PhaForce: Phase-Scheduled Visual–Force Policy Learning with Slow Planning and Fast Correction for Contact-Rich Manipulation

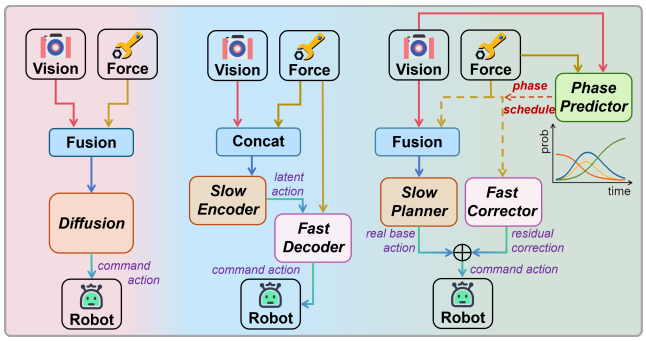
Mingxin Wang[1], Zhirun Yue[1,3], Renhao Lu[1,2], Yizhe Li[2], Zihan Wang[1], Guoping Pan[2], Kangkang Dong[1], Jun Cheng[3], Yi Cheng[2], Houde Liu[1,*]

***Abstract*— Contact-rich manipulation requires not only vision-dominant task semantics but also closed-loop reactions to force/torque (F/T) transients. Yet, generative visuomotor policies are typically constrained to low-frequency updates due to inference latency and action chunking, underutilizing F/T for control-rate feedback. Furthermore, existing force-aware methods often inject force continuously and indiscriminately, lacking an explicit mechanism to schedule *when / how much / where* to apply force across different task phases. We propose PhaForce, a phase-scheduled visual–force policy that coordinates low-rate chunk-level planning and high-rate residual correction via a unified *contact/phase* schedule. PhaForce comprises (i) a contact-aware phase predictor (CAP) that estimates contact probability and phase belief, (ii) a Slow diffusion planner that performs dual-gated visual–force fusion with *orthogonal residual injection* to preserve vision semantics while conditioning on force, and (iii) a Fast corrector that applies *control-rate* phase-routed residuals in interpretable corrective subspaces for within-chunk micro-adjustments. Across multiple real-robot contact-rich tasks, PhaForce achieves an average success rate of 86% (+40 pp over baselines), while also substantially improving contact quality by regulating interaction forces and exhibiting robust adaptability to OOD geometric shifts.**

I. INTRODUCTION

Diffusion-based visuomotor policies [1] and recent VLA models [2], [3] have achieved strong performance on vision-dominant manipulation tasks such as pick-and-place, rearrangement, and folding. However, many real-world skills are inherently *contact-rich*: success depends not only on geometric alignment but also on interaction dynamics such as friction, jamming, and transient impacts [4]–[9]. In such scenarios, vision is often ambiguous or occluded, and critical signals emerge as short-horizon 6D force/torque (F/T) events. For instance, in insertion, being fully seated versus being jammed on the rim can be visually indistinguishable at millimeter scale, while wrench transients reveal misalignment and recovery cues [10]. Similarly, in wiping, visual observations rarely reveal whether the tool is slightly detached or over-pressed [8].

This motivates incorporating F/T (wrench) sensing as physical feedback for contact-rich manipulation. Most force-aware policies encode a short F/T history and fuse it with vision (e.g., concatenation or attention), then use the fused multimodal representation in a chunked generative policy [11]–[14].



(a) Prior force-aware policy (b) RDP slow-fast policy (c) PhaForce

Fig. 1. Comparison of three force-aware policy architectures. Prior works fuse vision and force into a single generative policy, while RDP adopts a slow–fast decomposition without explicit phase scheduling. **PhaForce** introduces an explicit contact/phase schedule to coordinate force usage for both chunk-level planning (Slow) and within-chunk correction (Fast).

However, a key structural mismatch remains underexplored **(Gap-1: timescale mismatch of force feedback)**: F/T is a feedback signal whose value lies in rapid closed-loop correction, while generative policies are typically constrained to low-frequency updates by inference latency and action chunking. When force is primarily consumed at the action-chunk update rate, short-horizon interaction transients (e.g., stick–slip, micro-impacts, early jamming) can be under-reacted. This calls for an explicit closed-loop correction layer that reacts to force feedback within an action chunk.

Reactive Diffusion Policy [15] takes an important step towards slow–fast execution by coupling chunk-level generative planning with within-chunk reactivity. However, such reactive designs remain largely *phase-agnostic*, often applying high-frequency corrections without explicitly distinguishing which motion channels should be corrected at different stages. Contact-rich manipulation is inherently multi-phase: different stages (e.g., planar search versus normal insertion) demand orthogonal or even mutually exclusive corrective subspaces. Without an explicit phase schedule, high-rate reactivity can introduce spurious corrections in irrelevant subspaces, degrading alignment and potentially leading to jamming behaviors. Prior work therefore lacks an explicit phase schedule to dynamically route force feedback—leaving a critical gap in deciding *when* to trust force, *how much* to use it, and critically, *where* (in which corrective subspace) to apply it **(Gap-2: explicit phase scheduling)** [16], [17].

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China.
[2]Zerith Robotics, Shenzhen 518055, China.
[3]Shenzhen Institutes of Advanced Technology, Shenzhen 518055, China.
*Corresponding author: Houde Liu (liu.hd@sz.tsinghua.edu.cn)
This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001) and the Shenzhen Science and Technology Program (Grant No. ZDCY20250901104207008).

In this work, we propose **PhaForce**, a *phase-scheduled* slow–fast policy that uses an explicit contact probability and a task-defined phase-belief distribution to coordinate force usage for both chunk-level planning and within-chunk closed-loop correction. Fig. 1 provides an intuitive comparison of three force-aware policy architectures. PhaForce consists of three components: (1) a *contact-aware phase predictor* that outputs a continuous contact probability and a soft distribution over phases, providing an explicit semantic schedule signal; (2) a **Slow** diffusion planner that performs *dual-gated* visual–force fusion for long-horizon action-chunk generation, where contact gates the overall force injection and phase belief modulates the fused representation to maintain vision-dominant semantics via *orthogonal residual injection*; and (3) a **Fast** residual corrector that applies control-rate corrections in *phase-routed* corrective subspaces, trained with physically motivated supervision constructed from the *virtual target pose*. The final control command is obtained by composing the **Slow** real base action with the **Fast** residual correction.

