# FA-RDP: A Frequency-Adaptive Reactive Diffusion Policy for Contact-Rich Manipulation

Lifeng Zhuo[*1] Wendi Chen[*1,2] Han Xue[1] Shirun Tang[3] Jun Lv[3]
Cewu Lu[1,2,3†] Chuan Wen[1†]

***Abstract*— In contact-rich manipulation, action multimodality and reactivity dominate different stages of a single episode. Before contact, multiple trajectories might be equally valid, making it important to preserve diverse action modes. After contact, geometric constraints and force limits narrow the solution space, while successful execution demands rapid responses to force feedback. However, standard diffusion policies use a fixed inference frequency and sampling steps throughout the episode, forcing a fundamental compromise: low-frequency, multi-step sampling better preserves pre-contact multimodality but responds slowly to force feedback, whereas high-frequency sampling improves reactivity but tends to collapse distinct pre-contact modes. To resolve this tradeoff, we present FA-RDP, a frequency-adaptive reactive diffusion policy. A shared multi-frequency visual-force Transformer predicts action chunks at both low and high frequencies, while a learned multimodality indicator dynamically selects multi-step low-frequency sampling before contact and one-step high-frequency sampling as action ambiguity decreases. We further introduce Manifold Consistency Distillation (MCD), which reparameterizes the diffusion network to predict actions on the robot action manifold while retaining DDPM-based residual supervision. Experiments on three contact-rich manipulation tasks show that FA-RDP achieves the highest success rate while preserving diverse pre-contact trajectory modes. Code and demos are available at fa-rdp.github.io.**

## I. Introduction

Contact-rich manipulation requires a robot policy to solve two phase-dependent control problems: **diversity** before contact and **reactivity** after contact [1]. As shown in Fig. 1, pre-contact visual observations can admit several valid approach or contact modes, requiring the policy to preserve diverse pre-contact trajectory modes. By contrast, once contact is established, the robot must maintain contact and regulate the applied force under physical constraints, requiring rapid action updates using measured wrench feedback.

Robot policies such as Diffusion Policy [2] predict temporally coherent action sequences, but control within each chunk still remains open-loop. After contact, the policy cannot use newly observed force signals to update actions, so small pose errors can quickly cause force spikes, slip, or contact loss. Closed-loop execution shortens this feedback delay by conditioning each executed action on immediate force observations, maintaining contact before errors accumulate. Previous works like RDP [3] realize closed-loop execution using a hierarchical slow-fast design, but information can be lost between its slow and fast policies because the fast policy relies only on compressed latent actions. ImplicitRDP [4] removes this explicit hierarchy by learning visual and force modalities in an end-to-end diffusion network. However, the multi-step diffusion sampling limits the closed-loop inference frequency. These limitations motivate our central question: Can an end-to-end visual-force diffusion policy achieve high-frequency closed-loop response while preserving pre-contact multimodality?

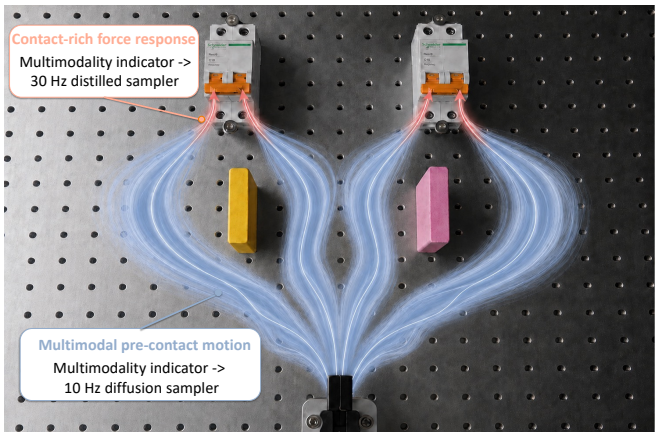


Fig. 1. Multimodality-guided frequency adaptation across manipulation phases. Before contact, low indicator value reflects multiple possible approach trajectories and selects the low-frequency diffusion sampler. After contact, the indicator value rises as physical constraints reduce action ambiguity, selecting the high-frequency distilled sampler for rapid force-feedback response.

To raise the closed-loop frequency, the most direct approach is to reduce the number of denoising steps through distillation [5, 6], which improves reactivity. But applying a one-step sampler throughout the episode sacrifices multimodality: it can collapse the multimodal action distribution needed before contact and degrade action smoothness, which may break contact or violate force constraints. The policy must therefore preserve multimodal sampling before contact while producing smooth actions at a high control rate after contact. This calls for adapting the inference frequency to the manipulation phase, rather than fixing it throughout the episode.

We propose FA-RDP, a frequency-adaptive reactive visual-force diffusion policy for contact-rich manipulation, built from three components. First, a single visual-force diffusion backbone is shared across frequencies through frequency-aware positional encoding, so the same network predicts both low-frequency and high-frequency action chunks without training separate models. Second, a learned multimodality

[1]Shanghai Jiao Tong University [2]Shanghai Innovation Institute
[3]Noematrix Ltd. [*]Equal contribution [†]Corresponding authors.

indicator, computed from the visual input, estimates how multimodal the current pre-contact behavior is, and selects the low-frequency multi-step sampler while this ambiguity is high and the high-frequency distilled sampler once contact constraints reduce it. Third, because the selected high-frequency sampler would still require multiple denoising steps, we distill it with manifold consistency distillation for one-step inference.

Experiment results show that FA-RDP achieves the highest average success rate of $81.7\%$, 30.0 percentage points higher than baseline. The component analyses further support the phase structure in Fig. 1: FA-RDP preserves diverse pre-contact modes, while a higher indicator value in lower-multimodality contact-rich phases activates the distilled high-frequency sampler.

