

DexMani: Human-Derived Manipulability Guidance for Dexterous Rotation

Xiaoyang Chen^{1,2,3*}, Shengcheng Luo^{2,3*}, Haoran Guo², Jiaming Jiang^{2,3},
Wanlin Li³, Ziyuan Jiao^{3,4†}, Chenxi Xiao^{2†}

¹Shanghai Jiao Tong University ²ShanghaiTech University

³Beijing Institute for General Artificial Intelligence (BIGAI) ⁴Beihang University

Abstract

Dexterous object rotation is a sequential contact problem: each support, release, and re-contact decision must both produce the desired object motion, and prepare the hand configuration for continued rotation. Existing reinforcement learning methods discover such movement patterns through trial and error on specific robotic hand embodiments, without explicitly accounting for how each contact transition affects the hand’s ability to sustain object rotation in subsequent steps. We introduce **DexMani**, a framework that transfers human demonstrations as *contact-conditioned manipulability evolution*. This prior captures how successful human contact transitions reshape the object-rotation directions available to the hand. DexMani then learns this manipulability evolution and uses it to guide downstream reinforcement learning, enabling rotation skills to be acquired across robot embodiments with distinct kinematics and active-contact configurations. Across the Shadow Hand, Allegro Hand, and XHand, DexMani achieves the highest success rates in every evaluated setting for both seen and unseen objects. DexMani reaches an average success rate of 57.5% on LEAP Hand, outperforming other baselines and producing smoother rotatory motions.

Project site — <https://dexmani.github.io/>

1 Introduction

Dexterous rotation tasks, such as unscrewing a cap, turning a valve, or spinning an object, often require the fingers to repeatedly break and re-establish contact (Morgan et al. 2022). Each contact transition must not only advance the object’s rotation but also leave the hand in a new configuration that support continued rotation about the desired axis. We refer to this configuration-dependent capability as *contact-conditioned rotational manipulability*. Sustained rotation therefore depends not only on the immediate rotational progress of each transition, but also on how this manipulability is maintained and reshaped over successive transitions.

Despite its importance for generating rotation motion, the evolution of contact-conditioned rotational manipulability (hereafter referred to simply as *manipulability*) has received little attention in existing learning-based methods. For instance, reinforcement learning methods can discover effective contact transitions through extensive trial

and error, but generally do not explicitly consider how manipulability evolves with the hand’s contact state (Qi et al. 2023; Yang et al. 2024). Similarly, imitation via demonstration-retargeting approaches transfer human motions into embodiment-specific robot trajectories (Qin et al. 2022; Li et al. 2025a), but have not yet incorporated contact-conditioned manipulability evolution as an explicit guidance or conditioning signal.

To bridge the gap, we propose **DexMani**, a framework that distills successful human rotation demonstrations into a prior over the short-horizon evolution of rotational manipulability, and uses it to guide downstream robot learning (Fig. 1). DexMani first learns an energy model that evaluates changes in manipulability based on patterns observed in human demonstrations. During robot control, the model scores the robot manipulability changes induced by candidate actions, and then generate residual action guidance that favors human-like evolution. Because this guidance is expressed through a manipulability representation that is transferable across embodiments, DexMani can generalize across rotation tasks and dexterous hands.

To evaluate its performance, we conducted comprehensive evaluations of DexMani on three contact-rich rotation tasks: Unscrew Cap, Rotate Object, and Turn Faucet. The results demonstrate strong cross-embodiment and cross-task generalization, with DexMani achieving the highest success rates on both seen and unseen objects in every evaluated setting. It achieves an average success rate of 57.5% across the three LEAP Hand tasks and 43.4% for cap unscrewing across the Shadow, Allegro, and XHand embodiments. We additionally demonstrate closed-loop deployment on a physical dexterous hand, highlighting the practical applicability of the proposed approach to real-world robotic systems.

In summary, our main contributions are:

- We introduce contact-conditioned manipulability, a human-to-robot representation that captures how contact transitions reshape task-space rotational capabilities.
- We develop an energy-based prior that evaluates whether changes in manipulability are consistent with human demonstrations and conducive to sustained rotation.
- We propose and evaluate a reinforcement learning framework that transfers this shared prior across dexterous hands with different kinematics.

*These authors contributed equally.

†Corresponding author.

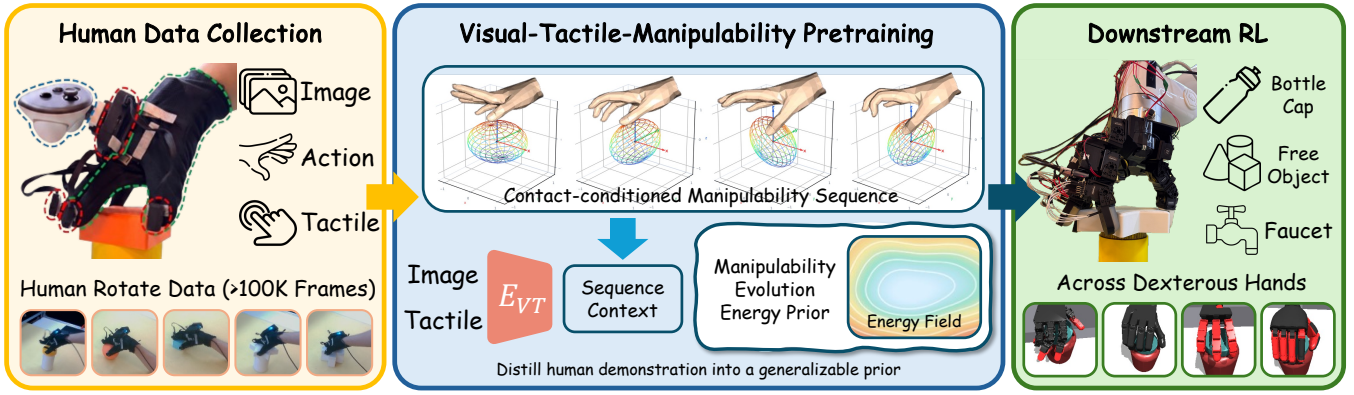


Figure 1: **DexMani overview.** Human visual–tactile rotation demonstrations are used to pretrain an energy-based prior over contact-conditioned manipulability evolution. The frozen prior then guides downstream reinforcement learning across rotation tasks and dexterous hand embodiments.

2 Related Work

Learning-Based Dexterous In-Hand Rotation

Learning-based methods generally acquire dexterous rotation skills from robot-generated interaction data, human demonstrations, or a combination of both. For instance, reinforcement learning has been applied to a variety of rotation tasks, including in-hand reorientation (Qi et al. 2023), tactile-only control (Yin et al. 2023), rapid finger gaiting (Wang et al. 2024), and articulated-object rotation (Yang et al. 2024). More recent work has explored transferable interaction representations learned from large-scale visual–tactile data (Ye et al. 2026; Huang et al. 2026; Zhang et al. 2026), leveraging privileged simulation information, policy distillation, and tactile perception to enable effective rotation skills. However, these methods must still discover effective contact transitions through either trial-and-error exploration or imitation of human demonstrations. They lack an explicit prior that encodes the desired motion trend throughout the rotation process.

Human Demonstration Priors and Motion Transfer

Recent research has increasingly used human demonstrations to guide robot learning. These approaches often convert demonstrations into hand-specific actions that a robot can reproduce. For example, a commonly used method is to retarget human motions into robot-specific joint commands, or fingertip trajectories (Qin et al. 2022; Li et al. 2025a). Recent work has further improved retargeting through contact priors, or by models for correcting retarget errors (Wu et al. 2026; Pan et al. 2026). Beyond retargeting, demonstration-guided reinforcement learning incorporates human-motion priors through policy pretraining, or additional reward objectives (Rajeswaran et al. 2018; Peng et al. 2021; Lum et al. 2025). However, these representations are often embodiment-specific: differences in hand kinematics and workspace necessitate retargeting or policy relearning for each new hand. In contrast, DexMani transfers an embodiment-agnostic prior over the desired evolution of manipulability, allowing each hand to realize this trend through its own actions and contact transitions.

Manipulability and Cross-Embodiment Guidance

Manipulability characterizes the directions of motion enabled by a hand configuration (Friedman and Flash 2007; Yokokohji, San Martin, and Fujiwara 2009). Prior work has investigated the geometry of manipulability ellipsoids (Jaquier et al. 2021), transferred desired manipulability profiles across embodiments (Jaquier, Rozo, and Calinon 2020; Reithmeir, Figueredo, and Haddadin 2022), and incorporated manipulability objectives into policy learning (Sun and Liao 2024; Li et al. 2025b). However, these methods do not explicitly account for how object contacts constrain manipulability during rotation. DexMani addresses this limitation with a contact-conditioned manipulability measure that captures a hand’s ability to sustain rotational motion under contact constraints, providing embodiment-agnostic guidance for contact-rich manipulation.

3 Methods

DexMani aims to transfer a preference over how rotational capability evolves during manipulation. As shown in Fig. 2, it first learns such a preference prior from human demonstrations. Then, during downstream reinforcement learning, the learned prior is used to provide residual action guidance, encouraging the policy to follow the learned preference.

