Geometric Interaction Primitive Diffusion for Contact-Rich Manipulation

Abstract*—Contact-rich manipulation requires policies that encode task geometry while adapting rapidly to uncertain contact dynamics. We present *Geometric Interaction Primitive Diffusion (GIPD)*, a structured policy-controller interface that represents Cartesian position, orientation, symmetric positive-definite stiffness, and interaction direction in a phase-aligned primitive. A diffusion model predicts dense trajectories, while a geometric ProDMP decoder produces smooth, boundary-consistent motion. At inference, the policy uses vision and proprioception without direct wrench input. During execution, measured force is projected along the predicted interaction direction and drives bounded correction either to restore surface contact or advance through resistance, without modifying the nominal primitive in the orthogonal subspace or prescribing a desired force. To determine when this correction should engage and when execution should stop, we introduce a failure-driven curriculum initialized from task-level interval annotations. Subsequent rollouts are executed autonomously, while recorded task events and frozen predecessor models generate dense labels as the reliable task prefix is extended from Failed through Nice Try to Successful outcomes. This process refines kinesthetic demonstrations that fail under direct replay without repeated frame-by-frame annotation. Experiments on box flipping, MCB switch toggling, and LEGO insertion show improved overall task success over vision-only and compliance-learning baselines.

***Index Terms*—Learning from Demonstration, Compliance and Impedance Control, Machine Learning for Robot Control, Deep Learning Methods**

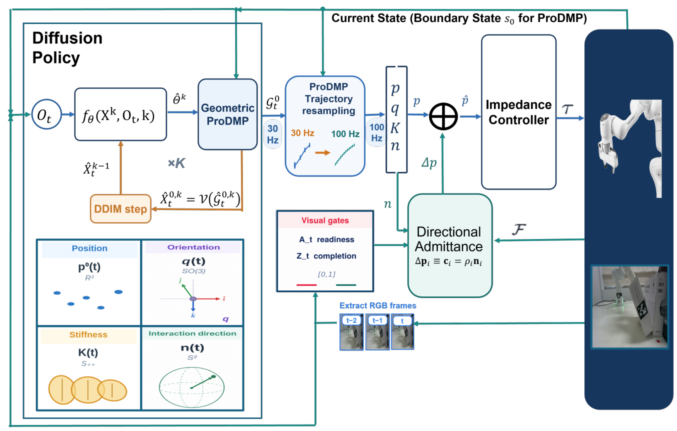


Fig. 1. Overview of the GIPD framework. From visual, proprioceptive observations and a noisy trajectory sample, the denoising network predicts geometric ProDMP parameters whose decoded primitive jointly represents position, orientation, stiffness, and interaction direction. The primitive is resampled from the policy rate to the control rate, while RGB-derived predictions of admittance need and task completion govern a directional adaptive-compliance correction driven by measured force. The corrected pose and stiffness are then executed by a Cartesian impedance controller on the Franka FR3. Wrench feedback deforms the primitive only during execution and is not an input to the diffusion policy.

I. INTRODUCTION

Contact-rich manipulation remains a barrier to deploying robots in real-world settings. Assembly, disassembly, switch operation, and nonprehensile reorientation require more than pose following. A learned policy must capture human stiffness modulation, recognize execution failure, and apply real-time corrections while preserving contact and task progress as friction, compliance, mechanism resistance, and contact state change. It must therefore represent the intended interaction geometry yet adapt to physical responses difficult to infer from vision alone.

Most force-aware learning systems place contact adaptation at one of two endpoints. Learned policy approaches either predict adaptive compliance from wrench observations [1] or use fast force feedback within slower visual action generation, requiring the policy to map dynamics-dependent wrench histories to corrective actions [2], [3]. Controller-centric methods instead preserve primarily geometric policy outputs, using predicted contact information only to configure a downstream interaction controller, leaving the structured trajectory connecting planning and execution unchanged by measured interaction [4].

We investigate the intermediate layer between these endpoints. We define a *Geometric Interaction Primitive (GIP)* as a phase-aligned representation of the intended interaction comprising end-effector position, orientation, stiffness, and interaction direction. Separate visual predictors determine when directional correction is required and when the task is complete. The learned policy predicts a nominal GIP from vision and proprioception, while measured wrench adapts only its rollout during execution. In this separation, the policy specifies *what interaction should occur*, whereas force feedback determines *how that interaction is physically realized* under the current contact response.

Movement primitives provide a compact interface between learned trajectory generation and feedback-driven execution. ProDMPs generate smooth trajectories with explicit boundary conditioning [5], [6]. Their geometric extensions represent orientation and stiffness on non-Euclidean manifolds [7], while force-coupled formulations show that measured interaction can adapt the primitive rollout itself [8]–[10]. Drawing on the structured decoding concept of Movement Primitive Diffusion [11], *Geometric Interaction Primitive Diffusion (GIPD)* integrates a geometric ProDMP decoder into the denoising process to jointly generate pose, stiffness, and interaction direction for contact-rich manipulation. Each GIP is defined on $\mathbb{R}^3 \times SO(3) \times \mathcal{S}_{++} \times S^2$, whose factors represent Cartesian position, orientation, SPD stiffness, and interaction direction, respectively. A common ProDMP phase synchronizes these components over the prediction horizon. At each reverse diffusion step, ProDMP parameters serve only as an internal parameterization of the clean estimate. Decoding produces a dense GIP trajectory, while the diffusion state and final sample remain in trajectory space.

Kinesthetic demonstrations define the motion axis from the dominant direction of local TCP velocity and the contact axis from the mean reaction force orthogonal to it. Demonstration wrench is used offline to construct direction and stiffness supervision, but it is not provided directly to the neural policy at inference. During execution, its force projection along the task-selected axis drives bounded correction. Along the contact axis, weakening support triggers inward correction while nominal motion advances the task. Along the motion axis, resistance modulates progress until visual completion. The orthogonal subspace remains governed by the nominal primitive and stiffness, and neither mode tracks a desired force. We test fixed policies with a nearly fivefold box-mass increase, reduced MCB tool-tip friction, and diagonal LEGO initialization.