Our contributions are summarized as follows:

- We propose **PhaForce**, a *phase-scheduled* slow–fast policy that unifies force-aware chunk-level generative planning with control-rate residual correction.
- We introduce an explicit scheduling signal (contact probability + phase belief) that decides *when / how much* to use force for planning and *where* to correct during execution, realized by dual-gated fusion with orthogonal residual injection in **Slow** and phase-routed corrective subspaces in **Fast**.
- We validate PhaForce on multiple real-robot contact-rich manipulation tasks, showing consistent improvements over strong baselines in both ID and OOD settings.

II. RELATED WORK

A. Force-Aware Visuomotor Policy Learning

A prevalent paradigm for force-aware imitation learning encodes a short history window of 6D wrench measurements and fuses the resulting force representation with visual features—typically via feature concatenation [11], [14], [18], [19] or attention-based cross-modal interaction [12], [20]–[22], treating the multimodal feature as the observation for a visuomotor policy. FoAR [16] modulates force usage with a predicted future contact probability to suppress noisy wrench signals in free-space and amplify force under contact. ForceVLA [13] employs a force-aware Mixture-of-Experts block, where expert routing varies with task progress and can implicitly specialize across interaction stages.

Beyond fusion, Stepputtis et al. [5] introduce a continuous phase variable (from 0 to 1) to represent task progress and feed it to skill primitives for contact-rich manipulation. TA-VLA [10] highlights torque transients as reliable event signals that reveal contact outcomes and naturally support intent switching (e.g., detect failure and retry).

However, existing approaches still lack an explicit, task-defined *probabilistic phase belief* mechanism to interpretably schedule *when / how much* force should be fused with vision and to exploit force cues without degrading visual task semantics.

B. Slow–Fast Policy Learning under Action Chunking

Despite recent progress in force-aware visuomotor policy learning, many methods still rely on action-chunk generation, resulting in near open-loop execution within each chunk and delaying the use of wrench transients that often signal contact anomalies and intent switches. ManipForce [17] introduces frequency-aware multimodal representations, but still follows the chunked diffusion paradigm without dedicated within-chunk force-driven correction.

Reactive Diffusion Policy [15] adopts a slow–fast architecture, where a slow diffusion model predicts low-rate latent action chunks and a fast decoder leverages high-rate wrench feedback to autoregressively generate fine-grained control commands within each chunk. Subsequent analyses [23] suggest that latent chunk compression may reduce free-space motion precision and degrade millimeter-level approach/contact localization. More broadly, wrench feedback in existing designs is still primarily used for within-chunk local refinement, leaving open a structured, task-semantic mechanism for long-horizon intent switching and phase-dependent correction. ImplicitRDP [23] further explores end-to-end structural slow–fast learning, yet explicit and interpretable phase-level scheduling remains underexplored.

Overall, it remains underexplored how to close the loop at high control rates while coordinating long-horizon planning and residual correction in an interpretable, phase-dependent manner.

III. METHOD

In this section, we present **PhaForce**, a phase-scheduled visual–force policy for contact-rich manipulation (Fig. 2). We first formalize the problem and specify the slow–fast execution pipeline under action chunking (Sec. III-A). We then introduce a *contact-aware phase predictor* that outputs a continuous contact probability and a phase-belief distribution (Sec. III-B). Finally, we describe how this phase belief coordinates both low-frequency task-intent planning in the **Slow** planner (Sec. III-C) and high-frequency residual correction in the **Fast** corrector (Sec. III-D).

A. Problem Formulation and Preliminaries

Observation inputs. At each timestep t, we define two observation views tailored to slow–fast execution. The *planner observation* $o_t^p = (\mathcal{I}_t,\ w_t^{\text{hist}},\ s_t)$ includes multi-view RGB images $\mathcal{I}_t$, wrench history w_t^{hist} expressed in the TCP frame, and proprioception s_t. The *corrector observation* $o_t^c = (w_t^{\text{hist}},\ s_t)$ excludes images and uses only low-dimensional signals. Here we define H_w as the wrench-history window length, capturing short-term interaction dynamics.

Policy outputs. We learn a slow–fast policy pair $(\pi_{\text{slow}}, \pi_{\text{fast}})$ under action chunking with two update rates. As shown in Fig. 2(c), f_c denotes the *control frequency* at which the robot is commanded and the **Fast** corrector is evaluated,

