30%). When $\tau = 0.8$, the average weight will focus more on the oldest

predictions, which makes the model behavior over-smoothed and hurt reactive ability (failure case 3 in Fig. 8 (b)). Thus, it is very hard to balance temporal consistency and reactivity with temporal ensemble.

As shown in Fig. 10, the relative trajectory prediction performs much better compared to the absolute action prediction in *Peeling* Task. It may be because relative trajectory are easier to learn for a smaller, fast policy, which brings a more generalizable reactive strategy from tactile feedback. In addition, the relative trajectory also compresses the latent space, facilitating the learning process of the latent diffusion policy. We also find that latency matching also contributes a lot to the policy performance (see Fig. 10) by ensuring smooth action transition between action chunks and reducing out-of-distribution (OOD) behaviors.

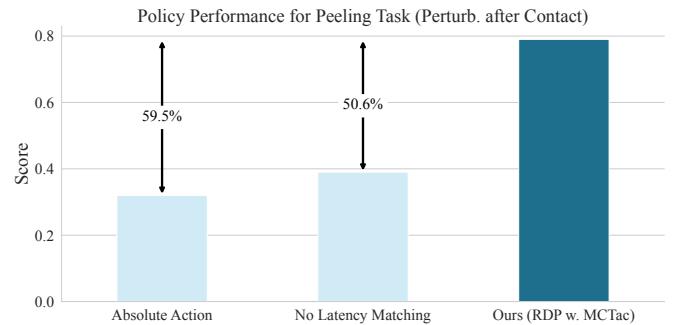


Fig. 10: Ablation Study.

TABLE V: Effects of Chunk Size and Temporal Ensemble

	action chunk size	temporal ensemble [42] factor	Wiping	
			Perturb. after Contact Grasp	Score
DP w. tactile emb.	8	-	100%	0.15
	2	-	20%	0.10
DP w. tactile emb.	8	$\tau=0.2$	30%	0.05
	8	$\tau=0.5$	0%	0.00
RDP (GelSight)	8	$\tau=0.8$	100%	0.15
	8	-	100%	0.50

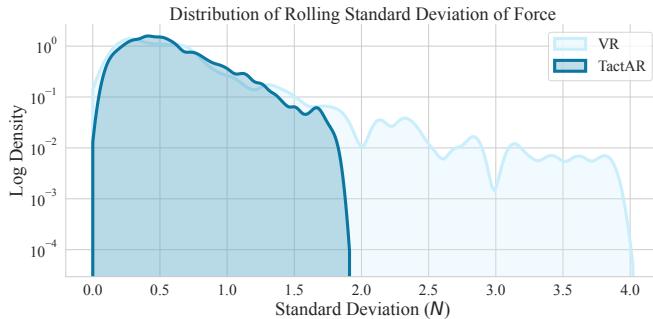


Fig. 11: Data Collection Quality Comparison. Data collected with TactAR has higher stability of contact forces compared to traditional VR teleoperation.

Tactile / force feedback in TactAR improves data quality in contact-rich tasks by improving the stability of contact forces (Q6). We use two different teleoperation settings to collect data (10 demos) for *Peeling* task with the same volunteer: (1) VR (TactAR without tactile / force feedback) (2) TactAR with tactile / force feedback, then we calculate the Rolling Standard Deviation of the recorded force curve with a window size of 10 steps, and the results are shown in Fig. 11. We can observe that using TactAR to collect data helps avoid large rolling standard deviation, indicating reduced temporal fluctuations. Data with higher stability of contact forces will help the model to discover useful patterns more easily. From an alternative perspective, the excellent performance of our RDP model also indirectly validates the effectiveness of the data collected by TactAR.

VI. LIMITATIONS AND FUTURE WORKS

While TactAR and RDP have demonstrated efficacy in numerous challenging tasks, they still have certain limitations. First, although TactAR can provide some degree of tactile / force feedback in AR, it is not as intuitive or efficient as direct human-hand operations. Future work could improve teleoperation efficiency by further reducing sensor and system latency. Second, our TactAR system is designed for two-finger grippers. Expanding our TactAR system and RDP algorithm to dexterous hands equipped with tactile sensors presents a promising direction for future research. Third, the fast policy in the RDP algorithm is currently limited to responding to high-frequency tactile / force input signals but cannot swiftly process high-frequency image inputs. Future research could consider incorporating high-frequency visual inputs into the fast network, enabling applicability to a broader range of task types. Lastly, the RDP algorithm is currently limited to single-task scenarios. Future work could integrate RDP with the Vision-Language Action (VLA) model by replacing the VLA tokenizer with an asymmetric tokenizer similar to the one in the RDP algorithm. This integration could introduce reactive behavior for closed-loop control with real-time tactile / force feedback within the VLA.