The main contributions are as follows.

- **Geometric interaction primitive.** A phase-aligned primitive on $\mathbb{R}^3 \times SO(3) \times \mathcal{S}_{++} \times S^2$, jointly representing position, orientation, SPD stiffness, and interaction direction through a differentiable geometric ProDMP decoder.
- **Primitive-level force adaptation.** A task-conditioned rank-one coupling that uses force projected onto the predicted interaction direction to restore engagement or advance through resistance, while leaving the orthogonal subspace governed by the nominal primitive and keeping wrench outside neural policy inference.
- **Failure-driven timing curriculum.** An initially annotated curriculum of Failed, Nice Try, and Successful rollouts that propagates supervision for admittance activation and task termination to subsequent autonomous rollouts, without repeated frame-by-frame annotation.

II. Related Work

A. Movement Primitives and Generative Trajectory Policies

Dynamic Movement Primitives (DMPs) encode motions as dynamical systems, while Probabilistic DMPs (ProDMPs) provide compact trajectory representations with conditioning and explicit boundary control [5], [6]. Variants also support reversible execution without learning separate forward and backward primitives [12]. Geometry-aware extensions operate in local coordinates for quantities such as orientation and stiffness and reconstruct trajectories on their underlying manifolds [7]. These properties make movement primitives a useful interface between learned trajectory generation and high-rate interaction control.

Force-adaptive DMP formulations modify learned motions using stored force profiles, coupling terms, or coupled motion and force dynamical systems [13]–[15]. Admittance-coupled DMPs further demonstrate that force feedback can alter the trajectory generator itself [8], [16].

Movement Primitive Diffusion (MPD) combines trajectory-space diffusion with a ProDMP-parameterized clean action estimate to generate smooth, boundary-consistent action sequences [11]. FA-ProDMP models force and motion correlations and uses event-triggered replanning for contact-rich skills [17]. Beyond trajectory structure, explicit image-space cues for the robot action frame can improve the spatial generalization of visuomotor policies [18]. GIPD instead grounds interaction geometry in a synchronized primitive and incorporates measured wrench into its rollout only at execution time.

B. Force-Aware Policies and Interaction Control

Classical force, impedance, admittance, and hybrid force and motion controllers regulate physical interaction through explicitly designed feedback structures. Unified force and impedance control combines compliant motion and exact wrench regulation while addressing passivity and contact-loss stabilization [19]. Object-centric tactile programming transfers task constraints into such a controller, whereas tactile ergodic coverage couples surface-level coverage planning with task-space impedance and desired-force tracking [20], [21]. Our objective is narrower. GIPD maintains task-relevant engagement while a nominal geometric trajectory progresses, without tracking a desired wrench.

Learning-based methods place force and compliance at different layers. Adaptive Compliance Policy predicts task-dependent stiffness and virtual targets from visual, proprioceptive, and force observations [1]. RDP and ImplicitRDP use hierarchical or contactly causal visual and force policies to react to force within action chunks [2], [3]. Other multimodal approaches generate motion and desired force for admittance execution, refine diffusion-generated motion intent at a faster tactile-control rate, or fuse force into a reactive flow-matching policy [22]–[24]. Semantic-Contact Fields combine visual semantics with dense contact estimates as observations for diffusion-based tactile tool manipulation [25]. Diffusion-Based Impedance Learning learns wrench-consistent impedance behavior through a generative model, while a diffusion contact model predicts contact trajectories to optimize variable stiffness [26], [27]. These approaches place force or tactile information inside the learned policy or learned contact/impedance model.

Direction Matters instead predicts contact state and interaction direction as geometry-related quantities, then combines them with a manually selected force magnitude in a downstream admittance controller [4].

III. Method

GIPD separates nominal interaction generation from contact-responsive execution as follows.

$$\begin{aligned} \mathcal{G}_t^0 &= \pi_\theta(\mathcal{O}_t), \\ \mathcal{U}_t &= \mathcal{E}(\mathcal{G}_t^0, \mathbf{f}_{\text{ext}}). \end{aligned} \tag{1}$$

The learned policy π_θ samples a nominal GIP $\mathcal{G}_t^0$ from visual and proprioceptive observations. The executor $\mathcal{E}$ then uses measured wrench to adapt the position component and produce the executed command horizon $\mathcal{U}_t$. Wrench is never provided to the denoising network.

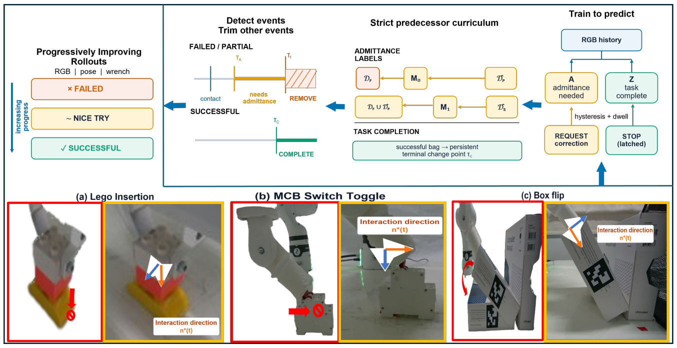


Fig. 2. Progress-driven rollout curriculum for learning interaction timing and task completion. Failed rollouts reveal where directional admittance must begin, Nice Try rollouts extend task progress, and successful rollouts identify completion. Repeating this progression supervises visual predictors of admittance need and task completion while the interaction direction remains determined by demonstration geometry and task requirements.

