

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‡}

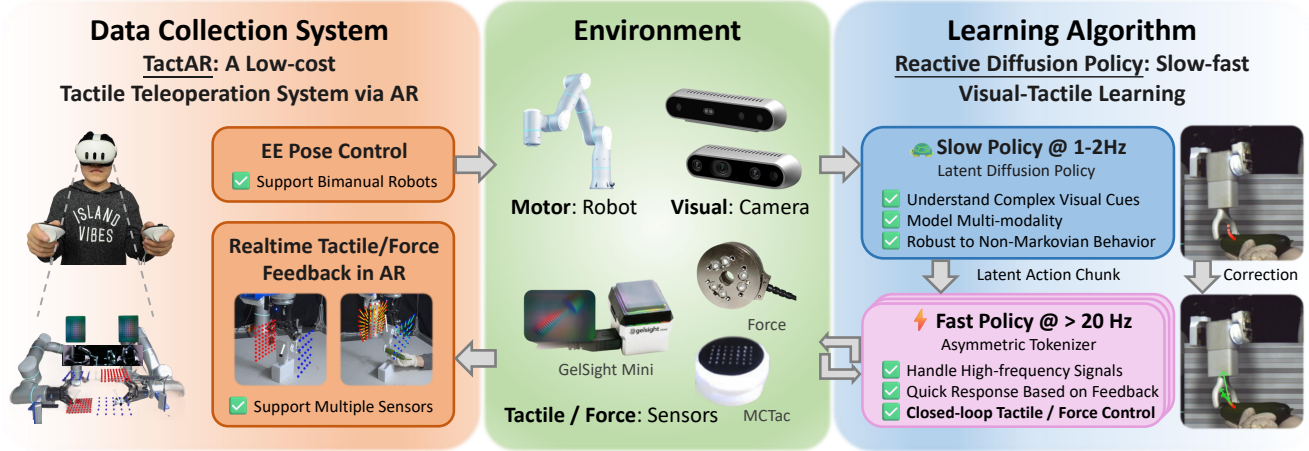


Fig. 1: **TactAR** is a low-cost and versatile teleoperation system which can provide real-time tactile / force feedback via Augmented Reality (AR). **Reactive Diffusion Policy (RDP)** is a slow-fast imitation learning algorithm that can model complex behaviors with a slow policy network and achieve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

I. INTRODUCTION

Contact-rich manipulation tasks that appear simple to humans remain challenging for robots. Research in neuroscience [4, 10, 11] shows that human manipulation involves both **predictive action** and **closed-loop fine-tuning**. While recent visual imitation learning [2, 19] methods have shown promise through action chunking, they operate in open-loop during chunk execution, limiting their reactivity. In addition, most approaches lack integration of fine-grained tactile feedback, restricting them to low-precision tasks.

To address these challenges, we propose:

- **TactAR**: A **teleoperation system** providing real-time tactile feedback through Augmented Reality (AR).
- **RDP**: An **imitation learning algorithm** that combines a slow policy for complex trajectory modeling with a fast policy for closed-loop tactile feedback control.

Our approach enables both complex action modeling and quick reactive behavior within a unified framework. Experiments on three challenging contact-rich tasks demonstrate significant performance improvements over baselines, while maintaining applicability on three different tactile / force sensors (e.g., Gelsight Mini [8] and joint torque sensors). More videos and analysis are available in the appendices and on reactive-diffusion-policy.github.io.

II. TELEOPERATION SYSTEM: TACTAR

TactAR is an AR-based teleoperation system that provides real-time tactile / force feedback for contact-rich tasks with three key features:

- **Real-time tactile feedback via AR**: 3D deformation fields are rendered and attached to the robot end-effector in AR, providing intuitive tactile / force feedback.
- **Cross-sensor compatibility**: TactAR supports multiple tactile / force sensors (GelSight Mini [8], MCTac [16], and joint torque sensors).
- **Low-cost**: Requires only a consumer-level Meta Quest 3 headset (\$500).

Please see Appendix I and our [website](#) for more details.

III. LEARNING ALGORITHM: RDP

We introduce Reactive Diffusion Policy (RDP), a slow-fast imitation learning algorithm that responds instantly to feedback while maintaining powerful action modeling capabilities. Please see Appendix II for more details.

A. Tactile / Force Representation

For optical tactile sensors, we extract a low-dimensional representation from the 2D deformation field matrix using principal component analysis (PCA). For force sensors, we directly use the 6D wrench vector.