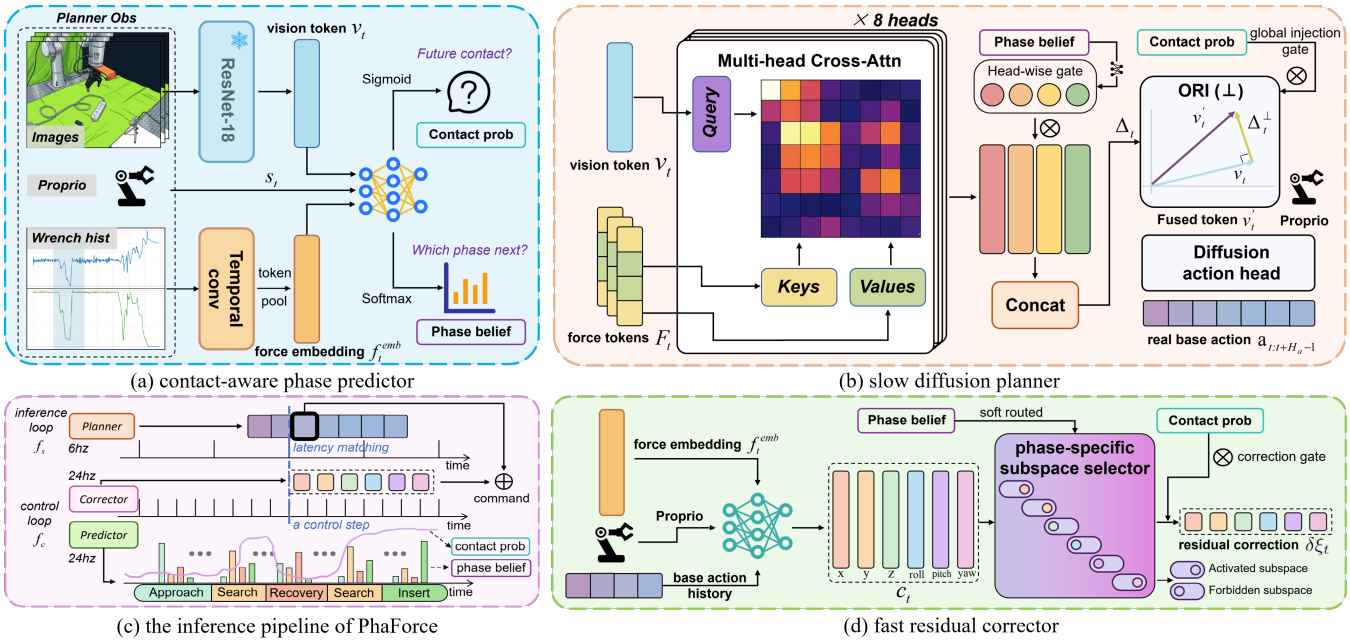


Fig. 2. **PhaForce Architecture.** The **Slow** diffusion planner runs at f_s=6 Hz to generate action chunks, while **CAP** and the **Fast** corrector run at the control rate f_c=24 Hz for contact/phase prediction and within-chunk closed-loop correction. In **Slow**, dual-gated vision–force fusion with orthogonal residual injection preserves vision-dominant task semantics. In **Fast**, phase-belief soft routing activates corrective subspaces and outputs a residual correction that is composed with the Slow base action to obtain the executed command.

and f_s denotes the *inference frequency* of the **Slow** planner, typically $f_s \ll f_c$ due to inference latency. π_{slow} predicts a nominal action chunk; we denote by $T_t^{\text{slow}} \in SE(3)$ the corresponding nominal TCP pose at control step t. In contrast, π_{fast} consumes o_t^c at f_c and predicts a small delta pose $\Delta T_t^{\text{fast}} \in SE(3)$ in the TCP frame for high-rate residual correction. The executed pose is composed on $SE(3)$ as

$$T_t = T_t^{\text{slow}} \circ \Delta T_t^{\text{fast}}. \tag{1}$$

In practice, we represent each pose $T_t \in SE(3)$ by position and a unit quaternion, and send the low-level command $a_t \in \mathbb{R}^8$ with gripper width.

B. Contact-Aware Phase Predictor

Beyond contact state, contact-rich tasks are inherently multi-phase; different phases demand different force usage and corrective subspaces (e.g., planar search vs. normal compliance). To explicitly represent such task progress, for each task we define K task-specific phases (e.g., *approach/search/recovery/insert/done* in plug-in tasks) and predict a continuous contact probability $p_t^c \in [0,1]$ and a phase belief $\mathbf{p}_t \in \Delta^{K-1}$ (with $\sum_{k=1}^{K} \mathbf{p}_t^{(k)} = 1$), which are evaluated at f_c and used to schedule force usage in **Slow** and **Fast** (Secs. III-C–III-D).

Inputs and outputs. As shown in Fig. 2(a), we introduce a lightweight predictor **CAP**, denoted as π_{CAP}, which takes the planner observation $o_t^p = (\mathcal{I}_t, w_t^{\text{hist}}, s_t)$ as input and outputs $(p_t^c, \mathbf{p}_t)$. We use a ResNet-18 for each RGB view without weight sharing to extract visual features, which are fused with force/proprio features by a small MLP, followed by a binary contact head and a categorical phase head.

Force encoder. To encode the wrench history w_t^{hist} while capturing both abrupt interaction transients and short-term temporal dependencies, we use a lightweight TCN-style temporal encoder shared across **CAP**, **Slow**, and **Fast**. It is implemented as a stack of dilated 1D convolutions with residual connections, providing multi-scale temporal receptive fields that are sensitive to short-lived F/T changes. For **Slow**, the encoder outputs a sequence of force tokens $\{f_{t-i}\}_{i=0}^{H_w-1}$ with $f_{t-i} \in \mathbb{R}^d$; for modules that require a single vector (**CAP/Fast**), we additionally apply temporal pooling to obtain a compact force embedding $f_t^{\text{emb}} \in \mathbb{R}^{d_f}$.

Targets and loss. Importantly, π_{CAP} is trained for *anticipation* rather than instantaneous judgment. We supervise the contact head using a future-window label indicating whether contact will occur within the next K_f control steps: $y_t^c = \bigvee_{i=1}^{K_f} \text{contact}_{t+i}$ and supervise the phase head using a future offset label $y_t^{\text{phase}} = \text{phase}_{t+\delta}$. Let $\ell_t^c \in \mathbb{R}$ and $\ell_t^\phi \in \mathbb{R}^K$ denote the contact/phase logits, with $p_t^c = \sigma(\ell_t^c)$ and $\mathbf{p}_t = \text{softmax}(\ell_t^\phi)$. We optimize a multi-task objective:

$$\mathcal{L}_{\text{CAP}} = \mathcal{L}_{\text{BCE}}(y_t^c, \ell_t^c) + \lambda_\phi \mathcal{L}_{\text{CE}}(y_t^{\text{phase}}, \ell_t^\phi), \tag{2}$$

where $\mathcal{L}_{\text{BCE}}$ is binary cross-entropy (with logits) for future contact prediction and $\mathcal{L}_{\text{CE}}$ is cross-entropy for phase prediction. All labels are automatically generated via scripts using wrench signals and TCP pose, avoiding manual annotation.