Given an observation history

$$\mathcal{O}_t = \{o_{t-h_o+1}, \ldots, o_t\}, \tag{2}$$

GIPD samples a nominal future primitive

$$\mathcal{G}_t^0 = \{g_{t+i}^0\}_{i=0}^{H-1}, \tag{3}$$

where each primitive state is

$$g_i^0 = (\mathbf{p}_i^0, \mathbf{q}_i, \mathbf{K}_i, \mathbf{n}_i). \tag{4}$$

Here, $\mathbf{p}_i^0 \in \mathbb{R}^3$ is the nominal Cartesian position, $\mathbf{q}_i \in S^3$ is a unit-quaternion representation of orientation in $SO(3)$, $\mathbf{K}_i \in \mathcal{S}_{++}$ is the Cartesian stiffness matrix and $\mathbf{n}_i \in S^2$ is the task-selected interaction direction supervised from offline motion and wrench geometry. Image classifiers separately estimate the probability that admittance is required, A_i, and the task-completion probability Z_i. Neither is part of the GIP. The diffusion state, training target, and final sample are dense trajectory coordinates. ProDMP parameters appear only as an internal structured parameterization of each clean GIP estimate.

A. Interaction Direction and Failure-Driven Curriculum

The interaction direction $\mathbf{n}_i \in S^2$ defines the one-dimensional task subspace in which measured wrench may adapt the nominal GIP rollout. We obtain this direction in two steps. The demonstration first provides candidate motion and contact axes, and the task mechanics then determine which axis should carry the admittance correction. The direction therefore specifies *where* force feedback may act. It neither prescribes a desired force nor determines when correction is active.

Interaction-axis construction. From synchronized kinesthetic motion and wrench, the dominant direction of local TCP velocity defines the motion axis $\mathbf{m}_i$. The mean reaction force remaining after its motion-axis component is removed defines the contact axis $\mathbf{s}_i$.

$$\begin{aligned}
\mathbf{m}_i &= \operatorname{eig}_1\left(\sum_{k\in\mathcal{W}_i} w_{ik}\mathbf{v}_k\mathbf{v}_k^\top\right), \\
\bar{\mathbf{f}}_i &= \sum_{k\in\mathcal{W}_i} w_{ik}\widetilde{\mathbf{f}}_k, \\
\mathbf{s}_i &= \frac{(\mathbf{I}-\mathbf{m}_i\mathbf{m}_i^\top)\bar{\mathbf{f}}_i}{\|(\mathbf{I}-\mathbf{m}_i\mathbf{m}_i^\top)\bar{\mathbf{f}}_i\|_2+\epsilon}, \\
\mathbf{n}_i &\parallel \begin{cases} \mathbf{s}_i, & \text{maintaining surface contact,} \\ \mathbf{m}_i, & \text{advancing resisted motion.} \end{cases}
\end{aligned} \tag{5}$$

These axes are initially signless. We orient the selected axis into the contact surface or toward demonstrated progress. BoxFlip selects $\mathbf{s}_i$, so correction restores weakening support while the nominal GIP supplies the surface motion. MCB toggling and LEGO insertion select $\mathbf{m}_i$, so projected resistance modulates progress until visual completion while the nominal GIP and stiffness govern the orthogonal subspace.

Offline stiffness construction. Following the approximate-compliance principle of ACP [1], we derive stiffness supervision offline from the demonstrated wrench. Let

$$\alpha_i = \operatorname{clip}\left(\frac{\|\widetilde{\mathbf{f}}_i\|_2 - f_{\text{low}}}{f_{\text{high}} - f_{\text{low}}}, 0, 1\right). \tag{6}$$

The stiffness along the selected interaction direction decreases with contact force as

$$\begin{aligned}
k_{\parallel,i} &= (1-\alpha_i)k_{\max} + \alpha_i k_{\min}, \\
\mathbf{K}_i &= k_\perp \mathbf{I} + (k_{\parallel,i} - k_\perp)\mathbf{n}_i\mathbf{n}_i^\top.
\end{aligned} \tag{7}$$

This produces an SPD stiffness trajectory that remains stiff for nominal tracking and becomes compliant along the task-selected interaction direction under contact. The resulting trajectory supervises the stiffness component of the GIP.

Failure-driven interaction timing. Direct replay of a kinesthetic trajectory can still fail because the controller does not know when force-responsive correction should modify the nominal trajectory. Initial task-level interval annotations identify the first persistent stall, slip, or loss of engagement and the interval leading to it; the post-failure suffix is discarded. These labels fit the first timing model, M_0, and correction is introduced along $\mathbf{n}_i$ before the detected failure frontier. Frozen M_0 then labels Nice Try (partial-progress) rollouts. Failure labels and accepted M_0 predictions fit M_1, which labels subsequent successful rollouts. Each autonomous rollout receives a coarse episode outcome, whereas recorded task events and the frozen predecessor model supply its dense timing targets. Ambiguous transition frames are excluded.

This Failed–Nice Try–Successful progression provides dense supervision for two visual predictors. The admittance-needed predictor activates or deactivates force-responsive correction along the already selected direction. The separate completion predictor terminates execution after a successful terminal event. The curriculum propagates timing labels but never changes the task-selected interaction direction.

B. Geometric ProDMP Decoder

Dynamic Movement Primitives (DMPs) provide stable nonlinear dynamical systems for representing demonstrated motions, while Probabilistic DMPs (ProDMPs) express trajectories through compact basis-function parameters and support conditioning, blending, and uncertainty-aware generation [5], [6]. For channel coordinate $\mathbf{z}_c(t)$, we write the ProDMP decoder as

$$\hat{\mathbf{z}}_c(t) = \xi_1(t)\mathbf{z}_{c,0} + \xi_2(t)\tau_m\dot{\mathbf{z}}_{c,0} + \mathbf{H}_c(t)\boldsymbol{\theta}_c, \qquad \boldsymbol{\theta}_c = [\mathbf{w}_c; \mathbf{g}_c], \tag{8}$$

where $\mathbf{w}_c$ and $\mathbf{g}_c$ are the basis weights and goal, $\mathbf{H}_c(t)$ is the channel basis matrix, τ_m is the movement duration, and ξ_1, ξ_2 encode the boundary conditions. This provides a smooth decoder that adapts to the current initial state.