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

*Equal contribution [†]Equal contribution [‡]Equal advising

TABLE I: 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.	0.60	0.49	0.16	0.41
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 II: 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

B. Slow-Fast Policy Learning

Previous action chunking approaches operate in open-loop during chunk execution, preventing real-time feedback incorporation. To overcome this limitation, we propose a hierarchical framework with two key components:

1) *Fast Policy*: The Asymmetric Tokenizer (AT) consists of a 1D-CNN encoder \mathcal{E} and a GRU decoder \mathcal{D} . The encoder downsamples action chunks to latent space: $\mathbf{Z} = \mathcal{E}(\mathbf{A})$. The decoder reconstructs actions using both latent vectors \mathbf{Z} and tactile feedback $\mathbf{F}^{reduced}$: $\hat{\mathbf{A}} = \mathcal{D}(\text{concat}([\mathbf{Z}, \mathbf{F}^{reduced}]))$. This asymmetric design ensures latent chunks contain high-level strategies while precise actions are predicted with tactile information. The AT is trained with:

$$L_{AT} = \mathbb{E}_{(\mathbf{A}, \mathbf{F}^{reduced}) \in \mathcal{D}_{policy}} \left[\|\mathbf{A} - \hat{\mathbf{A}}\|_1 + \lambda_{KL} L_{KL} \right] \quad (1)$$

Importantly, the fast policy achieves sub-millisecond inference time, enabling high-frequency control.

2) *Slow Policy*: The Latent Diffusion Policy (LDP) operates on latent action chunks using a diffusion model framework. This approach reduces computational costs while keeping the powerful capability of modeling complex actions according to visual inputs. The training objective is:

$$L_{LDP} = \mathbb{E}_{(\mathbf{O}, \mathbf{A}^0) \in \mathcal{D}_{policy}, k, \epsilon^k} \|\epsilon^k - \epsilon_{\theta}(\mathbf{O}, \mathbf{Z}^0 + \epsilon^k, k)\|_2 \quad (2)$$

IV. EXPERIMENTS

Please see Appendix III for more details.

A. Tasks and Evaluation

We evaluate on three challenging contact-rich tasks:

- **Peeling**: Requires precision and fast response to perturbations
- **Wiping**: Requires adaptive force control with rotation and fast response
- **Bimanual Lifting**: Requires precise force control and bimanual coordination

We conduct experiments with three different sensors: GelSight Mini [8], MCTac [16], and joint torque sensors of Flexiv Rizon 4 arms [7].

TABLE III: Policy Performance for Bimanual 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

B. Results and Analysis

Tactile integration approach matters. Simply adding tactile signals to observations does not necessarily improve performance. As shown in Table I, whether using raw tactile images or using tactile embeddings in standard Diffusion Policy (DP) performs similarly to visual-only approaches. This suggests that effective tactile integration requires architectural changes beyond simply adding inputs.

RDP significantly outperforms baselines. As shown in Tab. I, Tab. II and Tab. III, 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 bimanual coordination (*Bimanual Lifting*). We believe these capabilities are highly related to closed-loop adjustments with high-frequency tactile / force feedback.

Cross-sensor applicability. RDP performs well with different tactile / force sensors despite their varying characteristics. Performance with GelSight Mini and MCTac is comparable (0.90 vs 0.88 on Peeling), and RDP can even utilize different sensors simultaneously in bimanual tasks. Force sensor-based RDP consistently achieves the best results, possibly due to lower latency and dimensionality.

Fast reactivity to perturbations. RDP shows superior performance under perturbations, especially after contact is established. In the Peeling task, RDP achieves 0.80 score under post-contact perturbations compared to 0.15 for DP with tactile embedding. This demonstrates the effectiveness of the fast policy in providing immediate corrections based on tactile feedback.

Tactile / force feedback in TactAR improves data quality. We have conducted a user study on how tactile / force feedback in TactAR helps the data collection process. We invited 10 users with different levels of experience in VR teleoperation and Imitation Learning (IL). We perform 200 trials in total for quantitative analysis. Results show that most of the users ($\geq 70\%$) found that tactile / force AR feedback is very helpful in data collection. We also find that tactile / force feedback in TactAR can greatly improve the data quality from both the normalized peeling length ($0.72 \rightarrow 0.91$) and the ratio of stable contact force ($0.58 \rightarrow 0.87$).

V. CONCLUSION

In this paper, we present TactAR and RDP to collect high-quality data and learn reactive policy for contact-rich manipulation. RDP addresses the trade-off between sequence modeling and closed-loop control through its slow-fast design and outperforms SOTA visual IL baselines in experiments. We believe that this work takes an important step toward making visual-tactile IL more practical and accessible.

REFERENCES

- [1] ATI mini45 force/torque sensor. https://www.ati-ia.com/products/ft/ft_models.aspx?id=mini45, 2024. **5**
- [2] 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. **1, 4, 5, 7**
- [3] 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. **4**
- [4] 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. **1**
- [5] Flexiv Grav force control gripper. <https://www.flexiv.com/products/rizon>, 2024. **4**
- [6] Flexiv RDK (Robotic Development Kit). <https://www.flexiv.com/software/rdk>, 2024. **5**
- [7] Flexiv Rizon 4 robotic arm. <https://www.flexiv.com/products/rizon>, 2024. **2, 4, 5**
- [8] Gelsight Mini tactile sensor. <https://www.gelsight.com/gelsightmini/>, 2024. **1, 2, 4, 5, 6**
- [9] Gelsight SDK. <https://github.com/gelsightinc/gelsightinc>, 2024. **4**
- [10] 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. **1**
- [11] 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. **1**
- [12] 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. **6, 7**
- [13] mctac 2.0. Mctac-2.0, 01 2025. URL <https://github.com/Tacxels/MCTac-2.0>. **5**
- [14] OpenCV. <https://opencv.org/>, 2024. **4**
- [15] 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. **4**
- [16] 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. **1, 2, 5**
- [17] 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. **4**
- [18] 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. **4**
- [19] 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. **1, 6, 7**