The main contributions of this work are:

- We propose FA-RDP, a frequency-adaptive reactive visual-force diffusion policy, which is guided by a multimodality indicator and preserves pre-contact diversity while enabling post-contact reactivity.
- We introduce manifold consistency distillation, which reparameterizes the diffusion network to predict action chunks on the robot action manifold, rather than noise-like epsilon, score, or velocity targets, while retaining DDPM-based residual supervision for stable one-step high-frequency inference.
- We evaluate FA-RDP on three contact-rich manipulation tasks, showing improved task success, preserved pre-contact multimodality, and the benefit of higher-rate closed-loop control in contact-rich tasks.

## II. Related Work

### A. Robot Learning with Force Input

Vision-only policies such as Diffusion Policy [2], ACT [7], and 3D Diffusion Policy [8] do not use immediate force feedback, which limits their reactivity during contact.

To address this limitation, robot learning methods increasingly incorporate force or tactile input for contact-rich manipulation. DexForce [9] and 3D-ViTac [10] condition action prediction on force or contact signals; other works [11, 12, 13, 14] learn force-aware representations; and recent works [15, 16] scale tactile and video-tactile data toward generalist policies. Collectively, these methods show that contact signals carry information unavailable from vision, but action execution within each predicted chunk still remains open-loop, limiting real-time reaction to force feedback.

Previous works implement high-frequency force reaction in different ways: Compliant Residual DAgger [17] and Force Policy [18] rely on a separate low-level force controller, RDP [3] uses a hierarchical slow-fast design but relies on a compressed latent interface, and ImplicitRDP [4] learns visual and force modalities end-to-end but still requires multi-step denoising. FA-RDP instead keeps an end-to-end visual-force policy and makes the inference frequency adaptive.

### B. Multi-Frequency Policy Learning

Inference frequency is a key design choice for robot policies. Low-frequency inference supports longer-horizon and multimodal behavior, whereas high-frequency inference improves force reactivity but increases sampling cost and training difficulty. Prior work studies this tradeoff through different mechanisms: HiPolicy [19] uses hierarchical multi-frequency action chunking, high-frequency latent chunk learning [20] moves dense actions into a latent space, DVAC [21] adapts replanning from denoising variance, and AHA-WAM [22] separates low-frequency world planning from high-frequency action execution. ManipForce [23] uses frequency-aware representations to fuse asynchronous visual and force observations.

Existing multi-frequency policies are primarily motivated by hierarchical control, latent action learning, adaptive replanning, or asynchronous sensing. FA-RDP is instead motivated by the phase-dependent requirements of contact-rich manipulation: multimodal pre-contact behavior and rapid force-feedback response after contact. Accordingly, it uses frequency-aware positional encoding in a shared visual-force diffusion backbone to predict low-frequency and high-frequency action chunks.

### C. Robot Policy Distillation

Predicting high-frequency action chunks alone is not sufficient: closed-loop execution also requires the sampler to be fast enough for real-time control. DDIM [24] provides deterministic denoising trajectories for diffusion models, and progressive distillation [5] reduces the number of sampling steps by training a student sampler to match longer teacher trajectories. Consistency Models [25] learn mappings that support one-step or few-step generation. In robotics, ManiCM [26], Consistency Policy [6], and Hybrid Consistency Policy [27] adapt consistency-based acceleration to visuomotor policies. Flow-based policies pursue the same low-latency goal from another parameterization: MeanFlow [28] learns average velocity fields for one-step generation, Mean Flow Policy [29] applies mean-velocity objectives to robot manipulation, and FreqPolicy [30] improves flow-based visuomotor policies with frequency consistency.

These methods demonstrate effective one-step or few-step inference, but epsilon, score, or velocity targets are difficult to distill for robot action prediction. Inspired by JiT [31] and Pixel Mean Flows [32], FA-RDP instead predicts action chunks on the robot action manifold.

## III. Preliminary

We adopt the consistent reactive diffusion inference mechanism from prior end-to-end visual-force diffusion policies [4]. This mechanism combines slow visual-proprioceptive tokens and fast force tokens in one Transformer. Its structural slow-fast learning separates observations into a slow part for visual context and a fast part for dense force feedback. During training, a causal action-force mask ensures that each action token can attend only to current and past force tokens,

preventing future contact leakage while keeping parallel diffusion training.

At inference, the consistent mechanism realizes closed-loop force control within an action chunk. DDIM with $\eta = 0$ makes the denoising trajectory deterministic for a fixed initial noise $\mathbf{A}_K$. The policy therefore caches the slow context and $\mathbf{A}_K$ at the beginning of a chunk, then refreshes fast force tokens at each control step, runs multi-step denoising with the cached context and updated force tokens, and executes the newest predicted action. However, running multi-step denoising at each step limits the closed-loop force-feedback frequency, restricting rapid correction during continuous contact-rich interaction.

## IV. Method

FA-RDP addresses the low force-feedback frequency of prior end-to-end policies with a multi-frequency visual-force diffusion backbone. It preserves multimodal pre-contact behavior while enabling high-frequency control through three components: a multi-frequency visual-force Transformer that supports multi-frequency action prediction (Sec. IV-A), multimodality-based frequency selection that chooses the appropriate sampling mode (Sec. IV-B), and manifold consistency distillation (MCD) that stabilizes one-step high-frequency action prediction (Sec. IV-C). Implementation details are provided in Sec. IV-D.

### A. Multi-Frequency Visual-Force Transformer

FA-RDP realizes frequency-adaptive visual-force control through a shared multi-frequency visual-force Transformer. The same backbone is used for both frequency modes. A single forward pass is conditioned on one frequency mode $\nu \in \{\ell, h\}$ and predicts the action chunk for that mode. This shared backbone is trained in Stage 1 with joint low-frequency and high-frequency diffusion objectives.

The key design is frequency-adaptive positional encoding, as shown in Fig. 2. The frequency mode is not embedded as an extra token; instead, it selects temporal indices on a shared temporal grid. The high-frequency mode uses consecutive positions, whereas the low-frequency mode uses sparse positions, so both modes are expressed on the same underlying clock. This design lets the backbone share weights across frequencies while preserving each mode's temporal layout. We use causal action-force mask to avoid future-contact leakage.