To represent non-Euclidean quantities, we use local tangent coordinates before applying the ProDMP decoder and map the decoded trajectory back to the original manifold. For orientation, the quaternion trajectory $\mathbf{q}_i$ is represented around the initial orientation $\mathbf{q}_0$ by

$$\mathbf{r}_i = 2\,\mathrm{Log}(\mathbf{q}_i\mathbf{q}_0^{-1}), \qquad \mathbf{q}_i = \mathrm{Exp}\left(\frac{1}{2}\mathbf{r}_i\right)\mathbf{q}_0. \tag{9}$$

For stiffness, each symmetric positive definite matrix $\mathbf{K}_i \in \mathcal{S}_{++}$ is represented in the tangent space of a reference stiffness $\mathbf{K}_0$ as

$$\begin{aligned} &\mathbf{U}_i = \mathrm{Log}_{\mathbf{K}_0}(\mathbf{K}_i), \qquad \mathbf{z}_{K,i} = \mathrm{Mandel}(\mathbf{U}_i), \\ &\mathbf{K}_i = \mathrm{Exp}_{\mathbf{K}_0}(\mathbf{U}_i). \end{aligned} \tag{10}$$

For interaction direction, a fixed chart at the signed initial direction $\mathbf{n}_0$ gives

$$\begin{aligned} &\mathbf{u}_{n,i} = \mathrm{Log}_{\mathbf{n}_0}(\mathbf{n}_i) \in T_{\mathbf{n}_0}S^2, \\ &\boldsymbol{\Theta}_n = (\mathbf{I} - \mathbf{n}_0\mathbf{n}_0^\top)\widetilde{\boldsymbol{\Theta}}_n, \qquad \hat{\mathbf{n}}_i = \mathrm{Exp}_{\mathbf{n}_0}(\mathrm{ProDMP}(\boldsymbol{\Theta}_n; t_i)). \end{aligned} \tag{11}$$

The projection keeps the direction primitive tangent to S^2. Position is decoded in Euclidean coordinates, while orientation, stiffness, and direction are reconstructed through their manifold maps. A common phase produces a synchronized GIP on $\mathbb{R}^3 \times SO(3) \times \mathcal{S}_{++} \times S^2$ [5], [7].

C. Trajectory-Space Diffusion With a ProDMP-Parameterized x_0 Estimate

Each demonstration-derived training horizon is vectorized as $\mathbf{X}_t^0 = \mathcal{V}(\mathcal{G}_t^0)$. Its dense coordinates comprise position, local rotation vectors, Mandel-encoded SPD tangents, and signed S^2 tangents at $\mathbf{n}_0$. The forward process adds Gaussian noise directly to this trajectory,

$$\mathbf{X}_t^k = \sqrt{\bar{\alpha}_k}\mathbf{X}_t^0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{12}$$

where k is the diffusion step. The denoising network does not make ProDMP parameters the diffusion state. Instead, it predicts internal parameters for the clean GIP estimate,

$$\hat{\boldsymbol{\Theta}}^k = f_\theta(\mathbf{X}_t^k, \mathcal{O}_t, k), \qquad \hat{\boldsymbol{\Theta}}^k = \{\hat{\boldsymbol{\theta}}_p^k, \hat{\boldsymbol{\theta}}_q^k, \hat{\boldsymbol{\theta}}_K^k, \hat{\widetilde{\boldsymbol{\Theta}}}_n^k\}. \tag{13}$$

The differentiable decoder maps these parameters to a dense clean primitive,

$$\hat{\mathcal{G}}_t^{0,k} = \mathcal{D}_{\mathrm{geom}}(\hat{\boldsymbol{\Theta}}^k; s_t), \qquad \hat{\mathbf{X}}_t^{0,k} = \mathcal{V}(\hat{\mathcal{G}}_t^{0,k}), \tag{14}$$

and the reverse sampler updates the noisy trajectory as

$$\mathbf{X}_t^{k-1} = \mathcal{S}_k(\mathbf{X}_t^k, \hat{\mathbf{X}}_t^{0,k}), \tag{15}$$

where $\mathcal{S}_k$ is the DDPM or DDIM update used by the implementation and s_t denotes the current boundary state supplied to the decoder. Thus, the forward process, reverse process, supervision target, and final sample all remain in trajectory space. The internal ProDMP parameterization constrains each x_0 estimate to a smooth, boundary-consistent GIP and allows the sampled primitive to be evaluated at the required execution timestamps.

D. Geometric Trajectory Objective

The denoising model is trained with an x_0-prediction objective applied after geometric ProDMP decoding. Rather than supervising internal parameters, we compare the decoded trajectory with the demonstration-derived target GIP,

$$\mathcal{L} = \mathbb{E}_{t,k,\boldsymbol{\epsilon}}\left[\lambda_p\mathcal{L}_p + \lambda_R\mathcal{L}_R + \lambda_K\mathcal{L}_K + \lambda_n\mathcal{L}_n\right]. \tag{16}$$

Here $\mathcal{L}_p$ is the mean squared Euclidean position error and $\mathcal{L}_R$ is the squared quaternion geodesic error, respecting $\mathbf{q} \equiv -\mathbf{q}$. Stiffness is compared using the mean affine-invariant distance

$$d^2_{\mathrm{AIRM}}(\mathbf{A}, \mathbf{B}) = \|\,\mathrm{logm}(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}})\|_F^2. \tag{17}$$

For signed interaction direction, we use the validity- and confidence-weighted angular error

$$\mathcal{L}_n = \frac{\sum_i \nu_i\gamma_i\,\mathrm{atan2}(\|\hat{\mathbf{n}}_i \times \mathbf{n}_i\|_2, \hat{\mathbf{n}}_i^\top\mathbf{n}_i)}{\sum_i \nu_i\gamma_i + \epsilon}, \tag{18}$$

where ν_i and γ_i denote geometric validity and confidence. This preserves the distinction between $\mathbf{n}$ and $-\mathbf{n}$.

Applying the loss after decoding trains the network to generate accurate GIPs rather than to match a particular set of internal ProDMP parameters. The geometric reconstruction keeps orientation predictions on $SO(3)$, stiffness commands in $\mathcal{S}_{++}$, and interaction directions on S^2 for subsequent force coupling.