APPENDIX I FEATURES OF TACTAR

A. 3D Deformation Field Extraction

The gel surface’s marker array (Fig. 3) captures rich contact information, but deriving forces from 2D optical flow requires complex calibration with expensive sensors. To improve accessibility, we instead visualize the 3D deformation field. From tactile images I_t , we extract normalized marker positions D_t using OpenCV [14]. We use a score-based tracking algorithm [9] to calculate 2D optical flow between the initial frame D_0 and the current frame D_t :

$$F_t = [\mathbf{d}_x, \mathbf{d}_y] = \text{Flow}(D_0, D_t) \quad (3)$$

The 3D deformation field $V_t = [\mathbf{d}_x, \mathbf{d}_y, \mathbf{o}_z]$ (with z-offset \mathbf{o}_z) is then rendered in AR. For force sensors, we directly visualize $V_t = [\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z]$.

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

The Meta Quest 3 VR headset has a native refresh rate of 90Hz, and it has a built-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.

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.

APPENDIX II COMPONENTS OF RDP

A. Slow-Fast Policy Learning

As shown in Fig. 6, 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. 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.

1) *Fast Policy*: 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. It is also 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. During training, we keep the coefficient λ_{KL} small as in LDM [17] because we want to smooth the latent space of the AT rather

than turning it into a generative model. As shown in Tab. IV, our fast policy only takes less than 1ms for inference, which can even support higher-frequency inputs ($> 300\text{Hz}$) theoretically.

2) *Slow Policy*: We model the slow policy as a Diffusion Policy [2] operating on latent action chunks, which is called Latent Diffusion Policy (LDP). Diffusion Policy is a generative model that iteratively denoises the noisy action \mathbf{A}^k to a clean one $\hat{\mathbf{A}}^0$ through Stochastic Langevin Dynamics [18] 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 a network ϵ_θ . We use CNN-base Diffusion Policy with FiLM-based [15] condition injection as the network architecture.

TABLE IV: Inference Time of Different Modules

Diffusion Policy	Slow Policy (LDP)	Fast Policy (AT)
120ms	100ms	$< 1\text{ms}$

3) *Implementing Suggestions for Slow-Fast Policy*: Compared to the standard Diffusion Policy [2], 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 and proprioception, which has been proven to be effective even in complex tasks by UMI [3].
- **Latency matching.** This method has been mentioned in UMI [3] 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.

APPENDIX III EXPERIMENTS

A. Setup

1) *Hardware*: The experimental platform consists of two Flexiv Rizon 4 [7] robotic arms with joint torque sensors and two Flexiv Grav [5] grippers. For single-arm tasks, we 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** [8] (Robotics Package) optical tactile sensor with 8MP resolution at 25 FPS, and it has a 7×9 marker dot array on the surface.

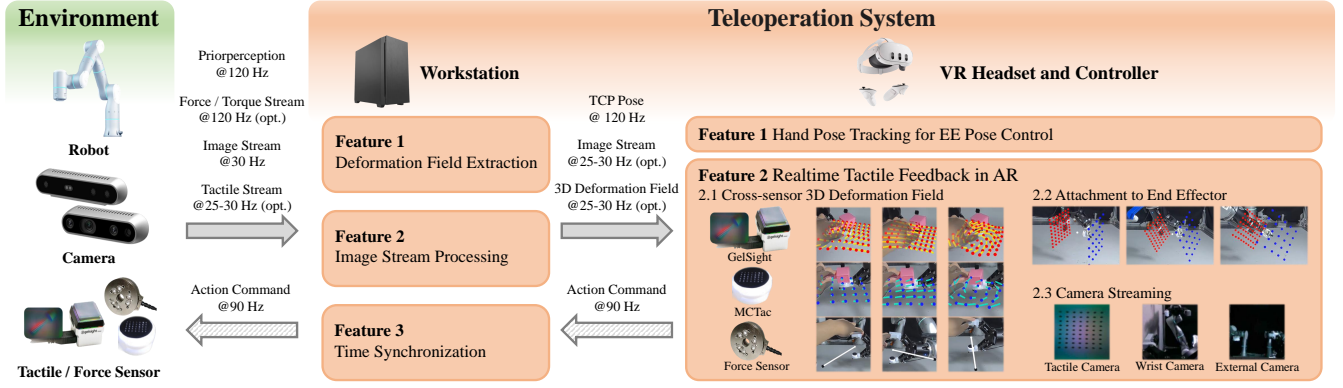


Fig. 2: Overview of **TactAR** teleoperation system. It can provide real-time tactile / force feedback via Augmented Reality (AR). The tactile feedback is represented as the 3D deformation field, which is a universal representation applicable to multiple different tactile / force sensors. The 3D deformation 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.