3.1 Human Demonstration Collection

High-quality robot demonstrations of dexterous rotation are difficult to collect due to the intractability of accurately conveying contact perception during teleoperation (Qin et al. 2023; Luo et al. 2025). We therefore collect human demonstrations instead by having participants directly rotate the target objects via a tactile glove. For this purpose, we develop the data-collection system shown in Fig. 3. A Meta Quest 3 controller (Meta 2023) attached to the back of the hand tracks the global hand pose. A Manus Quantum Meta-Glove (Manus Meta 2024) captures finger joint poses, while a JQ-Industries tactile glove (Juqiao Industrial 2024) measures contact pressure across the fingertips using a piezoresistive pressure array. Using this device, We concurrently extract

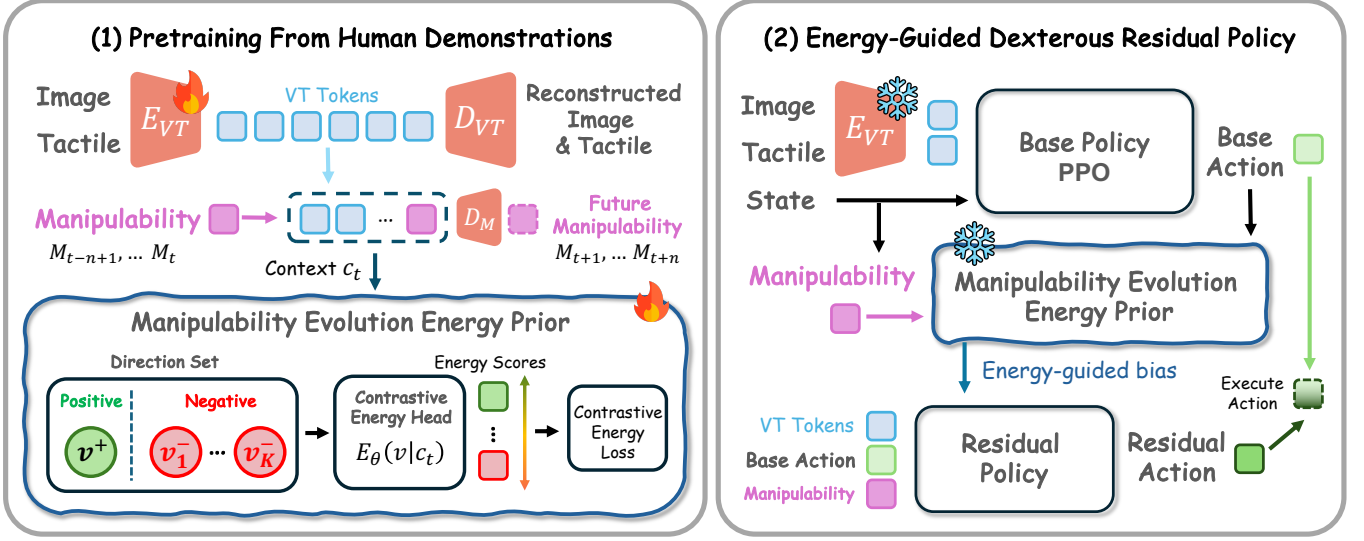


Figure 2: **DexMani pipeline.** DexMani learns a contact-conditioned manipulability-evolution energy prior from human visual-tactile demonstrations (left). During robot reinforcement learning, the frozen prior scores action-induced manipulability changes and provides hand-specific guidance to a residual policy (right).

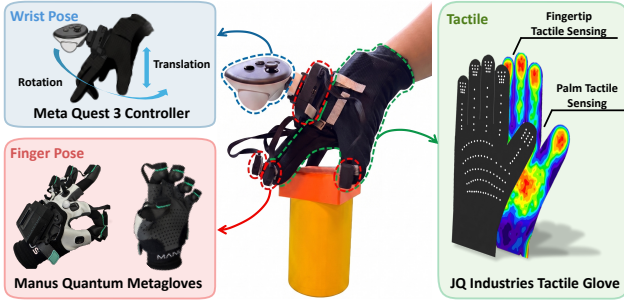


Figure 3: **Multimodal data collection system.** We collect synchronized hand pose and tactile pressure using a Manus Quantum MetaGlove and a JQ-Industries tactile glove.

fingertip pressure measurements, as well as the hand pose, and RGB observations. In total, we collect over 100,000 frames of human rotation data. These synchronized recordings allow us to determine active contact states and compute the corresponding contact-conditioned manipulability labels used for pretraining.

3.2 Analytic Manipulability from Human

Using collected human data, we calculate the contact-conditioned manipulability introduced as follows. The overall pipeline for such calculation comprises three components: (1) a visual-tactile encoder that fuses RGB and tactile histories into a compact representation, (2) contact-conditioned manipulability, an analytic, tactile-gated manipulability descriptor that provides supervision labels from demonstrations, and (3) an auxiliary prediction decoder that is discarded after pretraining. We describe each component below.

Visual-Tactile Encoder. Given a human demonstration sequence, the encoder maps these observations to

$$(z_t^{VT}, h_v, h_c) = F_\eta(V_{t-H+1:t}, T_{t-H+1:t}), \quad (1)$$

where visual and tactile histories are denoted as $V_{t-H+1:t}$ and $T_{t-H+1:t}$, respectively, H is the history length, z_t^{VT} is the fused visual-tactile representation, and h_v and h_c are modality-specific token features. Using this encoder, the RGB observations are embedded as visual patch tokens, while the tactile readings are projected into contact tokens. A learnable transformer-based (Vaswani et al. 2017) fusion network F_η then integrates the two modalities. Thus, z_t^{VT} compactly describes both the visually observed hand-object interaction and where contact is currently occurring.

Contact-conditioned Manipulability. We introduce contact-conditioned manipulability as a cross-embodiment representation of rotational capability evolution. Conventional configuration based manipulability indices characterize a hand’s kinematic capabilities but do not explicitly capture how active contacts constrain object motion. To address this limitation, we introduce contact-conditioned manipulability defined as following. Let $J_{c,t}^{\text{hum}}$ denote the stacked Jacobian of the active fingertip contacts, where W_t denote the diagonal matrix of tactile-confidence weights, H_h is a diagonal joint-space scaling matrix, and G_t^+ the damped pseudoinverse of the grasp matrix. Extending the conventional manipulability formulation of (Chiacchio et al. 1991) with tactile-induced kinematic constraints, we further define the contact capability $C_{c,t}^{\text{hum}}$ and rotational manipulability matrix $M_{\omega,t}^{\text{hum}}$ as

$$\begin{aligned} C_{c,t}^{\text{hum}} &= W_t^{1/2} J_{c,t}^{\text{hum}} H_h (J_{c,t}^{\text{hum}})^\top W_t^{1/2}, \\ M_{\omega,t}^{\text{hum}} &= P_\omega (G_t^+)^\top C_{c,t}^{\text{hum}} G_t^+ P_\omega^\top + \epsilon_m I_3. \end{aligned} \quad (2)$$

where P_ω selects the rotational component and $\epsilon_m I_3$ ensures positive definiteness. We represent $M_{\omega,t}^{\text{hum}}$ with the log-Euclidean coordinate $m_t^{\text{hum}} = \text{vech}(\log M_{\omega,t}^{\text{hum}}) \in \mathbb{R}^6$. Although m_t is computed using each hand’s own contact Jacobian, we define the wrist-frame axes consistently across hands, so that its dimensions describe rotational capabilities about the same x -, y -, and z -directions. Contact positions are measured relative to the weighted center of the tactilely detected contacts, avoiding the need to estimate the object center. This shared task-space definition allows human evolution prior to be transferred to robot-induced changes without joint-level correspondence.

3.3 Manipulability-Evolution Energy Prior

Using the contact-conditioned manipulability descriptors extracted from human demonstrations, we learn a prior over the evolution of manipulability during successful object rotation. The model is capable of evaluating candidate directions of manipulability change, assigning low energy to directions consistent with human demonstrations and high energy to incompatible directions.

To train this prior model, we use three complementary components: (1) a contrastive energy objective, (2) structured negative sampling, and (3) an auxiliary objective for predicting the magnitude of the change. All these components are detailed as following.

Contrastive Manipulability-Evolution Energy. We define a conditional energy model $E_\theta(\hat{v} \mid c_t)$ that measures the compatibility between a candidate manipulability-change direction $\hat{v}$ and the current interaction context $c_t = [z_t^{VT}, m_t, d_t]$, where z_t^{VT} encodes the visual–tactile observations, m_t denotes the current manipulability state, and d_t specifies the desired rotation direction. A lower energy indicates that the candidate direction is more consistent with the manipulability evolution observed in human demonstrations.

We obtain the positive direction from the demonstrated short-horizon change:

$$\hat{v}_t^+ = \frac{m_{t+\Delta} - m_t}{\|m_{t+\Delta} - m_t\|_2 + \epsilon_{\text{num}}}. \quad (3)$$

This produces a unit vector in six-dimensional manipulability space, allowing the energy model to focus on the direction in which manipulability should evolve. We exclude near-stationary frames from the directional objective because their normalized directions are dominated by measurement noise.

Structured Negatives and Energy Loss. To teach the model which directions are incompatible with the current context, we follow the sampling-based contrastive strategy of IBC (Florence et al. 2021). We contrast each demonstrated direction with K structured negatives. These negatives comprise random directions, reversed directions, and directions drawn from mismatched coordinate frames. Random negatives provide broad coverage of the direction space, whereas reversed and mismatched directions provide incorrect instances within interaction context.

Let $\mathcal{V}_t = \{\hat{v}_t^+\} \cup \{\hat{v}_{t,k}^-\}_{k=1}^K$ denote the resulting candidate

set. We optimize the contrastive energy loss

$$\mathcal{L}_E = -\log \frac{\exp[-E_\theta(\hat{v}_t^+ \mid c_t)/\tau]}{\sum_{\hat{v} \in \mathcal{V}_t} \exp[-E_\theta(\hat{v} \mid c_t)/\tau]}, \quad (4)$$

where τ is the contrastive temperature. This objective encourages the demonstrated direction to have lower energy than incompatible alternatives, thereby shaping an energy landscape that captures context-dependent manipulability evolution.

Magnitude Prediction and Full Pretraining Loss. In addition to the aforementioned directional target (normalized, it does not preserve the magnitude of desired motion), we introduce an additional auxiliary head that predicts the log-scaled magnitude:

$$\alpha_t = \log(1 + \|m_{t+\Delta} - m_t\|_2), \quad \mathcal{L}_\alpha = |\hat{\alpha}_t - \alpha_t|. \quad (5)$$

Thus, the complete pretraining objective is

$$\mathcal{L}_{\text{pre}} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{pred}} \mathcal{L}_{\text{pred}} + \lambda_E \mathcal{L}_E + \lambda_\alpha \mathcal{L}_\alpha, \quad (6)$$

where $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{pred}}$ train the auxiliary decoder and learned representation, respectively, while $\mathcal{L}_E$ and $\mathcal{L}_\alpha$ supervise the direction and magnitude of manipulability evolution. After pretraining, the visual–tactile encoder F_η and the energy model E_θ are frozen and subsequently used during robot reinforcement learning.

3.4 Manipulability Energy-Guided Residual Policy

Finally, we transfer the manipulability-evolution preference learned from human demonstrations to robot control. Rather than directly executing the direction preferred by the energy model, we use it as a local action-space hint for a residual policy. This design preserves the task-solving capability of the base policy while biasing exploration toward actions that induce human-like, contact-aware manipulability evolution.