E. Task-Conditioned Directional Admittance

The GIP decoder and admittance controller are separate modules. At each execution step, the decoded GIP provides the nominal ProDMP position $\mathbf{p}_{d,i}^0$ and policy-predicted, offline-supervised interaction direction $\hat{\mathbf{n}}_i$. The controller uses this direction to constrain where force-dependent corrections may modify the nominal trajectory. After normalization and signed-direction validation,

$$\mathbf{n}_i = \frac{\hat{\mathbf{n}}_i}{\|\hat{\mathbf{n}}_i\| + \epsilon}, \qquad f_{c,i} = \max\left(0, -\tilde{\mathbf{f}}_i^\top\mathbf{n}_i\right), \tag{19}$$

where $\tilde{\mathbf{f}}_i$ is the calibrated base-frame force and $f_{c,i}$ is the nonnegative reaction resolved against the signed interaction direction. When $\mathbf{n}_i = \mathbf{s}_i$, this projection measures support compression. When $\mathbf{n}_i = \mathbf{m}_i$, it measures resistance opposing task progress. Invalid directions and abrupt directional changes are rejected before admittance is enabled.

The correction is represented by a scalar displacement ρ_i along the interaction direction,

$$\mathbf{c}_i = \rho_i \mathbf{n}_i, \qquad \mathbf{p}_{d,i} = \mathbf{p}_{d,i}^0 + \mathbf{c}_i. \tag{20}$$

This construction guarantees $(\mathbf{I} - \mathbf{n}_i \mathbf{n}_i^\top)\mathbf{c}_i = \mathbf{0}$, so the nominal trajectory is modified only in the one-dimensional interaction subspace.

A supervisory gate combines the temporally filtered admittance prediction $\bar{A}_i$, latched completion prediction $\bar{Z}_i$, eligible phase e_i, direction validity d_i, and safety/freshness state s_i.

$$\begin{aligned} g_i &= \bar{A}_i(1 - \bar{Z}_i) e_i d_i s_i, \qquad \bar{A}_i = \mathcal{H}_A(A_i), \\ \bar{Z}_i &= \max_{j \le i} \mathcal{H}_Z(Z_j). \end{aligned} \tag{21}$$

where $\mathcal{H}_A$ and $\mathcal{H}_Z$ apply temporal hysteresis and dwell. The gate enables directional correction only when admittance is requested, completion has not been detected, and the phase, direction, and safety checks are valid. The two predictions have distinct roles: when $\bar{A}_i$ is off, directional correction is disabled, whereas a latched $\bar{Z}_i$ terminates the task and holds the final safe command. The factor $(1 - \bar{Z}_i)$ therefore disables correction during termination. While the gate is active, the controller selects a target correction velocity from two interaction modes,

$$v_i^\star = \begin{cases} -v_{\mathrm{ret}}, & f_{c,i} \ge f_{\mathrm{over}}, \\ v_{\mathrm{search}} \mathbb{I}[f_{c,i} < f_{\mathrm{loss}}], & \mu_i = \text{search}, \\ v_{\min} + \Delta v \dfrac{f_{c,i}}{f_s + f_{c,i}}, & \mu_i = \text{progress}, \end{cases} \tag{22}$$

where μ_i denotes the task mode associated with the selected direction and $\Delta v = v_{\max} - v_{\min}$. In contact search, $\mathbf{n}_i = \mathbf{s}_i$, and weak projected reaction triggers bounded inward motion while the nominal motion advances the task. In resistance progress, $\mathbf{n}_i = \mathbf{m}_i$, and the correction rate increases with projected resistance while the orthogonal subspace remains governed by the nominal trajectory and stiffness. Excessive compression commands a bounded retreat. Neither mode tracks a desired force. Outside the active gate, the correction is held or released smoothly according to the interaction state.

To avoid abrupt Cartesian commands, the selected velocity is integrated through an acceleration- and displacement-limited scalar state.

$$\begin{aligned} v_i &= v_{i-1} + \mathrm{clip}(v_i^\star - v_{i-1}, -a_{\max}\Delta t_i, a_{\max}\Delta t_i)\,, \\ \rho_i &= \mathrm{clip}(\rho_{i-1} + v_i \Delta t_i, -\rho_{\max}, \rho_{\max})\,. \end{aligned} \tag{23}$$

The correction is introduced and removed smoothly, remains continuous through brief readiness dropouts and policy replanning, and stops at completion or when a lateral-force, torque, or signal-freshness limit is reached. Consequently, wrench feedback modifies the reference only along $\mathbf{n}_i$, while the nominal ProDMP governs the orthogonal subspace.

IV. Experiments

We evaluate GIPD on BoxFlip, MCB switch toggling, and LEGO insertion, assessing visual timing prediction, nominal task success, and robustness under task-specific challenging conditions. The tasks are shown in Fig. 3.

TABLE I
TASK-SPECIFIC INTERACTION-DIRECTION LABELS.

Task	Interaction-direction label
BoxFlip	Contact line $\mathbf{s}$, constructed from reaction orthogonal to motion and signed toward the contact surface.
MCB	Motion line $\mathbf{m}$, signed from the initial state toward the toggled state.
LEGO	Insertion line $\mathbf{m}$, signed toward the seated state.

A. Experimental Setup

Platform and demonstrations.. Experiments use a Franka Emika FR3 and Intel RealSense D405 RGB camera, two cameras wrist-mounted for LEGO and one fixed for MCB and BoxFlip. Kinesthetic demonstrations are collected in gravity-compensation mode. The FR3 estimates the cartesian wrench from seven torque sensors, one at each joint. We calibrate the payload to reduce gravity-induced bias and spurious cross-axis components. RGB, end-effector pose, and base-frame wrench are recorded [28], [29]. All methods execute through the same Cartesian impedance controller.

Curriculum and training data.. Initial kinesthetic episodes are recorded in ACP format and used to train ACP. Each is fit with a geometric ProDMP and replayed on the robot using its pose, interaction direction, and stiffness; the latter two are computed offline. Time intervals immediately before replay failure are marked as requiring admittance. Later rollouts use these labels for directional correction and progress from Failed to Nice Try to Successful. Frozen stage models propagate admittance labels; uncertain and post-failure frames are omitted. Four early BoxFlip boundaries are checked in RGB. Final frames from the ACP episodes seed completion labels, which are extended to successful rollouts. DP and GIPD train on this curriculum, while ACP uses the original demonstrations.