For each mode $\nu$, let $\mathbf{A}^\nu$ denote the normalized demonstration action sequence for sample $n$, conditioned on slow observation $\mathbf{o}_n^s$ and force observation $\mathbf{o}_n^{\text{force}}$. The forward diffusion process samples a timestep $k$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, then constructs

$$\mathbf{A}_k^\nu = \sqrt{\bar{\alpha}_k}\mathbf{A}^\nu + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}. \tag{1}$$

Diffusion timesteps run from 0 to $K$, where $K$ is the initial-noise timestep; we omit the subscript 0 for clean demonstration and predicted action chunks. Throughout, $\mathbf{A}$ denotes a given or constructed action state, whereas $\hat{\mathbf{A}}$ denotes an action state predicted by a network. We parameterize the network to predict the action chunk $\hat{\mathbf{A}}^\nu = \pi_\theta(\mathbf{A}_k^\nu, k, \mathbf{o}_n^s, \mathbf{o}_n^{\text{force}}, \nu)$. The implied noise prediction is

$$\hat{\boldsymbol{\epsilon}} = \frac{\mathbf{A}_k^\nu - \sqrt{\bar{\alpha}_k}\hat{\mathbf{A}}^\nu}{\sqrt{1-\bar{\alpha}_k}}, \tag{2}$$

and we optimize the per-mode diffusion objective

$$\mathcal{L}_\epsilon^\nu = \mathbb{E}_{\mathbf{A}^\nu, k, \boldsymbol{\epsilon}}\left[\|\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}\|_2^2\right]. \tag{3}$$

The shared backbone is trained with a joint multi-frequency diffusion objective that applies the same loss to the low-frequency and high-frequency targets,

$$\mathcal{L}_{\text{MF}} = \mathcal{L}_\epsilon^\ell + \mathcal{L}_\epsilon^h. \tag{4}$$

Thus, the network predicts action samples, while the training objective still uses epsilon supervision.

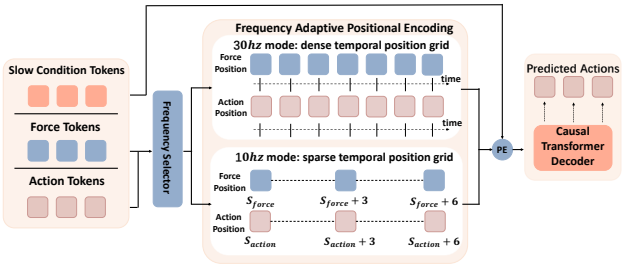


Fig. 2. Multi-frequency visual-force Transformer. The frequency mode selects frequency-adaptive positional indices on a shared temporal grid: dense positions for high-frequency action chunks and sparse positions for low-frequency action chunks. The causal decoder supports closed-loop updates by preventing action tokens from accessing future action or force tokens.

### B. Multimodality-Based Frequency Selection

The multi-frequency Transformer provides both low-frequency and high-frequency diffusion samplers, but the policy still needs to decide which sampler to use at different phases. We use a learned indicator that estimates local multimodality instead of relying on a phase label.

Stage 2 trains the multimodality indicator head on an independent calibration set. The Stage 1 visual-force policy is frozen, so this training does not change the multi-frequency backbone. For each calibration example, we fix the same visual-force condition and demonstration target, independently sample the low-frequency policy $N = 8$ times with different initial diffusion noise, and compute an empirical action residual:

$$e(\mathbf{o}_n) = \frac{1}{N}\sum_{m=1}^{N}\left\|\hat{\mathbf{A}}_m^\ell - \mathbf{A}^\ell\right\|_2^2. \tag{5}$$

Here, $\hat{\mathbf{A}}_m^\ell$ is the $m$-th action chunk sampled from the frozen Stage 1 low-frequency policy, and $\mathbf{A}^\ell$ is the corresponding demonstration. A Transformer indicator head maps slow observation tokens to a scalar logit $s(\mathbf{o}_n)$, with $\sigma(\mathbf{o}_n) = 1 + \exp(s(\mathbf{o}_n))$. Following the confidence-weighted regression form of VGGT [33], we train the indicator with

$$\mathcal{L}_{\text{MM}} = \mathbb{E}_{\mathbf{o}_n}\left[\gamma e(\mathbf{o}_n)\sigma(\mathbf{o}_n) - \alpha_{\text{MM}}\log\sigma(\mathbf{o}_n)\right], \tag{6}$$

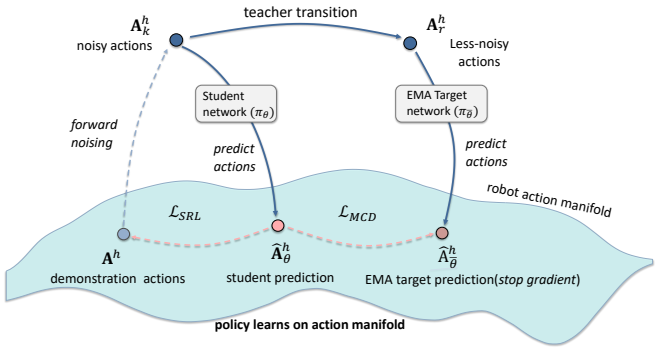


Fig. 3. Manifold consistency distillation for the high-frequency sampler. A teacher transition maps $\mathbf{A}_k^h$ to a less noisy state $\mathbf{A}_r^h$. Both networks predict action chunks on the robot action manifold. MCD aligns the student prediction with the stop-gradient target prediction, while SRL regularizes the student prediction to stay close to the demonstration action.

where $\gamma$ and $\alpha_{\text{MM}}$ control the error penalty and regularizer.

At inference time, FA-RDP computes the indicator value from the current slow tokens and selects the sampler with a fixed threshold. This makes execution frequency-adaptive, but the selected high-frequency diffusion sampler still requires multi-step denoising. We therefore distill the high-frequency sampler for fast force-feedback response.

