

Force-Modulated Dexterity: Unified Compliance for Contact-Rich Manipulation

Ying Yuan^{1,2}, David Held², Jitendra Malik¹ and Haozhi Qi¹

¹Amazon FAR, ²Carnegie Mellon University

<https://yingyuan0414.github.io/fomo-dex/>

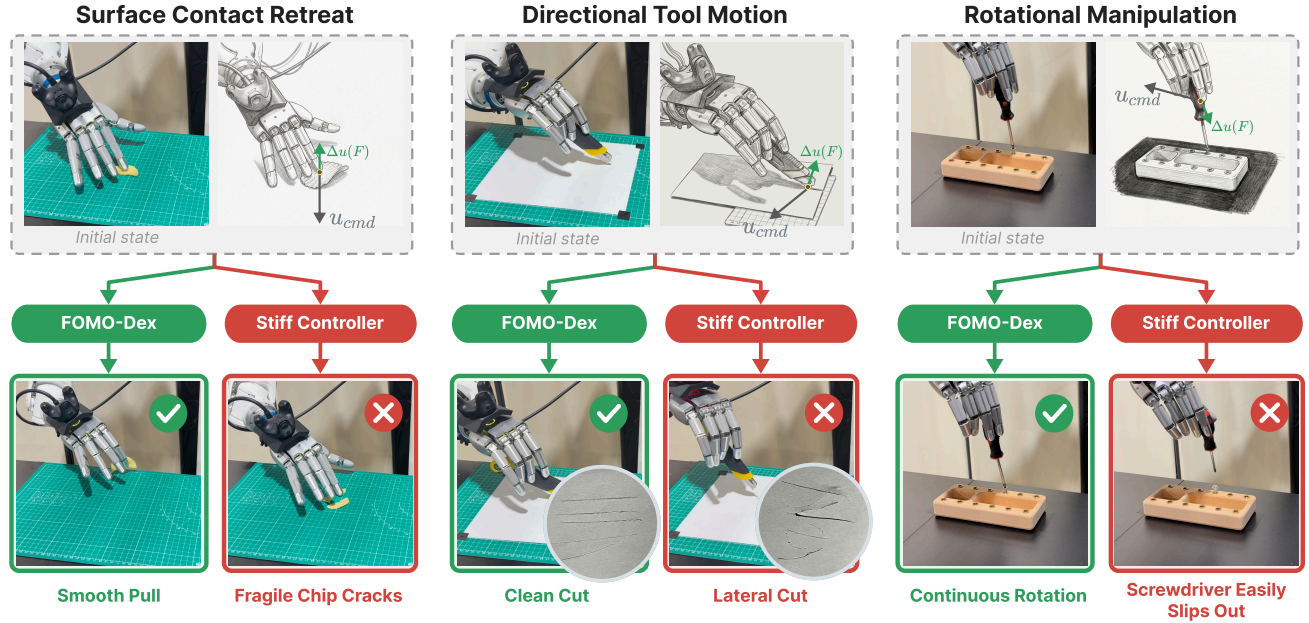


Figure 1: *Force-Modulated Dexterity (FOMO-Dex)* uses force feedback to adjust selected components of a motion command while leaving the others unchanged. The sketches show nominal commands (gray) and force-driven corrections (green). The same principle relieves pressure on a fragile object (left), reduces lateral loading during cutting (middle), and regulates grip during screwdriving (right). The outcomes illustrate how these corrections avoid the object damage and loss of tool control seen with stiff control.

Abstract. Teleoperation is widely used to collect high-quality demonstrations for robot learning. Dexterous manipulation, however, requires coordinated control of motion and contact forces. Operators can specify motion, but struggle to reproduce the subtle force adjustments of human manipulation without feeling the robot’s contacts. We introduce *Force-Modulated Dexterity (FOMO-Dex)*, a unified compliance formulation for contact-rich manipulation. FOMO-Dex expresses nominal actions in coordinates defined by task geometry and contact structure. It applies force-driven motion corrections along selected directions while leaving the remaining command components unchanged. Across card picking, paper cutting, screwdriving, and nut-and-bolt assembly, FOMO-Dex reduces excessive contact forces compared with stiff control. A user study shows that operators collect more successful demonstration data per hour after brief practice. For policy learning, imitating the corrected actions improves card-picking success over imitating raw operator commands. Policies trained on FOMO-Dex demonstrations also achieve more rotation in nut-and-bolt manipulation and screwdriving than those trained on demonstrations collected with the stiff controller.

1. Introduction

Realizing human-like dexterity has long been a central goal of robotics. Recent data-driven approaches offer a promising path toward this goal, but their progress depends on collecting high-quality demonstrations. Teleoperation provides a direct way to collect such data [9, 23, 32], yet collecting reliable contact-rich demonstrations remains difficult. Human manipulation relies on subtle sensorimotor adjustments that are largely automatic and difficult to command explicitly. In teleoperation, the operator cannot easily reproduce these adjustments without directly feeling the robot’s contacts. The resulting errors become especially consequential during contact, where small inaccuracies can produce large and inconsistent forces. Maintaining a stable force or deliberately changing it as the task requires is therefore difficult through motion commands alone.

Compliance provides a natural way to accommodate uncertainty during physical interaction [3, 11, 24]. It has been used to regulate object motion and grasp forces in multi-fingered hands [18, 21] and to learn contact-rich manipulation skills [6, 13]. Applying the same idea to dexterous manipulation, however, is not straightforward. A multi-fingered hand can simultaneously make and break several contacts. These contacts impose different motion constraints and are mechanically coupled through the manipulated object. For a tool, the robot should preserve motion along the task direction while responding to forces in the constrained directions. For a grasp, it should regulate fingertip forces without suppressing the motion that turns the object. Simply making the arm and hand more compliant does not specify how to achieve both. We must determine which action components should respond to contact and which should remain unchanged.

We propose *Force-Modulated Dexterity* (FOMO-Dex), which treats the commanded motion as a nominal action and uses force feedback to modify selected components. We define interaction directions from the task geometry and current contacts, then convert force errors along these directions into motion corrections. Directions excluded from modulation retain the nominal command. The corrections can act through either the arm or the fingers, using wrist force-torque, fingertip tactile, or joint-torque measurements as appropriate for the contact. Because the formulation operates on motion commands, the nominal action can come from either a teleoperator or a learned policy.

The same principle takes a different form in each task. To pick up a thin card, the contacting finger retreats when its load against the supporting surface becomes excessive. During cutting, a utility knife yields to lateral forces while retaining the commanded motion along the cut. In screwdriving and nut-and-bolt manipulation, the controller regulates fingertip forces while leaving the commanded rotation unchanged. The force targets and feedback laws differ, but each realization follows the same principle: task and contact geometry determine where force feedback may change the motion.