B. Evaluation Protocol

Each method is evaluated in 20 nominal trials per task. The same trained policies are then evaluated without retraining in 10 trials per task. BoxFlip uses a visually unchanged box nearly five times its training mass, MCB uses a slippery tool tip, and LEGO begins from a diagonal alignment. The first two conditions alter the contact response, whereas the LEGO condition tests geometric recovery before constrained insertion. Physical completion is the robot metric. Separately, we report episode-grouped agreement of the timing predictors with curriculum-generated supervision. Complete rollouts, rather than neighboring frames, define the split groups.

C. Compared Methods

The comparison evaluates each complete training-and-execution pipeline. All methods use the same robot, camera arrangement, and low-level controller, with the training-data paths specified in the experimental setup.

1) **Diffusion Policy (DP).** Trained on the curriculum trajectory dataset, the standard U-Net-based robomimic [30] implementation predicts Cartesian action chunks from RGB and proprioception, without wrench input,

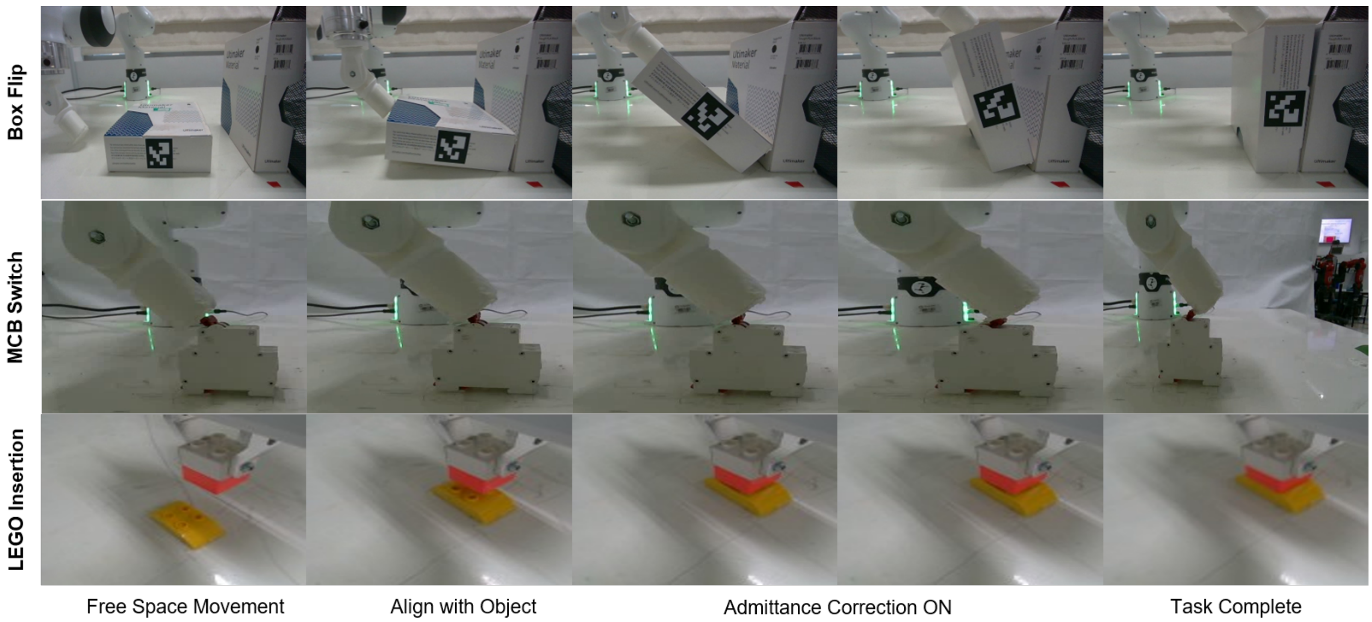


Fig. 3. Real-robot contact-rich tasks used for evaluation, shown left to right over task progress. The top row shows BoxFlip, which requires establishing and maintaining surface contact while reorienting the box. The middle row shows MCB switch toggling, which requires engaging a small lever and traversing its resistive transition without losing tip contact. The bottom row shows LEGO insertion, which requires alignment, contact establishment, and sustained axial progress until the blocks are fully seated.

geometric primitive decoding, or execution-time force coupling.

2) **Adaptive Compliance Policy (ACP).** The authors' released implementation [1] is trained on the initial ACP-format demonstrations and predicts motion together with compliance from visual, proprioceptive, and wrench histories.
3) **GIPD (ours).** Trained on the same curriculum trajectories as DP, GIPD predicts a geometric primitive from RGB and proprioception and uses measured wrench only in the gated executor, outside neural inference.

ACP implementation and 1-kHz force encoding.. We retain the released ACP architecture but adapt its input from the original 7-kHz ATI sensor to the Franka's 1-kHz estimated TCP wrench. A 73-ms STFT window and 55-ms hop preserve its $6{\times}30{\times}18$ encoder input; training-set percentiles normalize the six wrench-axis maps, and stiffness thresholds are recalibrated to the Cartesian impedance controller. Reported ACP results use a fixed safety filter that rejects action chunks whose predicted increment exceeds $0.524\,\mathrm{rad}$ in orientation or $0.03\,\mathrm{m}$ in position. Rejected chunks are not sent to the controller, the preceding command remains active until the next policy query.

D. Curriculum and Timing-Prediction Results

The two visual targets have distinct execution roles.The admittance-required prediction enables or disables directional correction, whereas the latched completion prediction terminates task execution. Table II reports episode-grouped agreement with the curriculum targets.

Agreement is high for both targets on BoxFlip and MCB and for LEGO completion. LEGO admittance onset is substantially harder to infer from RGB since neighboring pre-insertion and early-contact configurations can appear similar even when occluded contact determines whether correction is required. Thus, visual admittance timing remains a limitation for LEGO despite reliable completion prediction.