### C. Manifold Consistency Distillation for One-Step Control

Manifold consistency distillation (MCD) turns the selected high-frequency sampler into a one-step sampler. Directly distilling a model that predicts epsilon, score, or velocity requires the student to fit high-frequency, noise-like targets, which makes stable action prediction difficult. Inspired by the sample-prediction view of JiT [31] and Pixel Mean Flows [32], we reparameterize the denoising model to predict action chunks on the robot action manifold. The fixed DDPM conversion in Eq. 2 retains residual supervision without making the network predict a noise-like target, reducing the student's learning difficulty. Fig. 3 illustrates the resulting distillation process.

The consistency objective follows the general goal of reducing diffusion NFEs via progressive distillation [5] and is related to Consistency Models [25] and Consistency Policy [6]. Under the shared condition $\mathbf{c}_n^h$, let $\hat{\mathbf{A}}_\theta^h \equiv \pi_\theta(\mathbf{A}_k^h, k, \mathbf{c}_n^h)$ and $\hat{\mathbf{A}}_{\bar{\theta}}^h \equiv \pi_{\bar{\theta}}(\mathbf{A}_r^h, r, \mathbf{c}_n^h)$ denote the two action predictions. In Stage 3, the student is trained with

$$\begin{aligned}\mathcal{L}_{\text{distill}} =& \lambda_{\text{MCD}} \left\| \hat{\mathbf{A}}_\theta^h - \text{sg}[\hat{\mathbf{A}}_{\bar{\theta}}^h] \right\|_2^2 \\ &+ \lambda_{\text{SRL}} \left\| \hat{\mathbf{A}}_\theta^h - \mathbf{A}^h \right\|_2^2 .\end{aligned} \tag{7}$$

Here, $k > r$ are adjacent grid timesteps for the noisy state $\mathbf{A}_k^h$ and less-noisy teacher state $\mathbf{A}_r^h$, conditioned on $\mathbf{c}_n^h = (\mathbf{o}_n^s, \mathbf{o}_n^{\text{force}}, h)$. $\theta$ and $\bar{\theta}$ are the student and EMA target parameters, $\text{sg}[\cdot]$ stops gradients through the target prediction, and $\mathbf{A}^h$ is the demonstrated high-frequency action chunk. $\lambda_{\text{MCD}}$ and $\lambda_{\text{SRL}}$ weight the MCD and sample regression loss (SRL) terms. After distillation, FA-RDP uses the multi-frequency checkpoint for the low-frequency sampler and the distilled checkpoint for the high-frequency sampler. The final inference procedure follows ImplicitRDP's consistent inference mechanism [4] and is summarized in Alg. 1: cache slow context and initial noise at chunk start, select the frequency using the indicator, refresh wrench tokens at each step, run multi-step DDIM for the low-frequency mode or direct one-step action prediction for the high-frequency mode, and execute the newest valid action.

**Algorithm 1** FA-RDP Consistent Frequency-Adaptive Inference

**Require:** Policy networks $\pi_{\theta_\ell}$ and $\pi_{\theta_h}$, slow observation encoder $\mathcal{E}_{\text{slow}}$, fast observation encoders $\mathcal{E}_{\text{fast}}^\ell$ and $\mathcal{E}_{\text{fast}}^h$, slow observation horizon $h_o$, execution horizons $h_e^\ell$ and $h_e^h$, latency steps $l^\ell$ and $l^h$, multimodality indicator $\sigma$, threshold $\tau$

Initialize step $t \leftarrow 0$
// slow loop: action chunk modeling
**while** Task not done **do**
  // cache slow context and noise and select frequency
  $\mathbf{O}_{\text{slow}} \leftarrow \text{GetSlowObservation}(\text{len} = h_o)$
  $\mathbf{Z}_{\text{slow}} \leftarrow \mathcal{E}_{\text{slow}}(\mathbf{O}_{\text{slow}})$
  $q \leftarrow \sigma(\mathbf{Z}_{\text{slow}})$
  **if** $q > \tau$ **then**
    $\nu \leftarrow h$, $h_e \leftarrow h_e^h$, $l \leftarrow l^h$
    $\pi \leftarrow \pi_{\theta_h}$, $\mathcal{E}_{\text{fast}} \leftarrow \mathcal{E}_{\text{fast}}^h$
  **else**
    $\nu \leftarrow \ell$, $h_e \leftarrow h_e^\ell$, $l \leftarrow l^\ell$
    $\pi \leftarrow \pi_{\theta_\ell}$, $\mathcal{E}_{\text{fast}} \leftarrow \mathcal{E}_{\text{fast}}^\ell$
  **end if**
  $\mathbf{A}_K^\nu \leftarrow \mathcal{N}(0, \mathbf{I})$
  // fast loop: closed-loop control within the horizon
  **for** $i \leftarrow 0$ to $h_e - 1$ **do**
    // update fast context
    $\mathbf{O}_{\text{fast}} \leftarrow \text{GetFastObservation}(\nu, \text{len} = h_o + i + l)$
    $\mathbf{Z}_{\text{fast}} \leftarrow \mathcal{E}_{\text{fast}}(\mathbf{O}_{\text{fast}})$
    // get a noisy action sequence of a certain length
    $\mathbf{A}_{\text{pre}}^\nu \leftarrow \text{Prefix}(\mathbf{A}_K^\nu, i + h_o + l)$
    // consistent denoising with cache
    **if** $\nu = h$ **then**
      $\hat{\mathbf{A}}^\nu \leftarrow \pi(\mathbf{A}_{\text{pre}}^\nu, K, \mathbf{Z}_{\text{slow}}, \mathbf{Z}_{\text{fast}}, \nu = h)$
    **else**
      $\hat{\mathbf{A}}^\nu \leftarrow \text{DDIM}(\pi, \mathbf{Z}_{\text{slow}}, \mathbf{Z}_{\text{fast}}, \mathbf{A}_{\text{pre}}^\nu, \nu = \ell, \eta = 0)$
    **end if**
    // execute the current step action
    $\mathbf{a}_t \leftarrow \hat{\mathbf{A}}^\nu[-1]$
    Execute $\mathbf{a}_t$
    $t \leftarrow t + 1$
    **if** Task done **then**
      **break**
    **end if**
  **end for**
**end while**

### D. Implementation Details

**100 Hz force compensation.** Our demonstrations record robot states rather than commanded targets. During deployment, impedance control can therefore create a tracking gap between a policy-predicted state and the robot's realized motion. To compensate for this execution error, all methods share a 100 Hz command layer that linearly interpolates policy outputs and adjusts each command using the latest measured world-frame external force:

$$\mathbf{p}_{\text{cmd}} = \mathbf{p}_{\text{policy}} - \lambda \mathbf{f}_{\text{ext}}, \quad \lambda = 10^{-4}. \tag{8}$$

This translation-only compensation is applied before sending targets through the Flexiv Cartesian motion-force interface;

unlike ImplicitRDP, which additionally finetunes the low-level controller for precise position tracking, all methods here share the same default control layer; this controller-level difference causes the performance gap.

