Contact-Grounded Policy: Dexterous Visuotactile Policy with Generative Contact Grounding

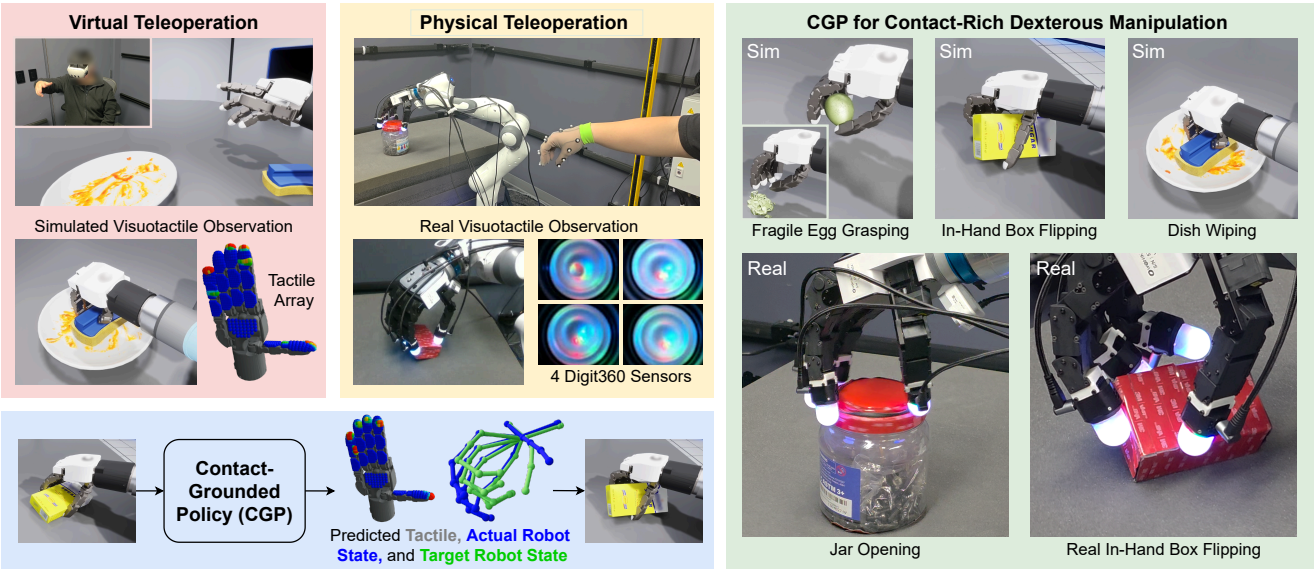

Figure 1. Contact-Grounded Policy (CGP) enables fine-grained, contact-rich dexterous manipulation by grounding multi-point contacts through predicting the actual robot state and tactile feedback, and by using a learned contact-consistency mapping to convert these predictions into controller-executable targets for a compliance controller. CGP supports both dense tactile arrays and vision-based tactile sensors mounted on the hand. We collect demonstrations via teleoperation in both simulation and on a physical robot, and evaluate CGP across multiple dexterous manipulation tasks.

Abstract

Contact-rich dexterous manipulation with multi-finger hands remains an open challenge in robotics because task success depends on multi-point contacts that continuously evolve and are highly sensitive to object geometry, frictional transitions, and slip. Recently, tactile-informed manipulation policies have shown promise. However, most use tactile signals as additional observations rather than modeling contact state or how their action outputs interact with low-level controller dynamics. We present Contact-Grounded Policy (CGP), a visuotactile policy that grounds multi-point contacts by predicting coupled trajectories of actual robot state and tactile feedback, and using a learned contact-consistency mapping to convert these predictions into executable target robot states for a compliance controller. CGP consists of two components: (i) a conditional diffusion model that forecasts future robot state and tactile feedback in a compressed latent space, and (ii) a learned contact-consistency mapping that converts the predicted robot state-tactile pair into executable targets for a compliance controller, enabling it to realize the intended contacts. We evaluate CGP using a physical four-finger Allegro V5 hand with Digit360 fingertip tactile sensors, and a simulated five-finger Tesollo DG-5F hand with dense whole-hand tactile arrays. Across a range of dexterous tasks including in-hand manipulation, delicate grasping, and tool use, CGP outperforms visuomotor and visuotactile diffusion-policy baselines.

1. Introduction

Dexterous manipulation with multi-finger robotic hands remains one of the most challenging frontiers in robotics. Unlike rigid end-effectors or simple grippers, dexterous manip-

ulation with hands requires the robot to continuously regulate rich, high-dimensional contact interactions between multiple fingers and the object. These interactions are distributed across multiple contact points and evolve rapidly as the hand and object move. They are highly nonlinear and partially observable, and their outcomes are sensitive to contact geometry, frictional state transitions, and slip.

Existing learning approaches to dexterous manipulation generally fall into three primary paradigms. First, grasp-centric pipelines focus on generating secure grasp configurations [6, 31, 34, 37]. These grasp-centric pipelines are effective for rigid pick-and-place tasks. However, once a grasp is established, they typically constrain subsequent finger motion, limiting their applicability to contact-rich behaviors that require continuous reconfiguration and active contact regulation, such as in-hand manipulation and tool use. Second, reinforcement learning (RL) can discover complex contact strategies [14, 17, 21, 35]. However, RL policies often face major hurdles in sim-to-real transfer, particularly with visual and tactile observations, and frequently require extensive reward engineering to achieve delicate manipulation on new tasks.