C. Slow Diffusion Planner

As shown in Fig. 2(b), the **Slow** diffusion planner π_{slow} runs at rate f_s with an augmented planner input $\tilde{o}_t^p = (o_t^p, p_t^c, \mathbf{p}_t)$ and outputs an executable action chunk of

horizon H_a in control steps:

$$\mathbf{a}_{t:t+H_a-1} \sim \pi_{\text{slow}}(\cdot \mid \tilde{o}_t^p). \tag{3}$$

Encoders. We encode the multi-view RGB observation into a single visual token $v_t \in \mathbb{R}^d$ by concatenating per-view global embeddings extracted by ResNet-18 encoders, and encode the wrench history into force tokens $F_t \in \mathbb{R}^{H_w \times d}$ using the force encoder described in Sec. III-B.

Dual-gated fusion. We fuse vision and force via a multi-head cross-attention block, which uses the visual token as a query to attentively aggregate the force tokens.

Let $Q = v_t W_Q \in \mathbb{R}^{1\times d_k}$ and $K = F_t W_K,\ V = F_t W_V \in \mathbb{R}^{H_w \times d_k}$ where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d_k}$. For a single head,

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{4}$$

To make force usage phase-dependent and interpretable, we introduce a phase-dependent *head-wise gate*

$$g_t^{\text{head}} = \sigma(\text{MLP}(\mathbf{p}_t)) \in [0,1]^H, \tag{5}$$

where $g_t^{\text{head}}(h)$ denotes its h-th element. We reweight per-head outputs by g_t^{head} and obtain the cross-attention output

$$\Delta_t = W_O\big[\, g_t^{\text{head}}(1)\,\text{Attn}_1;\ \ldots;\ g_t^{\text{head}}(H)\,\text{Attn}_H\,\big], \tag{6}$$

where Attn_h denotes the output of head h computed by Eq. (4) with head-specific projections, and W_O is the output projection. In addition, a *global injection gate* $g_t^c = p_t^c$ controls the injection strength in Eq. (8), suppressing the influence of noisy wrench signals in free-space.

Orthogonal residual injection (*ORI*). Rather than overwriting the visual feature with Δ_t, we inject it as a *residual* and retain only its component *orthogonal* to the visual token, preserving vision-dominant semantics and mitigating semantic drift.

$$\Delta_t^\perp = \Delta_t - \text{Proj}_{v_t}(\Delta_t) = \Delta_t - \frac{\langle \Delta_t, v_t\rangle}{\langle v_t, v_t\rangle + \epsilon}\, v_t, \tag{7}$$

where $\epsilon = 10^{-6}$ is a numerical stabilizer. The fused token is then

$$v_t' = v_t + \alpha \cdot g_t^c \cdot \Delta_t^\perp, \tag{8}$$

where α is a learnable scalar gain that is clipped to a bounded range for stability. Intuitively, p_t^c controls *when / how much* force should influence planning, while $\mathbf{p}_t$ controls *which heads* are emphasized under the current phase.

Diffusion-based chunk planning. Given the conditioning $z_t = \text{cond}(v_t', s_t)$, we use a diffusion action head to generate an action chunk by progressively denoising a noisy action trajectory into executable commands [1], [24], [25]. In training, the rotational component is represented by 6DRot.

D. Fast Residual Corrector

As shown in Fig. 2(d), the **Fast** corrector π_{fast} runs at rate f_c with an augmented corrector input $\tilde{o}_t^c = (\, o_t^c,\ \mathbf{h}_t^{\text{slow}},\ p_t^c,\ \mathbf{p}_t\,)$, where $\mathbf{h}_t^{\text{slow}}$ denotes a short history of base actions produced by **Slow**. **Fast** predicts an intermediate within-chunk *channel-wise* residual c_t:

$$c_t = \pi_{\text{fast}}(\tilde{o}_t^c) \in \mathbb{R}^6, \tag{9}$$

where $c_t = [c_x,\ c_y,\ c_z,\ c_{\text{roll}},\ c_{\text{pitch}},\ c_{\text{yaw}}]^\top$.

Phase-routed corrective subspaces. The channel residual c_t specifies residual increments along interpretable correction channels. For each phase $k \in \{1,\ldots,K\}$, we predefine a *phase-specific subspace selector* $B_k \in \mathbb{R}^{6\times 6}$ as a diagonal binary channel mask, i.e., $B_k = \text{diag}(m_k)$ with $m_k \in \{0,1\}^6$, which disables *forbidden* dimensions and keeps only the *activated* channels in that phase. For example, in plug-in tasks, the *search* phase activates (x, y, yaw) (e.g., $m_{\text{search}} = [1,1,0,0,0,1]$), whereas the *insert* phase activates normal compliance along (z, yaw). We then obtain the residual twist by *softly routing* the raw channel-wise residual c_t using the phase belief, with contact gating by p_t^c:

$$\delta\xi_t = p_t^c \left(\sum_{k=1}^{K} \mathbf{p}_t^{(k)}\, B_k\right) c_t \in \mathbb{R}^6, \tag{10}$$

For execution, $\delta\xi_t$ is converted to the pose increment ΔT_t^{fast}. Unlike admittance controllers that typically rely on fixed gains and hand-designed DOF switching, phase-belief soft routing smoothly interpolates among phase-specific corrective subspaces. Moreover, contact gating suppresses spurious corrections induced by free-space wrench noise without additional heuristic thresholds or filtering.