VII. CONCLUSION

In this paper, we presented TactAR, a low-cost teleoperation system providing real-time tactile / force feedback through AR, and Reactive Diffusion Policy (RDP), a novel slow-fast imitation learning algorithm for contact-rich manipulation. TactAR demonstrates that high-quality tactile /force feedback can be achieved through AR visualization without expensive specialized hardware. RDP successfully addresses the trade-off between sequence modeling and closed-loop control through its hierarchical design - using a slow network for complex trajectory planning and a fast network for reactive tactile feedback control. Through extensive experiments on three challenging contact-rich tasks, we demonstrated that RDP significantly outperforms state-of-the-art visual IL baselines in terms of both task completion and reactivity to tactile feedback. The cross-sensor experiments further validated that our approach generalizes well across different tactile / force sensors. We believe this work takes an important step toward making visual-tactile imitation learning more practical and accessible.

REFERENCES

- [1] Malek Aburub, Cristian C Beltran-Hernandez, Tatsuya Kamijo, and Masashi Hamaya. Learning diffusion policies from demonstrations for compliant contact-rich manipulation. *arXiv preprint arXiv:2410.19235*, 2024.
- [2] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Cheston Tan, Yunzhu Li, and Jiajun Wu. Robopack: Learning tactile-informed dynamics models for dense packing. 2024. URL <https://arxiv.org/abs/2407.01418>.
- [3] ATI mini45 force/torque sensor. https://www.ati-ia.com/products/ft_ft_models.aspx?id=mini45, 2024.
- [4] Tapomayukh Bhattacharjee, Advait Jain, Sarvagya Vaish, Marc D Killpack, and Charles C Kemp. Tactile sensing over articulated joints with stretchable sensors. In *2013 World Haptics Conference (WHC)*, pages 103–108. IEEE, 2013.
- [5] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *5th Annual Conference on Robot Learning*, 2021.
- [6] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Er-ciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. Anyskin: Plug-and-play skin sensing for robotic touch. *arXiv preprint arXiv:2409.08276*, 2024.
- [7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.
- [8] William van den Bogert, Madhavan Iyengar, and Nima Fazeli. Built different: Tactile perception to overcome cross-embodiment capability differences in collaborative manipulation. *arXiv preprint arXiv:2409.14896*, 2024.
- [9] Thanpimon Buamanee, Masato Kobayashi, Yuki Uranshi, and Haruo Takemura. Bi-act: Bilateral control-based