Imitation learning from human demonstrations is well-suited to contact-rich dexterous manipulation because humans naturally regulate distributed, time-varying contacts through coordinated finger motion and compliant interaction. It avoids complex reward design, scales with additional demonstrations, and can generalize across objects and tasks when the dataset covers diverse contact patterns. Motivated by these advantages, recent imitation-learning methods have shown impressive performance by directly leveraging human demonstrations [9, 29, 33]. However, many proposed visuomotor policies still struggle with complex, contact-rich dexterous manipulation [1]. A core limitation is that they primarily predict kinematic trajectories without explicit contact semantics, making it difficult to represent and control evolving multi-point contacts.

Addressing this limitation requires policies to go beyond kinematics-only prediction. First, they must leverage multimodal perception, reasoning over tactile feedback alongside vision, to reduce ambiguity under partial observability. Second, policies must go beyond using tactile signals as additional observations and instead model contact state and how action outputs interact with low-level controller dynamics. Concretely, the policy should produce controller-compatible targets that reflect contact evolution and can be faithfully executed by the compliance controller during deployment. Otherwise, learned outputs can become physically inconsistent with the environment, leading to slip, overly stiff interactions, and ultimately unreliable execution. Bridging high-level task intent and low-level contact-rich control therefore requires deploying contact predictions through the compliance controller.

In this paper, we introduce **Contact-Grounded Policy (CGP)**, a supervised policy learning framework for dexterous visuotactile manipulation, as shown in Fig. 1. CGP casts dexterous manipulation as a contact-grounding problem: rather than using tactile signals only as additional observations, CGP grounds evolving multi-point contacts by predicting coupled trajectories of the actual robot state and tactile feedback, and by using a learned contact-consistency mapping to convert these predictions into controller-executable target robot states that account for the dynamics of the compliance controller. This representation produces concrete, contact-consistent targets that enable real-time execution to realize the intended contact evolution.

Our contributions are as follows:

1. Contact-Grounded Policy Framework. We introduce Contact-Grounded Policy (CGP), a visuotactile policy that grounds evolving multi-point contacts for dexterous manipulation. CGP does so by predicting coupled trajectories of the actual robot kinematic state and tactile feedback, and then translating these predictions into executable control targets. Specifically, a learned contact-consistency mapping converts each predicted state-tactile pair into a target robot state that accounts for the compliance controller, enabling real-time realization of the intended contact evolution. Across a range of dexterous tasks including in-hand manipulation, delicate grasping, and tool use, CGP outperforms visuomotor and visuotactile diffusion-policy baselines.

2. Efficient Tactile Prediction for Contact Grounding. We utilize a latent tactile prediction model integrated into CGP that delivers high-fidelity tactile forecasts. For efficiency, we compress tactile observations with a KL-regularized variational autoencoder (VAE) and perform prediction in the resulting compact latent space. This design captures diverse contacts while remaining lightweight at runtime, and we demonstrate its effectiveness across both tactile arrays and vision-based tactile sensors. Ablations show that the KL-regularized latent space stabilizes generation and improves downstream policy performance.

2. Related Work

Tactile-informed policy learning has become a prominent paradigm for contact-rich manipulation [12, 18, 32, 33, 38]. A growing body of work goes beyond direct action regression by introducing tactile/force prediction objectives to shape representations, providing richer supervision when action labels are low-dimensional or ambiguous [9, 13, 19]. However, most approaches treat tactile signals as additional observations or auxiliary prediction targets, rather than modeling contact state or how action outputs interact with low-level controller dynamics.

Predicting dense tactile signals alone does not guarantee

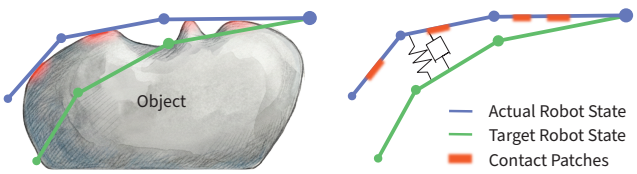

Figure 2. Schematic of contact grounding using a 3-DoF revolute finger, illustrating the actual robot state, target robot state, and the resulting contact patches. We assume that each joint is controlled by a low-level proportional-derivative (PD) controller, which can be viewed as a virtual spring-damper that maps the tracking error between target and actual joint angles to motor torques, enabling compliant motion.

that the intended contacts will be physically realized during rollout [9, 13, 19]. In many approaches, tactile prediction is only loosely coupled to the control stack; as a result, predicted contact patterns may not be reproduced by the commands executed by the low-level controller. This can yield disconnected contact awareness: models anticipate tactile patterns yet fail to ground them into controller references.

Related work moves toward more executable contact representations by predicting control-relevant interaction targets, such as adaptive compliance references, rather than treating tactile or force prediction as an isolated modeling objective [5, 11, 30]. However, many such formulations rely on simplified contact abstractions (e.g., a single arm end-effector force vector or a single force vector per gripper finger), which limits their extensibility to distributed contact points across a multi-finger hand. Similarly, [3, 36] predict sparse fingertip forces executed by a low-level controller, whereas real dexterous interaction involves dense, multi-patch contacts across the hand with high-DoF finger motion.

3. Method

This section presents CGP, a supervised policy learning framework for dexterous visuotactile manipulation that casts dexterous manipulation as contact grounding. CGP grounds high-level task intent into contact-consistent control targets by predicting (i) future actual robot states, (ii) the corresponding expected tactile feedback, and (iii) executable target robot states for the compliance controller. Coupled prediction produces a structured description of the intended contact evolution.