Across these four tasks, FOMO-Dex reduces excessive contact forces compared with stiff control. In screwdriving, it also increases mean rotation from 4.2 to 34.7 turns per episode while reducing tool slips (Table 2). A user study further shows that operators collect more successful demonstration data per hour with FOMO-Dex after brief practice (Table 3). These results evaluate compliance not only by the forces it reduces, but also by the manipulation and demonstration collection it enables.

We also study how these demonstrations should be used for policy learning. During teleoperation, we record both the nominal commands and the force-modulated actions sent to the robot. In card picking, directly imitating the corrected actions produces higher lift success than learning raw commands and applying force modulation at deployment. For nut-and-bolt manipulation and screwdriving, policies trained on FOMO-Dex demonstrations achieve more rotation than policies trained on stiff-control demonstrations

(Table 4). Force modulation thus supports both demonstration collection and policy learning.

The contributions of this work are:

- a unified force-modulation interface for dexterous manipulation that adjusts nominal arm and finger commands in directions defined by task geometry and contact structure;
- several realizations of this formulation for unilateral load relief, translational tool compliance, and multi-contact force regulation, demonstrated across four contact-rich dexterous tasks; and
- a teleoperation-to-learning approach that records paired nominal and force-modulated actions and uses the corrected actions to train autonomous visuomotor policies.

2. Related Work

2.1. Dexterous Teleoperation

Recent dexterous teleoperation systems map tracked human motion to high-DoF robot hands [22, 25]. These interfaces provide an intuitive way to command dexterous motion, but motion tracking alone does not regulate contact forces. Small errors can produce excessive forces or unstable contact, and operators may struggle to correct them without directly feeling the robot’s interaction with the environment. To help operators respond to contact, Bunny-VisionPro and DOGlove provide haptic feedback [7, 35]. DexUMI instead collects demonstrations through a wearable hand exoskeleton, allowing the demonstrator to interact directly with objects and feel their response [29].

DexScrew [14] uses simulation-trained finger-motion primitives to assist teleoperation and collect demonstrations for tactile policy learning. Robot-side compliance instead adjusts the response to contact without requiring the operator to make every correction. Zeng et al. [34] adapt joint impedance and feedforward forces while tracking teleoperated hand postures. DexTeleop-0 [19] uses tactile measurements and contact locations to optimize joint-command corrections that regulate fingertip loads and balance the object wrench. FOMO-Dex shares this use of robot-side force feedback. Our focus is a common formulation for fingertip loading and arm-mediated tool contact. Depending on the interaction, selected directions either yield above a force limit or regulate force toward a target, while the remaining command components are unchanged.

2.2. Compliance and Hybrid Position-Force Control

Compliance has a long history in robotic manipulation. Hybrid position-force control separates directions governed by motion and force [24], while impedance and admittance control specify how motion and interaction forces are related [11, 16]. Task-frame formulations express motion and force constraints in coordinates aligned with the interaction geometry [3]. Optimally-Conditioned Hybrid Servoing chooses force- and velocity-controlled directions from a contact model and a desired system motion to improve kinematic conditioning [12].

For multi-fingered hands, object-level impedance control already separates grasp stability from object motion. Li et al. [18] combine grasp and manipulation impedance and learn the latter from demonstrations. Pfanne et al. [21] regulate object pose while optimizing internal forces under friction constraints, including during contact reconfiguration. These methods control a desired object motion through the fingers. FOMO-Dex instead takes nominal arm and finger commands and corrects their response to contact without requiring an object-pose reference.

Minimalist Compliance Control [26] makes task-space admittance available on position-controlled robots by estimating external wrenches from motor signals. It supports dexterous hands as well as arms and

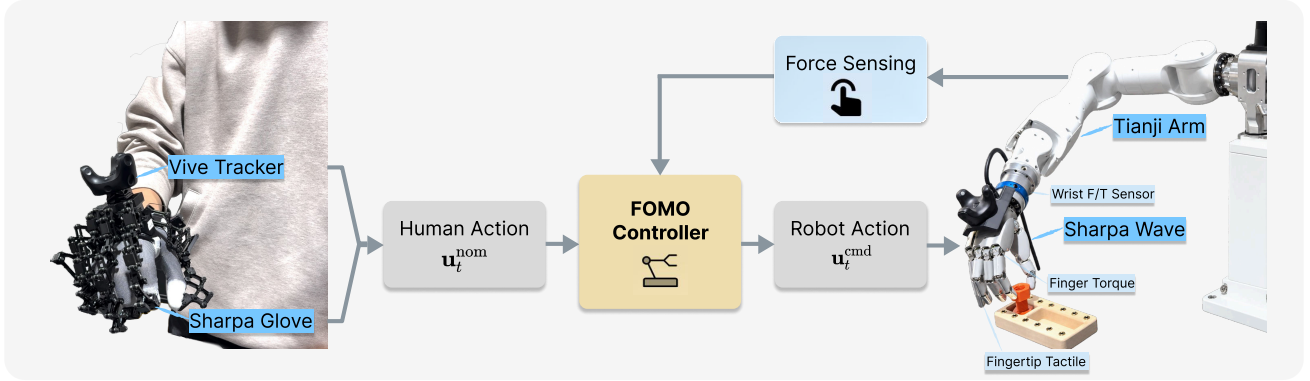


Figure 2: System overview. A teleoperator or learned policy provides nominal arm and finger commands. FOMO-Dex uses feedback from the wrist force-torque sensor and the hand’s tactile and joint-torque sensors to adjust selected motion components before sending the commands to the robot.

can execute geometry-based compliance profiles. FOMO-Dex also builds on established task-frame and admittance-control principles, rather than introducing a new admittance law. We use them to specify task-dependent force corrections during dexterous teleoperation and record the corrected commands as demonstrations for policy learning.

2.3. Learning Compliant Manipulation

Earlier work on variable impedance learning uses human demonstrations to estimate stiffness or impedance profiles and adapt them during execution [1, 2, 36, 37]. More recent diffusion policies incorporate force measurements as observations for contact-rich manipulation [15, 27]. Other approaches also predict compliance parameters. Adaptive Compliance Policy (ACP) learns spatially and temporally varying compliance by jointly predicting motion targets and approximate stiffness [13]. UMI-FT extends this approach to in-the-wild multimodal demonstrations, predicting position targets, grasp force, and stiffness [6]. VIDP jointly predicts robot motion and task-dependent compliance [17].

Beyond compliance gains, Force Policy [8] learns an interaction frame and motion/force selection for hybrid control from demonstrations. Compliant Residual DAgger [30] uses a compliant interface to collect human corrections and learns force-aware residual policies. In FOMO-Dex, the task configuration specifies the force-response directions and feedback laws. They act on nominal commands without requiring a learned compliance or residual model.