TABLE II
EPISODE-GROUPED AGREEMENT WITH CURRICULUM SUPERVISION. F1 IS REPORTED FOR THE POSITIVE CLASS; BACC. DENOTES BALANCED ACCURACY

	Adm.-required		Task complete	
Task	F1	BAcc.	F1	BAcc.
BoxFlip	.967	.964	.982	.990
MCB	.991	.989	.852	.995
LEGO	.522	.768	1.000	1.000

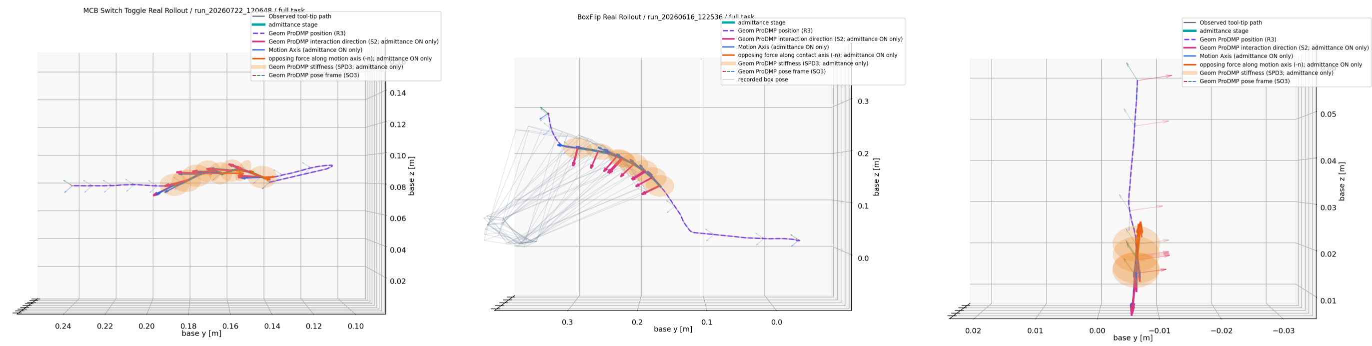
TABLE III
NOMINAL-CONDITION TASK SUCCESS OVER 20 REAL-ROBOT TRIALS.

Method	BoxFlip	MCB	LEGO
DP	$5/20$	$0/20$	$3/20$
ACP	$15/20$	$4/20$	$5/20$
GIPD (ours)	$20/20$	$19/20$	$16/20$

E. Real-Robot Task Success

Table III reports physical task completion under the nominal conditions.

GIPD achieves the best overall performance across the three tasks. On BoxFlip, DP completes only $5/20$ trials, whereas ACP $15/20$ and GIPD achieves $20/20$. The task requires the tool to maintain support against the box while the nominal motion reorients it. Because the visual appearance does not reveal whether sufficient support is maintained, DP cannot



(a) MCB switch toggling (b) BoxFlip (c) LEGO insertion

Fig. 4. GIPD policy predicted rollouts for (a) MCB switch toggling, (b) box flipping, and (c) LEGO insertion. The trajectories show the predicted tool-pose frames, stiffness ellipsoids, motion and contact axes, and interaction directions. Directional quantities are shown only while admittance is predicted to be active, as indicated by the blue/teal trajectory segments. The interaction direction is shown in pink, with the opposing force along this direction in light orange. During interaction, notice that the stiffness ellipsoid predicted by the policy shrinks along the interaction direction in which the admittance correction is applied, while remaining stiff in the orthogonal directions to rigidly track the predicted pose.

react when contact weakens and recorded rollouts frequently lose engagement before the flip is complete. ACP addresses this sustained-contact behavior through its learned compliance. GIPD instead selects the contact axis s as the interaction direction such that the measured reaction drives a bounded correction that restores support, while the nominal primitive continues the surface motion and governs the orthogonal subspace.

The advantage is larger on MCB toggling, where GIPD achieves $19/20$, compared with $4/20$ for ACP and $0/20$ for DP. This task combines a small contact region with a resistive switch transition. DP has no physical signal indicating whether the tool tip has remained engaged and therefore exhibits contact loss or incomplete actuation. ACP uses wrench observations, but its unsuccessful rollouts frequently fail to overcome the switch resistance while still maintaining contact jointly predicting motion and compliance. In GIPD, the visually gated executor projects the measured resistance onto the predicted motion axis and modifies progress only along that direction. The orthogonal directions remain governed by the predicted pose and stiffness, helping the tool preserve alignment while traversing the switch resistance.

For LEGO insertion, GIPD completes $16/20$ trials, while DP and ACP complete $5/20$ and $3/20$ respectively. Small pose errors at contact generate lateral loading that can stall or jam the insertion. DP cannot use this contact response to modify its rollout, and recorded ACP failures include lateral loading and jamming. GIPD assigns the insertion axis as the interaction direction, lowers stiffness and applies correction along this axis, and retains stiff pose tracking in the orthogonal directions. As mentioned in the section IV-D, LEGO has the weakest admittance-timing agreement in Table II, its $16/20$ physical success indicates that disagreement around the precise onset does translate into task failure if the interaction is not engaged accurately. To reduce this we used contrasting colored blocks to improve visual estimation. However, occluded contact onset remains as a remaining limitation.

Figure 4 visualizes the synchronized pose, candidate axes, stiffness, and active correction intervals for one rollout testing GIPD policy predictions from each task.

TABLE IV
CHALLENGING-CONDITION TASK SUCCESS OVER 10 REAL-ROBOT TRIALS.

Method	BoxFlip	MCB	LEGO
DP	$0/10$	$0/10$	$0/10$
ACP	$7/10$	$0/10$	$0/10$
GIPD (ours)	$10/10$	$7/10$	$5/10$

F. Challenging-Condition Results

Table IV evaluates each fixed policy without retraining under the task-specific conditions described in section IV-B.