Predicting Action-Induced Manipulability Change. At time t , the robot receives an RGB observation V_t^r , tactile signals T_t^r , and joint state q_t . The frozen visual–tactile encoder produces

$$z_t^r = F_\eta(V_t^r, T_t^r), \quad (7)$$

and the base policy proposes a nominal continuous action

$$a_t^0 \sim \pi_{\text{base},\psi}(\cdot \mid z_t^r, q_t). \quad (8)$$

We represent the robot’s current rotational manipulability using the same descriptor as for the human demonstrations: $m_t^r = \text{vech}(\log M_{\omega,t}^r) \in \mathbb{R}^6$, where $\text{vech}(\cdot)$ stacks the six unique entries of a symmetric 3×3 matrix.

To evaluate how a candidate action changes manipulability, let $u_t(a, q_t)$ denote the local joint displacement induced by action a under the robot controller. We define the resulting normalized manipulability-change direction as

$$\hat{v}_t^r(a) = \frac{m^r(q_t + u_t(a, q_t)) - m_t^r}{\|m^r(q_t + u_t(a, q_t)) - m_t^r\|_2 + \epsilon}. \quad (9)$$

This one-step estimate keeps the current contact geometry fixed, providing an efficient local approximation without requiring contact-dynamics rollouts. Contact changes are incorporated at the next control step through the updated tactile observations and robot state.

Energy-Guided Action Bias. We define the robot-side energy context as $c_t^r = [z_t^r, m_t^r, d_t]$, where d_t denotes the desired task rotation axis. To identify a locally preferred action without replacing the base policy, we construct a candidate set around its nominal action:

$$\mathcal{A}_t = \{a_t^0\} \cup \left\{ \text{clip}_{\mathcal{A}}(a_t^0 + \xi_{t,k}) \right\}_{k=1}^K, \quad (10)$$

$$\xi_{t,k} \sim \mathcal{N}(0, \sigma_a^2 I).$$

Each candidate is mapped to its induced manipulability-change direction through Eq. (9) and evaluated by the frozen energy model. The lowest-energy candidate then defines a local action-space bias:

$$a_t^* = \arg \min_{a \in \mathcal{A}_t} E_{\theta}(\dot{v}_t^r(a) \mid c_t^r), \quad (11)$$

$$b_t^E = \text{sg}[\text{clip}_{b_{\max}}(a_t^* - a_t^0)].$$

Here, $\text{sg}(\cdot)$ denotes the stop-gradient operator, and b_t^E indicates which local deviation from the nominal action is predicted to yield a more desirable manipulability evolution. Clipping limits the influence of this guidance and prevents the energy model from overriding the base policy.

Residual Policy and Training. Because the one-step energy score captures only a local preference and does not account for long-horizon task outcomes, we do not execute a_t^* directly. Instead, we condition a learned residual policy on the energy-guided bias:

$$\Delta a_t \sim \pi_{\text{res}, \phi}(\cdot \mid z_t^r, q_t, a_t^0, m_t^r, \text{sg}(b_t^E)), \quad (12)$$

$$a_t = \text{clip}_{\mathcal{A}}(a_t^0 + \lambda_R \Delta a_t).$$

where λ_R controls the magnitude of the residual correction, and $\text{clip}_{\mathcal{A}}(\cdot)$ projects the final command onto the valid action space $\mathcal{A}$.

During robot learning, the encoder F_{η} and energy model E_{θ} remain frozen, while the base and residual policies are optimized solely using the original task reward. Notably, our framework guides policy learning through an embodiment-agnostic preference over manipulability evolution, which differs from conventional approaches that use human demonstrations for hand-specific action imitation, reward shaping, or online optimization. Additional definitions, hyperparameters, and implementation details are provided in the supplementary material.

4 Experiments

We evaluate whether contact-conditioned manipulability evolution provides effective and reusable guidance for dexterous rotation. We first measure task success and generalization to held-out objects across three LEAP Hand tasks, as well as cap unscrewing with the Shadow, Allegro, and XHand, using the same frozen human-derived prior. We then conduct controlled ablations to isolate the benefit of state-aligned evolution guidance. Finally, we assess execution quality and demonstrate closed-loop deployment in the real world.

4.1 Experimental Setup

We evaluate our approach on three contact-rich rotation tasks: (1) *Unscrew Cap*, (2) *Rotate Object*, and (3) *Turn*

Faucet. A trial is considered successful if the cap, object, or faucet handle rotates by at least 2π about the target axis.

We pretrain the human-derived preference prior using demonstrations of *Unscrew Cap* and *Rotate Object*. To assess cross-task generalization, we exclude *Turn Faucet* from prior pretraining. We further enforce two criteria to evaluate object-level generalization rigorously: (1) objects appearing in the human demonstrations are excluded from robot-policy training and evaluation; and (2) the objects for each robot task are divided into *Seen* objects used for policy training and *Unseen* objects reserved exclusively for evaluation.

We conduct three categories of evaluation. (1) First, to determine whether the proposed prior improves task learning, we train a separate LEAP-hand policy for each task and evaluate its performance. (2) Second, to assess cross-task transfer, we train and evaluate a policy on *Turn Faucet* using the same frozen prior (this task is excluded from prior pretraining). (3) Third, to examine cross-hand transfer, we train Shadow, Allegro, and XHand policies for *Unscrew Cap* using the same task definition, reward function, and frozen preference prior. These evaluations collectively assess whether a single human-derived prior can facilitate policy learning across tasks, objects, and hand morphologies.

We evaluate three aspects of performance: (1) *task success*, (2) *generalization to unseen objects*, and (3) *execution quality*. We report the success rate (SR, %) separately for *Seen* and *Unseen* objects. Avg. SR denotes the unweighted mean across all conditions. All results are reported as the mean $\pm$ standard deviation over three independent evaluation seeds. For each seed, we evaluate 1,000 episodes per object with randomized initial hand poses.

4.2 Performance of the Proposed Approach

Generalization across tasks and objects. We compare DexMani against five baselines: (1) *PPO*, which learns solely from the task reward; (2) *VT Pretraining*, which learns a visual-tactile policy representation from human demonstrations via pretraining, without manipulability supervision or online guidance; (3) *VTM*, which additionally predicts contact-conditioned manipulability during pretraining, but does not utilize manipulability during policy learning; (4) *VTA*, which additionally guides policy learning using retargeted human actions; and (5) *VTA-E*, which additionally learns an energy model over retargeted actions and provides guidance through the same residual interface as DexMani. The details, and additional differences between these baselines are reported in the supplementary material.

The results are reported in Table 1 and reveal three main findings. First, DexMani achieves the highest SR across all six task-split conditions, attaining an Avg. SR of 57.5% and outperforming VTM, the strongest baseline, by 5.6 percentage points. Second, the improvements are particularly pronounced on *Unseen* objects, indicating that the learned guidance generalizes beyond the object instances and categories encountered during policy training. Third, DexMani performs strongly on *Turn Faucet*, despite this task being excluded from human-prior pretraining. This result demonstrates that the same frozen prior can transfer to a previously unseen rotation task.

Method	Unscrew Cap		Rotate Object		Turn Faucet		Avg. SR
	Seen	Unseen	Seen	Unseen	Seen	Unseen	
PPO (Schulman et al. 2017)	11.8 $\pm$ 2.1	5.0 $\pm$ 1.9	26.5 $\pm$ 0.7	21.5 $\pm$ 0.6	33.2 $\pm$ 3.5	20.5 $\pm$ 1.0	19.8
VT Pretraining	29.0 $\pm$ 1.5	12.7 $\pm$ 3.1	42.6 $\pm$ 1.4	34.0 $\pm$ 4.6	62.4 $\pm$ 4.7	53.5 $\pm$ 4.8	39.0
VTM	47.0 $\pm$ 0.9	27.8 $\pm$ 1.8	49.5 $\pm$ 3.0	37.9 $\pm$ 3.4	79.0 $\pm$ 1.2	70.1 $\pm$ 3.2	51.9
VTA	40.1 $\pm$ 0.8	25.1 $\pm$ 1.7	37.4 $\pm$ 5.8	20.3 $\pm$ 1.8	77.2 $\pm$ 2.2	63.5 $\pm$ 7.7	43.9
VTA-E	24.2 $\pm$ 2.0	11.6 $\pm$ 3.6	29.0 $\pm$ 6.3	19.1 $\pm$ 3.0	67.6 $\pm$ 1.9	42.0 $\pm$ 5.1	32.3
DexMani	65.1 $\pm$ 2.4	39.3 $\pm$ 2.5	50.7 $\pm$ 2.2	39.2 $\pm$ 3.1	80.2 $\pm$ 2.0	70.3 $\pm$ 2.9	57.5

Table 1: **Performance of generalization across tasks and objects.** Using a single frozen human-derived prior, DexMani achieves the highest mean SR in all six task-split settings, including Turn Faucet, which is absent from human pretraining.

Method	Shadow Hand		Allegro Hand		XHand		Avg. SR
	Seen	Unseen	Seen	Unseen	Seen	Unseen	
PPO (Schulman et al. 2017)	20.8 $\pm$ 2.6	6.8 $\pm$ 2.6	11.8 $\pm$ 1.6	5.2 $\pm$ 2.4	32.5 $\pm$ 3.3	17.2 $\pm$ 1.6	15.7
VTM	66.4 $\pm$ 4.4	40.9 $\pm$ 9.2	24.3 $\pm$ 3.9	12.5 $\pm$ 4.2	58.0 $\pm$ 1.4	22.7 $\pm$ 8.4	37.5
VTA-E	52.7 $\pm$ 3.2	24.5 $\pm$ 6.7	17.6 $\pm$ 3.5	9.8 $\pm$ 1.7	45.2 $\pm$ 2.8	18.2 $\pm$ 7.5	28.0
DexMani	70.0 $\pm$ 3.1	48.2 $\pm$ 4.8	34.6 $\pm$ 4.0	14.8 $\pm$ 2.3	63.4 $\pm$ 4.5	29.5 $\pm$ 1.9	43.4

Table 2: **Performance of Cross-hand generalization on cap-unscrewing task.** The same human-derived DexMani prior is reused across the Shadow, Allegro, and XHand platforms, with a separate policy trained in each hand-specific action space.

Method	Unscrew		Faucet		Avg. SR
	Seen	Unseen	Seen	Unseen	
Zero Guidance	18.6	9.4	48.6	20.2	24.2
Context-Shuffled	14.0	5.3	22.5	7.3	12.3
Greedy-M	54.5	36.3	67.8	44.7	50.8
DexMani	65.1	39.3	80.2	70.3	63.7

Table 3: **Ablation Studies.** All methods use the same training setup, differing only in the guidance signal.