- imitation learning via action chunking with transformer. *arXiv preprint arXiv:2401.17698*, 2024.
- [10] Weihang Chen, Jing Xu, Fanbo Xiang, Xiaodi Yuan, Hao Su, and Rui Chen. General-purpose sim2real protocol for learning contact-rich manipulation with marker-based visuotactile sensors. *IEEE Transactions on Robotics*, 2024.
- [11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [12] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.
- [13] Kyunghyun Cho. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- [14] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.
- [15] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 653–660. IEEE, 2024.
- [16] Hongjie Fang, Hao-Shu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15031–15038. IEEE, 2024.
- [17] J Randall Flanagan, Miles C Bowman, and Roland S Johansson. Control strategies in object manipulation tasks. *Current opinion in neurobiology*, 16(6):650–659, 2006.
- [18] Flexiv Grav force control gripper. <https://www.flexiv.com/products/rizon>, 2024.
- [19] Flexiv RDK (Robotic Development Kit). <https://www.flexiv.com/software/rdk>, 2024.
- [20] Flexiv Rizon 4 robotic arm. <https://www.flexiv.com/products/rizon>, 2024.
- [21] Yuni Fuchioka and Masashi Hamaya. An electromagnetism-inspired method for estimating in-grasp torque from visuotactile sensors. *arXiv preprint arXiv:2404.15626*, 2024.
- [22] Gelsight Mini tactile sensor. <https://www.gelsight.com/gelsightmini/>, 2024.
- [23] Gelsight SDK. <https://github.com/gelsightinc/gsrobotics>, 2024.
- [24] Oliver Glauser, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Deformation capture via soft and stretchable sensor arrays. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019.
- [25] Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13825–13832. IEEE, 2024.
- [26] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu. Foar: Force-aware reactive policy for contact-rich robotic manipulation. *arXiv preprint arXiv:2411.15753*, 2024.
- [27] Tess Hellebrekers, Oliver Kroemer, and Carmel Majidi. Soft magnetic skin for continuous deformation sensing. *Advanced Intelligent Systems*, 1(4):1900025, 2019.
- [28] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. *arXiv preprint arXiv:2410.09309*, 2024.
- [29] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d vitac:learning fine-grained manipulation with visuo-tactile sensing. In *Proceedings of Robotics: Conference on Robot Learning(CoRL)*, 2024.
- [30] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024.
- [31] Roland S Johansson. Sensory input and control of grip. In *Novartis Foundation Symposium 218-Sensory Guidance of Movement: Sensory Guidance of Movement: Novartis Foundation Symposium 218*, pages 45–63. Wiley Online Library, 2007.
- [32] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, 2009.
- [33] Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding. *arXiv preprint arXiv:2501.04693*, 2025.
- [34] Tatsuya Kamijo, Cristian C Beltran-Hernandez, and Masashi Hamaya. Learning variable compliance control from a few demonstrations for bimanual robot with haptic feedback teleoperation system. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12663–12670. IEEE, 2024.
- [35] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [36] Masato Kobayashi, Thanpimon Buamanee, and Takumi Kobayashi. Alpha-alpha and bi-act are all you need: Importance of position and force information/control for imitation learning of unimanual and bimanual robotic manipulation with low-cost system. *arXiv preprint*

- arXiv:2411.09942*, 2024.
- [37] Naveen Kuppuswamy, Alex Alspach, Avinash Uttamchandani, Sam Creasey, Takuya Ikeda, and Russ Tedrake. Soft-bubble grippers for robust and perceptive manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9917–9924. IEEE, 2020.
- [38] Lucy Lai, Ann Zixiang Huang, and Samuel J Gershman. Action chunking as policy compression. 2022.
- [39] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.
- [40] Changyi Lin, Ziqi Lin, Shaoxiong Wang, and Huazhe Xu. Dtact: A vision-based tactile sensor that measures high-resolution 3d geometry directly from darkness. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10359–10366. IEEE, 2023.
- [41] Changyi Lin, Han Zhang, Jikai Xu, Lei Wu, and Huazhe Xu. 9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation. *IEEE Robotics and Automation Letters*, 2023.
- [42] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. *arXiv:2404.16823*, 2024.
- [43] Wenhui Liu, Junbo Wang, Yiming Wang, Weiming Wang, and Cewu Lu. Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. *arXiv preprint arXiv:2410.07554*, 2024.
- [44] mctac 2.0. Mctac-2.0, 01 2025. URL <https://github.com/Tacxels/MCTac-2.0>.
- [45] Miquel Oller, Dmitry Berenson, and Nima Fazeli. Tactile-driven non-prehensile object manipulation via extrinsic contact mode control. *arXiv preprint arXiv:2405.18214*, 2024.
- [46] OpenCV. <https://opencv.org/>, 2024.
- [47] Venkatesh Pattabiraman, Yifeng Cao, Siddhant Haldar, Lerrel Pinto, and Raunaq Bhirangi. Learning precise, contact-rich manipulation through uncalibrated tactile skins. *arXiv preprint arXiv:2410.17246*, 2024.
- [48] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [49] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023.
- [50] Jieji Ren, Jiang Zou, and Guoying Gu. Mc-tac: Modular camera-based tactile sensor for robot gripper. In *International Conference on Intelligent Robotics and Applications*, pages 169–179. Springer, 2023.
- [51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [52] Sigma-7 haptic interface. <https://www.forcedimension.com/products/sigma>, 2024.
- [53] Andrea Sipos, William van den Bogert, and Nima Fazeli. Gelslim 4.0: Focusing on touch and reproducibility. *arXiv preprint arXiv:2409.19770*, 2024.
- [54] Stefano Stassi, Valentina Cauda, Giancarlo Canavese, and Candido Fabrizio Pirri. Flexible tactile sensing based on piezoresistive composites: A review. *Sensors*, 14(3):5296–5332, 2014.
- [55] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, 2019.
- [56] Ian H Taylor, Siyuan Dong, and Alberto Rodriguez. Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787. IEEE, 2022.
- [57] Tito Pradhono Tomo, Massimo Regoli, Alexander Schmitz, Lorenzo Natale, Harris Kristanto, Sophon Somlor, Lorenzo Jamone, Giorgio Metta, and Shigeki Sugano. A new silicone structure for uskin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot icub. *IEEE Robotics and Automation Letters*, 3(3):2584–2591, 2018.
- [58] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.
- [59] Nicholas Wettels, Veronica J Santos, Roland S Johansson, and Gerald E Loeb. Biomimetic tactile sensor array. *Advanced robotics*, 22(8):829–849, 2008.
- [60] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.
- [61] Te-Yen Wu, Lu Tan, Yuji Zhang, Teddy Seyed, and Xing-Dong Yang. Capacitivo: Contact-based object recognition on interactive fabrics using capacitive sensing. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*, pages 649–661, 2020.
- [62] Boxin Xu, Luoyan Zhong, Grace Zhang, Xiaoyu Liang, Diego Virtue, Rishabh Madan, and Tapomayukh Bhattacharjee. Cushsense: Soft, stretchable, and comfortable tactile-sensing skin for physical human-robot interaction. *arXiv preprint arXiv:2405.03155*, 2024.
- [63] Daniel Xu, Andreas Tairych, and Iain A Anderson.