Force information can also be obtained directly from human demonstrations. OSMO [33] records normal and shear forces through a tactile glove, while ForceBand [10] estimates fingertip forces from wrist sEMG. Both enrich human demonstrations with force information for policy learning. FOMO-Dex instead uses robot-side force feedback to correct actions during demonstration collection.

Force feedback also affects what actions a policy should imitate. DexForce [4] reconstructs fingertip position targets from measured positions and contact forces in kinesthetic demonstrations, then learns policies that output these force-informed targets. Contact-Grounded Policy [31] predicts robot-state and tactile trajectories and learns a mapping from these predictions to executable controller targets. We share the premise that measured motion alone does not specify the action needed to reproduce contact. Our demonstrations already contain both the nominal commands and the force-modulated commands sent

to the robot. We directly imitate the latter, rather than reconstructing targets from measured states or learning a separate state-to-command mapping.

3. Force-Modulated Dexterity

3.1. Problem Formulation

We consider a robotic arm and a multi-fingered hand teleoperated through human motion tracking. The robot receives nominal arm and finger commands together with force feedback from the wrist and hand. Directly executing these commands during contact can produce excessive force, unstable tool motion, or grasp loss. Our goal is to regulate contact forces toward task-specific targets or limits while retaining the commanded motion in directions that should remain stiff.

3.2. System Setup

Our system consists of a 7-DoF Tianji Marvin arm and a 22-DoF Sharpa Wave dexterous hand (Fig. 2). A six-axis force-torque sensor at the wrist measures arm-level interaction forces and torques. The hand provides fingertip tactile and joint-torque measurements for finger-level feedback. A Sharpa Avatar glove tracks the operator’s finger motion, and an HTC Vive Tracker attached to the glove tracks the hand pose.

3.3. Unified Force-Modulation Controller

At time step t , the nominal command $\mathbf{u}_t^{\text{nom}}$ consists of a Cartesian end-effector pose and a vector of finger-joint targets. The controller computes a force-derived correction $\Delta\mathbf{u}_t$ and combines it with the nominal command to obtain the command sent to the robot,

$$\mathbf{u}_t^{\text{cmd}} = \mathbf{u}_t^{\text{nom}} \oplus \Delta\mathbf{u}_t, \quad (1)$$

where $\oplus$ denotes addition for finger-joint commands and pose composition for Cartesian end-effector commands. The operator provides the nominal motion, and the controller adjusts its response to contact.

To construct $\Delta\mathbf{u}_t$, we first choose interaction directions from the task geometry and current contacts (Section 3.3.1). We project force measurements onto these directions and compare them with the corresponding targets or limits. A direction-wise admittance law converts the resulting force errors into motion corrections (Section 3.3.2). We then map these corrections to arm or finger commands (Section 3.3.3). Figure 3 illustrates this process and its task-specific realizations.

3.3.1. Task-Frame Compliance Representation

At each time step t , we use the task geometry and contact structure to define a set of r unit interaction directions,

$$\mathcal{A}^{(t)} = \{\mathbf{a}_1^{(t)}, \dots, \mathbf{a}_r^{(t)}\}, \quad \|\mathbf{a}_i^{(t)}\|_2 = 1. \quad (2)$$

These directions identify the motion components relevant to the interaction, following task-frame motion decomposition [3, 24]. Each direction $\mathbf{a}_i^{(t)}$ is associated with a contact-force measurement $\mathbf{F}_i^{(t)}$ from the relevant sensor. The force component along that direction is

$$f_i^{(t)} = \mathbf{a}_i^{(t)\top} \mathbf{F}_i^{(t)}. \quad (3)$$

We denote the indices of the directions selected for force modulation by $\mathcal{C} \subseteq \{1, \dots, r\}$. Directions intended to remain stiff are excluded from $\mathcal{C}$, so their nominal motion components receive no compliance-induced correction.

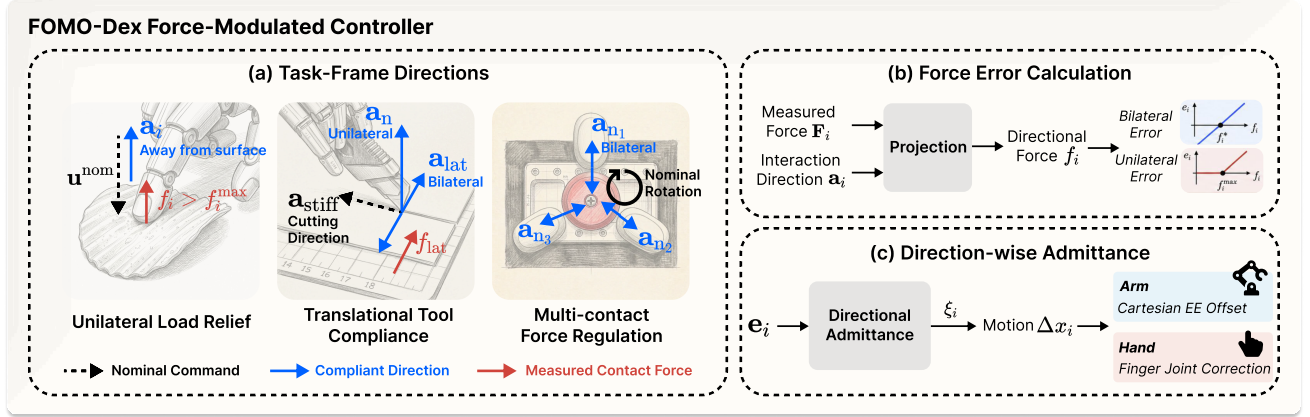


Figure 3: Overview of the FOMO-Dex controller and its task-specific realizations. Task geometry and contact structure define interaction directions, along which force measurements are converted into unilateral or bilateral force errors. Direction-wise admittance then produces arm or hand motion corrections, enabling unilateral load relief, translational tool compliance, and multi-contact force regulation within the same formulation.

3.3.2. Force Error and Admittance Modulation

For each selected direction $i \in \mathcal{C}$, we define a force error relative to the desired contact condition. When the goal is to track a target force f_i^* , we use the signed error

$$e_i^{(t)} = f_i^{(t)} - f_i^*. \quad (4)$$

When the goal is only to limit excessive loading, we use a unilateral error that is zero below the prescribed force limit $f_i^{\max}$,

$$e_i^{(t)} = [f_i^{(t)} - f_i^{\max}]_+, \quad [x]_+ = \max(x, 0). \quad (5)$$

We convert the force error into a scalar motion correction ξ_i through a direction-wise admittance law [16],

$$m_i \ddot{\xi}_i + d_i \dot{\xi}_i + k_i \xi_i = e_i, \quad i \in \mathcal{C}, \quad (6)$$

where m_i , d_i , and k_i are the virtual mass, damping, and stiffness parameters for direction $\mathbf{a}_i$. This equation gives the second-order mass-spring-damper response used for directional tool motion. Setting $m_i = 0$ gives a first-order response, while setting $m_i = k_i = 0$ gives the integrating response used for other interactions.