The baseline comparisons provide two additional insights. First, the limited performance of VT Pretraining and VTM indicates that representation pretraining on human demonstrations alone, even with manipulability supervision, is insufficient to fully exploit the transferred knowledge without online policy guidance. Second, the performance of VTA and VTA-E suggests that retargeted human actions transfer less effectively across embodiments than the proposed manipulability-based representation. In contrast, DexMani transfers a preference over how rotational manipulability should evolve, allowing each robot to realize this preference according to its own morphology, and kinematic constraints. Consequently, a single prior can guide multiple rotation tasks while generalizing to previously unseen objects.

Generalization across hands. To assess whether the learned prior transfers across embodiments, we evaluate DexMani on three dexterous hands: (1) Shadow, (2) Allegro, and (3) XHand. For all three embodiments, we use the same *Unscrew Cap* task, reward function, and preference prior model, while training a separate policy in each hand’s native action space. As shown in Table 2, DexMani achieves the highest SR on

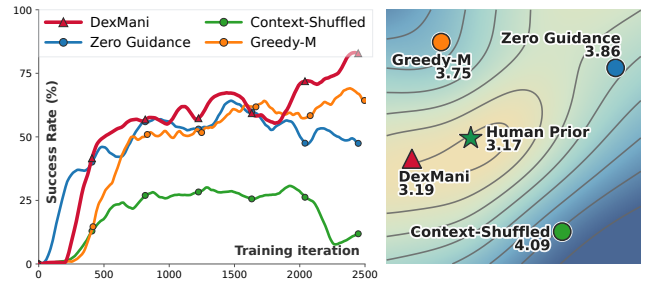


Figure 4: **Learning and energy analysis.** Left: Turn Faucet learning curves. Right: Projected action-energy landscape and selected actions under different guidance strategies.

both *Seen* and *Unseen* objects for all three hands. These consistent improvements demonstrate that manipulability-based guidance transfers across distinct hand kinematics.

4.3 Ablation Study

To understand why the manipulability preference prior improves policy performance, we investigate three questions: (1) Does manipulability-based guidance improve action selection? (2) If so, must this guidance be conditioned on the correct interaction context? (3) Is guiding a residual policy using the short-horizon evolution of human manipulability preferences more effective than greedily maximizing the robot’s instantaneous rotational manipulability? We address these questions using three ablations: *Zero Guidance*, which removes guidance from the residual policy’s input; *Context-Shuffled*, which replaces the correct guidance with mismatched interaction context; and *Greedy-M*, replaces the learned evolution guidance with a bias toward the candi-

Method	Unscrew Cap			Rotate Object			Turn Faucet		
	TCI $\uparrow$	LDLJ $\uparrow$	SPARC $\uparrow$	TCI $\uparrow$	LDLJ $\uparrow$	SPARC $\uparrow$	TCI $\uparrow$	LDLJ $\uparrow$	SPARC $\uparrow$
PPO	0.29 $\pm$ 0.11	-25.6 $\pm$ 1.4	-6.0 $\pm$ 0.5	0.43 $\pm$ 0.11	-14.5 $\pm$ 0.1	-2.6 $\pm$ 0.2	0.22 $\pm$ 0.05	-23.2 $\pm$ 2.3	-4.8 $\pm$ 0.8
VT Pretrain	0.41 $\pm$ 0.13	-24.5 $\pm$ 1.1	-5.5 $\pm$ 0.6	0.41 $\pm$ 0.23	-14.4 $\pm$ 0.1	-2.8 $\pm$ 0.2	0.23 $\pm$ 0.06	-23.2 $\pm$ 2.0	-4.5 $\pm$ 0.9
VTM	0.44 $\pm$ 0.17	-24.1 $\pm$ 0.8	-5.3 $\pm$ 0.7	0.26 $\pm$ 0.18	-15.5 $\pm$ 0.8	-3.3 $\pm$ 0.3	0.27 $\pm$ 0.15	-23.4 $\pm$ 1.8	-4.4 $\pm$ 0.4
VTA	0.45 $\pm$ 0.12	-23.7 $\pm$ 1.6	-5.0 $\pm$ 0.1	0.29 $\pm$ 0.09	-15.6 $\pm$ 0.9	-3.1 $\pm$ 0.2	0.23 $\pm$ 0.04	-23.9 $\pm$ 2.4	-4.1 $\pm$ 1.2
VTA-E	0.43 $\pm$ 0.05	-23.6 $\pm$ 0.4	-5.0 $\pm$ 0.3	0.23 $\pm$ 0.10	-13.9 $\pm$ 0.2	-2.8 $\pm$ 0.5	0.20 $\pm$ 0.02	-24.8 $\pm$ 0.4	-4.5 $\pm$ 0.8
DexMani	0.50 $\pm$ 0.13	-23.1 $\pm$ 0.5	-4.5 $\pm$ 0.4	0.36 $\pm$ 0.05	-11.9 $\pm$ 1.8	-2.3 $\pm$ 0.7	0.28 $\pm$ 0.07	-22.4 $\pm$ 0.5	-3.9 $\pm$ 0.1

Table 4: Quantitative comparison of execution quality and efficiency across three dexterous rotation tasks on the LEAP hand.

Method	Unscrew Cap	Rotate Object	Turn Faucet
DexMani	6/10	3/10	1/10

Table 5: Real-world DexMani deployment results.

date that maximizes instantaneous rotational manipulability, while retaining the same residual-policy interface.

As shown in Table 3, DexMani achieves the highest success rate across all conditions. Its improvements over *Zero Guidance* and *Context-Shuffled* confirm the benefits of manipulability-based guidance matched to the interaction context. The advantage over *Greedy-M* shows that the gain does not come from simply increasing instantaneous rotational capability. Instead, guidance learned from successful manipulability evolution provides a more effective signal for sustained rotation. Consistently, Figure 4 shows that DexMani learns faster and performs best on *Turn Faucet*. The action-energy landscape further reveals that DexMani selects an action in a low-energy region, whereas *Greedy-M* is only partially aligned with the human-derived prior. These results demonstrate the importance of interaction-conditioned, temporally informed guidance.

4.4 Motion Quality and Deployment

Beyond task success, we further assess execution quality in terms of rotational capability and motion smoothness. We quantify rotational capability using the Task Compatibility Index (TCI) (Chiu 1988), which measures the ability of the hand-object system to generate motion about the target rotation axis; a higher TCI indicates greater task-aligned rotational capability. We evaluate motion smoothness in both the time and frequency domains using log dimensionless jerk (LDLJ) (Hogan and Sternad 2009) and spectral arc length (SPARC) (Balasubramanian et al. 2015), respectively. Higher LDLJ and SPARC values indicate smoother motion. As shown in Table 4, DexMani achieves the best LDLJ and SPARC scores across all three tasks, indicating smoother and more coordinated object motions. It also obtains the highest TCI on *Unscrew Cap* and *Turn Faucet*, demonstrating a stronger ability to generate rotational motion aligned with the target axis.

Next, we deploy DexMani on a physical system consisting of a 16-DoF LEAP Hand (Shaw, Agarwal, and Pathak 2023) mounted on a 6-DoF xArm, as shown in Figure 5. The fingertips are equipped with TwinTac tactile sensors (Huang, Xu,



Figure 5: **DexMani real-world deployment.** The sequence shows closed-loop execution at 20 Hz in the real world.

and Xiao 2025), whose measurements are binarized to indicate contact. At each control step, the policy integrates these tactile signals with visual and proprioceptive observations to predict joint-position targets. To facilitate sim-to-real transfer, we apply domain randomization (Tobin et al. 2017) during simulation training, improving robustness to variations in physical dynamics and sensory observations. As reported in Table 5, DexMani successfully performs closed-loop rotation tasks on the physical system, demonstrating its ability to transfer from simulation to the real world despite discrepancies in sensing and dynamics.

5 Conclusion

We introduced DexMani, a framework that transfers human rotational manipulation experience through contact-conditioned manipulability evolution. By representing human preferences as manipulability evolution rather than hand-specific joint trajectories, DexMani improves task success, motion smoothness, and rotational capability while generalizing across tasks and objects. We also demonstrated its successful deployment on a physical LEAP Hand. These results establish manipulability evolution as a transferable representation of human experience across tasks, objects, and hand morphologies. One main limitation of this work is that, although the prior is shared, each task-hand configuration still requires a separately trained policy, precluding direct transfer of a unified policy across embodiments. Future work will focus on improving Sim2Real performance and developing unified policies that support direct cross-embodiment transfer.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 52305007), the Natural Science Foundation of Shanghai (Grant No. 25ZR1402370), the Artificial Intelligence Project of the State Key Laboratory of General Artificial Intelligence, BIGAI, Peking University, Beijing, China (Project No. SKLAGI2025OP19), the State Key Laboratory of Mechanical System and Vibration (Grant No. MSV202519) and the MoE Key Laboratory of Intelligent Perception and Human-Machine Collaboration (KLIP-HuMaCo).