3.1. Problem Setup

We consider an arm-hand manipulation system equipped with tactile sensing on the hand (e.g., fingertip or full-hand coverage). At each time step t, the robot receives multi-modal observations $\mathbf{o}_t = \{\mathbf{i}_t, \mathbf{u}_t, \mathbf{x}_t\}$. Here, the vision observation $\mathbf{i}_t$ stacks N_i RGB views (e.g., agent-view and wrist-view) and is represented as $\mathbf{i}_t \in \mathbb{R}^{N_i \times H_i \times W_i \times 3}$. The tactile observation $\mathbf{u}_t$ can take two forms: (i) tactile RGB images from N_u sensors, represented as $\mathbf{u}_t \in \mathbb{R}^{N_u \times H_u \times W_u \times 3}$, or (ii) tactile arrays with N_{tac} tactile units and per-unit feature dimension d_{tac}, represented as $\mathbf{u}_t \in \mathbb{R}^{N_{\text{tac}} \times d_{\text{tac}}}$. The robot state $\mathbf{x}_t$ consists of the end-effector pose and hand joint angles, where the end-effector translation is in $\mathbb{R}^3$ and the rotation is parameterized by a unit quaternion in $\mathbb{R}^4$; overall, we represent $\mathbf{x}_t \in \mathbb{R}^{(3+4+n_h)}$, where n_h is the number of hand joints.

The policy interfaces with the robot through a low-level tracking controller by outputting a target robot state $\mathbf{a}_t$, which serves as the controller reference. For example, $\mathbf{a}_t$ specifies an operational space impedance controller setpoint for the arm and joint space targets for the hand proportional-derivative (PD) controller. We parameterize the target translation in $\mathbb{R}^3$ and represent the target rotation with the 6D continuous rotation representation in $\mathbb{R}^6$ (rot6D), which facilitates stable action regression [4, 39]; thus, $\mathbf{a}_t \in \mathbb{R}^{(3+6+n_h)}$. The low-level controller converts the target state $\mathbf{a}_t$ into commanded joint torques.

3.2. Contact Grounding

Contact-rich dexterous manipulation involves distributed, time-varying contacts over a high-DoF hand, so modeling contact via designed parameterizations (e.g., contact locations, modes, or fixed contact points) often fails to capture the full variability and combinatorial complexity of evolving multi-point contact. CGP instead represents contact through coupled state and tactile signals that are directly measurable and can be mapped to controller-executable targets.

A key observation is that, under a specific tactile sensor and compliance controller setup, contact can be represented by the triplet $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{a}_t)$ without explicitly specifying contact locations or modes, where $\mathbf{x}_t$ is the actual robot state, $\mathbf{u}_t$ is tactile feedback, and $\mathbf{a}_t$ is the compliance controller reference (i.e., the target robot state). Under closed-loop tracking, interaction manifests as a discrepancy between $\mathbf{a}_t$ and $\mathbf{x}_t$, while the resulting contact outcome is reflected in $\mathbf{u}_t$ (Fig. 2). This representation naturally adapts to tactile coverage: full-hand sensing captures distributed, multi-patch contacts, whereas dense fingertip tactile sensing (e.g., vision-based tactile images) can be sufficient when interaction is primarily localized to the fingertips.

Motivated by this coupling, we learn a controller-specific contact-consistency mapping: given $(\mathbf{x}_t, \mathbf{u}_t)$, it predicts a target state $\mathbf{a}_t$ whose tracking would reproduce the observed interaction under the compliance controller

$$\mathbf{a}_t = \mathcal{M}_\phi(\mathbf{x}_t, \mathbf{u}_t). \tag{1}$$

Eq. (1) avoids explicitly modeling contact locations/modes or system dynamics; instead, it learns a compact, implicit, and setup-dependent mapping. Because this mapping is tied

to a specific embodiment, sensing configuration, and compliance controller dynamics, it can be learned in a purely data-driven manner, while remaining flexible to distributed, evolving multi-point contacts that are difficult to capture with hand-designed contact parameterizations or explicit dynamics models.

3.3. Policy Pipeline Overview

3.3.1. Architecture

Fig. 3 summarizes the CGP pipeline. We denote a short history of observations by $\mathbf{O}_t = \{\mathbf{o}_{t-T_o+1}, \ldots, \mathbf{o}_t\}$. Here, T_o denotes the observation horizon. CGP consists of two coupled components that implement contact grounding in a structured manner. The first component is a contact-consistency mapping $\mathcal{M}_\phi$, which infers the controller reference consistent with the current interaction, as shown in Eq. (1). The second component is a conditional trajectory generator π_θ that predicts how the interaction should evolve over a future horizon. Conditioned on $\mathbf{O}_t$, it samples coupled trajectories of future actual robot state and tactile feedback,

$$(\hat{\mathbf{X}}_t, \hat{\mathbf{U}}_t) \sim \pi_\theta(\cdot \mid \mathbf{O}_t),$$

where $\hat{\mathbf{X}}_t = \{\hat{\mathbf{x}}_{t+1}, \ldots, \hat{\mathbf{x}}_{t+T}\}$ and $\hat{\mathbf{U}}_t = \{\hat{\mathbf{u}}_{t+1}, \ldots, \hat{\mathbf{u}}_{t+T}\}$. Here, T denotes the prediction horizon. In our implementation, π_θ follows a diffusion-policy-style design and is parameterized by a U-Net denoiser [4].

This factorization is intentional. Rather than directly mapping observations to controller references, CGP first predicts interaction targets in future actual robot state and tactile feedback that describe the desired contact evolution, and then converts them into executable target robot states via $\mathcal{M}_\phi$. This separation allows the trajectory generator to model contact evolution as paired state-tactile trajectories, while the mapping realizes the intended contacts by producing low-level controller references.

3.3.2. Residual Mapping

We formulate the contact-consistency mapping $\mathcal{M}_\phi$ in residual form. Rather than regressing the target state directly, it outputs an offset from the current actual state, which anchors learning, improves conditioning, and yields more robust targets under the compliance controller.

3.3.3. Inference and Execution

At test time, CGP samples future actual robot state and tactile feedback trajectories from π_θ and converts them into target robot states step-wise,

$$\hat{\mathbf{a}}_{t+k} = \mathcal{M}_\phi(\hat{\mathbf{x}}_{t+k}, \hat{\mathbf{u}}_{t+k}), \quad k = 1, \ldots, T.$$

The compliance controller tracks the next target robot state, and the policy replans in a receding-horizon manner.