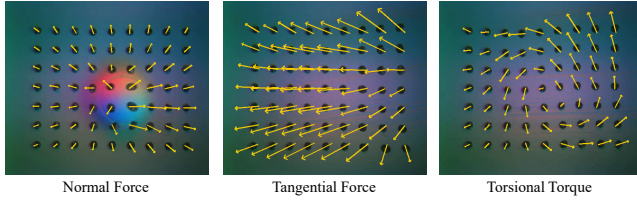


Fig. 3: Examples of marker deformation field in GelSight Mini [8] during different contact modes.

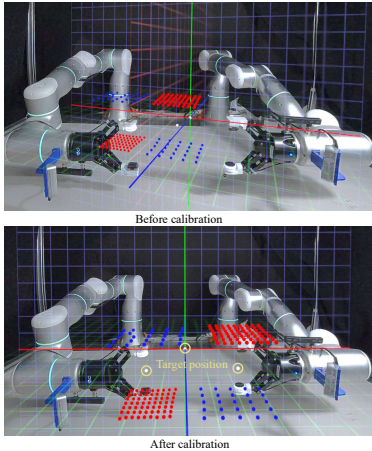


Fig. 4: Calibration process in AR. The user adjust the translation and rotation of the virtual coordinate system such that it can align with the pre-defined TCP position (the white sphere) and the origin of the world coordinate system.

- **MCTac** [16] optical tactile sensor with 2MP resolution at 30 FPS, and it has a 5×7 marker dot array on the surface. To acquire a stable and sensitive tactile signal, we have improved the camera-based tactile sensor based on the open source MCTac [13, 16], details are shown in Fig. 7.
- Built-in **joint torque sensors** in Flexiv Rizon 4 [7] robotic arm. We use the estimated TCP force/torque calculated by Flexiv RDK [6] 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[1]) due to inaccurate dynamics model, which further challenges the learning algorithm.

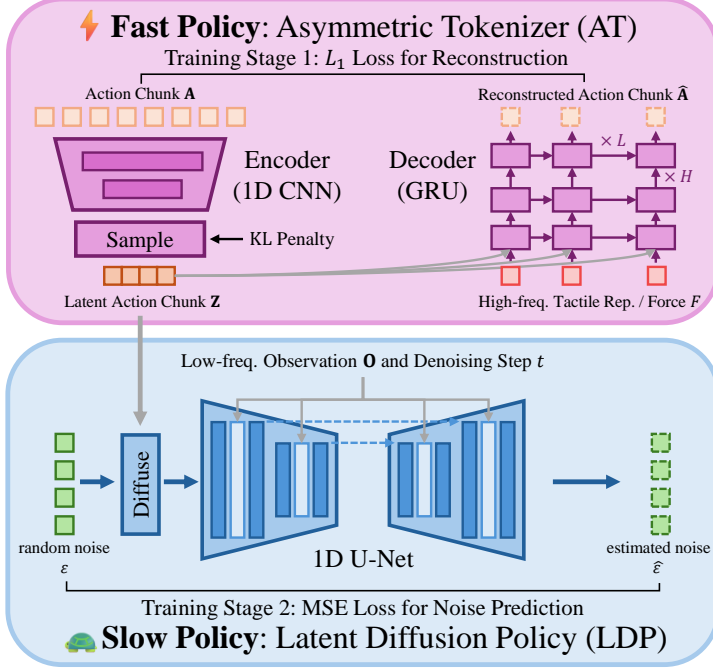
In order to evaluate policy performance under different tactile / force sensors, we attach MCTac and GelSight Mini to different fingertips of the same gripper. In this way, we can collect synchronized data from MCTac, GelSight Mini and force/torque sensors simultaneously. The TactAR teleoperation uses 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 [2] 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.