The nearly fivefold increase in box mass changes the contact response without providing a corresponding visual cue. Under this shift, DP falls to $0/10$ and ACP achieves $7/10$, whereas GIPD retains $10/10$ success. The fixed GIPD policy does not need to infer the new mass from RGB, the directional admittance component measures the changed reaction along the predicted contact axis and adjusts support while leaving the nominal flipping motion unchanged. This separation between visual motion generation and execution-time wrench response explains why the same primitive can remain effective under the heavier payload.

Reducing the MCB tool-tip friction directly weakens the narrow contact needed to actuate the switch. DP and ACP fail all 10 trials, whereas GIPD completes $7/10$. The result is consistent with the directional executor using measured resistance to regulate progress through the switch transition rather than relying on vision alone. At the same time, the reduction from $19/20$ nominal success to $7/10$ shows the limit of a rank-one correction: when low friction produces slip outside the selected interaction direction, axis-aligned admittance cannot fully recover the lost contact.

The diagonal LEGO initialization tests a different form of generalization. It changes the initial geometry before constrained insertion, rather than only the contact response. DP and ACP achieve $0/10$, while GIPD succeeds in $5/10$ trials.

In the successful GIPD rollouts, the visual geometric primitive must first recover the diagonal alignment and then once contact is established can the directional executor support axial progress. The result therefore reflects the complete pipeline rather than the one-dimensional force correction alone. The remaining failures also show that recovery from unseen initial geometry remains more difficult than adaptation to altered contact dynamics.

Taken together, these challenging-condition results show that a fixed GIPD policy can respond to substantial changes in contact dynamics and can retain partial capability under an initial-geometry shift.

V. Conclusion

We presented GIPD, which uses a geometric ProDMP decoder to jointly represent pose, SPD stiffness, and signed interaction direction. Vision and proprioception generate the nominal primitive, while measured wrench provides bounded correction only along the predicted direction. A failure-driven rollout curriculum supervises admittance activation and task completion without frame-by-frame annotation. Real-robot results show improved task success over visual and compliance-learning baselines and robustness to changed contact dynamics. Future work will integrate GIPD into the diffusion action heads of vision-language-action models and into task and motion planning frameworks, where the policy learns task actions and the classifiers learn their preconditions and postconditions.

References

[1] 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," *arXiv preprint arXiv:2410.09309*, 2024.

[2] 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.

[3] 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, accepted June 2026; arXiv:2512.10946.

[4] Y. Yang, A. Chen, Z. Zhu, K. Xu, Y. Mao, Y. Wei, L. Chen, R. Xiong, and Y. Wang, "Direction matters: Learning force direction enables sim-to-real contact-rich manipulation," 2026.

[5] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, pp. 1133 – 1184, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231846758

[6] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[7] F. J. Abu-Dakka and V. Kyrki, "Geometry-aware dynamic movement primitives," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4421–4426, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212717906

[8] A. Kramberger, E. Shahriari, A. Gams, B. Nemec, A. Ude, and S. Haddadin, "Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6023–6028.

[9] W. Amanhoud, M. Khoramshahi, and A. G. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Robotics: Science and Systems XV*, 2019.

[10] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. G. Billard, "Force adaptation in contact tasks with dynamical systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6841–6847.

[11] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *IEEE Robotics and Automation Letters*, 2024.

[12] I. Iturrate, C. Sloth, A. Kramberger, H. G. Petersen, E. H. Østergaard, and T. R. Savarimuthu, "Towards reversible dynamic movement primitives," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5063–5070.

[13] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, pp. 199–217, 2015.

[14] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Robotics: Science and Systems*, 2019.

[15] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. Billard, "Force adaptation in contact tasks with dynamical systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6841–6847.

[16] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 136–142.

[17] P. W. L"odige, M. X. Li, and R. Lioutikov, "Use the force, bot! – force-aware prodmp with event-based replanning," *arXiv preprint arXiv:2409.11144*, 2024.

[18] J. Jang, Y. Sung, W. Joung, D. Chae, S. Lee, S. Kim, J. Kim, and J. Lee, "AxisGuide: Grounding robot action coordinate system in RGB observations for robust visuomotor manipulation," in *Robotics: Science and Systems (RSS)*, 2026, arXiv:2606.06761.

[19] S. Haddadin and E. Shahriari, "Unified force-impedance control," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2112–2141, 2024.

[20] K. Karacan, R. J. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin, "Tactile robot programming: Transferring task constraints into constraint-based unified force-impedance control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 204–210.

[21] C. Bilaloglu, T. L"ow, and S. Calinon, "Tactile ergodic coverage on curved surfaces," *IEEE Transactions on Robotics*, vol. 41, pp. 1421–1435, 2025, arXiv:2402.04862.

[22] 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*, vol. 72, pp. 11 520–11 529, 2025.

[23] D. Zhu, Y. Li, Y. Li, and S. Guo, "PGDP: Physics-guided diffusion policy for multi-rate control in contact-rich manipulation," *IEEE Robotics and Automation Letters*, vol. 11, no. 7, pp. 8833–8840, 2026.

[24] S. Zhang, Y. Yuan, H. Tang, Y. Zheng, Q. Yu, P. Li, G. Huang, H. Huang, X. Quan, and J. Hao, "ForceFlow: Learning to feel and act via contact-driven flow matching," 2026.

[25] K. Y. Ma, H. Zhang, W. Lin, M. Z. Shou, and Y. Wu, "Semantic-contact fields for category-level generalizable tactile tool manipulation," 2026.

[26] N. Geiger, T. Asfour, N. Hogan, and J. Lachner, "Diffusion-based impedance learning for contact-rich manipulation tasks," 2026.

[27] M. Okada, M. Komatsu, and T. Taniguchi, "A contact model based on denoising diffusion to learn variable impedance control for contact-rich manipulation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 7286–7293, arXiv:2403.13221.

[28] Franka Robotics, "libfranka," https://github.com/frankarobotics/libfranka, 2018–2026.

[29] J. Frisch and contributors, "franka_trajectory_recorder," https://github.com/janfrischi/franka_trajectory_recorder, 2024–2026.

[30] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning (CoRL)*, 2021.