**Algorithm settings.** Frequency-adaptive execution requires the low-frequency and high-frequency modes to span the same prediction horizon at different temporal resolutions. We therefore use a low-frequency mode at 10 Hz with $H_\ell = 16$ and a high-frequency mode at 30 Hz with $H_h = 48$, so both modes cover the same 1.6 s horizon with different action densities. Camera observations are recorded at 10 Hz, and the force/torque stream used by the policy is recorded at 30 Hz. During execution, the slow loop refreshes every 1 s, corresponding to $h_e^\ell = 10$ for the low-frequency mode and $h_e^h = 30$ for the high-frequency mode. The diffusion process uses 100 training timesteps, indexed from 0 to 99. To make the high-frequency mode fast enough for closed-loop use, we distill it on the six-timestep grid $\mathcal{G} = \{99, 79, 59, 39, 19, 0\}$.

**Hardware settings.** We run policy inference on an Intel Core Ultra 9 285K CPU and NVIDIA GeForce RTX 5090 GPU. The policy inference latency is below 30 ms for high-frequency execution and below 50 ms for low-frequency execution.

## V. Experiments

We evaluate FA-RDP on three real-world contact-rich manipulation tasks to answer four key questions:

- **Q1:** How does FA-RDP compare against visual-only diffusion policy, hierarchical visual-force policy, fixed-frequency end-to-end visual-force policy, and a non-diffusion regression baseline?
- **Q2:** Does FA-RDP preserve multimodal ability?
- **Q3:** Does the multimodality indicator improve performance by switching between frequency modes?
- **Q4:** Does our distillation perform better than other distillation methods?

### A. Experimental Setup

Our hardware setup uses a Flexiv Rizon 4R leader arm for teleoperation and a Rizon 4s follower arm for task execution. We collect kinematic demonstrations through TDK-enabled teleoperation, record contact forces from the follower's end-effector F/T sensor, and capture visual observations with a wrist-mounted iPhone fisheye camera [34] and a fixed third-view USB camera. Fig. 4 shows the hardware setup and task objects used in our experiments.

We design three contact-rich manipulation tasks to evaluate multimodality and reactivity jointly. In each task, two front blocks create multiple valid approaches, requiring the policy to choose a collision-free trajectory before contact and then apply reactive control during a force-sensitive flip, toggle, or press:

1) **Dual Box Flipping:** The robot chooses a valid side approach and pushes a box against the fixture while maintaining contact.
2) **Dual Switch Toggling:** The robot reaches the correct switch contact location, maintains directional contact along the switch motion, and applies sufficient force to complete the toggle.
3) **Dual Button Pressing:** The robot aligns with the button axis at a nearly fixed contact point and applies force along the button axis until the actuation threshold is reached, with little tangential motion.

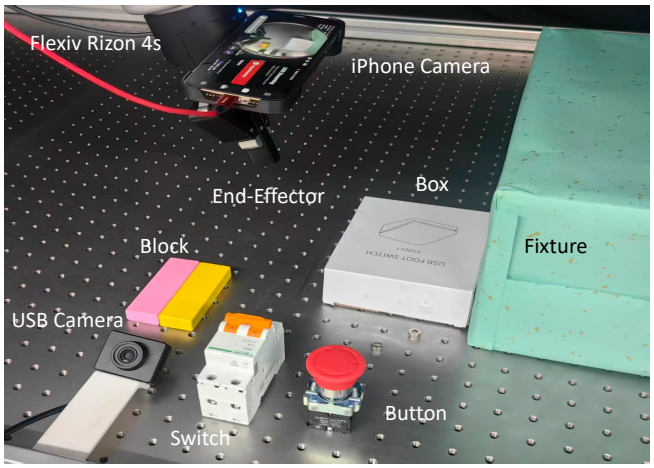


Fig. 4. Robot workspace and task objects. The Rizon 4s follower arm executes tasks using observations from a wrist-mounted iPhone camera and a fixed third-view USB camera, while the workspace contains the target box, switch, button, front blocks, and fixture.

For each task, we collect 60 demonstrations and evaluate 20 trials per method.

### B. Baselines

We compare FA-RDP against the following main baselines in Q1:

- **Diffusion Policy (DP):** vision-only CNN diffusion policy with open-loop action chunks.
- **Reactive Diffusion Policy (RDP):** hierarchical slow-fast visual-force policy.
- **ImplicitRDP:** fixed-frequency end-to-end visual-force diffusion policy.
- **Regression Policy w/ Force:** non-diffusion visual-force regression policy that predicts high-frequency action chunks using the same backbone as ImplicitRDP and an MSE objective.
- **FA-RDP (Ours):** multimodality-guided selection between the low-frequency diffusion sampler and high-frequency distilled sampler.

For Q2–Q4, we further evaluate sampler and distillation variants:

- **High-frequency alone policy:** high-frequency diffusion policy without MCD or multimodality-guided selection.
- **High-frequency distilled alone policy:** high-frequency manifold-distilled policy without multimodality-guided selection.
- **MeanFlow Policy:** one-step policy whose backbone predicts mean and instantaneous velocity fields and updates the noisy trajectory with the mean velocity to predict an action chunk [29].