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

(a) Training Pipeline of Reactive Diffusion Policy



(b) Inference Pipeline of Slow-Fast Policy

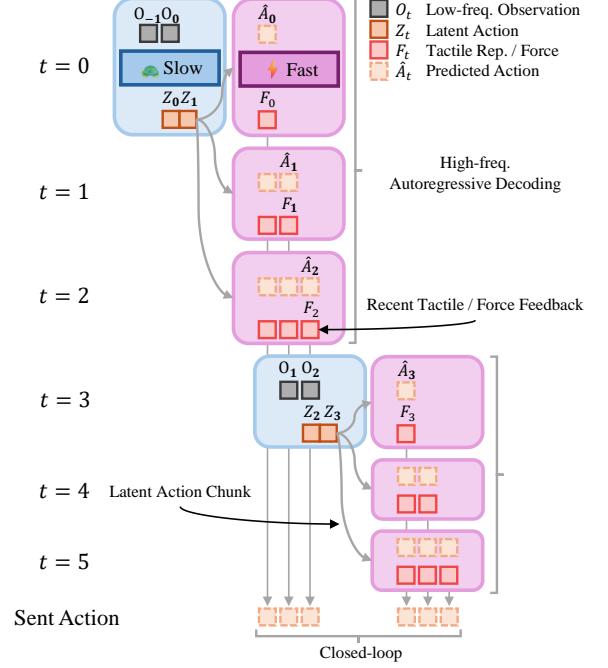


Fig. 5: Overview of **Reactive Diffusion Policy (RDP)** framework. (a) The training pipeline of RDP, comprising the first stage for training the fast policy (Asymmetric Tokenizer) and the second stage for training the slow policy (Latent Diffusion Policy). (b) The inference pipeline of RDP. The slow 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.

test-time variations for *Bimanual Lifting* task: (a) *soft paper cup*. (b) *hard paper cup*. We run 10 trials for each test-time variation.

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 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.

3) *Details of the Data Collection Process*: 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 proactively recorded some reactive behaviors to enhance the robustness of the model.

4) *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 12 FPS action sequences for

each action chunk. They will periodically (1-2Hz) predict new inference results at time intervals determined by the action chunk (about 0.67s in real-world time). The fast policy (AT) takes tactile / force observations at 24 FPS and outputs 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 [8], which is 25 FPS. Our RDP algorithm can also be applied to higher frequency tactile / force signals in theory.

B. Results

Slow-fast hierarchy, relative trajectory and latency matching are essential for RDP performance. As shown in Fig. 6, 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. 9 (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 [12, 19] can perform semi-closed-loop control by averaging predictions from multiple timesteps. We have also tried different temporal ensemble factors τ in

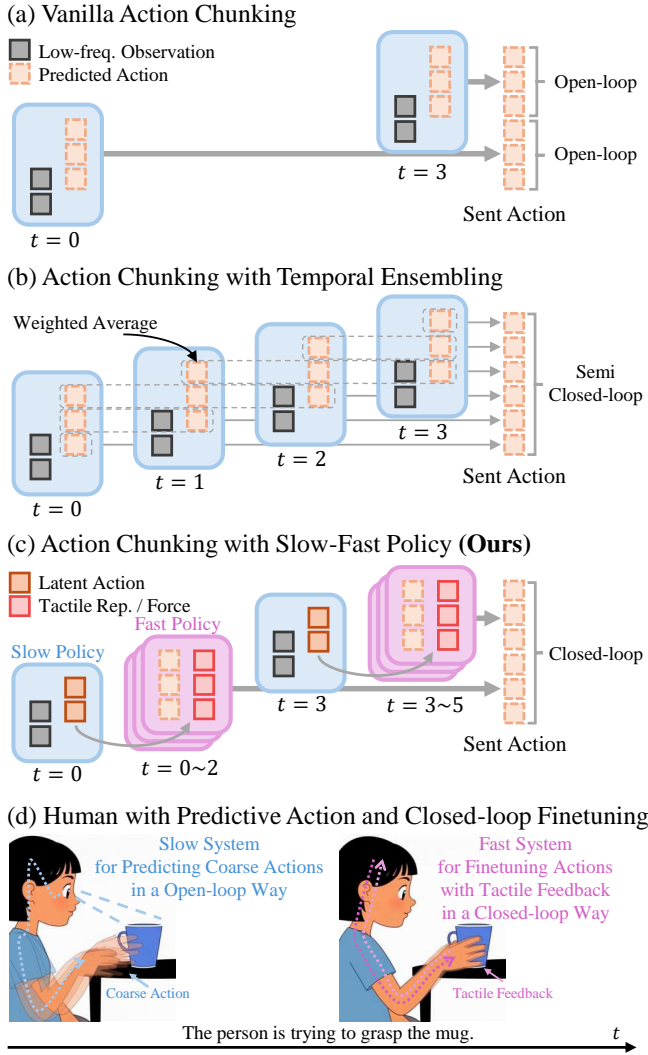


Fig. 6: Comparison among various pipelines. (a) Vanilla action chunking [2] with open-loop control during the chunk execution. (b) Action chunking enhanced with temporal ensembling [12, 19] for semi-closed-loop control. (c) Our slow-fast inference pipeline, showcasing closed-loop capabilities with fast responsive adjustments. (d) Human control patterns in contact-rich tasks.