3.4. Latent Tactile Generation

Raw tactile observations for dexterous hands can be high-dimensional, especially for vision-based tactile sensors and dense tactile arrays, making long-horizon generation computationally expensive. To enable efficient real-time generation while preserving contact variability, CGP generates tactile trajectories in a learned latent space.

3.4.1. Tactile Compression with VAE

We train a variational autoencoder (VAE) [15] to compress raw tactile observations $\mathbf{u}_t$ into a compact latent representation, $\mathbf{h}_t = E(\mathbf{u}_t), \hat{\mathbf{u}}_t = G(\mathbf{h}_t)$, where E and G denote the encoder and decoder, respectively. A compact, well-behaved latent space is critical for stable and efficient diffusion-based generation; therefore, we use KL regularization to encourage learning such a compact latent representation [23]. The VAE backbone can be adapted to different tactile modalities (e.g., tactile arrays or tactile images) while sharing a common training objective and pipeline. In all cases, tactile observations are mapped to a fixed-dimensional latent $\mathbf{h}_t \in \mathbb{R}^M$.

3.4.2. Coupled Diffusion over State and Tactile Latent

After compression, tactile trajectories are generated in latent space together with actual robot states. We define the coupled future trajectory as $\mathbf{Y}_t = \left[\mathbf{x}_{t+1:t+T},\ \mathbf{h}_{t+1:t+T}\right]$, which concatenates future actual robot states and tactile latent states over the horizon.

We model the conditional distribution of $\mathbf{Y}_t$ given the observation history $\mathbf{O}_t$ using a diffusion model. Following standard DDPM/DDIM training [10, 25], we sample a diffusion step $j \in \{1, \ldots, J\}$ and perturb the clean trajectory $\mathbf{Y}_t^0$ with Gaussian noise, $\mathbf{Y}_t^j = \alpha_j \mathbf{Y}_t^0 + \sigma_j \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\{\alpha_j, \sigma_j\}$ are fixed noise-schedule coefficients. A U-Net denoiser π_θ is trained to predict the injected noise conditioned on $\mathbf{O}_t$,

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{(\mathbf{O}_t, \mathbf{Y}_t^0), \boldsymbol{\epsilon}, j}\left[\|\boldsymbol{\epsilon} - \pi_\theta(\mathbf{O}_t, \mathbf{Y}_t^j, j)\|^2\right].$$

3.5. Implementation Details

This section describes key implementation choices in CGP.

3.5.1. Tactile Encoders and Decoders

CGP employs tactile encoders across multiple components, including diffusion conditioning, contact-consistency mapping, and tactile compression, while decoders are used only for tactile compression. Since these components are trained with different objectives and at different stages, encoder weights are not shared, though all encoders follow the same architectural template.

We use different tactile modalities in simulation and on real hardware: dense tactile arrays in simulation and vision-based tactile sensors (RGB tactile images) on the physical

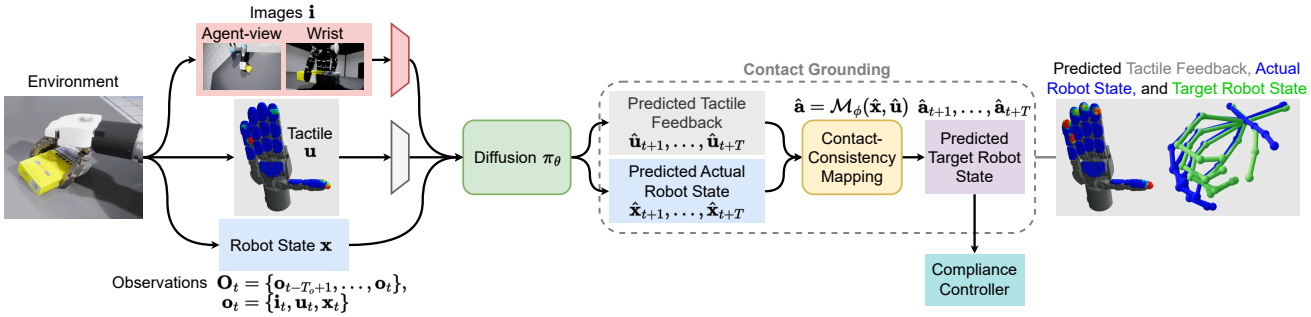

Figure 3. Overview of Contact-Grounded Policy (CGP). CGP grounds multi-point contacts by predicting coupled trajectories of actual robot state and tactile feedback, and using a learned contact-consistency mapping to convert these predictions into executable target robot states for a compliance controller.

system. We validate the proposed pipeline on both sensor types.

In simulation, tactile sensing is provided as a dense tactile array $\mathbf{u}_t \in \mathbb{R}^{N_{\text{tac}} \times d_{\text{tac}}}$ (e.g., d_{tac}=3 for 3D force vectors). We use a 1D ResNet-style architecture that applies convolutions along the tactile-unit dimension N_{tac}, treating d_{tac} as channels. For decoding, the latent code is projected to a compact 1D feature map and progressively upsampled with transposed convolutions and residual refinement, followed by a final convolution to reconstruct tactile features.

On real hardware, tactile sensing is provided by N_u vision-based tactile sensors, each producing an RGB image. We adopt a per-sensor design with shared 2D ResNet-style encoder-decoder weights, processing each tactile image independently to produce consistent fingertip representations. The encoder consists of a convolutional stem followed by strided residual blocks and global pooling, while the decoder mirrors this structure with progressive upsampling and residual refinement. A final convolutional layer reconstructs the RGB tactile image.