Physical-prior supervision. In contact-rich tasks, beyond the nominal TCP pose, we consider a *virtual target pose* that the robot would track under compliant interaction, as implied by force feedback [11]. Rather than explicitly estimating this target pose, we treat the desired *pose offset* as a residual twist and specify it via a phase-dependent physical prior, yielding automatic supervision for **Fast**. Concretely, we construct phase-wise physically motivated residual targets $\delta\xi_{t,k}^* \in \mathbb{R}^6$ from wrench signals to capture desired corrective trends in each phase. For example, during planar *search*, the target drives the robot to relieve tangential friction and mitigate jamming torques:

$$\delta\xi_{t,\text{search}}^* = [-\alpha_x F_{x,t},\ -\alpha_y F_{y,t},\ 0,\ 0,\ 0,\ -\alpha_{\text{yaw}}\tau_{z,t}]^\top, \tag{11}$$

where we treat roll/pitch as a *forbidden subspace* to promote stable execution. During *wiping*, the target enforces normal compliance by tracking a desired normal force F_z^*:

$$\delta\xi_{t,\text{wiping}}^* = [0,\ 0,\ \alpha_z(F_z^* - F_{z,t}),\ 0,\ 0,\ 0]^\top. \tag{12}$$

Similar targets can be defined for rotation channels using measured torques. To keep supervision consistent with the soft-routed correction in Eq. (10), we compute a single target

by phase-belief–weighted averaging of phase-wise residual priors, with correction gating by p_t^c:

$$\delta\xi_t^* = p_t^c \sum_{k=1}^{K} \mathbf{p}_t^{(k)}\, \delta\xi_{t,k}^*. \tag{13}$$

Training loss. We regress $\delta\xi_t$ to $\delta\xi_t^*$ with an ℓ_1 loss:

$$\mathcal{L}_{\text{fast}} = \mathbb{E}\left[\|\delta\xi_t - \delta\xi_t^*\|_1\right]. \tag{14}$$

IV. EXPERIMENTS

A. Experimental Setup

Our experiments are conducted on a Flexiv Rizon 4s robotic arm equipped with a 6-axis force/torque sensor at the end effector. We use one wrist-mounted and two external Intel RealSense D435 cameras to provide multi-view RGB observations. We collect 80 expert teleoperated demonstrations per task using TactAR [15], which provides real-time *wrench visualization* to refine contact behaviors. All devices are connected to a workstation with an Intel Core i7-14700F CPU and an NVIDIA RTX 4060 Ti GPU for data collection and policy evaluation.

B. Tasks and Metrics

Tasks. We evaluate PhaForce on five real-robot contact-rich tasks with task-defined phases for **CAP**, capturing phase-dependent corrective subspaces beyond normal-force compliance (e.g., tangential friction and torque cues for planar alignment and jamming relief). Table I summarizes the phase-specific activated subspaces for **Fast**.

(i) Charger Plug-in. Phases: {*approach, search, recovery, insert, done*}. *Search* performs planar hole-alignment, where tangential friction forces and torques reveal misalignment and hole rim contact; thus corrections mainly lie in a planar subspace. *Recovery* indicates severe sticking or large wrench transients and requires an explicit retreat-and-retry intent switch, handled by the **Slow** planner rather than within-phase **Fast** micro-correction. Notably, *Recovery* may not occur in every episode; it is activated only when the above conditions are detected, while many successful trials proceed with *search* followed by *insert*. *Insert* emphasizes continuous advancement toward a fully seated insertion.

(ii) USB Plug-in. Phases: same as **(i)**. In our setup, USB is more sensitive to small planar/yaw misalignment, often exhibiting friction-induced stick–slip and transient torques due to edge contact under tight tolerances.

(iii) Drawer Opening. Phases: {*pick, unlock, pull, done*}. *Unlock* overcomes initial stiction. *Pull* follows the drawer's constraint-guided sliding motion, where wrench feedback enforces directional compliance—driving along the opening direction while suppressing lateral forces that cause binding.

(iv) Wiping (ID). Phases: {*pick, approach, wiping, done*}. Only *wiping* is force-critical: vision is ambiguous about contact quality, as slight detachment yields ineffective wiping while over-pressing increases friction and induces oscillation; thus normal force feedback provides a direct signal for maintaining effective contact.

TABLE I
CORRECTIVE SUBSPACES FOR FORCE-CRITICAL PHASES ACROSS TASKS

subspace	search	insert	unlock	pull	wiping
x	✓		✓	✓	
y	✓		✓	✓	
z		✓			✓
roll				✓	
pitch				✓	
yaw	✓	✓			

(v) Wiping (OOD). Same phases as ID, but the board is raised by 3 cm at test time while demonstrations are collected at the original height, creating an out-of-distribution contact condition. This further tests whether a policy can leverage force feedback to compensate for unseen contact geometry and maintain stable wiping.

Metrics. For each method and each task, we run 20 evaluation trials with randomized initial conditions and report the success rate (SR). For plug-in tasks, we regard partial insertion that does not reach a fully seated state as failure. For wiping, we additionally evaluate contact quality and wiping effectiveness in Sec. IV-E.

C. Baselines and Implementation

Baselines. We compare against four methods: (i) *Diffusion Policy (DP)* [1], a strong vision-only imitation learning policy; (ii) *DP (force-concat)*, which directly concatenates the force feature with the vision feature for action generation; (iii) *RDP* [15], a representative slow–fast diffusion policy that leverages high-rate wrench/tactile feedback for within-chunk reactive execution; and (iv) *PhaForce (Ours)*.

Implementation. For all methods, diffusion runs at $f_s = 6$ Hz and we execute at a control rate of $f_c = 24$ Hz with an action-chunk horizon $H_a = 16$; we adopt latency matching by discarding the first few steps following UMI [26] and RDP [15], and send interpolated actions to the low-level controller at > 500 Hz. For *PhaForce*, we use a wrench-history window of $H_w = 36$ ($\approx 1.5\,s$) to capture short-term interaction dynamics. Each RGB view is encoded into a 512-d embedding and concatenated into a visual token of dimension $d = 1536$. For **CAP**, we set $\delta = 3$, $K_f = 8$, and $\lambda_\phi = 2$. In **Slow**, we use multi-head cross-attention with $H = 8$ heads (per-head dimension $d_k = d/H = 192$). For diffusion, we use a DDIM scheduler with ϵ-prediction, using 100 timesteps at training and 10 timesteps at inference. For the physical-prior teachers in **Fast**, we set $\alpha_x = \alpha_y = \alpha_z = 5 \times 10^{-5}\,\mathrm{m/N}$ and $\alpha_{\text{roll}} = \alpha_{\text{pitch}} = \alpha_{\text{yaw}} = 3 \times 10^{-2}\,\mathrm{rad/(N \cdot m)}$. The average inference time per run is $\sim$120 ms (Slow), $\sim$3 ms (CAP), and $<$1 ms (Fast).