Each admittance state $(\xi_i, \dot{\xi}_i)$ remains associated with the same contact as its direction $\mathbf{a}_i^{(t)}$ changes. The scalar state is retained, and the correction rotates with the direction. We reset the state when a contact is created or lost, or when its direction enters or leaves $\mathcal{C}$. This prevents corrections from carrying over between contact modes.

We choose the sign of each interaction direction consistently with the measured force, so that a positive force error drives a correction along $\mathbf{a}_i$ that relieves excess contact load.

The resulting correction associated with interaction direction i is

$$\Delta \mathbf{x}_i = \xi_i \mathbf{a}_i. \quad (7)$$

Table 1: Task-specific FOMO-Dex controller configurations. “–” denotes not applicable. Contact on/off denotes the force thresholds used to activate and deactivate contact-dependent modulation, respectively.

	Card Picking	Paper Cutting	Bolt-nut/Screwdriver
Controller rate (Hz)	90	90	90
Force target f^* (N)	–	–	0.5–2.4
Force limit $f^{\max}$ (N)	3.0	3.5	3.0
Contact on/off (N)	0.6/0.6	5.0/2.0	0.25/0.15
Max correction	0.05 rad	0.06 m	0.10 rad

Thus, task geometry determines *where* force feedback acts, while the admittance law determines *how* force error modifies the nominal motion.

3.3.3. Mapping to Arm and Hand Commands

The task determines whether the arm or the fingers apply the interaction-space corrections.

For arm-mediated compliance, let $\mathcal{C}_{\text{arm}}$ contain the indices of the directions assigned to the arm. We sum their corrections,

$$\Delta \mathbf{x}_{\text{arm}} = \sum_{i \in \mathcal{C}_{\text{arm}}} \Delta \mathbf{x}_i, \quad (8)$$

and used to shift the Cartesian equilibrium pose supplied to the arm impedance controller,

$$\mathbf{T}_t^{\text{cmd}} = \mathbf{T}_t^{\text{nom}} \oplus \Delta \mathbf{T}_{\text{arm}}, \quad (9)$$

where $\Delta \mathbf{T}_{\text{arm}}$ is the pose correction corresponding to $\Delta \mathbf{x}_{\text{arm}}$.

For hand-mediated compliance, let $\mathcal{C}_j$ contain the indices of the directions associated with fingertip j . Their corrections define the desired fingertip displacement,

$$\Delta \mathbf{p}_j = \sum_{i \in \mathcal{C}_j} \Delta \mathbf{x}_i, \quad (10)$$

We map these displacements to a joint correction $\Delta \mathbf{q}$ using damped least-squares inverse kinematics [28]. With fingertip Jacobian $\mathbf{J}_j$, candidate joint correction $\mathbf{z}$, and regularization weight λ , we solve

$$\Delta \mathbf{q} = \arg \min_{\mathbf{z}} \sum_j \|\mathbf{J}_j \mathbf{z} - \Delta \mathbf{p}_j\|_2^2 + \lambda \|\mathbf{z}\|_2^2, \quad (11)$$

and add $\Delta \mathbf{q}$ to the nominal finger-joint command as in Eq. 1.

Both mappings therefore use the same interaction-space correction, but apply it through the command interface of the arm or hand.

3.3.4. Task-Specific Compliance Configuration

We instantiate the controller for each task by specifying interaction directions and their force targets or limits. Figure 3(a) shows the directions, and Table 1 lists the controller settings. The three cases below differ in the contact condition they regulate and whether the correction is applied through the arm or fingers.

Unilateral load relief. During card picking, excessive fingertip force against the supporting surface can disturb the pickup motion or damage the object. We choose an interaction direction pointing away from the surface and use a unilateral force error. Above the prescribed load limit, this error drives a retreat correction through the finger joints. Below the limit, the force error is zero. The correction thus responds to excessive surface loading without requiring the fingertip to maintain a fixed contact force.

Translational tool compliance. When cutting paper with a utility knife, a mismatch between the commanded motion and blade orientation can generate lateral forces that deflect the blade or tear the paper. We exclude the cutting direction from force modulation and apply compliance in the lateral and surface-normal directions. Bilateral feedback lets the arm respond to lateral forces from either side. Along the surface normal, a unilateral force error drives retreat when the pressing force exceeds its limit. This leaves the commanded cutting motion unchanged while reducing lateral and excessive normal loads.

Multi-contact force regulation. During rotational manipulation, the fingers must maintain contact as they move around the rotation axis. Small errors in the nominal motion can cause contact loss or excessive compression. We use the local fingertip contact normals $\mathbf{n}_i$ as interaction directions and assign a compressive force target f_i^* to each contact. These targets are derived from a shared grip level g , which we slowly adapt based on contact loss, friction utilization, and excessive loading. We distribute this grip level across active contacts according to the current grasp geometry, favoring equal loading while minimizing the net wrench induced by the contact-normal forces. Bilateral admittance increases compression when the measured force is below its target and reduces compression when the force is too large. The commanded rotational motion is left unchanged while the controller adjusts the fingertip loads.

3.4. Visuomotor Policy Training

We train a visuomotor policy by behavior cloning to evaluate whether the collected demonstrations support autonomous manipulation. The policy takes RGB images from the head ZED camera together with robot proprioception and force measurements. A frozen pretrained DINOv2 encoder [20] extracts image features, and a diffusion policy [5] predicts joint-space actions.

The training targets are the force-modulated joint commands sent to the robot, rather than the nominal teleoperation commands. The policy therefore learns from both the operator’s motion and the force-dependent corrections in the demonstrations. During training, we randomly mask the proprioceptive inputs to encourage the policy to use visual and force observations.

4. Experiments

We evaluate FOMO-Dex in teleoperation and autonomous policy execution through three questions:

- 1) *How does force modulation affect contact forces and task progress during teleoperation?*
- 2) *Across different users, does our controller increase successful-demonstration throughput relative to stiff control?*
- 3) *How do demonstration collection and action targets affect autonomous policy performance?*

Table 2: **Evaluation of our force-modulated controller across four benchmark tasks compared with stiff and naive-compliance baselines over 50 trials.** Exc. Frames denotes the fraction of frames in which the measured contact force exceeds the task-specific safe threshold, and Force Exceedance is defined in Eq. 12. $\uparrow/\downarrow$ indicate higher/lower is better; bold and underline denote the best and second-best results, respectively.