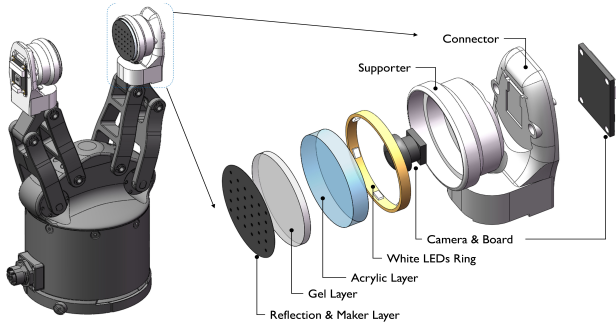


Fig. 7: Improved MCTac Sensor for our task. The left part is the gripper integrated illustration, and the right part is the detailed structure and components of the camera-based tactile sensor.

HATO [12], 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. 9 (b)). Thus, it is very hard to balance temporal consistency and reactivity with temporal ensemble.

As shown in Fig. 11, 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. 11) by ensuring smooth action transition between action chunks and reducing out-of-distribution (OOD) behaviors.

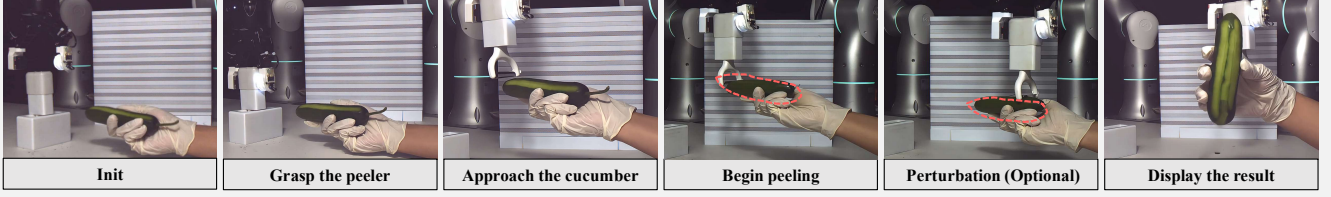
TABLE V: Effects of Chunk Size and Temporal Ensemble

	action chunk size	temporal ensemble [12] 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
	8	$\tau=0.8$	100%	0.15
RDP (GelSight)	8	-	100%	0.50

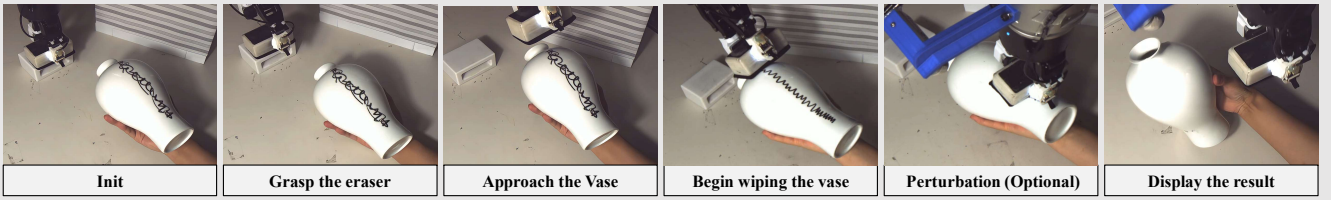
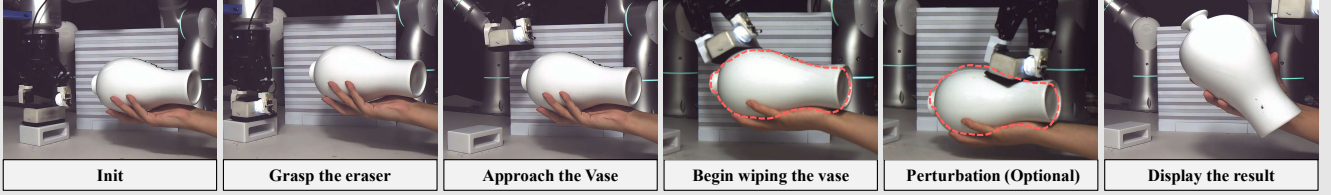
Tactile / force feedback in TactAR improves data quality in contact-rich tasks by improving the stability of contact forces. For a more detailed quantitative analysis of contact forces, we collect data (10 demos) for *Peeling* task with the same user by VR teleoperation and TactAR respectively, then 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. 14. We can observe that using TactAR to collect data helps avoid a large rolling standard deviation, indicating reduced temporal fluctuations and more stable contact forces.

Data with higher stability of contact forces will help the model to discover useful patterns more easily. We have collected the same number of demonstrations (60) for *Peeling* task with traditional VR teleoperation without tactile / force feedback and trained RDP (force) with these data. The results in Fig. 15 show that **data quality has a large influence on policy performance** (the score decreases by more than 30%). We observe that policies trained with low-quality data exhibited more unstable performance, such as more unstable force during peeling and a higher likelihood of breaking halfway through. A possible explanation is that the Fast Policy in RDP is designed to identify associations between tactile / force signals and trajectories from the data and learn reactive behavior. When contact forces in the data