References

- Balasubramanian, S.; Melendez-Calderon, A.; Roby-Brami, A.; and Burdet, E. 2015. On the analysis of movement smoothness. *Journal of neuroengineering and rehabilitation*, 12(1): 112.
- Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; and Dollar, A. M. 2015. Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set. *IEEE Robotics Automation Magazine*, 22(3): 36–52.
- Chiacchio, P.; Chiaverini, S.; Sciacivico, L.; Siciliano, B.; et al. 1991. Global task space manipulability ellipsoids for multiple-arm systems. *IEEE Transactions on Robotics and Automation*, 7(5): 678–685.
- Chiu, S. L. 1988. Task compatibility of manipulator postures. *The international journal of robotics research*, 7(5): 13–21.
- Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2021. Implicit Behavioral Cloning. *arXiv:2109.00137*.
- Friedman, J.; and Flash, T. 2007. Task-dependent selection of grasp kinematics and stiffness in human object manipulation. *Cortex*, 43(3): 444–460.
- Hogan, N.; and Sternad, D. 2009. Sensitivity of Smoothness Measures to Movement Duration, Amplitude and Arrests. *Journal of motor behavior*, 41: 529 – 534.
- Hsu, P.; Li, Z.; and Sastry, S. 1988. On grasping and coordinated manipulation by a multifingered robot hand. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 384–389 vol.1.
- Huang, X.; Xu, Z.; and Xiao, C. 2025. TwinTac: A Wide-Range, Highly Sensitive Tactile Sensor with Real-To-Sim Digital Twin Sensor Model. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12286–12292.
- Huang, Y.; Wu, J.; Jiang, J.; Lin, H.; Aierken, A.; Wang, Y.; Cheng, K.; Jiao, Z.; and Zhong, Y. 2026. HT-Bench: Benchmarking and Learning Dexterous Full-Hand Tactile Representations with Egocentric Vision. *arXiv preprint arXiv:2606.19161*.
- Jaquier, N.; Rozo, L.; Caldwell, D. G.; and Calinon, S. 2021. Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research*, 40(2-3): 624–650.
- Jaquier, N.; Rozo, L.; and Calinon, S. 2020. Analysis and transfer of human movement manipulability in industry-like activities. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11131–11138. IEEE.
- Juqiao Industrial. 2024. Juqiao Tactile Glove. Flexible piezoresistive fiber sensors for robotics. <https://jq-industries.com/>.
- LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, A.; and Huang, F. J. 2006. A Tutorial on Energy-Based Learning.
- Li, K.; Li, P.; Liu, T.; Li, Y.; and Huang, S. 2025a. Manip-trans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.
- Li, Z.; Liu, J.; Li, D.; Teng, T.; Li, M.; Calinon, S.; Caldwell, D.; and Chen, F. 2025b. ManiDP: Manipulability-Aware Diffusion Policy for Posture-Dependent Bimanual Manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9956–9962.
- Liu, Q.; Cui, Y.; Sun, Z.; Li, G.; Chen, J.; and Ye, Q. 2025. VTDexManip: A Dataset and Benchmark for Visual-Tactile Pretraining and Dexterous Manipulation with Reinforcement Learning. In *ICLR*.
- Lum, T. G. W.; Lee, O. Y.; Liu, K.; and Bohg, J. 2025. Crossing the Human-Robot Embodiment Gap with Sim-to-Real RL using One Human Demonstration. In *Conference on Robot Learning*, 4418–4441. PMLR.
- Luo, S.; Peng, Q.; Lv, J.; Hong, K.; Driggs-Campbell, K. R.; Lu, C.; and Li, Y. 2025. Human-Agent Joint Learning for Efficient Robot Manipulation Skill Acquisition. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 1370–1377.
- Makoviychuk, V.; Wawrzyniak, L.; Guo, Y.; Lu, M.; Storey, K.; Macklin, M.; Hoeller, D.; Rudin, N.; Allshire, A.; Handa, A.; and State, G. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. *arXiv:2108.10470*.
- Manus Meta. 2024. Manus VR Gloves. <https://www.manus-meta.com/>. Accessed: 2026-04.
- Meta. 2023. Meet Meta Quest 3, Our Mixed Reality Headset. <https://about.fb.com/news/2023/09/meet-meta-quest-3-mixed-reality-headset/>. Accessed: 2026-07-27.
- Morgan, A. S.; Hang, K.; Wen, B.; Bekris, K.; and Dollar, A. M. 2022. Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics and Automation Letters*, 7(2): 4821–4828.
- Pan, C.; Wang, C.; Qi, H.; Liu, Z.; Bharadhwaj, H.; Sharma, A.; Wu, T.; Shi, G.; Malik, J.; and Hogan, F. 2026. SPIDER: Scalable Physics-Informed Dexterous Retargeting. *arXiv:2511.09484*.
- Peng, X. B.; Ma, Z.; Abbeel, P.; Levine, S.; and Kanazawa, A. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Trans. Graph.*, 40(4).
- Qi, H.; Yi, B.; Suresh, S.; Lambeta, M.; Ma, Y.; Calandra, R.; and Malik, J. 2023. General In-Hand Object Rotation with Vision and Touch. In *Conference on Robot Learning (CoRL)*.

- Qin, Y.; Wu, Y.-H.; Liu, S.; Jiang, H.; Yang, R.; Fu, Y.; and Wang, X. 2022. DexMV: Imitation Learning for Dexterous Manipulation from Human Videos. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, 570–587. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-19841-0.
- Qin, Y.; Yang, W.; Huang, B.; Van Wyk, K.; Su, H.; Wang, X.; Chao, Y.-W.; and Fox, D. 2023. AnyTeleop: A General Vision-Based Dexterous Robot Arm-Hand Teleoperation System. In *Robotics: Science and Systems*.
- Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2018. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*.
- Reithmeir, A.; Figueredo, L.; and Haddadin, S. 2022. Human-to-Robot Manipulability Domain Adaptation with Parallel Transport and Manifold-Aware ICP. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5218–5225.
- Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6).
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Shaw, K.; Agarwal, A.; and Pathak, D. 2023. LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning. In *Robotics: Science and Systems*.
- Sun, D.; and Liao, Q. 2024. A Framework of Robot Manipulability Learning and Control and Its Application in Telerobotics. *IEEE Transactions on Fuzzy Systems*, 32(1): 266–280.
- Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 23–30. IEEE.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, 6000–6010. ISBN 9781510860964.
- Wang, J.; Yuan, Y.; Che, H.; Qi, H.; Ma, Y.; Malik, J.; and Wang, X. 2024. Lessons from Learning to Spin “Pens”. In *CoRL*.
- Wu, J.; Yao, S.; He, G.; Liu, X.; Zeng, Z.; Jiang, X.; Yang, H.; Zhang, W.; and Zhao, H. 2026. TopoRetarget: Interaction-Preserving Retargeting for Dexterous Manipulation. *arXiv:2606.16272*.
- Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; Yi, L.; Chang, A. X.; Guibas, L. J.; and Su, H. 2020. SAPIEN: A SimulATED Part-based Interactive ENvironment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yang, M.; chenghua lu; Church, A.; Lin, Y.; Ford, C. J.; Li, H.; Psomopoulou, E.; Barton, D. A.; and Lepora, N. F. 2024. AnyRotate: Gravity-Invariant In-Hand Object Rotation with Sim-to-Real Touch. In *8th Annual Conference on Robot Learning*.
- Ye, Q.; Liu, Q.; Wang, S.; Chen, J.; Cui, Y.; Jin, K.; Chen, H.; Cai, X.; Li, G.; and Chen, J. 2026. Visual-tactile pretraining and online multitask learning for humanlike manipulation dexterity. *Science Robotics*, 11(110).
- Yin, Z.-H.; Huang, B.; Qin, Y.; Chen, Q.; and Wang, X. 2023. Rotating without Seeing: Towards In-hand Dexterity through Touch. In *Robotics: Science and Systems*. ISBN 978-0-9923747-9-2.
- Yokokohji, Y.; San Martin, J.; and Fujiwara, M. 2009. Dynamic manipulability of multifingered grasping. *IEEE Transactions on Robotics*, 25(4): 947–954.
- Zhang, C.; Cai, P.; Xi, Z.; Yuan, H.; Luo, H.; Zhang, W.; Zheng, S.; Xu, C.; and Lu, Z. 2026. Human-Centric Transferable Tactile Pre-Training for Dexterous Robotic Manipulation. *arXiv:2607.01067*.

Supplementary Material

1 Human Dataset and Hardware Details

Human demonstrations provide supervision for both the visual–tactile encoder and the prior model of contact-conditioned rotational manipulability evolution. To explain how these data are collected, we briefly describe the data collection system and the processing procedure of the dataset.

Data collection system. We collect synchronized multi-modal recordings of human object rotation using a wearable acquisition system. The system captures RGB video, right-hand kinematics, and dense tactile measurements at 30 Hz. Hand kinematics are recorded using a Manus Quantum MetaGlove (Fig. 6). The Manus hand model provides the three-dimensional positions of 21 hand joints. Contact signals are captured using a 256-channel Jujiao piezoresistive tactile glove. At each frame, the tactile glove outputs a pressure vector $u_t \in \{0, \dots, 255\}$, where each element corresponds to a tactile taxel and larger values indicate stronger local pressure responses. Two RGB cameras record the hand–object interaction at a resolution of 640×480 . All sensor streams are temporally aligned using timestamps.



Figure 6: **Hardware for human data collection.** Left: the Jujiao tactile glove. Right: the Manus Quantum MetaGlove.

As shown in Fig. 7, the human-demonstration dataset contains 27 objects, including common geometric primitives and everyday household items. This collection encompasses diverse object geometries and contact patterns for learning the evolution of rotational manipulability.

Visual processing. We use the recorded human-hand trajectories to animate a MANO hand model (Romero, Tzionas, and Black 2017). The resulting hand motions are rendered in simulation to provide additional RGB observations. During pretraining, both the captured RGB images and the MANO-rendered images are used as visual inputs.

Tactile processing. We partition the raw tactile taxels into five semantic fingertip regions corresponding to the thumb, index, middle, ring, and little fingers. For each finger i , we aggregate the responses of the taxels assigned to its fingertip region. The corresponding taxel set, S_i , is determined by the



Figure 7: Objects used for human demonstration collection.

physical layout of the Jujiao tactile glove. The finger-level tactile response is defined as

$$f_{i,t} = \sum_{\ell \in S_i} |u_{\ell,t}|. \quad (13)$$

We compute a contact threshold separately for each recording r :

$$\tau_r = 0.15 \text{ Percentile}_{95}(\{f_{i,t} \mid i \in \{1, \dots, 5\}, t \in r\}). \quad (14)$$

The binary contact state is then given by

$$c_{i,t} = \mathbb{I}[f_{i,t} > \tau_r]. \quad (15)$$

Only fingertips for which $c_{i,t} = 1$ are included when constructing the contact Jacobian and grasp matrix (Hsu, Li, and Sastry 1988). For visual–tactile pretraining, the tactile input to the encoder is

$$T_t = [(c_{1,t}), \dots, (c_{5,t})]. \quad (16)$$

Thus, the encoder receives five finger-level tokens rather than the raw tactile vector.

During RL training, we do not match individual taxels across different hands. Instead, we group the tactile sensors by finger and convert each finger’s response into a binary contact state. We use the same contact threshold for all robot hands, setting the simulated force threshold to 0.01 N. For four-fingered hands, the little-finger entry is set to zero. Consequently, the method requires only finger-level contact information and does not rely on identical sensor layouts, taxel-level correspondence, or contact-normal estimation.

2 Additional Method Details

This section provides the mathematical details omitted from the main text for clarity. We first describe how human manipulability labels are analytically computed from hand kinematics and tactile contacts. We then detail the manipulability-evolution energy objective and the robot-side realization of the residual action guidance.