Task	Metric	Stiff	Naive Comp.	Ours
Card Picking	Success (%) $\uparrow$	<u>90.9</u>	37.5	93.8
	Duration (s) $\downarrow$	22.1 $\pm$ 4.2	<u>17.2</u> $\pm$ 2.4	16.9 $\pm$ 4.2
	Exc. Frames (%) $\downarrow$	13.0 $\pm$ 5.4	<u>7.4</u> $\pm$ 7.9	5.9 $\pm$ 2.8
	Force Exceedance (N) $\downarrow$	0.32 $\pm$ 0.29	<u>0.10</u> $\pm$ 0.13	0.05 $\pm$ 0.03
Paper Cutting	Cut Rate (cm/s) $\uparrow$	2.47 $\pm$ 0.15	3.16 $\pm$ 1.12	<u>2.63</u> $\pm$ 0.41
	Lat. Force (N) $\downarrow$	2.00 $\pm$ 0.41	<u>1.61</u> $\pm$ 0.43	1.41 $\pm$ 0.14
	Norm. Force (N) $\downarrow$	<u>4.39</u> $\pm$ 0.94	5.28 $\pm$ 0.77	3.62 $\pm$ 0.18
Bolt-Nut	Turns/ep $\uparrow$	23.7 $\pm$ 1.2	17.0 $\pm$ 2.1	<u>20.3</u> $\pm$ 1.2
	Exc. Frames (%) $\downarrow$	27.4 $\pm$ 4.1	8.0 $\pm$ 2.0	<u>23.3</u> $\pm$ 4.3
	Force Exceedance (N) $\downarrow$	0.64 $\pm$ 0.18	0.09 $\pm$ 0.03	<u>0.57</u> $\pm$ 0.21
Screwdriving	Turns/ep $\uparrow$	4.2 $\pm$ 2.2	<u>14.6</u> $\pm$ 6.9	34.7 $\pm$ 5.7
	Slip Events/ep $\downarrow$	1.00 $\pm$ 0.00	<u>0.75</u> $\pm$ 0.43	0.28 $\pm$ 0.46
	Exc. Frames (%) $\downarrow$	0.4 $\pm$ 0.9	0.2 $\pm$ 0.2	<u>0.3</u> $\pm$ 1.2
	Force Exceedance (N) $\downarrow$	0.186 $\pm$ 0.073	0.001 $\pm$ 0.001	<u>0.002</u> $\pm$ 0.009

4.1. Experimental Setup

4.1.1. Baselines

We compare our force-modulated controller described in Section 3 with the following baselines. The **Stiff** controller directly executes the operator command with hand position control and arm impedance control, i.e., $\Delta \mathbf{u}_t = 0$ in Eq. 1. The **Naive Compliant** controller uses the same admittance parameters as FOMO-Dex wherever applicable, but applies them isotropically rather than selectively along task- and contact-relevant directions. All controllers share the same teleoperation interface, command rate, sensing, low-level control, and safety limits.

4.1.2. Benchmark Problems

We evaluate our method on four contact-rich dexterous manipulation tasks that require different forms of force-aware compliance. We report task-progress metrics and interaction-based metrics. Let f_t denote the relevant measured contact force and f^{safe} its task-dependent threshold. We report the fraction of frames exceeding this threshold and the mean force exceedance

$$E_f = \frac{1}{T} \int_0^T \max(0, |f_t| - f^{\text{safe}}) dt, \quad (12)$$

where T is the episode duration. This metric captures both the magnitude and duration of force above the threshold.

Card Picking The robot picks a poker card from a flat surface while avoiding excessive fingertip loading against the support. We measure pickup success, episode duration, the fraction of frames with excessive contact force, and force exceedance using Eq. 12.

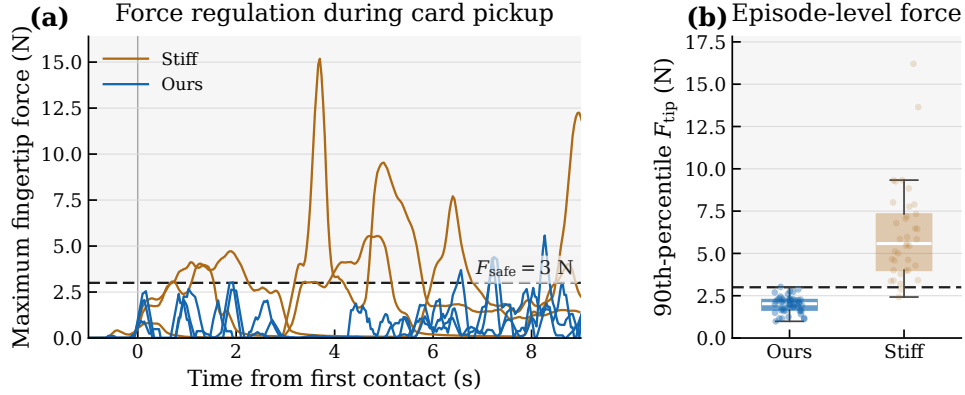


Figure 4: Fingertip forces during card picking with stiff control and FOMO-Dex. (a) Maximum force across fingertips over five representative episodes per controller, aligned at first contact. (b) Distribution of the per-episode 90th-percentile fingertip force. Dashed lines mark the 3 N force threshold.

Paper Cutting The robot uses a utility knife to follow a marked path through a sheet of paper while maintaining appropriate contact with the surface. We measure cut rate and the lateral and normal interaction forces applied to the tool.

Bolt–Nut Screwing The robot rotates a nut along a threaded bolt while maintaining fingertip contact. We measure the number of completed turns per episode, the fraction of frames with excessive contact force, and force exceedance using Eq. 12.

Screwdriving The robot turns a screw with a screwdriver while maintaining tip engagement and axial contact. We measure the number of completed turns per episode, screwdriver slip events, the fraction of frames with excessive contact force, and force exceedance using Eq. 12.

4.1.3. Trial protocol

For each task, the object and robot are reset to a common initial configuration. We record operator commands, post-compliance commands, robot state, wrist wrench and fingertip forces at 30 Hz. The thresholds used to compute excessive force are fixed during data collection.

4.2. Controller Analysis on Teleoperation Data

We first evaluate how force modulation affects the execution of nominal teleoperation commands across four contact-rich tasks. Table 2 compares our task-frame force-modulated controller against baseline controllers. Stiff control tracks the operator command without explicit force-dependent corrections. This leads to excessive interaction forces in tasks such as card picking and paper cutting, and can also destabilize contact in screwdriving. Naive compliance alleviates some of these forces, but indiscriminately relaxing the commanded motion can interfere with task progress. This is particularly evident in card picking, where reduced contact force does not translate into reliable task completion.