Task1: Peeling



Task2: Wiping



Task3: Bimanual Lifting

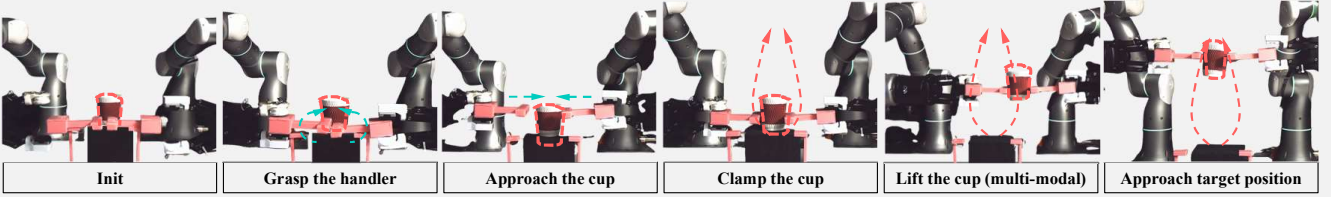


Fig. 8: Three experiment tasks including *Peeling*, *Wiping* and *Bimanual Lifting*.

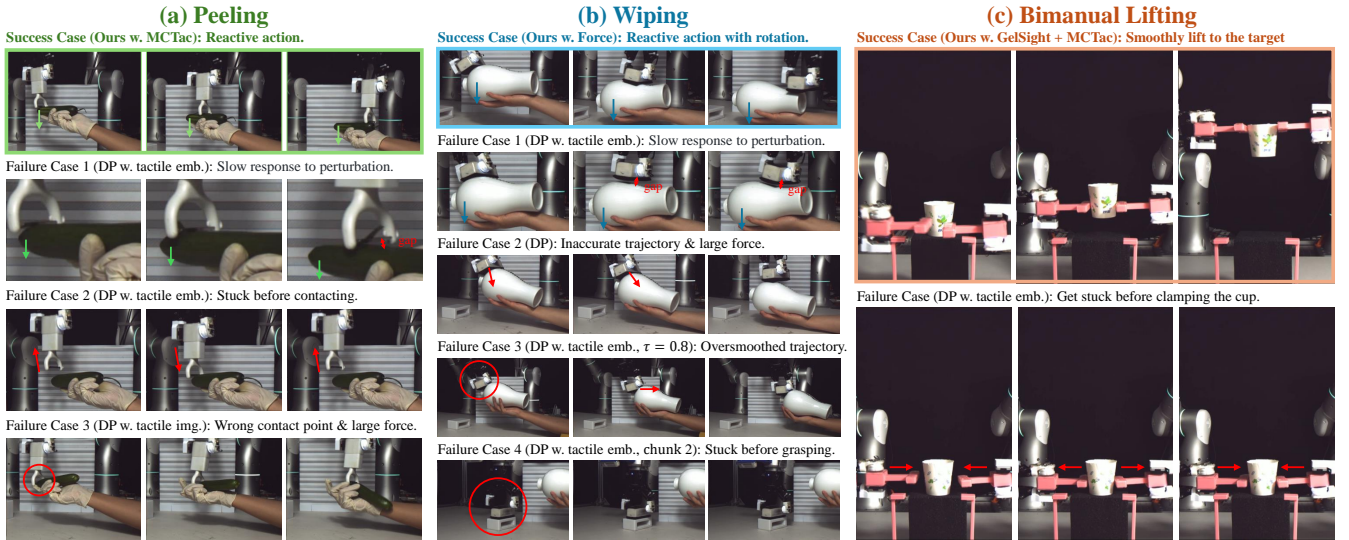


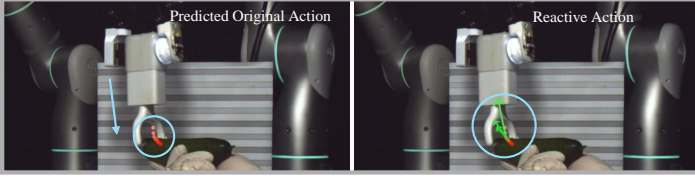
Fig. 9: Evaluation results and failure cases of baselines. Please see the [website](#) for more details.

are highly unstable, the Fast Policy struggles to identify reasonable associations, which reduces performance.

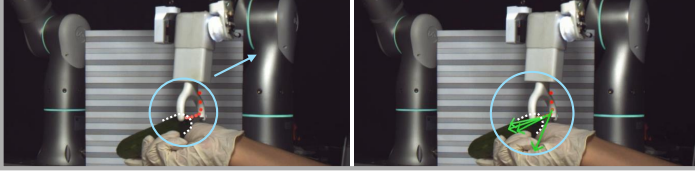
C. Case Study and Explainability

We conduct several case studies and visualize the RDP inference process. Please refer to Fig. 9 and Fig. 10.

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.