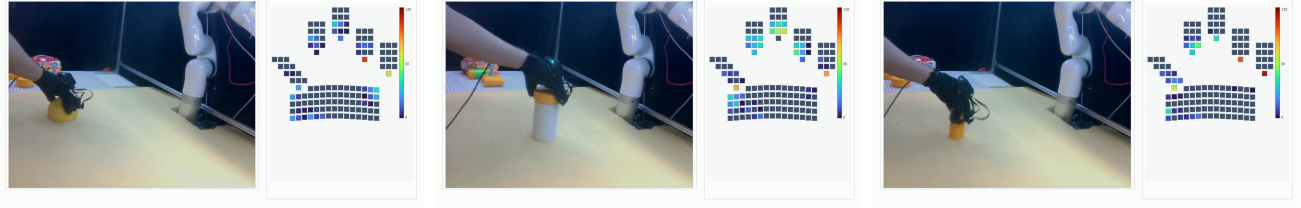


Figure 8: **Human Dataset Visualization.** Each pair shows the RGB observation and the corresponding tactile-pressure map.

2.1 Contact-conditioned Manipulability

We compute contact-conditioned manipulability from the human hand configuration and the tactile contact state. The computation contains three steps. We first construct the human hand Jacobian from the Manus hand model. Then we modulate the Jacobians with the active contact information. Finally, we map the resulting motion capability to the object rotational space.

Human hand Jacobian. The Manus glove provides hand poses at each frame. Each finger is represented as a serial kinematic chain. For an active fingertip contact i , its positional Jacobian is obtained by differentiating the forward kinematics of the corresponding finger:

$$J_{i,t} = \frac{\partial p_{i,t}}{\partial q_t}, \quad (17)$$

where q_t denotes the human hand configuration and $p_{i,t}$ denotes the fingertip position. We stack the Jacobians of all active contacts into

$$J_{c,t}^{\text{hum}} = \begin{bmatrix} J_{1,t} \\ \vdots \\ J_{N_t,t} \end{bmatrix}, \quad \dot{p}_t = J_t \dot{q}_t, \quad (18)$$

where N_t is the number of active contacts.

We compute the human contact-conditioned rotational manipulability as

$$\begin{aligned} C_{c,t}^{\text{hum}} &= W_t^{1/2} J_{c,t}^{\text{hum}} H_h (J_{c,t}^{\text{hum}})^\top W_t^{1/2}, \\ M_{\omega,t}^{\text{hum}} &= P_\omega (G_t^+)^{\top} C_{c,t}^{\text{hum}} G_t^+ P_\omega^\top + \epsilon_m I_3. \end{aligned} \quad (19)$$

The terms in Eq. (19) are defined below.

Contact activation matrix W_t . In the general formulation, W_t is a block-diagonal matrix that removes inactive fingertip contacts. In our implementation, inactive contacts are removed before constructing $J_{c,t}^{\text{hum}}$ and G_t . Therefore,

$$W_t = I_{3N_t}, \quad (20)$$

and no additional contact masking is applied.

Joint-space scaling matrix H_h . The matrix $H_h = \text{diag}(h_1, \dots, h_{n_h})$ controls the contribution of each human joint direction. A larger h_j gives the j -th joint direction a larger contribution to $C_{c,t}^{\text{hum}}$. In our implementation, we set $H_h = I_{n_h}$ and weight all joint directions equally.

Grasp matrix G_t . Let $p_{i,t}$ be the position of active contact $i \in \mathcal{A}_t$ in the wrist frame. We use the center of the active contacts as the local reference point:

$$\bar{p}_t = \frac{1}{N_t} \sum_{i \in \mathcal{A}_t} p_{i,t}. \quad (21)$$

The relative contact position is

$$r_{i,t} = p_{i,t} - \bar{p}_t. \quad (22)$$

The standard grasp matrix is

$$G_t = \begin{bmatrix} I_3 & \cdots & I_3 \\ [r_{1,t}]_\times & \cdots & [r_{N_t,t}]_\times \end{bmatrix}, \quad (23)$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix. We use the damped pseudoinverse

$$G_t^+ = G_t^\top (G_t G_t^\top + \lambda_G I_6)^{-1}. \quad (24)$$

The term $(G_t^+)^{\top} C_{c,t}^{\text{hum}} G_t^+$ maps the contact-space motion capability into the object-motion space. The grasp mapping first produces a 6×6 object-space manipulability matrix. Its first three dimensions describe translation, and its last three dimensions describe rotation. We use

$$P_\omega = [0_{3 \times 3} \quad I_3] \quad (25)$$

to select the rotational component. The resulting $M_{\omega,t}^{\text{hum}} \in \mathbb{R}^{3 \times 3}$ describes the available object rotation about the three wrist-frame axes.

Finally, we convert the rotational manipulability matrix into a six-dimensional descriptor:

$$m_t^{\text{hum}} = \text{vech}(\log M_{\omega,t}^{\text{hum}}) \in \mathbb{R}^6. \quad (26)$$

DexMani learns the normalized short-horizon changes of m_t^{hum} . These changes describe how rotational capability evolves with the hand configuration and active contacts.

Cross-embodiment convention. Here, we explain why the contact-conditioned manipulability enables cross-embodiment applications. Specifically, the descriptor $m_t = \text{vech}(\log M_{\omega,t})$ is not invariant to arbitrary coordinate changes. We instead use a shared wrist-frame convention for human demonstrations and robot rollouts. Fingertip positions and Jacobians are expressed in wrist frame before manipulability is computed. In all experiments, the target rotation direction is represented by the hand-frame z -axis, i.e.,

$d = [0, 0, 1]$. Tactile readings are binarized and used only to select active fingertip contacts.

After this alignment, the components of m_t refer to the same wrist-frame rotation axes for both the human and the robot. The human-trained energy model can therefore score the manipulability changes induced by robot actions in the same coordinate system. No joint-level correspondence or action retargeting is required. Each robot produces a preferred change through its own joint motion and contact transitions.

2.2 Robot-Side Residual Action

Algorithm 1: Robot-side update at one control step.

Require: Frozen encoder F_η and energy model E_θ

Require: Robot observation (V_t^r, T_t^r, q_t) and direction d_t

- 1: Compute the visual-tactile feature $z_t^r = F_\eta(V_t^r, T_t^r)$.
- 2: Compute the current descriptor $m_t^r = \text{vech}(\log M_{\omega,t}^r)$.
- 3: Sample the nominal action $a_t^0 \sim \pi_{\text{base},\psi}(\cdot \mid z_t^r, q_t)$.
- 4: Construct the local candidate set

$$\mathcal{A}_t = \{a_t^0\} \cup \{\text{clip}_{\mathcal{A}}(a_t^0 + \xi_k)\}_{k=1}^K.$$

- 5: Estimate $\hat{v}_t^r(a)$ for each candidate using the current contact geometry.
- 6: Discard candidates with near-zero predicted manipulability change.
- 7: Set the energy context $c_t^r = [z_t^r, m_t^r, d_t]$.
- 8: Score each remaining candidate by $S_t(a) = E_\theta(\hat{v}_t^r(a) \mid c_t^r)$.
- 9: Select $a_t^* = \arg \min_{a \in \mathcal{A}_t} S_t(a)$.
- 10: Compute the stop-gradient bias

$$b_t^E = \text{sg}[\text{clip}_{b_{\max}}(a_t^* - a_t^0)].$$

- 11: Sample the residual action

$$\Delta a_t \sim \pi_{\text{res},\phi}(\cdot \mid z_t^r, q_t, a_t^0, m_t^r, b_t^E).$$

- 12: Execute

$$a_t = \text{clip}_{\mathcal{A}}(a_t^0 + \lambda_R \Delta a_t).$$

- 13: Update $\pi_{\text{base},\psi}$ and $\pi_{\text{res},\phi}$ with PPO.

This section provides the implementation details used in our methods. At each control step, the base policy produces a normalized action a_t^0 . We sample eight nearby actions around a_t^0 . Each candidate action is rescaled to a target joint configuration in Isaac Gym.

For a candidate a , we compute the joint displacement

$$u_t(a, q_t) = q_t^{\text{tar}}(a) - q_t, \quad (27)$$

where q_t is the current joint configuration and $q_t^{\text{tar}}(a)$ is the target specified by a . We use the robot fingertip Jacobian to calculate the resulting fingertip motion:

$$J_{\text{tip},t} = \frac{\partial p_{\text{tip}}(q_t)}{\partial q_t}, \quad (28)$$

$$\Delta p_t(a) = J_{\text{tip},t} u_t(a, q_t).$$

The candidate contact positions are obtained by adding $\Delta p_t(a)$ to the current positions of the active fingertips, the

contact Jacobian is recomputed at the candidate joint configuration $q_t^{\text{tar}}(a)$, these contact positions and Jacobian are then used to construct the grasp matrix and the corresponding rotational manipulability $M_{\omega,t}^r(a)$. Finally, we convert $M_{\omega,t}^r(a)$ into its log-Euclidean descriptor and normalize its difference from the current descriptor to obtain $\hat{v}_t^r(a)$, following Eq. (9) in the main paper. The frozen energy-based model (EBM) (LeCun et al. 2006) scores the manipulability change induced by each of the nine candidates, including the nominal action. A lower energy indicates better agreement with the human-derived manipulability-evolution prior.

2.3 Computational Cost.

We profile one training iteration on an NVIDIA RTX 4090. PPO takes 5.77 s per iteration, while DexMani takes 11.50 s. The detailed runtime breakdown is shown in Table 6. The local manipulability computation and EBM scoring together take 0.434 s.

Method	Time (s)
PPO	5.773
VT Pretraining	9.256
VTM	10.447
VTA	9.436
VTA-E	10.823
DexMani	11.497
DexMani components	
Local manipulability computation	0.410
EBM context and candidate scoring	0.024
Total local guidance computation	0.434

Table 6: Runtime profiling.

3 Training Details.

We implement all baseline methods under the same task environments, object splits, reward functions, and PPO settings. The methods differ only in the human supervision used during pretraining and whether online guidance is provided during RL.

VT pretrain. We implement VT following the visual-tactile joint pretraining method of (Liu et al. 2025). As shown in Fig. 9, each training sample contains a visual-tactile history, $(V_{t-H+1:t}, T_{t-H+1:t})$, with $H = 8$. Each RGB frame is divided into image patches, while each tactile frame is represented by finger level tokens. Tokens from the temporal window are randomly masked, and a transformer encoder fuses the remaining visual and tactile tokens. A lightweight decoder reconstructs the masked inputs. The network and training settings are listed in Table 7. After pretraining, the decoder is discarded and the encoder is frozen. The fused visual-tactile feature is concatenated with robot proprioception and used as input to the PPO policy.