3.5.2. Visual Encoder and Diffusion

For visual encoding and diffusion, we follow the architecture of diffusion policy [4], using a U-Net-based conditional diffusion model and DDIM sampling [25] for efficient inference.

In simulation, visual features, tactile features, and the low-dimensional robot state are concatenated and injected into the diffusion U-Net via FiLM conditioning [20]. On real hardware, each tactile image is encoded using the shared tactile encoder (Sec. 3.5.1), and the resulting per-sensor features are aggregated via cross-sensor self-attention. The aggregated tactile feature is then concatenated with the visual feature and robot state, and applied through FiLM conditioning throughout the diffusion backbone.

3.5.3. Contact-Consistency Mapping

The contact-consistency mapping is a lightweight network. For tactile arrays, we prioritize tactile fidelity by decoding the diffusion-predicted tactile latent to raw tactile, re-encoding it, and concatenating the resulting feature with the actual robot state to predict the target state using an MLP. This design preserves detailed contact information for target prediction.

For vision-based tactile sensors, we use a more lightweight design for real-time deployment. Reconstructing tactile images over the prediction horizon is substantially more expensive, so we instead concatenate the diffusion-predicted tactile latent states with the actual robot states and feed them directly to an MLP head. This greatly reduces inference cost while retaining sufficient tactile information for contact-consistent execution.

We use different designs in simulation and on real hardware because the tactile modalities and runtime constraints differ: array decoding is lightweight and preserves high-frequency contact details, whereas image reconstruction is compute-heavy and can become a bottleneck for real-time inference.

4. System Overview

4.1. Simulation Setup

For the simulation environment, we integrate a real-time finite element solver with Unreal Engine via a native plugin. The solver uses reduced-order models with data-driven hyper-reduction to simulate contact among rigid and compliant objects at interactive rates, providing rich per-contact state for dexterous manipulation [2, 24, 26, 41]. Unreal Engine handles rendering and I/O; the physics step is advanced natively each frame. The simulated robot system consists of a UR5 arm under an operational space impedance controller [27, 40] and a Tesollo DG-5F five-finger 20-DoF hand under a joint space PD controller. To capture rich multi-contact interactions, we instrument the inner surfaces

of all fingers and the palm with a dense tactile array of 768 sensing points, each reporting a 3D force vector, yielding tactile observations in $\mathbb{R}^{768\times 3}$. Visual observations are provided by two RGB cameras, including an agent-view camera and a wrist-mounted camera.

Demonstrations are collected via VR teleoperation using a Meta Quest 3 headset. The tracked hand pose is mapped to the robot end-effector pose to define the arm target [8]. For the fingers, we retarget the operator's tracked fingertip positions to the DG-5F by scaling them to its kinematic range, and use RelaxedIK [22, 28] to solve inverse kinematics and produce finger joint targets.

4.2. Real-Robot Setup

On real hardware, we use a Franka Panda arm under an operational space impedance controller [27, 40] and an Allegro V5 four-finger 16-DoF hand under a joint space PD controller. Tactile sensing is provided by four Digit360 sensors [16] mounted on the fingertips. We use tactile images as the tactile modality. Visual sensing follows the same two-view design as in simulation.

Real-robot demonstrations are collected using an OptiTrack motion-capture system and an instrumented glove for hand tracking [7]. We map the tracked human hand pose to the robot end-effector pose to define the arm target. For hand teleoperation, we perform fingertip-based retargeting: we map glove-estimated fingertip positions into the Allegro hand workspace, ignore the human pinky, and solve inverse kinematics with RelaxedIK [22, 28] to obtain finger joint targets.

5. Experiments

We evaluate CGP through three complementary sets of experiments, each targeting a distinct component of the framework. First, we evaluate full policy rollouts on both simulated and real-world contact-rich dexterous manipulation tasks to assess end-to-end performance. Second, we study hand configuration prediction to directly validate the contact-consistency mapping in isolation. In this experiment, we highlight the necessity of several key design choices in the mapping, and provide high-level evidence that contact can be consistently represented through this mapping in a way that generalizes across diverse contact configurations. Finally, we analyze tactile reconstruction and compression to validate the design of the latent tactile representation used for real-time prediction, and quantify how alternative design choices impact policy performance.

5.1. Performance on Contact-Rich Dexterous Manipulation

We evaluate CGP in full closed-loop execution on three simulated tasks and two real-robot tasks involving contact-rich, dexterous manipulation. As baselines, we compare against

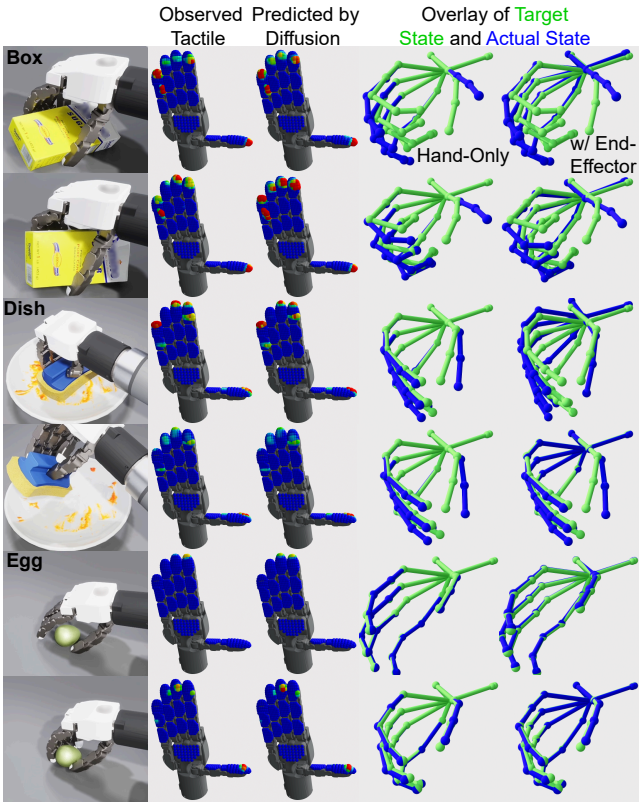

Figure 4. Snapshots of Contact-Grounded Policy (CGP) rollouts on three simulated tasks, showing time-aligned predicted and observed tactile feedback. At each inference step, the diffusion model predicts the next 16 steps of tactile feedback and actual states, which are mapped to target states and executed for 8 steps before the next inference. Predicted tactile is time-aligned with subsequent observations after execution, and the close match indicates that CGP executes contact-grounded targets and realizes the predicted contact evolution. Full rollout videos are provided in the supplementary material.