- Stretch not flex: programmable rubber keyboard. *Smart Materials and Structures*, 25(1):015012, 2015.
- [64] Youcan Yan, Zhe Hu, Zhengbao Yang, Wenzhen Yuan, Chaoyang Song, Jia Pan, and Yajing Shen. Soft magnetic skin for super-resolution tactile sensing with force self-decoupling. *Science Robotics*, 6(51):eabc8801, 2021.
 - [65] Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *8th Annual Conference on Robot Learning*, 2024.
 - [66] Zhenjun Yu, Wenqiang Xu, Siqiong Yao, Jieji Ren, Tutian Tang, Yutong Li, Guoying Gu, and Cewu Lu. Precise robotic needle-threading with tactile perception and reinforcement learning. In *Conference on Robot Learning*, pages 3266–3276. PMLR, 2023.
 - [67] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
 - [68] Ying Yuan, Haichuan Che, Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Kang-Won Lee, Yi Wu, Soo-Chul Lim, and Xiaolong Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6558–6565. IEEE, 2024.
 - [69] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
 - [70] Yanzhan Zhang, Zicheng Kan, Yang Yang, Yu Alexander Tse, and Michael Yu Wang. Effective estimation of contact force and torque for vision-based tactile sensors with helmholtz–hodge decomposition. *IEEE Robotics and Automation Letters*, 4(4):4094–4101, 2019.
 - [71] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
 - [72] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
 - [73] Bo Zhou, Ruixuan Jiao, Yi Li, Fang Fang, and Fu Chen. Contact compliance visuo-proprioceptive policy for contact-rich manipulation with cost-efficient haptic hand-arm teleoperation system. *arXiv e-prints*, pages arXiv–2409, 2024.

APPENDIX

A. Hardware Details

To acquire a stable and sensitive tactile signal, we have improved the camera-based tactile sensor based on the open source MCTac [50, 44], details are shown in Fig. 12. First, we redesign the mechanical interface module to fit the connector of Flexiv Grav Gripper [18]. We also design a mount for the camera for more convenient cable management. Second, because we pay more attention to the shear force in this work, we utilized the white side illumination in stand of the colorful LEDs, which not only highlight the markers and their displacement, but also simplified the system complexity. Finally, to increase the tracking stability and to avoid the tracking confusion (which can lead to abnormal values under large shear displacement), we have enlarged the gel marker size (1 mm) and their spacing (3 mm). This sparse marker array not only satisfied the tactile requirement, but also avoid the confusion and losing in marker tracking. After painting the reflection layer, we take the laser marking machine to process the surface with 5x6 circle markers array (can remove the reflection layer in these regions), then we spray black ink on the proceeded surface to increase the contrast of tactile image, as shown in [56]. After fabricating, we have assembled the customized camera-based tactile sensor, integrated it on the robot gripper, and token the USB-HUB to collect the observed tactile image stream.