VTM. As shown in Fig. 10, VTM extends VT with manipulability prediction during pretraining. The current contact-conditioned rotational manipulability and task axis are added

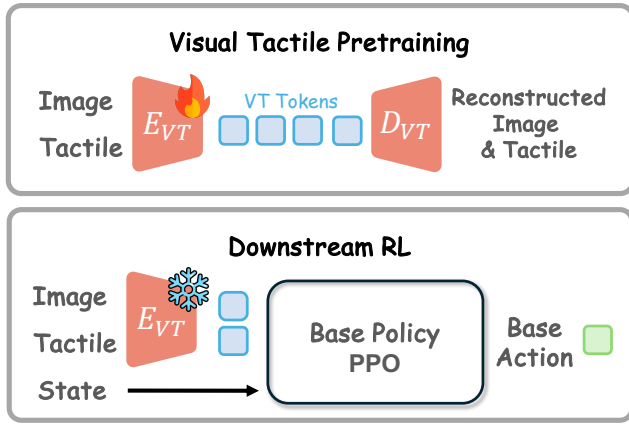


Figure 9: VT pretraining pipeline.

Hyperparameter	All Models
History length H	8
Image resolution per frame (H_v, W_v)	(224, 224)
Image patches per frame N_v	196
Tactile tokens per frame N_c	5
Patch resolution (P, P)	(16, 16)
Encoder embedding dimension d_{en}	384
Image mask ratio γ_v	0.75
Tactile mask ratio γ_c	0.50
Fusion dimension d	384
Decoder embedding dimension d_{de}	192
Image loss weight λ_v	1
Tactile loss weight λ_c	10
Learning rate	1.5×10^{-4}
Batch size	1024

Table 7: Hyperparameters of visual-tactile pretraining.

to the context c_t , predicts four future manipulability descriptors sampled at a stride of four raw frames. The model is trained with manipulability, direction, and magnitude prediction losses. The energy-loss weight is set to zero, so VTM does not learn an energy field. The detailed settings are listed in Table 8.

VTA. As shown in Fig. 11, VTA follows the same pretraining and downstream protocol as VTM. The main difference is the supervision used during pretraining. VTA replaces the manipulability token in the context c_t with the human action recorded in the dataset. The prediction head also predicts future human actions instead of future manipulability. All other pretraining settings, including the prediction horizon, loss weights, training epochs, batch size, and learning rate, are the same as those of VTM.

VTA-E. As shown in Fig. 12, VTA-E follows the same energy-guided training procedure as VTM-E. The main difference is the representation used by the energy model. During pretraining, VTA-E learns an energy field over human actions. Its auxiliary prediction head also predicts future human actions instead of future manipulability. During downstream

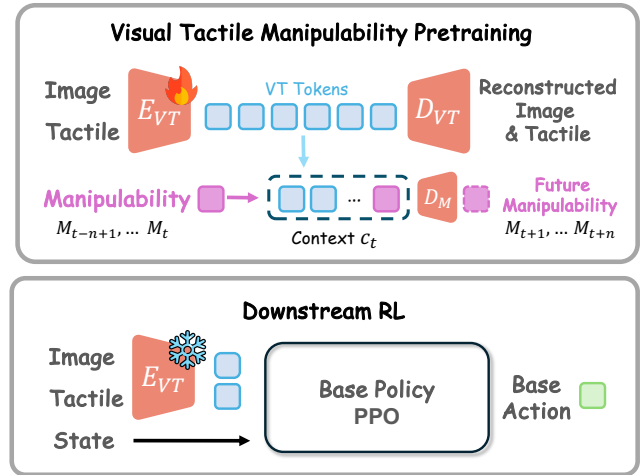


Figure 10: VTM pretraining pipeline.

Hyperparameters	Value
Future prediction horizon	4
Manipulability loss weight	1.0
Direction loss weight	0.2
Magnitude loss weight	0.1
Energy loss weight	0.0
Batch size	1024
Learning rate	1.5×10^{-4}

Table 8: Hyperparameters of VTM.

RL, both the visual-tactile encoder and the human-action energy model remain frozen. The current robot-hand pose and the poses induced by candidate actions are retargeted to the human-hand space using AnyTeleop (Qin et al. 2023). The energy model then scores the corresponding human-action changes, following the same procedure as VTM-E.

Manipulability Evolution Energy Learning. During pretraining, manipulability descriptors are sampled every four raw frames. We use the normalized change from m_t to m_{t+4} as the positive evolution direction. Each positive sample is paired with eight negatives, including its reversed direction, and randomly sampled directions. Conditioned the context c_t , the energy head assigns a scalar energy to each candidate direction, with lower values indicating better agreement with the human demonstrations. After pretraining, the visual-tactile encoder, context encoder, and energy head are frozen for downstream RL. The detailed settings are listed in Table 9.

PPO Training Details. All downstream policies are trained with PPO using the same optimization settings. We run 200 parallel environments and each policy update uses rollouts of 32 steps. The collected samples are divided into four mini-batches and optimized for ten epochs.

The actor and critic use multilayer perceptrons with hidden dimensions [1024, 1024, 512]. ELU is used as the activation function. The learning rate is 3×10^{-4} . The PPO clipping range is 0.2, and the gradient norm is clipped at 1.0. We

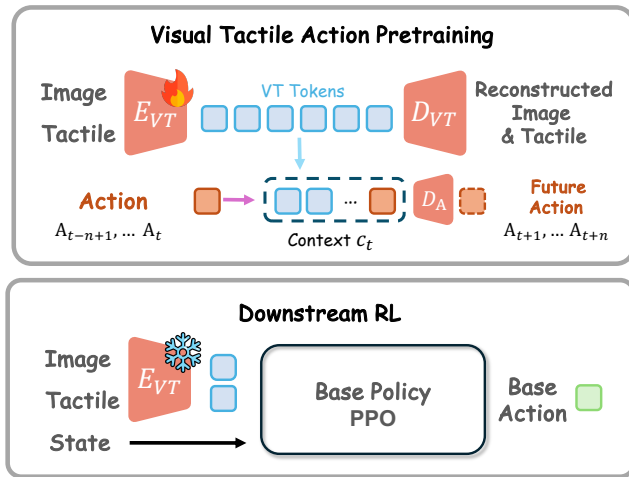


Figure 11: VTA pretraining pipeline.

Hyperparameter	Value
Temporal offset Δ	4
Prediction horizon	4
Number of negatives K	8
Negative composition	8
Stationary threshold	10^{-4}
Temperature τ	0.1
Energy loss weight λ_E	0.1
Magnitude loss weight λ_α	0.1
Learning rate	1.5×10^{-4}
Batch size	1024
Training epochs	400

Table 9: Hyperparameters for energy pretraining.

use a discount factor of $\gamma = 0.96$ and a GAE parameter of $\lambda_{\text{GAE}} = 0.95$. The initial policy noise standard deviation is 0.8. The desired KL divergence is 0.016, and no entropy bonus is applied.

The maximum episode length is 500 steps for Unscrew Cap and Turn Faucet, and 600 steps for Rotate Object. The same PPO settings are used for all methods. The pretrained encoders and energy models remain frozen during downstream policy training.

Domain Randomization. We apply domain randomization (Tobin et al. 2017) during simulation training to reduce the gap between simulated and real observations. The randomization covers proprioception, appearance, camera configuration, and tactile sensing.

Zero-mean Gaussian noise is added to the joint angles and joint velocities at each observation step. Object color and light direction are randomized when the environment is reset. We also perturb the camera position and look-at point at each reset. Image brightness, contrast, and pixel values are randomized independently for every frame. Finally, Gaussian noise is added to the tactile force readings before contact binarization. The complete settings are listed in Table 11.

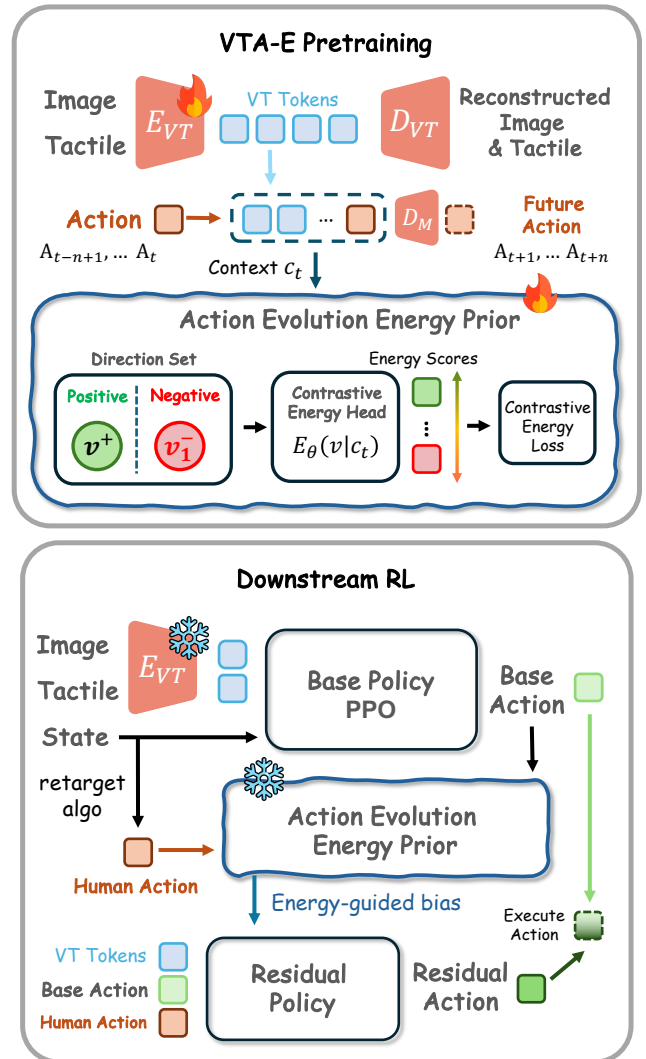


Figure 12: VTA-E pretraining pipeline.

At deployment, the real visual, tactile, and proprioceptive observations are processed using the same input format as in simulation. The tactile signals are binarized before being passed to the policy. No real-world policy fine-tuning is performed.