- **Consistency Policy:** one-step policy that uses consistency distillation and EDM preconditioning to predict an action chunk from the raw network output [6].

### C. Results and Analysis

*1) Comparison with Baselines (Q1):* As illustrated in Table I, FA-RDP achieves the highest success rate on all three tasks compared with visual-only diffusion, hierarchical visual-force, fixed-frequency end-to-end visual-force, and direct visual-force regression baselines.

TABLE I

MAIN SUCCESS RATES. BOX, BUTTON, AND SWITCH DENOTE THE THREE CONTACT-RICH TASKS.

| Method | Box | Button | Switch | Avg. |
|---|---|---|---|---|
| DP | 0/20 | 2/20 | 4/20 | 10.0% |
| RDP | 5/20 | 7/20 | 9/20 | 35.0% |
| ImplicitRDP | 8/20 | 11/20 | 12/20 | 51.7% |
| Regression Policy w/ Force | 2/20 | 4/20 | 6/20 | 20.0% |
| **FA-RDP (Ours)** | **14/20** | **18/20** | **17/20** | **81.7%** |

The failure cases in Figs. 5–7 reveal where each baseline breaks down. DP and ImplicitRDP both lose contact in all three tasks, separating from the box during flipping, leaving the switch before the toggle completes, and slipping off the button before the press finishes. DP fails because its visual-only open-loop chunks cannot use new force feedback, while ImplicitRDP predicts visual-force actions end to end but updates too slowly to keep pace with the changing wrench after contact.

RDP reaches the task objects but contacts the wrong location, pushing the wrong box region, reaching an incorrect switch point, and pressing away from the button axis. These failures are consistent with spatial information loss through its compressed slow-policy interface during precise approach.

Regression Policy w/ Force averages the multiple valid pre-contact approaches under its MSE objective and cannot commit to a single collision-free mode, so it knocks down the front block during target approach.

In contrast, FA-RDP separates the two phases: it keeps the multimodal low-frequency sampler for a collision-free approach and switches to the distilled high-frequency sampler after contact to react quickly to force feedback. This preserves pre-contact diversity while maintaining contact through the flip, toggle, and press.

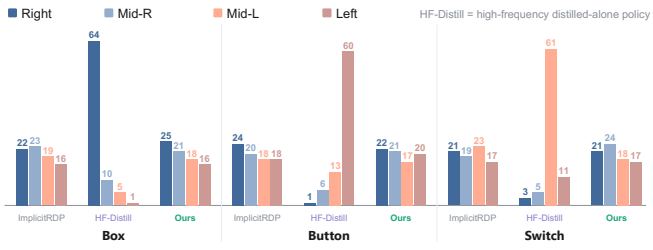


Fig. 8. Pre-contact mode distributions across the three tasks. The high-frequency distilled alone policy collapses to one dominant approach mode, whereas FA-RDP and ImplicitRDP preserve multimodal distribution over pre-contact approaches.

*2) Multimodal Preservation (Q2):* We count the observed pre-contact trajectories in the right, middle-right, middle-left, and left modes; the resulting distributions are visualized in Fig. 8. For the multimodality study, we run 80 trials per task; ImplicitRDP and FA-RDP cover all four modes, whereas the high-frequency distilled alone policy concentrates in one mode. This indicates that FA-RDP preserves the multimodal capability of multi-step diffusion before contact while enabling high-frequency force response after contact.

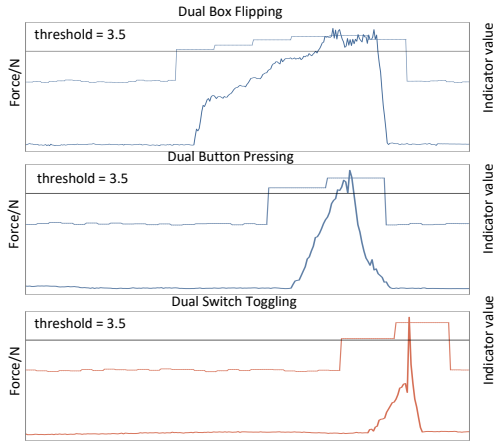


Fig. 9. Multimodality indicator value and force curves on the three contact-rich tasks. The indicator value remains lower before contact and during target approach, then rises with the force response in contact-rich phases, supporting the switch from the low-frequency sampler to the high-frequency distilled sampler.

*3) Indicator Accuracy (Q3):* Q3 evaluates whether the multimodality indicator improves performance by switching between the low-frequency sampler and the high-frequency distilled sampler. We compare FA-RDP with the high-frequency distilled alone policy in Table II. The latter always uses the distilled high-frequency policy, whereas FA-RDP uses the low-frequency sampler before contact and during target approach, then switches to the high-frequency distilled sampler when contact-rich feedback becomes informative.

TABLE II

INDICATOR-GUIDED SWITCHING COMPARISON. BOX, BUTTON, AND SWITCH DENOTE THE THREE TASKS.

| Method | Box | Button | Switch | Avg. |
|---|---|---|---|---|
| High-frequency distilled alone policy | 12/20 | 14/20 | 11/20 | 61.7% |
| **FA-RDP (Ours)** | **14/20** | **18/20** | **17/20** | **81.7%** |

The multimodality and force curves in Fig. 9 show the temporal behavior of the indicator. The indicator value stays low before contact, when multiple collision-free approaches remain valid, so FA-RDP keeps the low-frequency diffusion sampler active for smooth target approach before contact. The indicator rises with the force response after contact, when the robot must maintain contact and limit the applied force, so FA-RDP selects the high-frequency distilled sampler for reactive execution. Compared with the high-frequency distilled alone policy, this indicator-guided