D. Results and Analysis

Table II reports success rates across five real-robot tasks. Overall, **PhaForce** achieves the best (or tied-best) performance on all tasks. Averaged over the three baselines, **PhaForce** improves the mean SR by **+40** pp. Fig. 3 further shows **PhaForce**'s execution over time, highlighting phase switches throughout each task.

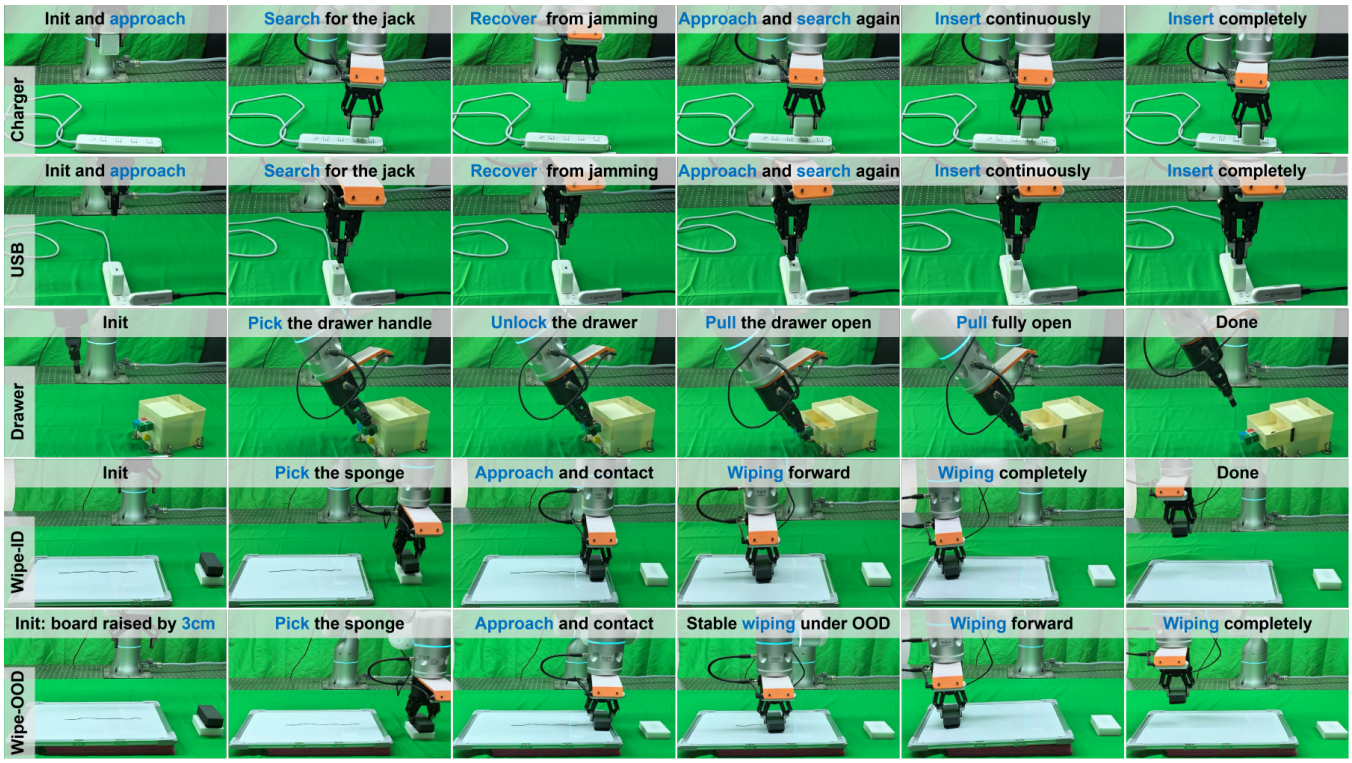
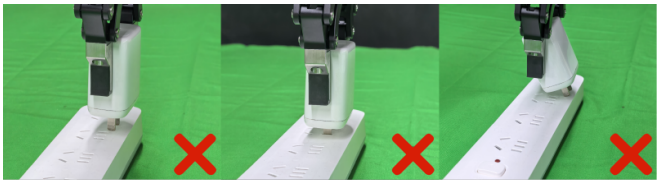


Fig. 3. We design five contact-rich tasks; each task exhibits *varying contact states and phase belief*, and each phase activates different corrective subspaces. Each row illustrates the phase transitions in a task. PhaForce not only excels on in-distribution tasks but also remains stable under OOD shifts.

TABLE II

SUCCESS RATE (SR, %) ACROSS DIFFERENT POLICIES.

Method	Charger	USB	Drawer	Wipe-ID	Wipe-OOD	Avg
DP	20	15	60	**95**	0	38
DP (force-concat)	20	20	50	85	0	35
RDP	50	55	65	85	75	66
PhaForce (ours)	**80**	**85**	**85**	**95**	**85**	**86**



(a) Stagnation (b) Partial-insertion (c) Slip-induced in-hand rotation

Fig. 4. Three common baseline failure modes in plug-in tasks.

Plug-in tasks. Insertion is highly sensitive to small pose errors and local contact geometry, and typically involves phase switches such as planar search and recovery. As shown in Fig. 4, we observe three common failure modes of baselines: *(i) Stagnation:* after a slight misalignment, the end-effector gets stuck at the socket entrance and fails to trigger planar search or retreat-and-retry. *(ii) Partial insertion:* the plug enters the socket but remains not fully seated, resulting in an incomplete insertion. *(iii) Slip-induced in-hand rotation:* rim collisions with excessive contact force can cause the connector to slip inside the gripper and rotate substantially, misorienting the plug. The last mode is an *unintended* disturbance rather than an intended search subspace, and is difficult to compensate without demonstrations covering large reorientation.

DP often fails to achieve millimeter-level alignment because it must infer the contact state and corrective direction purely from images. *DP (force-concat)* yields only limited gains because naive wrench concatenation lacks an explicit mechanism to convert high-frequency force feedback into timely within-chunk micro-corrections [15].