4 Experiment Details

DexMani uses human demonstrations to pretrain the visual-tactile encoder and the prior over contact-conditioned rotational manipulability evolution.

4.1 Task Details

We evaluate DexMani on three contact-rich object rotation tasks: *Unscrew Cap*, *Rotate Object*, and *Turn Faucet*. These tasks require different contact patterns and finger coordination strategies, while sharing the goal of sustained object rotation. Figure 13 shows the three LEAP Hand environments, and the setup used for cross-hand evaluation on the Shadow

Hyperparameter	Value
Parallel environments	200
Rollout length	32
Number of mini-batches	4
Optimization epochs	10
Hidden dimensions	[1024, 1024, 512]
Activation	ELU
Learning rate	3×10^{-4}
PPO clip range	0.2
Maximum gradient norm	1.0
Discount factor γ	0.96
GAE parameter λ_{GAE}	0.95
Initial action-noise std.	0.8
Desired KL divergence	0.016
Entropy coefficient	0

Table 10: PPO training hyperparameters.

Parameter	Randomization strategy
Joint angle	$+\mathcal{N}(0, 0.05)$, per observation
Joint velocity	$+\mathcal{N}(0, 0.05)$, per observation
Object color	Randomized at environment reset
Light direction	Randomized at environment reset
Camera eye position	$\mathcal{U}[-4 \text{ cm}, 4 \text{ cm}]$, at reset
Camera look-at position	$\mathcal{U}[-4 \text{ cm}, 4 \text{ cm}]$, at reset
Image brightness	$\mathcal{U}[-20, 20]$, per frame
Image contrast	$\mathcal{U}[-8, 8]$, per frame
Image pixel noise	$\mathcal{U}[-5, 5]$, per frame
Tactile force	$+\mathcal{N}(0, 0.1)$, before binarization

Table 11: **Domain randomization settings.** $\mathcal{N}$ and $\mathcal{U}$ denote Gaussian and uniform distributions, respectively.

Hand, Allegro Hand, and XHand in IsaacGym (Makoviychuk et al. 2021). Figure 14 shows the training and evaluation objects used in the simulation tasks. Objects inside the black dashed boxes are excluded from policy training and used only for unseen-object evaluation. The simulation environments and reward functions are adapted from VT DexManip (Liu et al. 2025). We adjust the task settings to match our evaluation protocol. The setup, success criterion, object split, and reward function of each task are described below.

Unscrew Cap. This task evaluates coordinated finger motion during bottle-cap rotation. Each bottle is fixed to the table. The cap has one rotational degree of freedom about its local z -axis. The goal is to rotate the cap counterclockwise by one full revolution within an episode.

The reward is

$$r = \lambda_1 r_p + \lambda_2 r_v + \lambda_3 r_d + \lambda_4 r_s. \quad (29)$$

The rotation term is $r_p = \min(\theta_{\text{joint}}, 7.0)$, where θ_{joint} is the cap rotation angle. The velocity term is $r_v = \text{clamp}(v_{\text{joint}}, -10, 10)$. The distance term is $r_d = \exp(-10d)$, where d is the summed distance from the fingertips to a reference point located 2 cm below the cap. The

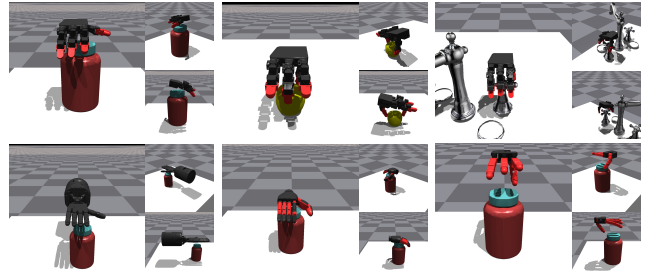


Figure 13: **Simulated task settings.** Top: Unscrew Cap, Rotate Object, and Turn Faucet on the LEAP Hand. Bottom: cap unscrewing with the Shadow, XHand, and Allegro Hand.

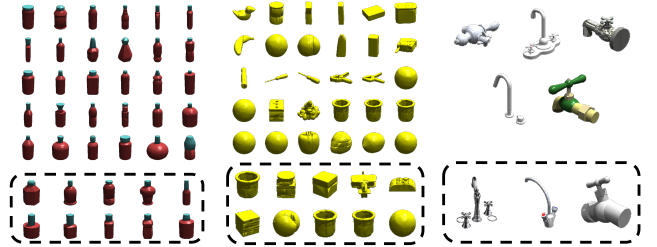


Figure 14: **Objects used in the three simulation tasks.** Objects inside the black dashed boxes are held out for unseen-object evaluation.

success term is $r_s = 5$. The reward weights are listed in Table 12.

Reward term	Weight
Rotation position	$\lambda_1 = 0.5$
Angular velocity	$\lambda_2 = 1$
Fingertip distance	$\lambda_3 = 0.5$
Success	$\lambda_4 = 1$

Table 12: Reward weights for Unscrew Cap.

The velocity reward is disabled when the cap has positive angular velocity but no tactile contact is active. This encourages the hand to rotate the cap through fingertip contact.

Rotate Object. In this task, the dexterous hand rotates an object on a table without toppling it. The task requires coordination between the thumb and the other fingers. The goal is to rotate the object by 2π while keeping it upright. We use thirty YCB objects (Calli et al. 2015) for policy training and ten held-out objects for evaluation.

The reward is

$$r = \lambda_1^\top r_d + \lambda_2 r_{\text{rot}} + \lambda_3 r_a + \lambda_4 r_v + r_b. \quad (30)$$

The distance term is $r_d = [\exp(-10d_1), d_2]^\top$. Here, d_1 is the vertical distance between the fingertips and the object. The term d_2 is the planar distance between the object and its target position. The term r_{rot} measures the difference between the current and target object poses. The action penalty is $r_a = \|a\|_2$. The velocity term is

$r_v = \text{clamp}(\omega_z, -10, 10)$, where ω_z is the object angular velocity about the z -axis. A success bonus of $r_b = 250$ is added when the task is completed. The reward weights are listed in Table 13.

Reward term	Weight
Distance	$\lambda_1 = [0.25, -10]^\top$
Pose difference	$\lambda_2 = 1$
Action penalty	$\lambda_3 = -2 \times 10^{-4}$
Angular velocity	$\lambda_4 = 1$
Success bonus	1

Table 13: Reward weights for Rotate Object.

Turn Faucet. This task requires the dexterous hand to rotate a faucet handle. Unlike Unscrew Cap, the fingers must push against the handle rather than twist a cylindrical cap. The goal is to rotate the handle clockwise by one full revolution. We use five faucet models from the SAPIEN dataset (Xiang et al. 2020) for policy training. All models have a rotation axis perpendicular to the ground. For evaluation, we use three unseen instances. The reward for faucet turning follows the same formulation as Unscrewing cap. The main difference is that the velocity reward is always applied and is not gated by tactile contact. In addition, we encourage the fingertips to approach task-specific target positions around the faucet handle. The distance term is computed as the sum of the distances from each fingertip to its corresponding target position.

4.2 Additional Analysis

Learning efficiency. Figures 15 and 16 compare learning progress in simulation. On the LEAP Hand, DexMani reaches strong cap-unscrewing performance earlier and converges to a higher final success rate than the baselines. The same trend holds across the Shadow, Allegro, and XHand embodiments, despite their different kinematics and action spaces. These results show that the human-derived guidance improves both learning efficiency and final policy performance, and that this benefit remains consistent across hand morphologies.

Figure 17 further shows six representative simulated rollouts. Across different settings, DexMani maintains object progress while releasing and re-establishing finger contacts, producing sustained rotation without prescribing a fixed contact sequence. Together with the learning curves, these examples illustrate that the human-derived guidance improves both policy learning and execution.

Real-world experiment. The Faucet task presents the largest distribution shift in our real-world evaluation. Neither faucet-turning demonstrations nor the physical faucet object appear in the human pretraining data, and the physical faucet also differs from the simulated objects used for robot-policy training. The policy must therefore handle a new task-specific contact pattern, unseen object geometry, and the sim-to-real sensing and dynamics gap simultaneously. We attribute the low success rate mainly to this combined task and object distribution shift.

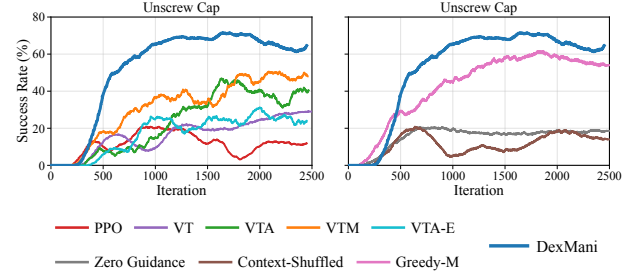


Figure 15: Learning curves of unscrew cap Task.

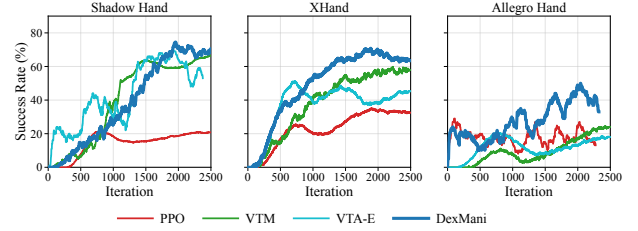


Figure 16: Learning curves of cross-hand unscrew cap Task.

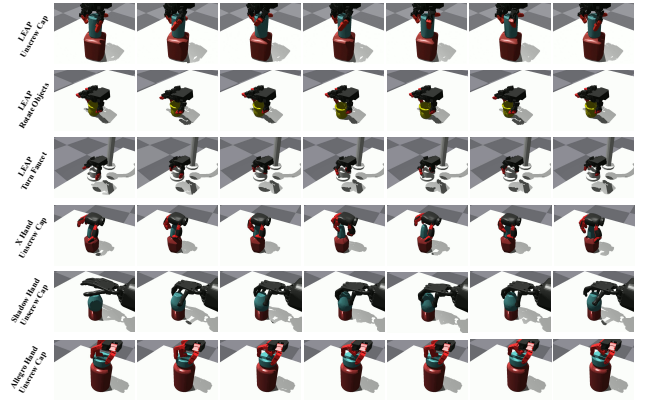


Figure 17: **Representative simulated rollouts.** Six sequences illustrate sustained rotation through repeated contact transitions across the evaluated settings.