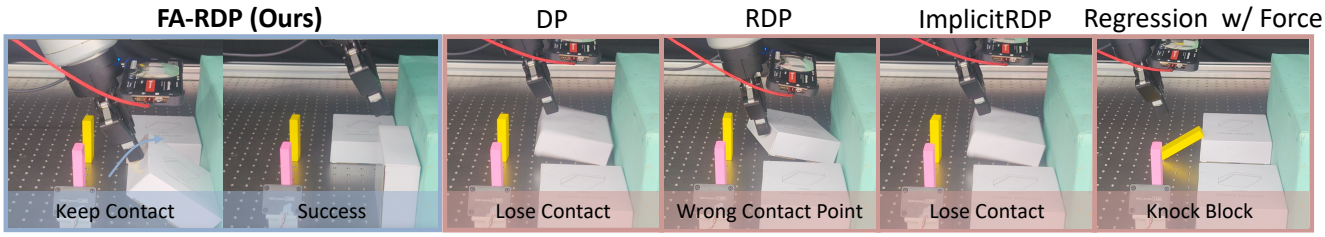


Fig. 5. Dual Box Flipping failure-case comparison. DP and ImplicitRDP can lose contact during the flip, RDP can contact the wrong box region, and Regression Policy w/ Force can knock down a front block before completing the task. FA-RDP keeps contact and completes the box flipping task.

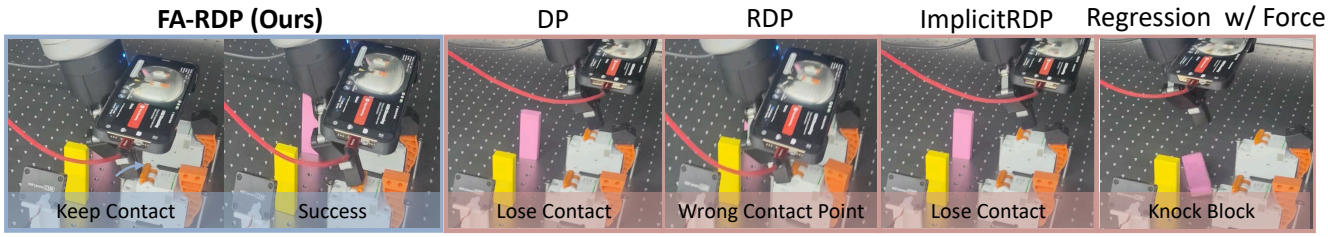


Fig. 6. Dual Switch Toggling failure-case comparison. DP and ImplicitRDP can lose contact during the toggle, RDP can contact the wrong switch region, and Regression Policy w/ Force can knock down a front block before completing the task. FA-RDP keeps contact and completes the switch toggling task.

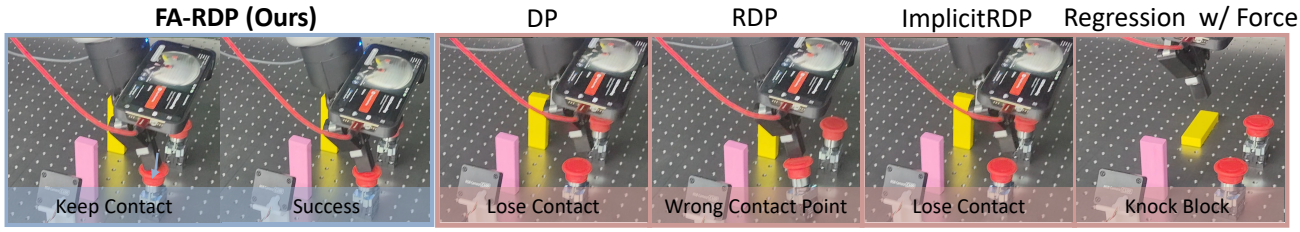


Fig. 7. Dual Button Pressing failure-case comparison. DP and ImplicitRDP can lose contact during the press, RDP can contact the wrong button region, and Regression Policy w/ Force can knock down a front block before completing the task. FA-RDP keeps contact and completes the button pressing task.

switching improves the average success rate from 61.7% to 81.7%.

*4) Distillation Method Comparison (Q4):* Q4 evaluates whether our distillation performs better than other distillation methods. All methods in Table III use the high-frequency mode, and multimodality-based selection is not used. The high-frequency alone policy is the original policy before distillation, and the other rows use the one-step objectives defined in Sec. V-B.

TABLE III

HIGH-FREQUENCY DISTILLATION COMPARISON. BOX, BUTTON, AND SWITCH DENOTE THE THREE TASKS.

| Method | Box | Button | Switch | Avg. |
|---|---|---|---|---|
| High-frequency alone policy | 4/20 | 3/20 | 5/20 | 20.0% |
| MeanFlow Policy | 0/20 | 0/20 | 1/20 | 1.7% |
| Consistency Policy | 0/20 | 0/20 | 1/20 | 1.7% |
| **High-frequency distilled alone policy** | **12/20** | **14/20** | **11/20** | **61.7%** |

Without distillation, running the original multi-step high-frequency policy with one-step sampling results in insufficient prediction accuracy, causing performance degradation. MeanFlow Policy and Consistency Policy each achieve only 1.7% average success, whereas our high-frequency distilled policy reaches 61.7%. These results indicate the effectiveness of manifold consistency distillation with SRL regularization for one-step high-frequency control.

## VI. CONCLUSION

We presented FA-RDP, a frequency-adaptive reactive diffusion policy that addresses the phase-dependent tradeoff between pre-contact diversity and post-contact reactivity. Its shared multi-frequency visual-force Transformer predicts multi-frequency action chunks, the multimodality indicator guides frequency selection, and manifold consistency distillation provides one-step high-frequency prediction on the robot action manifold. Across the three tasks, FA-RDP combines broad pre-contact mode coverage with the highest average success rate of 81.7%, demonstrating the effectiveness of our method. However, the current evaluation is limited to visual-force input without testing other sensing modalities, uses single-task policies rather than multi-task learning, and requires a three-stage training procedure.