The balance between force reduction and task progress differs across tasks. For *card picking*, the controller can yield to the supporting surface while preserving the motion needed to retrieve the card. Figure 4 overlays fingertip forces across five card-pickup episodes, showing that our controller consistently limits

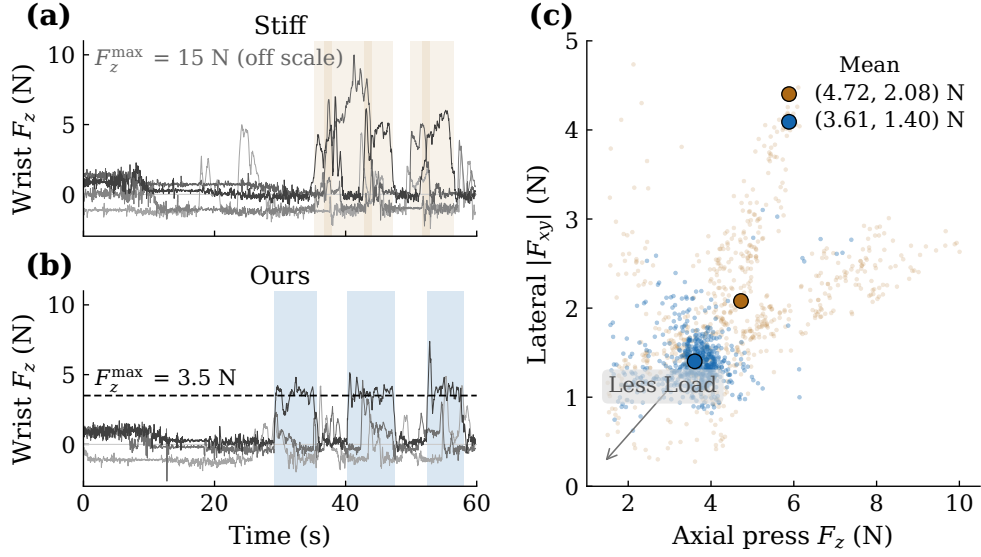


Figure 5: Interaction forces during paper cutting. (a,b) Wrist axial force over five representative episodes under stiff control and FOMO-Dex, respectively. (c) Axial versus lateral force samples for stiff control (orange) and FOMO-Dex (blue). Large markers indicate the means.

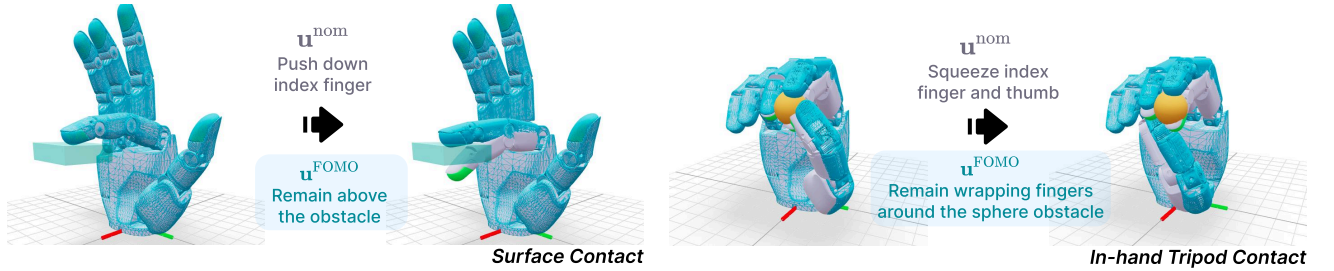


Figure 6: Hand configurations before and after force modulation. The gray mesh shows the nominal configuration commanded by the teleoperator, while the blue mesh shows the force-modulated command. For surface contact (left), the contacting fingertip retreats from the supporting surface. For in-hand tripod contact (right), force regulation adjusts the index–thumb contacts while preserving the remaining grasp configuration.

excessive contact forces, while stiff control exhibits frequent high-force interactions. During *paper cutting*, FOMO-Dex reduces lateral and normal loads relative to stiff control while maintaining a comparable cut rate. Figure 5 shows these force reductions across five episodes. In *bolt-nut manipulation*, stiff control favors rotational progress at the cost of larger contact forces, while naive compliance reduces contact forces but more strongly suppresses the commanded rotation. FOMO-Dex lies between these baselines in both completed turns and force exceedance. In *screwdriving*, force modulation substantially improves rotational progress and maintains tool engagement. Naive compliance achieves slightly lower force exceedance than FOMO-Dex, but completes fewer turns and has more tool slips. Lower force exceedance alone therefore does not imply better manipulation performance.

To further illustrate how force modulation changes the commanded hand configuration, Fig. 6 visualizes representative contact-dependent corrections. The gray mesh denotes the nominal hand configuration,

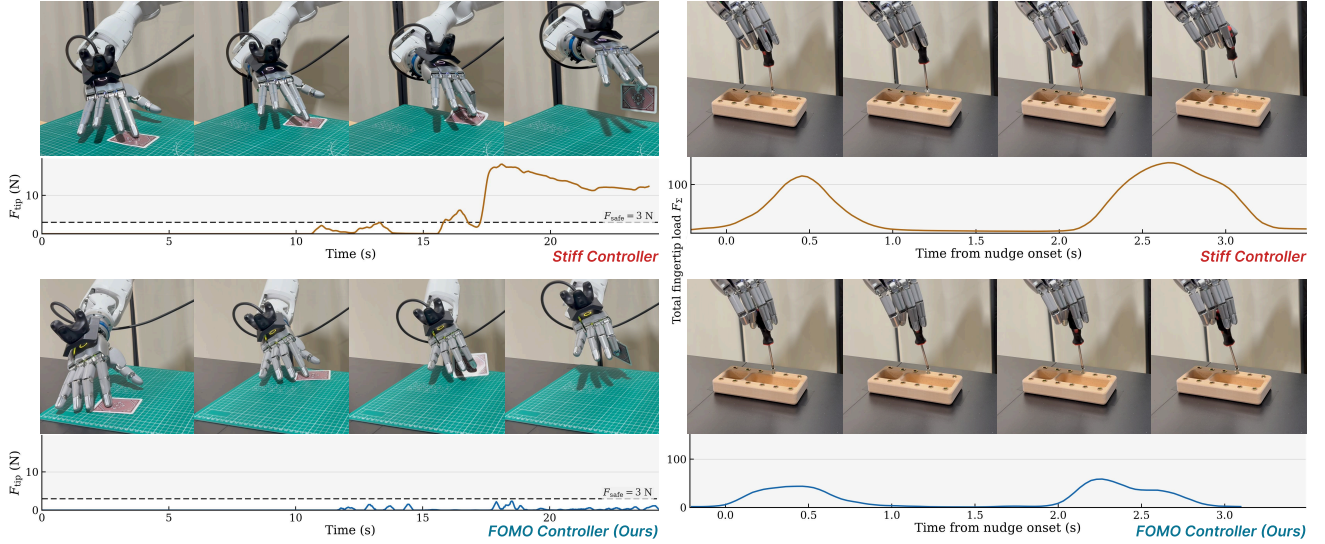


Figure 7: Real-world teleoperation with stiff control (top) and FOMO-Dex (bottom). In card picking (left), FOMO-Dex limits excessive fingertip loading against the supporting surface. In screwdriving (right), it reduces large variations in total fingertip load and maintains more stable tool engagement. Representative execution frames are shown with the corresponding force traces.