RDP improves performance via fast closed-loop refinement, yet it still underperforms **PhaForce** on fine insertion. We conjecture that, for such millimeter-sensitive tasks, latent action-space planning can degrade execution precision for contact alignment, consistent with the precision-loss effects attributed to latent compression in [23]. In contrast, **PhaForce**'s **Slow** predicts executable actions directly in the *real* action space, rather than latent space. Moreover, without an explicit phase belief, RDP may struggle to reliably switch into *recovery*, making it harder to escape stagnation at the socket entrance.

PhaForce mitigates these issues by using phase belief to condition and gate search/recovery behaviors, enabling timely retreat-and-retry instead of stagnation. As shown in Fig. 5, CAP's anticipatory contact/phase predictions align with wrench transients. Moreover, **PhaForce** applies targeted within-chunk residual corrections in task-relevant subspaces

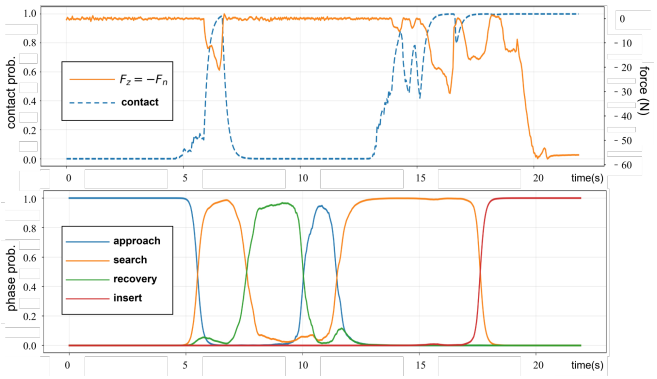


Fig. 5. We visualize the contact probability, phase belief, and z-axis force F_z in a USB Plug-in task (curves are smoothed for visualization).

TABLE III

WIPING (ID): DEEPER EVALUATION BEYOND SR.

Method	SR	Score	$\overline{F_n}$ (N)	Over	Under
DP	**95**	0.80	17.3	21.5%	7.4%
DP (force-concat)	85	0.75	17.0	16.8%	5.6%
RDP	85	0.65	13.6	5.7%	2.3%
PhaForce w/o Fast	**95**	0.70	15.1	14.3%	5.0%
PhaForce (ours)	**95**	**0.85**	12.3	**4.5%**	**2.0%**

while preserving the vision-conditioned semantics of the slow planning chunk. It also alleviates partial-insertion failures where baselines engage the hole yet fail to achieve a fully seated insertion. For slip events, we reduce their occurrence by avoiding excessive contact and by triggering recovery once abnormal wrench signals are detected; handling large in-hand rotations more fundamentally remains future work (e.g., explicit in-hand pose estimation or enriched demonstrations).

Drawer task. For Drawer Opening, **PhaForce** achieves a consistent improvement and we attribute this gain to phase-aware force utilization that facilitates compliant pulling (with small roll/pitch compliance) under friction variations and occasional binding, enabling timely adjustments rather than persisting with a misaligned pull.

E. Wiping: Contact Quality and Effectiveness

Success rate alone is insufficient for wiping, since a policy may complete the motion while applying excessive force or experiencing frequent contact dropouts. For both ID and OOD, we report a wiping score (1 for fully wiped, 0.5 for partial wiping, and 0 for no erasure), mean contact normal force $\overline{F_n}$, and the over-pressure ($F_n > 25$ N) / under-pressure ($F_n < 2.5$ N) time ratios. For reference, in our demonstrations the mean normal force during contact is 18.7 N; we empirically set the target normal force to $F_n^* = 12$ N, corresponding to $F_z^* = -12$ N under the convention $F_z = -F_n$, to reduce over-pressure while maintaining reliable erasure.

Wiping (ID). As shown in Table III, **PhaForce** achieves the best overall wiping outcome, with the highest score, the lowest under-/over-pressure ratios. By contrast, *DP* attains strong SR/Score, but its contact quality is unstable, with frequent over-pressure and contact dropouts, consistent with small teleoperation jitter in demonstrations manifesting as force fluctuations during contact. *DP (force-concat)* and *RDP* occasionally fail to grasp the sponge, highlighting the risk of using wrench feedback without phase scheduling: in non-contact stages, wrench signals are often noise-dominated, thereby hurting overall performance [16]. Meanwhile, *RDP* markedly reduces unstable contact events, yet yields a lower wiping score, plausibly because latent action-space planning can degrade fine-grained visual localization needed for precise erasing of the notes. Overall, **PhaForce** benefits from (i) **CAP** for unified phase-aware scheduling of force usage, (ii) **Slow** visual-force fusion via *ORI* that preserves vision-dominant task semantics, and (iii) the **Fast** corrector regulates the contact normal force via force feedback in the corrective subspace.

TABLE IV

WIPING (OOD): DEEPER EVALUATION BEYOND SR (– DENOTES A METRIC THAT IS NOT APPLICABLE WHEN SR=0).

Method	SR	Score	$\overline{F_n}$ (N)	Over	Under
DP	0	–	46.2	85.3%	–
DP (force-concat)	0	–	44.6	86.7%	–
RDP	75	0.65	9.2	7.2%	**0.22%**
PhaForce w/o Fast	0	–	48.3	80.3%	–
PhaForce (ours)	**85**	**0.75**	14.3	**7.0%**	0.34%

Wiping (OOD). As shown in Table IV, chunk-level diffusion planners without fast correction fail completely (*DP*, *DP (force-concat)*, and *PhaForce w/o Fast*), exhibiting sustained over-pressure and zero success. This collapse is mainly due to the height mismatch: the policy overfits to the demonstration height, causing the end-effector to either over-press and stall or stick to the board and drag quasi-statically under large friction, thus failing to execute the intended wiping motion. In contrast, *RDP* remains feasible under OOD, highlighting the advantage of a slow–fast design for contact adaptation, and **PhaForce** further improves wiping score with comparable contact stability, showing that **Fast** is key to compensating the OOD height mismatch.