## REFERENCES

[1] T. Tsuji, Y. Kato, G. Solak, H. Zhang, T. Petrič, F. Nori, and A. Ajoudani, "A survey on imitation learning for contact-rich tasks

in robotics," *The International Journal of Robotics Research*, p. 02783649261417694, 2025.
[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.
[3] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," *arXiv preprint arXiv:2503.02881*, 2025.
[4] W. Chen, H. Xue, Y. Wang, F. Zhou, J. Lv, Y. Jin, S. Tang, C. Wen, and C. Lu, "Implicitrdp: An end-to-end visual-force diffusion policy with structural slow-fast learning," *IEEE Robotics and Automation Letters*, 2026.
[5] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," *arXiv preprint arXiv:2202.00512*, 2022.
[6] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," *arXiv preprint arXiv:2405.07503*, 2024.
[7] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.
[8] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.
[9] C. Chen, Z. Yu, H. Choi, M. Cutkosky, and J. Bohg, "Dexforce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation," *IEEE Robotics and Automation Letters*, 2025.
[10] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li, "3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing," *arXiv preprint arXiv:2410.24091*, 2024.
[11] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak, "Factr: Force-attending curriculum training for contact-rich policy learning," *arXiv preprint arXiv:2502.17432*, 2025.
[12] S. Oh, J. J. Liu, T. Tao, P. Han, K. Shaw, S. Funabashi, R. Salakhutdinov, and D. Pathak, "Factr 2: Learning external force sensing for commodity robot arms improves policy learning," *arXiv preprint arXiv:2606.12406*, 2026.
[13] J. Yu, H. Liu, Q. Yu, J. Ren, C. Hao, H. Ding, G. Huang, G. Huang, Y. Song, P. Cai, *et al.*, "Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation," *Advances in Neural Information Processing Systems*, vol. 38, pp. 93 409–93 439, 2026.
[14] I. Guzey, B. Evans, S. Chintala, and L. Pinto, "Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play," *arXiv preprint arXiv:2303.12076*, 2023.
[15] H. Yuan, W. Yi, Z. Zhang, W. Chen, Y. Mo, J. Yin, X. Li, X. Zeng, C. Wen, C. Lu, *et al.*, "Vtam: Video-tactile-action models for complex physical interaction beyond vlas," *arXiv preprint arXiv:2603.23481*, 2026.
[16] C. Yuan, Z. Zhang, M. Zhou, W. Chen, Y. Wang, Z. Liu, D. Niu, S. Wang, H. Zhang, W. Zhang, *et al.*, "Ftp-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation," *arXiv preprint arXiv:2606.13102*, 2026.
[17] X. Xu, Y. Hou, Z. Liu, and S. Song, "Compliant residual dagger: Improving real-world contact-rich manipulation with human corrections," *Advances in Neural Information Processing Systems*, vol. 38, pp. 139 559–139 581, 2026.
[18] H. Fang, S. Tang, M. Mei, H. Qin, Z. He, J. Chen, Y. Feng, C. Wang, W. Liu, Z. He, *et al.*, "Force policy: Learning hybrid force-position control policy under interaction frame for contact-rich manipulation," *arXiv preprint arXiv:2602.22088*, 2026.
[19] J. Zhang, Z. Han, J. Wang, X. Wu, S. Lin, J. Li, H. Fan, R. Wu, D. Li, and H. Dong, "Hipolicy: Hierarchical multi-frequency action chunking for policy learning," *arXiv preprint arXiv:2604.06067*, 2026.
[20] K. Wang, Y. Zheng, Y. Zheng, J. Zhao, and W. Ding, "Learning high-frequency continuous action chunks in latent space," *arXiv preprint arXiv:2605.24931*, 2026.
[21] X. Feng, Y. Cheng, C. Shi, B. Han, Y. Yan, Y. Hong, Z. Tian, and L. Jiang, "Denoising tells when to replan: Denoising-variance adaptive chunking for flow-based robot policies," *arXiv preprint arXiv:2606.03847*, 2026.
[22] J. Cai, L. Ling, S. Chu, Z. Liu, J. Kang, Z. Liang, W. Xu, Y. Mao, W. Zhang, X. Yang, *et al.*, "Aha-wam: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing," *arXiv preprint arXiv:2606.09811*, 2026.
[23] G. Lee, Y. Lee, K. Kim, S. Lee, S. Noh, S. Back, and K. Lee, "Manipforce: Force-guided policy learning with frequency-aware representation for contact-rich manipulation," *arXiv preprint arXiv:2509.19047*, 2025.
[24] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.
[25] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," 2023.
[26] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, W. Ding, and Y. Tang, "Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation," *arXiv preprint arXiv:2406.01586*, 2024.
[27] Q. Zhao, Y. Shen, X. Zhai, D. Wu, J. Qi, C. Hao, J. Hu, and Q. Yu, "Hybrid consistency policy: Decoupling multi-modal diversity and real-time efficiency in robotic manipulation," *IEEE Robotics and Automation Letters*, 2026.
[28] Z. Geng, M. Deng, X. Bai, Z. Kolter, and K. He, "Mean flows for one-step generative modeling," *Advances in Neural Information Processing Systems*, vol. 38, pp. 75 460–75 482, 2026.
[29] J. Sheng, Z. Wang, P. Li, and M. Liu, "Mp1: Meanflow tames policy learning in 1-step for robotic manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 22, 2026, pp. 18 532–18 539.
[30] Y. Su, N. Liu, D. Chen, Z. Zhao, K. Wu, M. Li, Z. Xu, Z. Che, and J. Tang, "Freqpolicy: Efficient flow-based visuomotor policy via frequency consistency," *Advances in Neural Information Processing Systems*, vol. 38, pp. 27 769–27 797, 2026.
[31] T. Li and K. He, "Back to basics: Let denoising generative models denoise," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 36 115–36 125.
[32] Y. Lu, S. Lu, Q. Sun, H. Zhao, Z. Jiang, X. Wang, T. Li, Z. Geng, and K. He, "One-step latent-free image generation with pixel mean flows," *arXiv preprint arXiv:2601.22158*, 2026.
[33] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.
[34] H. Xue, N. Min, X. Liu, W. Chen, Y. Fang, J. Lv, C. Lu, and C. Wen, "Rethinking camera choice: An empirical study on fisheye camera properties in robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 35 059–35 069.