a standard visuomotor diffusion policy that uses only visual observations, and a visuotactile diffusion policy that follows the same formulation but additionally conditions on tactile observations. To ensure a fair comparison, we implement both visuomotor and visuotactile diffusion-policy baselines with identical visual and tactile encoders and the same U-Net backbone as CGP. All tasks require coordinated finger motion and force modulation while maintaining stable, continuously evolving contacts under the compliance controller. Task details are provided in the appendix and the supplementary video.

Table 1 reports success rates across all tasks. CGP consistently outperforms both visuomotor and visuotactile diffusion-policy baselines. The improvements are pronounced in tasks with sustained or delicate contact, such as dish wiping, in-hand box flipping, and jar opening, where

Method	Sim			Real	
	In-Hand Box Flipping *60 demos*	Fragile Egg Grasping *100 demos*	Dish Wiping *100 demos*	Jar Opening *45 demos*	Real In-Hand Box Flipping *90 demos*
Contact-Grounded Policy	**66.0%**	**74.8%**	**58.4%**	**93.3%**	**80.0%**
Visuotactile DP	58.0%	70.0%	43.6%	66.7%	60.0%
Visuomotor DP	53.2%	53.2%	42.4%	73.3%	60.0%

Table 1. Success rate over 5 challenging contact-rich, dexterous manipulation tasks. DP is short for diffusion policy. For the simulation tasks (left three), we report the mean success rate over the last 5 checkpoints, evaluated with 250 rollouts in total. For the real-world tasks, we report the success rate over 15 consecutive rollouts.

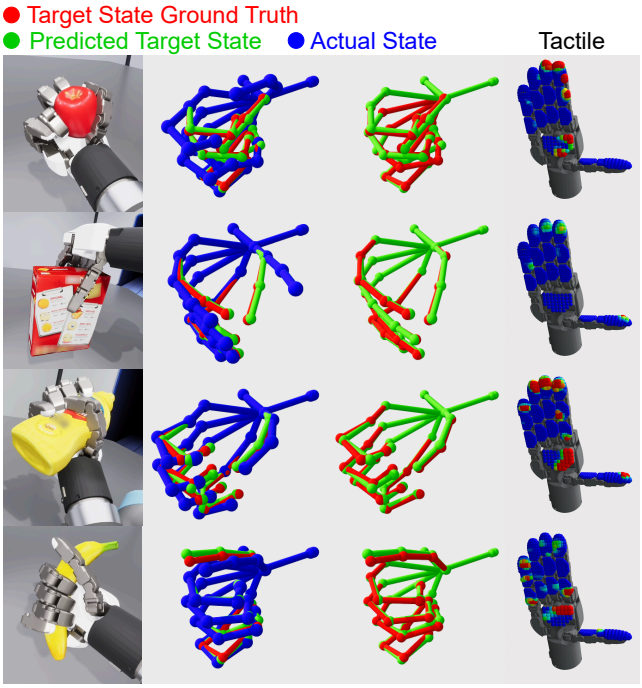

Figure 5. Hand configuration predictions by the contact-consistency mapping for unseen grasps (Section 5.2). This figure provides high-level evidence that contact can be consistently represented through this mapping in a way that generalizes across diverse contact configurations.

Input Modality	Tactile Encoder	Abs. Mode ↓	Residual Mode ↓
State + Tactile	ResNet1D	**8.80** $\pm$ 0.24	**5.94** $\pm$ 0.20
	MLP	12.50 $\pm$ 0.32	8.33 $\pm$ 0.32
	Transformer	14.39 $\pm$ 0.38	9.58 $\pm$ 0.48
State Only	-	16.05 $\pm$ 0.39	10.64 $\pm$ 0.38
Tactile Only	ResNet1D	35.93 $\pm$ 0.89	12.15 $\pm$ 0.20
	MLP	36.86 $\pm$ 0.25	12.72 $\pm$ 0.25
	Transformer	43.11 $\pm$ 0.91	14.62 $\pm$ 0.43

Table 2. Hand configuration prediction results (Section 5.2). MAE denotes mean absolute error. Values are reported in joint-angle space as MAE ($\times 10^{-3}$ rad), shown as mean $\pm$ std over 3 seeds. Model initializations and the train/validation splits are determined by the random seed. "Abs." stands for absolute.

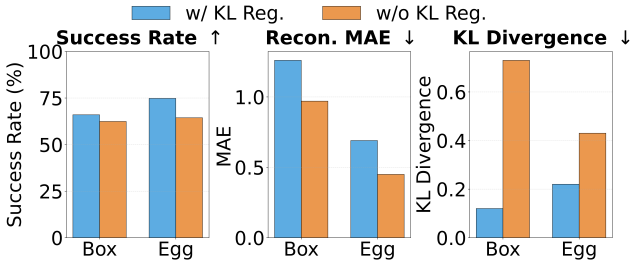

Figure 6. Ablation results of KL regularization for tactile compression. MAE stands for mean absolute error. "Recon." stands for reconstruction.

contact evolution plays a central role.