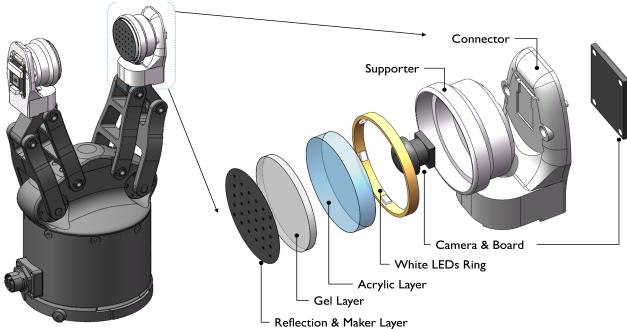


Fig. 12: Improved MCTac Sensor for our task. The left part is the gripper integrated illustration, and the right part is the detailed structure and components of the camera-based tactile sensor.

B. Implementation Details of TactAR

The VR headset will receive the latest robot TCP pose transformation matrix W_t^{TCP} , the 3D deformation field V_t and the undeformed marker locations D_0 . If we use multiple 3D arrows to represent the 3D deformation field in robot TCP coordinate system, we can calculate the start points P_{start}^{TCP} and the end points P_{end}^{TCP} of the 3D arrows by Eq. 4. Finally, we convert the 3D arrows from the robot TCP coordinate system to the world coordinate system in AR by Eq. 5, here s is the scale factor for better visibility.

$$P_{start}^{TCP} = D_0, P_{end}^{TCP} = D_0 + V_t \quad (4)$$

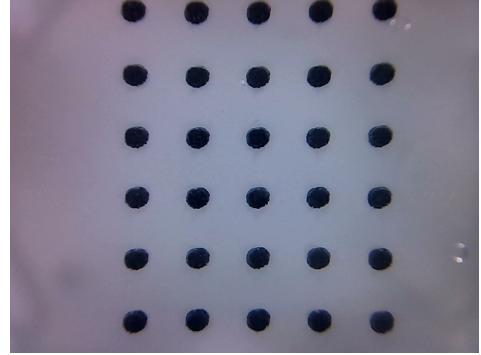


Fig. 13: The example image of our improved MCTac optical tactile sensor.

$$P_{start}^{AR} = sW_t^{TCP}D_0, P_{end}^{AR} = sW_t^{TCP}(D_0 + V_t) \quad (5)$$

Finally, we render the 3D arrows $P_{start}^{AR}, P_{end}^{AR}$ in AR with Quest3. We adaptively change the color of a 3D arrow based on its length to enhance its visibility. As shown in Fig. 3, the tangential force and the torsional torque can be easy to recognize from the 2D flow field, but the normal force that is typically manifested as an outward diffusion pattern can be challenging to observe intuitively. When the gripper is closed with large normal forces, the marker pattern may be dominant by the outward diffusion pattern. So we additionally implement an auto-detection mechanism, which automatically resets the visualized marker flow when the gripper is closed or open.

C. Details of the Tactile Representation

Specifically, we collect a small tactile dataset $\mathcal{D}_{tactile}$ of random interaction with different objects and calculate the deformation field matrix F_t for each frame t in $\mathcal{D}_{tactile}$. Please see the Appendix D for more details of $\mathcal{D}_{tactile}$. The dataset $\mathcal{D}_{tactile}$ has m frames, and we concatenate and reshape the deformation field F_t in all frames as a new matrix F_{concat} with shape $(m, 2n)$. We perform PCA on F_{concat} and get the matrix after dimensionality reduction $F_{reduced} = T_{proj}F_{concat}$ with shape (m, d) , where T_{proj} is the PCA projection matrix and d is the number of main components. When encountering a new frame $F_{t'}$, we can get the d -dim tactile PCA feature by $f_{t'}^{reduced} = T_{proj}F_{t'}$. In practice, we find $d = 15$ is enough for reconstruction without losing much details. Fig. 14 has shown the reconstruction results of PCA. Fig. 15 has shown the first few main components of PCA in our tactile representation.

