

VibeAct: Vibration to Actions for Contact-Rich Reactive Robot Dexterity

Yuemin Mao^{*,1}, Uksang Yoo^{*,1}, Jean Oh¹, Jonathan Francis^{1,2}, Jeffrey Ichnowski¹

¹Carnegie Mellon University

²Bosch Center for Artificial Intelligence

^{*}Equal contribution.

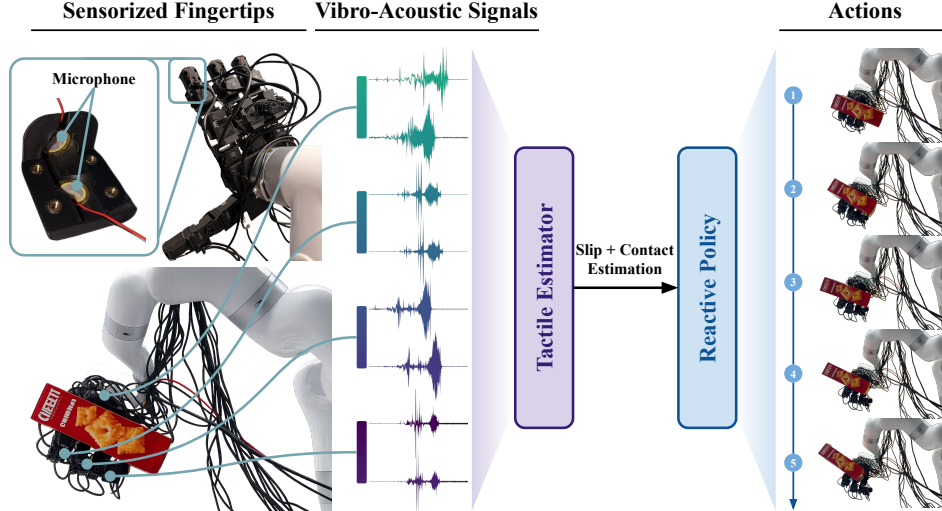


Figure 1: **VIBEACT** connects real vibrotactile sensing to simulation-based policy learning through an explicit intermediate representation of contact and slip. A tactile estimator infers this representation from microphone signals, and the policy learns to act on the same representation in simulation.

Abstract: Dexterous manipulation depends on contact events that are fast, local, and often visually occluded. Piezoelectric microphones offer a compact and high-bandwidth way to sense these interactions, but the resulting vibro-acoustic signals are difficult to simulate faithfully enough for end-to-end sim-to-real policy learning on dexterous robot hands. We propose VIBEACT, a framework that bridges real vibrotactile sensing and simulation-based reinforcement learning through a shared physical representation of contact and slip. In the real world, we embed piezoelectric microphones into a dexterous robot hand and collect vibro-acoustic data through teleoperation, then replay the recordings in a calibrated digital clone to automatically label per-finger contact and slip. A tactile estimator learns to predict contact and slip from real microphone waveforms, while manipulation policies are trained in simulation on the same representation computed directly from simulated contacts. This decoupling lets policies exploit rapid tactile feedback without simulating raw audio. Across five contact-rich tasks spanning regrasping, in-hand reorientation, and insertion, VIBEACT consistently outperforms a proprioception-and-point-cloud baseline in simulation, with the largest gains on tasks requiring sustained reactive control, where the continuous slip-magnitude channel proves the most informative observation. The learned policies transfer to a physical dexterous hand-arm platform, improving success rates on deployed tasks. Project videos and additional details are at vibeact.github.io.

Keywords: Dexterous Manipulation, Tactile Sensing

1 Introduction

Open-world manipulation requires rapid responses to contact and slip events [1]. When a fingertip first makes contact, or when a grasp transitions from sticking to sliding, the relevant hand-object interaction cues are often brief, subtle, or visually occluded [2]. We present VIBEACT, a framework that estimates these contact and slip interactions using piezoelectric microphones embedded in a robot hand and leverages them as tactile observations for learned dexterous manipulation policies.

Piezoelectric microphones offer a promising tactile sensing modality for robot hands. Compared with popular vision-based tactile sensors such as GelSight [3] and Digit [4], they are inexpensive, compact, and high-bandwidth. Additionally, they can be mounted away from the contact surface while still capturing structure-borne vibrations generated by impacts, stick-slip motion, and sliding [5, 6, 7, 8]. The cost of these advantages is that the recorded waveform depends not only on the underlying contact event but also on finger material, adhesive, mounting location, object texture, amplifier chain, and background motor vibrations [9]. As a result, simulating vibro-acoustic signals accurately enough for end-to-end policy learning is an open challenge.

On the other hand, learning dexterous manipulation directly from real-world vibrotactile observations is equally impractical. Collecting large-scale contact-rich demonstrations is labor-intensive, while online reinforcement learning is prohibitively expensive and potentially unsafe due to unstable exploration during early training. Together, these constraints motivate a representation that can bridge real vibrotactile sensing and simulation-based control.