F. Ablations

To validate the effectiveness of key components of **PhaForce**, we conduct ablations on two representative tasks: *USB Plug-in*, a multi-phase task with distinct phase transitions, and *Wiping (OOD)*, which requires robust contact adaptation under environment shifts. Specifically, we consider three ablation variants: (i) *PhaForce (w/o PB)*, removing phase belief by replacing $\mathbf{p}_t$ with a uniform prior over K phases during *both* training and testing, i.e., $p_t^{(k)} \equiv 1/K$; (ii) *PhaForce (w/o ORI)*, replacing the fused token v_t' with the cross-attention output Δ_t as diffusion conditioning; and (iii) *PhaForce (w/o Fast)*, removing the **Fast** residual corrector.

Results. Table V shows that removing phase belief (*w/o PB*) severely hurts *USB Plug-in* (SR 85→25), indicating that explicit phase scheduling is essential to *route* corrections to

TABLE V

ABLATION RESULTS OVER 20 REAL-ROBOT TRIALS PER METHOD.

Method	USB SR	Wipe-OOD SR	Wiping Score
PhaForce (w/o PB)	25	45	0.60
PhaForce (w/o ORI)	35	60	0.45
PhaForce (w/o Fast)	50	0	–
PhaForce (ours)	**85**	**85**	**0.75**

the right subspaces and *trigger* timely search/recovery/insert transitions in a multi-phase insertion. Replacing *ORI* with direct cross-attention conditioning (*w/o ORI*) degrades both SR and wiping score on *Wipe-OOD* (SR 85→60; score 0.75→0.45), supporting that *ORI* preserves vision-dominant semantics while injecting force information. Finally, removing **Fast** (*w/o Fast*) collapses *Wipe-OOD* (SR 0), confirming that high-rate residual correction is indispensable for stabilizing contact (rapidly compensating F/T transients) when the environment deviates from the demonstrations.

V. CONCLUSIONS

In this paper, we propose **PhaForce**, a phase-scheduled visual–force policy learning framework that integrates low-rate generative planning with high-rate reactive correction for contact-rich manipulation. PhaForce combines a contact-aware phase predictor (**CAP**) that delivers a global contact/phase schedule, a **Slow** diffusion planner that performs dual-gated vision–force fusion with orthogonal residual injection, and a **Fast** residual corrector that performs within-chunk, subspace-specific closed-loop refinement. Real-robot experiments across five tasks show that PhaForce consistently outperforms strong baselines in both ID and OOD settings, excelling in both phase-transition-intensive skills and sustained-contact skills beyond success rate alone. Future work will explore learning the Fast corrector with reinforcement learning beyond supervised residual targets, and extend PhaForce from single-task imitation to VLA models that generalize across diverse skills and embodiments.

REFERENCES

[1] 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.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024.

[3] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "$\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[4] Z. Sun, Y. Wang, D. Held, and Z. Erickson, "Force-constrained visual policy: Safe robot-assisted dressing via multi-modal sensing," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4178–4185, 2024.

[5] S. Stepputtis, M. Bandari, S. Schaal, and H. B. Amor, "A system for imitation learning of contact-rich bimanual manipulation policies," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 810–11 817.

[6] X. Zhang, C. Zhang, B. Zhang, Z. Peng, S. Cui, and S. Wang, "Dextac: Learning contact-aware visuotactile policies via hand-by-hand teaching," *arXiv preprint arXiv:2601.21474*, 2026.

[7] Y. Wu, Z. Chen, F. Wu, L. Chen, L. Zhang, Z. Bing, A. Swikir, S. Haddadin, and A. Knoll, "Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 11 831–11 837.

[8] C. Tsuji, E. Coronado, P. Osorio, and G. Venture, "Adaptive contact-rich manipulation through few-shot imitation learning with force-torque feedback and pre-trained object representations," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 240–247, 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] Z. Zhang, H. Xu, Z. Yang, C. Yue, Z. Lin, H.-a. Gao, Z. Wang, and H. Zhao, "Ta-vla: Elucidating the design space of torque-aware vision-language-action models," *arXiv preprint arXiv:2509.07962*, 2025.

[11] Y. Hou, Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, S. Feng, B. Burchfiel, and S. Song, "Adaptive compliance policy: Learning approximate compliance for diffusion guided control," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 4829–4836.

[12] J. Li, T. Wu, J. Zhang, Z. Chen, H. Jin, M. Wu, Y. Shen, Y. Yang, and H. Dong, "Adaptive visuo-tactile fusion with predictive force attention for dexterous manipulation," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 3232–3239.

[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," *arXiv preprint arXiv:2505.22159*, 2025.

[14] B. Zhou, R. Jiao, Y. Li, X. Yuan, F. Fang, and S. Li, "Admittance visuomotor policy learning for general-purpose contact-rich manipulations," *IEEE Transactions on Industrial Electronics*, 2025.

[15] 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.

[16] Z. He, H. Fang, J. Chen, H.-S. Fang, and C. Lu, "Foar: Force-aware reactive policy for contact-rich robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

[17] 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.

[18] T. Li, Y. Li, Z. Zhang, and N. Figueroa, "Flow with the force field: Learning 3d compliant flow matching policies from force and demonstration-guided simulation data," *arXiv preprint arXiv:2510.02738*, 2025.

[19] W. Liu, J. Wang, Y. Wang, W. Wang, and C. Lu, "Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 1105–1112.

[20] H. Ge, Y. Jia, Z. Li, Y. Li, Z. Chen, R. Huang, and G. Zhou, "Filic: Dual-loop force-guided imitation learning with impedance torque control for contact-rich manipulation tasks," *arXiv preprint arXiv:2509.17053*, 2025.

[21] H. Choi, Y. Hou, C. Pan, S. Hong, A. Patel, X. Xu, M. R. Cutkosky, and S. Song, "In-the-wild compliant manipulation with umi-ft," *arXiv preprint arXiv:2601.09988*, 2026.

[22] J. H. Kang, S. Joshi, R. Huang, and S. K. Gupta, "Robotic compliant object prying using diffusion policy guided by vision and force observations," *IEEE Robotics and Automation Letters*, 2025.

[23] 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," *arXiv preprint arXiv:2512.10946*, 2025.

[24] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[25] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[26] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.