D. Details of the Data Collection Process

For the tactile dataset used for PCA embedding, we collect 30 minutes of random interaction data with Gelsight Mini [22] and 40 minutes of random interaction data with MCTac [50] with 20 objects in Fig. 16.

For three tasks used in our experiments, we collect 60 demonstrations for *Peeling* task, 80 demonstrations for *Wiping* task and 50 demonstrations for *Bimanual Lifting* tasks with TactAR system. During the data collection process, we

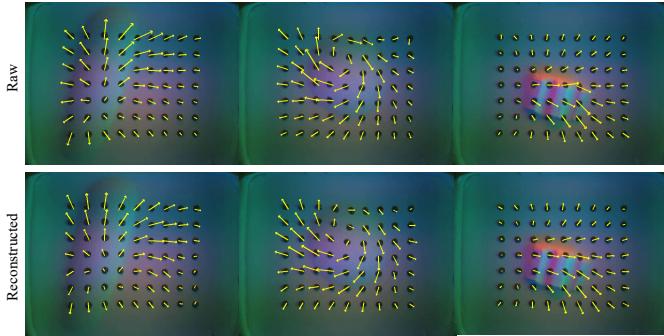


Fig. 14: Examples of PCA reconstruction results for marker deformation field of GelSight Mini[22]. The raw tactile images are showed as the background.

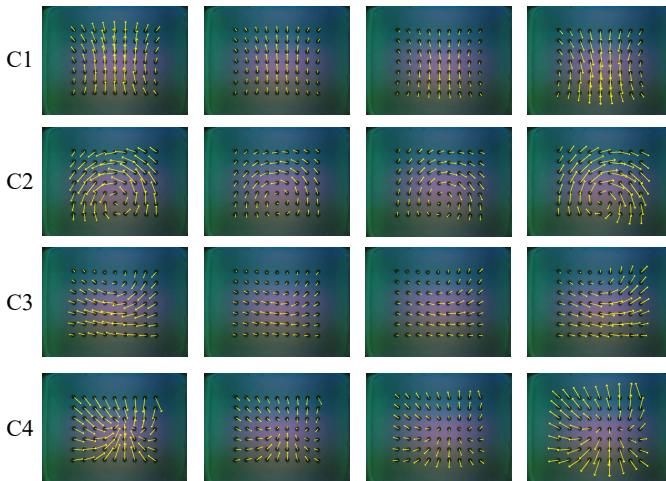


Fig. 15: The first four main components (C1-C4) of PCA in our tactile representation. These components are corresponding to tangential force (C1, C3), torsional torque (C2) and normal force (C4) respectively.

proactively recorded some reactive behaviors to enhance the robustness of the model.

E. Details of Evaluation Protocols

Due to the involvement of human interaction during testing, we employed a **single-blind** testing methodology to minimize the impact of subjective judgment. In each experiment, a model is randomly selected for evaluation, and the evaluator is unaware of which model is being assessed each time.

For *Peeling* task, we calculate the score based on the proportion of the peeled cucumber skin to the total length of the cucumber, normalized by the average score of the demonstration data. For *Wiping* task, we calculate the score based on the size of the remaining handwriting compared to the demonstration data. If the residue reaches the human demonstration level, the score is 1; If there is minor residue (less than one third of the handwriting length), the score is 0.5; If significant residue remains, the score is 0. For *Bimanual Lifting* task, if the paper cup is lifted into the air following



Fig. 16: The objects used in the tactile dataset.

the designated trajectory without significant compression, the score will be 1; If the paper cup is partially compressed in the air, the score will be 0.5; If the cup is not lifted up, or dropped in the air, the score will be 0.

F. Details of the Inference Process

We use observation $T_o = 2$ for all Diffusion Policy baselines and our Latent Diffusion Policy (LDP). The Diffusion Policy and our slow policy (LDP) predict open-loop 12FPS action sequences. The default action chunk size is about 0.67s in real-world time. The slow policy will periodically (1-2Hz) predict new inference results at time intervals determined by the action chunk. The actions are interpolated and sent to robots with a higher frequency (>500Hz). The fast policy (AE) takes tactile / force observations at 24 FPS and output action predictions at 24 FPS.