while the blue mesh shows the force-modulated configuration. For surface contact, the contacting fingertip retreats from the obstacle while the remaining fingers largely preserve the nominal motion. Under in-hand multi-contact interaction, the controller instead adjusts the active contacts while maintaining the overall grasp. These examples highlight that FOMO-Dex does not uniformly soften the hand, but selectively modifies motion components associated with the current contact constraints.

Figure 7 shows how these contact-dependent corrections translate to real-world teleoperation behavior. In card picking, stiff control produces a large fingertip-force spike as the hand interacts with the supporting surface, whereas FOMO-Dex relieves the contact load while allowing the pickup motion to continue. In screwdriving, stiff control exhibits large variations in total fingertip load, while FOMO-Dex attenuates these variations and maintains more stable tool engagement.

4.3. User Study

We conduct a user study with four participants to evaluate how force modulation affects teleoperated demonstration collection. Each participant performs three tasks—card picking, bolt–nut manipulation, and screwdriving—using both our controller and the stiff controller. Task order and controller order are randomized across participants.

For each task–controller condition, participants first perform 10 trials without prior practice (zero-shot). They are then given 10 minutes to practice the task using both controllers, after which they perform another 10 evaluation trials with each controller. Participants receive the same task instructions and success criteria across conditions.

We measure the amount of successful demonstration data collected per hour of teleoperation (Succ. Data) and the mean episode length. Succ. Data is computed as the total duration of successful demonstrations divided by the total teleoperation time, and scaled to minutes of successful data per hour of human

Table 3: **User-study comparison of demonstration collection with the stiff controller and FOMO-Dex, zero-shot and after 10 min of practice.** Succ. Data is successful demonstration time per hour of teleoperation. Mean Ep. Len. is episode duration; shorter is preferred for card picking, while longer is preferred for screwdriving, where episodes are capped at 60 s and longer durations indicate more stable sustained manipulation of the long screw.

Experience	Stiff		Ours	
	Succ. Data (min/h) $\uparrow$	Mean Ep. Len. (s)	Succ. Data (min/h) $\uparrow$	Mean Ep. Len. (s)
Card Picking	<i>Mean Ep. Len. $\downarrow$</i>			
Zero-shot	11.9 $\pm$ 5.2	32.7 $\pm$ 10.5	20.6 $\pm$ 4.0	29.1 $\pm$ 0.5
+ 10 min	24.3 $\pm$ 9.4	27.9 $\pm$ 1.2	29.3$\pm$5.6	24.9$\pm$10.2
Bolt-Nut	<i>Mean Ep. Len. = 60</i>			
Zero-shot	41.0 $\pm$ 1.1	60.0 $\pm$ 0.0	40.7 $\pm$ 1.0	60.0 $\pm$ 0.0
+ 10 min	30.3 $\pm$ 12.8	60.0 $\pm$ 0.0	50.2$\pm$7.0	60.0 $\pm$ 0.0
Screwdriving	<i>Mean Ep. Len. $\uparrow$</i>			
Zero-shot	11.8 $\pm$ 6.4	11.2$\pm$5.2	17.9 $\pm$ 15.2	10.9 $\pm$ 6.8
+ 10 min	12.4 $\pm$ 4.7	6.4 $\pm$ 2.0	30.7$\pm$10.9	10.4 $\pm$ 2.3

Table 4: **Evaluation of behavior-cloning policies trained on 50 demonstrations per task-controller setting.** (a) Card-picking ablation comparing stiff, raw-command training with FOMO-Dex inference (Comp. Raw), and post-modulation-action training (Comp. Applied) over 18 evaluation trials. (b) Cross-task comparison of policies trained from stiff and FOMO-Dex demonstrations, evaluated by turns per episode, over 10 evaluation trials.

(a) Card Picking Ablation				(b) Cross-task Comparison		
Success (%)	Reach	Retr.	Lift	Turns/ep.	Stiff	Ours
Stiff	100	100	56	Bolt-Nut	24.5 $\pm$ 4.2	60.8$\pm$4.5
Comp. Raw	100	89	0	Screwdriving	0.6 $\pm$ 0.8	6.3$\pm$2.2
Comp. Applied (Ours)	100	89	67			

teleoperation.

Table 3 shows that FOMO-Dex increases successful-data throughput across all three tasks after practice. For card picking, our controller increases successful-data throughput both zero-shot and after practice. For bolt-nut manipulation, the two controllers perform similarly zero-shot, while after practice participants collect substantially more successful data with FOMO-Dex. In screwdriving, FOMO-Dex improves throughput both before and after practice. After practice, participants also sustain longer episodes with FOMO-Dex than with stiff control.

4.4. Behavior-Cloning Results

We further investigate how force-modulated demonstrations should be used for policy learning. We consider two ways of incorporating our controller into behavior cloning. *Comp. Raw* trains the policy to predict the raw teleoperation commands and applies our force-modulation controller at inference time. *Comp. Applied* instead uses force-modulated actions as training targets and applies the stiff controller at inference time.

We first compare these strategies on card picking. As shown in Table 4(a), directly imitating the applied actions produces the highest lift success. The policy trained on raw commands completes no successful lifts, even with force modulation at deployment. Based on this result, we use post-modulation actions as the learning target in the remaining tasks.

We next evaluate whether demonstrations collected with force modulation improve policy learning on long-horizon rotational tasks. We compare policies trained on demonstrations collected with stiff teleoperation against those collected with FOMO-Dex, using the applied actions as targets for the latter. As shown in Table 4(b), policies trained on FOMO-Dex demonstrations achieve substantially more rotation in both bolt–nut manipulation and screwdriving.