Fig. 4 visualizes snapshots of rollouts and provides direct evidence of effective contact grounding. At each replanning step, CGP predicts a short-horizon tactile trajectory, maps it into contact-consistent targets, and executes these targets under compliant tracking. To verify that predicted contacts are actually realized during execution, we time-align tactile frames predicted at earlier replanning steps with tactile feedback observed at the corresponding future time steps. The aligned predicted and observed tactile signals exhibit alignment across contact regions and force patterns, despite the stochasticity of multi-contact interactions. This close correspondence indicates that CGP does not merely forecast plausible tactile outcomes, but generates interaction targets that the robot can reliably execute to reproduce the predicted contact evolution. Overall, the visualization confirms that tactile prediction in CGP is tightly coupled to execution through contact-consistent targets, rather than being treated as an auxiliary objective decoupled from control.

Moreover, Fig. 4 compares two overlays: aligning only the hand target/actual states (with a fixed hand-skeleton base) versus aligning the end-effector (which determines the hand base). The two overlays are similar for in-hand box flipping and egg grasping, suggesting that most interactions are hand-dominated; in contrast, dish wiping shows a clear target-actual pose offset when the sponge contacts the plate

Method	Box			Egg			Dish		
	MAE↓	Active MAE↓	KL↓	MAE↓	Active MAE↓	KL↓	MAE↓	Active MAE↓	KL↓
MLP (w/ KL)	1.37	12.24	0.21	0.77	6.70	**0.15**	1.86	8.14	**0.15**
MLP (w/o KL)	1.12	9.93	0.59	0.54	4.51	0.40	1.40	6.20	0.42
ResNet1D (w/ KL)	1.26	12.07	**0.12**	0.69	5.95	0.22	1.54	6.80	0.24
ResNet1D (w/o KL)	**0.97**	**9.91**	0.73	**0.45**	**3.92**	0.43	**1.02**	**4.49**	0.45
Transformer (w/ KL)	1.66	14.88	0.19	1.58	10.58	0.28	3.19	13.57	0.21
Transformer (w/o KL)	1.69	14.60	0.97	1.18	9.08	0.61	3.28	13.57	0.53

Table 3. Tactile reconstruction and compression results on the validation set. MAE stands for mean absolute error. MAE averages the per-unit force error over all 768 tactile units, where each unit's force magnitude is computed as the resultant of the 3-axis forces. Active MAE computes the same error but only over tactile units whose ground-truth force is non-zero (i.e., active contact units). MAE and Active MAE are reported in 10^{-2} N.

(Fig. 4, third row), indicating arm-driven force application and demonstrating that our formulation generalizes across both hand and arm-driven contact interactions.

5.2. Hand Configuration Prediction

We design a controlled hand configuration prediction task to validate the effectiveness of the contact-consistency mapping. This experiment evaluates the mapping from the actual robot state and tactile feedback to the target robot state, without involving downstream policy learning, and serves two purposes: (i) to justify the necessity of key design choices in the mapping, and (ii) to provide high-level evidence that contact can be consistently represented through this mapping in a way that generalizes across diverse contact configurations.

We collect 150 teleoperated grasping episodes with 4114 frames across 11 objects in simulation. Object details are provided in the appendix. Each episode corresponds to a unique grasp configuration with diverse contact patterns. To avoid data leakage across correlated frames, we split data at the episode level with an aggressive 1:1 train-validation split.

Table 2 shows that accurate prediction requires both actual robot state and tactile feedback. Removing either modality substantially increases error, supporting the contact-grounding hypothesis. Tactile-only inputs cannot resolve global hand configuration, while state-only inputs miss contact-induced variations. Together, they capture contact-consistent structure that transfers across objects and contact patterns.

We also ablate mapping designs. A ResNet-style tactile encoder outperforms MLP and Transformer variants, and predicting residual targets reduces error compared to absolute prediction, indicating that contact grounding is best modeled as contact-conditioned corrections around the actual state. These results motivate the final design of the contact-consistency mapping used in CGP. Qualitative results in Fig. 5 further illustrate that, under the ResNet encoder and residual formulation, the model can accurately predict complex and previously unseen target robot state when informative tactile feedback is available. This indicates that the learned mapping generalizes beyond memorized grasp poses and captures transferable, contact-consistent structure across diverse interactions.

5.3. Tactile Reconstruction, Compression, and Design Impact

We evaluate tactile reconstruction and compression to validate the latent tactile representation used for prediction and to quantify how key design choices affect downstream policy performance. Using the same demonstration dataset as policy training, we split episodes into train and validation with a 1:1 ratio. We size-match encoder and decoder capacity across methods.

Table 3 shows that ResNet-based encoders achieve strong reconstruction quality across tasks. While removing KL regularization slightly lowers reconstruction error, it produces a poorly structured latent space with much higher KL divergence. This effect carries over to policy learning. Fig. 6 shows that removing KL consistently degrades rollout performance despite better reconstruction metrics. KL regularization yields a more compact and better-conditioned latent space, which is also reflected by the reduced KL loss in Table 3. This improves the stability of diffusion-based prediction [23] and the fidelity of long-horizon tactile forecasts. Overall, these results validate KL-regularized latent tactile representations in CGP for reliable tactile prediction and contact-grounded control.

Additional quantitative and qualitative reconstruction results on real vision-based tactile sensors (Digit360) are provided in the appendix, highlighting both the effectiveness of our design and its applicability across tactile sensor types.

References

[1] Shan An, Ziyu Meng, Chao Tang, Yuning Zhou, Tengyu Liu, Fangqiang Ding, Shufang Zhang, Yao Mu, Ran Song, Wei Zhang, et al. Dexterous manipulation through imitation learning: A survey. *arXiv preprint arXiv:2504.03515*, 2025.