The central insight of this work is that contact and slip provide an interface between sensing and control for sim-to-real policy learning. Contact onset, binary slip, and scalar slip magnitude are low-dimensional and task-relevant quantities that can both be estimated from real microphone signals and computed directly in a contact simulator. Rather than training policies on raw audio, VIBEACT learns a tactile estimator that maps microphone waveforms to this physical representation, and trains reinforcement learning policies in simulation using the same representation as an observation channel. The estimator solves the sensing problem, while the simulator solves the control problem.

We instantiate VIBEACT on a dexterous hand mounted to a robot arm, with piezoelectric microphones embedded in each fingertip. We replay real-world recordings of robot trajectories and object poses in a calibrated digital clone, and use the simulator’s contact solver to generate contact and slip labels for training the tactile estimator. Using the same tactile representation alongside proprioceptive and point-cloud observations, we train PPO policies [10] entirely in simulation. We then evaluate the learned policies on contact-rich manipulation tasks, including in-hand reorientation and peg insertion.

This paper makes three contributions: **(i)** a sim-to-real framework for vibrotactile dexterity based on a shared contact-and-slip representation that bridges real vibrotactile sensing and simulation-based policy learning; **(ii)** a digital-clone data labeling pipeline that automatically generates per-finger contact and slip supervision from real-world demonstrations; and **(iii)** an empirical study showing that this representation serves as an effective tactile observation for reinforcement learning across contact-rich dexterous manipulation tasks.

2 Related Work

Tactile sensing for dexterous manipulation. Tactile sensing is critical for dexterous manipulation as many task-relevant events occur at the contact interface and are often difficult to infer from vision alone [11, 12]. Prior work has explored a wide range of tactile sensing modalities for robotic manipulation. Vision-based tactile sensors have enabled high-resolution reconstruction of contact geometry [3, 4], shear and slip [13, 14], surface texture [15, 16], and contact-rich manipulation behaviors [17, 18, 19, 20, 21]. However, scaling them to multi-fingered robot hands introduces challenges including mechanical bulk, camera bandwidth, illumination constraints, and substantial compute overhead. Other tactile sensors, such as magnetic tactile skins [22, 23, 24] and capacitive or resistive sensors [25, 26, 27, 28], provide lower-dimensional contact information that can be easier to

integrate into control pipelines, but often require modifying the fingertip structure or contact surface material. VIBEACT is complementary to these approaches: it uses piezoelectric microphones for high-bandwidth signals, embeds sensors inside fingertips without altering external contact geometry or surface texture, and exposes only a compact tactile representation to the policy.

Robot vibrotactile sensing. Contact interactions generate rich structure-borne vibrations that encode information about contact states and interaction dynamics, motivating the use of vibrotactile sensing in robotics [29, 30]. Prior works have leveraged vibrotactile sensing for tasks including material property estimation [31], contact estimation [7, 32], object and action recognition [33, 34], granular flow estimation [35], tool use [36], and slip estimation [8]. A growing body of work instead feeds audio-visual and vibrotactile signals directly into learned manipulation policies, typically through behavior cloning on paired observations and expert actions [37, 38, 39, 40, 41]. By training entirely in the real world, these methods avoid simulating vibro-acoustic signals, but they depend on teleoperated or handheld-grripper demonstrations and have not been extended to dexterous multi-fingered hands with embedded microphones, for which such demonstrations are typically difficult to collect. Reinforcement learning with scalable physics simulation offers a promising alternative that removes the need for demonstrations and naturally supports dexterous control [42, 43]. Learning dexterous policies directly from raw vibro-acoustic signals in simulation, however, remains impractical, since the measured vibration signals depend on difficult-to-model phenomena shaped by material properties, structural resonances, and actuator noise [9]. VIBEACT bridges vibrotactile sensing and simulation-based policy learning by training a real-world perception model that maps vibro-acoustic signals to a physically grounded tactile representation that is also simulatable, while using simulation only to train manipulation policies on the same representation.

3 Problem Formulation

We consider a dexterous hand manipulating rigid objects under partial observability. At each control step t , the policy observes proprioception q_t , an optional point cloud p_t , and a tactile representation z_t . It outputs a continuous action a_t for the arm and hand joints. The goal is to maximize task return for contact-rich manipulation tasks such as grasping, lifting, placing, inserting, and in-hand repositioning.