5. Conclusions

We present FOMO-Dex, a unified framework for force-modulated dexterous manipulation that treats teleoperation or policy output as a nominal action and selectively adjusts contact-relevant components using force feedback in task- and contact-relevant coordinates. Across four contact-rich tasks, FOMO-Dex improves contact behavior while preserving task progress, increases successful demonstration throughput in a user study, and enables behavior-cloning policies to benefit from force-modulated demonstrations. These results suggest that explicit force modulation provides a practical interface between motion commands and the contact responses required for dexterous manipulation. Future work will infer interaction directions and force targets online, incorporate uncertainty-aware contact estimation, and evaluate the framework across broader tasks and objects.

References

- [1] Fares J Abu-Dakka and Matteo Saveriano. Variable impedance control and learning—a review. *Frontiers in Robotics and AI*, 2020.
- [2] Fares J Abu-Dakka, Leonel Roza, and Darwin G Caldwell. Force-based variable impedance learning for robotic manipulation. *Robotics and Autonomous Systems*, 2018.
- [3] Herman Bruyninckx and Joris De Schutter. Specification of force-controlled actions in the "task frame formalism"-a synthesis. *IEEE Transactions on Robotics and Automation*, 1996.
- [4] Claire Chen, Zhongchun Yu, Hojung Choi, Mark Cutkosky, and Jeannette Bohg. DexForce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation. *RA-L*, 2025.
- [5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.
- [6] 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. In *ICRA*, 2026.
- [7] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-VisionPro: Real-time bimanual dexterous teleoperation for imitation learning. In *IROS*, 2025.
- [8] Hongjie Fang, Shirun Tang, Mingyu Mei, Haoxiang Qin, Zihao He, Jingjing Chen, Ying Feng, Chenxi Wang, Wanxi Liu, Zaixing He, Cewu Lu, and Shiquan Wang. Force policy: Learning hybrid force-position control policy under interaction frame for contact-rich manipulation. In *RSS*, 2026.

- [9] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020.
- [10] Botao He, Zhi Wang, Linna Kuang, Ishaan Ghosh, Jitendra Malik, Cornelia Fermuller, Tingfan Wu, Jiayuan Mao, Ruoshi Liu, Haozhi Qi, and Yiannis Aloimonos. ForceBand: Learning forceful manipulation with sEMG. In *CoRL*, 2026.
- [11] Neville Hogan. Impedance control: An approach to manipulation. In *ACC*, 1984.
- [12] Yifan Hou and Matthew T Mason. An efficient closed-form method for optimal hybrid force-velocity control. In *ICRA*, 2021.
- [13] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppaswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. In *ICRA*, 2025.
- [14] Elvis Hsieh, Wen-Han Hsieh, Yen-Jen Wang, Toru Lin, Jitendra Malik, Koushil Sreenath, and Haozhi Qi. Learning dexterous manipulation skills from imperfect simulations. In *ICRA*, 2026.
- [15] Jeon Ho Kang, Sagar Joshi, Ruopeng Huang, and Satyandra K Gupta. Robotic compliant object prying using diffusion policy guided by vision and force observations. *RA-L*, 2025.
- [16] Arvid QL Keemink, Herman Van der Kooij, and Arno HA Stienen. Admittance control for physical human-robot interaction. *IJRR*, 2018.
- [17] Hisham Khalil, Neil Fernandes, Thomas M Kwok, Hsiu-Chin Lin, and Yue Hu. VIDP: Variable impedance diffusion policy for compliant robot manipulation from diverse demonstrations. *arXiv:2608.06210*, 2026.
- [18] Miao Li, Hang Yin, Kenji Tahara, and Aude Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In *ICRA*, 2014.
- [19] Haichao Liu, Yuyao Jiang, Hyunsun Park, Yuanjiang Xue, and Ziwei Wang. DexTeleop-0: Force-aware bimanual dexterous teleoperation with ego-centric perception towards shared autonomy. *arXiv:2606.23431*, 2026.
- [20] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.
- [21] Martin Pfanne, Maxime Chalon, Freek Stulp, Helge Ritter, and Alin Albu-Schäffer. Object-level impedance control for dexterous in-hand manipulation. *RA-L*, 2020.
- [22] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *RA-L*, 2022.
- [23] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system. In *RSS*, 2023.

- [24] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 1981.
- [25] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *CoRL*, 2024.
- [26] Haochen Shi, Songbo Hu, Yifan Hou, Weizhuo Wang, Karen Liu, and Shuran Song. Minimalist compliance control. In *RSS*, 2026.
- [27] Rishabh Shukla, Raj Talan, Samrudh Moode, Neel Dhanaraj, Jeon Ho Kang, and Satyandra K Gupta. Force-conditioned diffusion policies for compliant sheet separation tasks in bimanual robotic cells. In *ICRA*, 2025.
- [28] Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 1986.
- [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 *CoRL*, 2025.
- [30] Xiaomeng Xu, Yifan Hou, Zeyi Liu, and Shuran Song. Compliant residual DAGger: Improving real-world contact-rich manipulation with human corrections. In *NeurIPS*, 2025.
- [31] Zhengtong Xu, Yeping Wang, Ben Abbatematteo, Jom Preechayasomboon, Sonny Chan, Nicholas Colonnese, and Amirhossein H Memar. Contact-grounded policy: Dexterous visuotactile policy with generative contact grounding. In *RSS*, 2026.
- [32] Shiqi Yang, Minghuan Liu, Yuzhe Qin, Runyu Ding, Jialong Li, Xuxin Cheng, Ruihan Yang, Sha Yi, and Xiaolong Wang. ACE: A cross-platform and visual-exoskeletons system for low-cost dexterous teleoperation. In *CoRL*, 2024.
- [33] Jessica Yin, Haozhi Qi, Youngsun Wi, Sayantan Kundu, Mike Lambeta, William Yang, Changhao Wang, Tingfan Wu, Jitendra Malik, and Tess Hellebrekers. OSMO: Open-source tactile glove for human-to-robot skill transfer. *RA-L*, 2026.
- [34] Chao Zeng, Shuang Li, Yiming Jiang, Qiang Li, Zhaopeng Chen, Chenguang Yang, and Jianwei Zhang. Learning compliant grasping and manipulation by teleoperation with adaptive force control. In *IROS*, 2021.
- [35] Han Zhang, Songbo Hu, Zhecheng Yuan, and Huazhe Xu. DOGlove: Dexterous manipulation with a low-cost open-source haptic force feedback glove. In *RSS*, 2025.
- [36] Yan Zhang, Fei Zhao, and Zhiwei Liao. Learning and generalizing variable impedance manipulation skills from human demonstrations. In *AIM*, 2022.
- [37] Zhengxue Zhou, Xingyu Yang, and Xuping Zhang. Variable impedance control on contact-rich manipulation of a collaborative industrial mobile manipulator: An imitation learning approach. *Robotics and Computer-Integrated Manufacturing*, 2025.