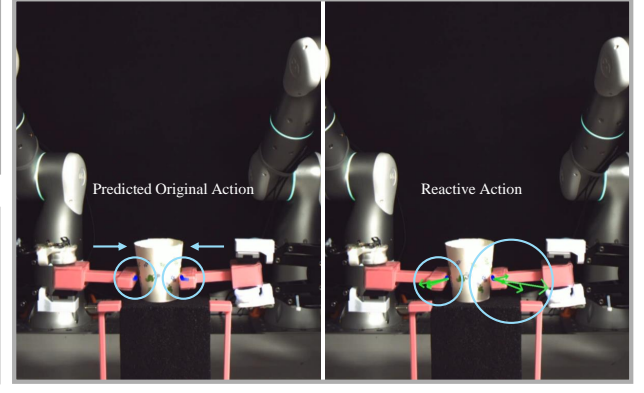


Fig. 10: Visualization of the RDP inference process. The red (left) and blue (right) dots can be seen as the predicted action chunk of the slow policy. The green arrow represents the correction direction and magnitude (scaled up for better visibility) of the reactive action predicted by the fast policy during inference. Please see the videos on the [website](#) for more details.

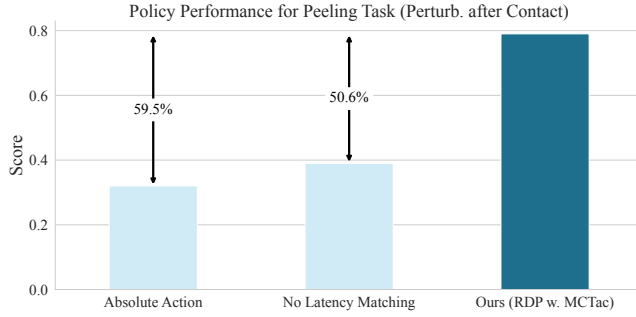


Fig. 11: Ablation Study.

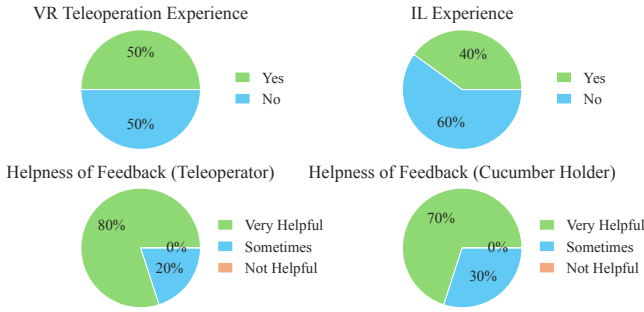


Fig. 12: User study results among 10 users on teleoperation w./w.o. tactile / force feedback in *Peeling* task.

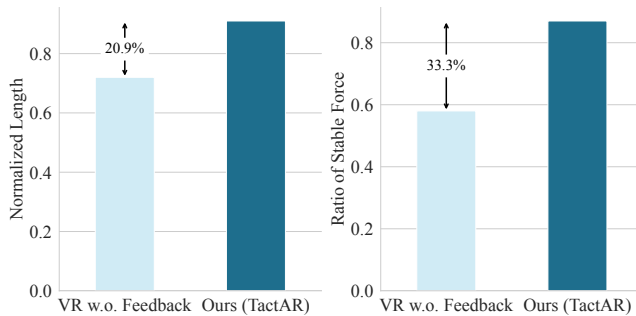


Fig. 13: Teleoperation data quality of 10 users on *Peeling* task (no perturb.) w./w.o. tactile / force feedback.

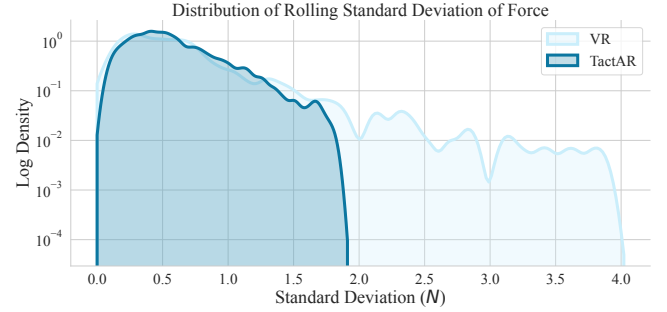


Fig. 14: The stability of contact forces with different teleoperation systems in *Peeling* task. Data collected with TactAR has higher stability of contact forces compared to traditional VR teleoperation.

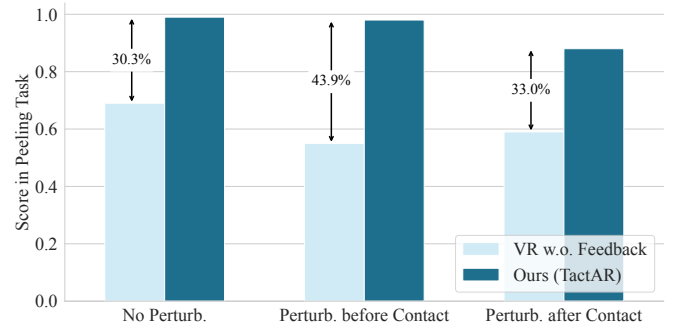


Fig. 15: How data quality influences policy performance of RDP (force) in *Peeling* task.