[2] Yue Chang, Peter Yichen Chen, Zhecheng Wang, Maurizio M Chiaramonte, Kevin Carlberg, and Eitan Grinspun. Licrom: Linear-subspace continuous reduced order modeling with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023.

[3] Claire Chen, Zhongchun Yu, Hojung Choi, Mark Cutkosky, and Jeannette Bohg. Dexforce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation. *IEEE Robotics and Automation Letters*, 10(6):6416–6423, 2025. doi: 10.1109/LRA.2025.3568318.

[4] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

[5] Hojung Choi, Yifan Hou, Chuer Pan, Seongheon Hong, Austin Patel, Xiaomeng Xu, Mark R Cutkosky, and Shuran Song. In-the-wild compliant manipulation with umi-ft. *arXiv preprint arXiv:2601.09988*, 2026.

[6] Hao-Shu Fang, Hengxu Yan, Zhenyu Tang, Hongjie Fang, Chenxi Wang, and Cewu Lu. Anydexgrasp: General dexterous grasping for different hands with human-level learning efficiency. *arXiv preprint arXiv:2502.16420*, 2025.

[7] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D Twigg, and Kenrick Kin. Online optical marker-based hand tracking with deep labels. *Acm Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

[8] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[9] Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation. *arXiv preprint arXiv:2506.15953*, 2025.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

[11] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4829–4836, 2025.

[12] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. In *8th Annual Conference on Robot Learning*, 2024.

[13] Zixuan Huang, Huaidian Hou, and Dmitry Berenson. Unified multimodal diffusion forcing for forceful manipulation. *arXiv preprint arXiv:2511.04812*, 2025.

[14] Gagan Khandate, Siqi Shang, Eric T Chang, Tristan Luca Saidi, Yang Liu, Seth Matthew Dennis, Johnson Adams, and Matei Ciocarlie. Sampling-based exploration for reinforcement learning of dexterous manipulation. In *Proceedings of Robotics: Science and Systems*, 2023.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Mike Lambeta, Tingfan Wu, Ali Sengul, Victoria Rose Most, Nolan Black, Kevin Sawyer, Romeo Mercado, Haozhi Qi, Alexander Sohn, Byron Taylor, et al. Digitizing touch with an artificial multimodal fingertip. *arXiv preprint arXiv:2411.02479*, 2024.

[17] Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. In *Conference on Robot Learning*, 2024.

[18] Jason Jingzhou Liu, Yulong Li, Kenneth Shaw, Tony Tao, Ruslan Salakhutdinov, and Deepak Pathak. Factr: Force-attending curriculum training for contact-rich policy learning. In *Proceedings of Robotics: Science and Systems*, 2025.

[19] Zhuoyang Liu, Jiaming Liu, Jiadong Xu, Nuowei Han, Chenyang Gu, Hao Chen, Kaichen Zhou, Renrui Zhang, Kai Chin Hsieh, Kun Wu, et al. Mla: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. *arXiv preprint arXiv:2509.26642*, 2025.

[20] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[21] Haozhi Qi, Brent Yi, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. From simple to complex skills: The case of in-hand object reorientation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[22] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. Relaxedik: Real-time synthesis of accurate and fea-

sible robot arm motion. In *Robotics: Science and Systems*, volume 14, pages 26–30, 2018.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[24] Cristian Romero, Dan Casas, Maurizio Chiaramonte, and Miguel A Otaduy. Learning contact deformations with general collider descriptors. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023.

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *Proceedings of International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=St1giarCHLP`.

[26] Yutian Tao, Maurizio Chiaramonte, and Pablo Fernandez. Interpolated adaptive linear reduced order modeling for deformation dynamics. *arXiv preprint arXiv:2509.25392*, 2025.

[27] UT-Austin-RPL. deoxys_control. GitHub Repository, 2022. URL `https://github.com/UT-Austin-RPL/deoxys_control`.

[28] Yeping Wang, Pragathi Praveena, Daniel Rakita, and Michael Gleicher. Rangedik: An optimization-based robot motion generation method for ranged-goal tasks. *arXiv preprint arXiv:2302.13935*, 2023.

[29] Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation. In *9th Annual Conference on Robot Learning*, 2025. URL `https://openreview.net/forum?id=XrgRvBklWu`.

[30] Xiaomeng Xu, Yifan Hou, Zeyi Liu, and Shuran Song. Compliant residual DAgger: Improving real-world contact-rich manipulation with human corrections. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=cjcm5LYVWm`.

[31] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[32] Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip Glen Crandall, Wan Shou, Dongyi Wang, and Yu She. Unit: Data efficient tactile representation with generalization to unseen objects. *IEEE Robotics and Automation Letters*, 2025.

[33] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems*, 2025.

[34] Jianglong Ye, Keyi Wang, Chengjing Yuan, Ruihan Yang, Yiquan Li, Jiyue Zhu, Yuzhe Qin, Xueyan Zou, and Xiaolong Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation. In *Proceedings of Robotics: Science and Systems*, 2025.

[35] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. In *Proceedings of Robotics: Science and Systems*, 2023.

[36] Di Zhang, Chengbo Yuan, Chuan Wen, Hai Zhang, Junqiao Zhao, and Yang Gao. Kinedex: Learning tactile-informed visuomotor policies via kinesthetic teaching for dexterous manipulation. In *9th Annual Conference on Robot Learning*, 2025. URL `https://openreview.net/forum?id=GKueYvjqSS`.

[37] Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *8th Annual Conference on Robot Learning*, 2024.

[38] Jialiang Zhao, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Edward Adelson. Polytouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 104–110, 2025. doi: 10.1109/ICRA55743.2025.11128816.

[39] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[40] Yifeng Zhu and Abhishek Joshi. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Proceedings of Conference on Robot Learning*, 2022.

[41] Zeshun Zong, Xuan Li, Minchen Li, Maurizio M Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.