The tactile hardware consists of piezoelectric microphones embedded in each robot fingertip (Fig. 1, left). At time t , the sensors produce an n -channel vibration signal $x_{t-w:t} \in \mathbb{R}^{n \times W}$ over a temporal window of length W . Directly training policies in simulation on x would require accurate simulation of vibration propagation and sensor transfer functions. We instead define a compact tactile representation:

$$z_t = \{z_t^i\}_{i=1}^4, \quad z_t^i = [b_t^i, m_t^i, e_t^i], \quad (1)$$

where b_t^i denotes a binary slip state, m_t^i denotes scalar slip magnitude, and e_t^i denotes a sparse contact-onset event for finger i . The real-world perception problem is to learn a mapping $f_\theta : x_{t-w:t} \mapsto z_t$ from vibro-acoustic signals to tactile representation. The policy learning problem is to learn a manipulation policy $\pi_\phi(a_t \mid q_t, p_t, z_t)$ in simulation, where the simulator computes z_t directly from contact dynamics.

4 Method

VIBEACT has three components (Fig. 1). First, a teleoperation pipeline records robot and object states, synchronized with vibro-acoustic signals, then replays trajectories in a calibrated digital clone

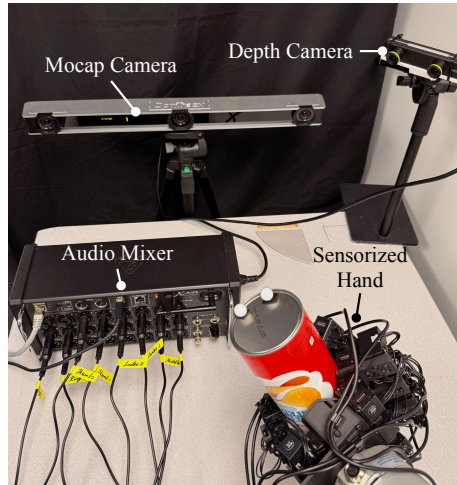


Figure 2: Robot and data collection setup.

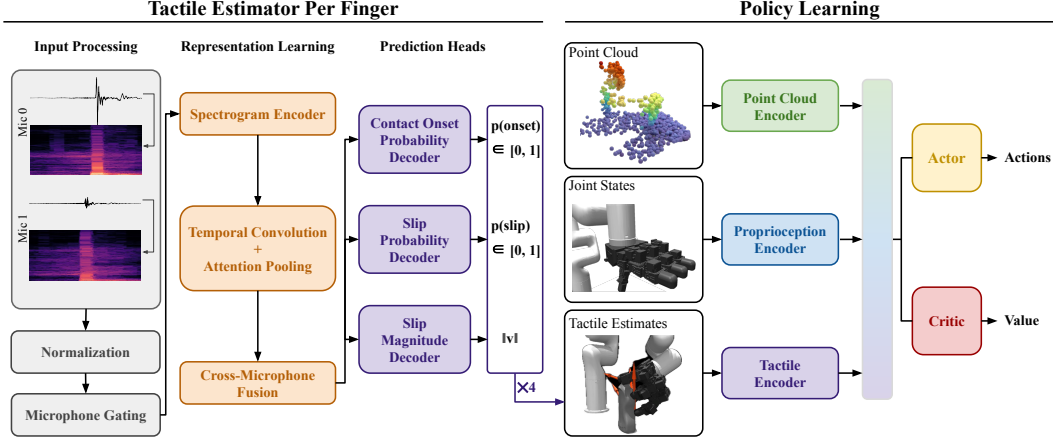


Figure 3: **An overview of VIBEACT.** We train a tactile estimator with four independent per-finger subnetworks using real-world data to map vibro-acoustic signals to a physically grounded contact and slip representation. We then train RL policies that use this representation as an additional observation modality alongside point clouds and proprioception.

in simulation to generate per-finger contact and slip labels as a tactile representation. Second, a tactile estimator learns to map real microphone signals to this tactile representation. Third, RL policies are trained in simulation using proprioception, point clouds, and the same tactile representation as observations. At deployment, the trained estimator replaces the simulator-derived tactile channel.

4.1 Real-World Vibrotactile Data Collection and Digital-Clone Labeling

The hardware setup consists of an xArm7 and a LEAP hand [44]. We embed two piezoelectric microphones in each fingertip to capture structure-borne vibrations within the finger body (Fig. 1, left), and synchronize all eight microphone channels with an audio mixer (Fig. 2). During data collection, we teleoperate the hand to interact with objects while recording 48 kHz microphone audio, arm and hand joint positions, joint velocities, and synchronized timestamps. We either fix objects rigidly to the table with calibrated transforms to the robot base, or track objects using a mocap system whose cameras are calibrated to the robot base.

We then replay the recorded robot trajectories, together with the stationary object poses or mocap-tracked object trajectories, in a calibrated MuJoCo [45] digital-clone environment. This replay recovers contact supervision from aligned robot and object states, producing per-finger contact and slip labels without manual annotation (Fig. 4). At each timestep, the simulator provides location, normal force, and relative tangential velocity for each active fingertip contact. When multiple contacts occur on the same fingertip, we aggregate them by marking contact onset if any new contact appears and taking the maximum tangential slip speed. The resulting representation forms a 12-D tactile label vector that matches the tactile observation space used by the policy.

4.2 Tactile Representation

Instantiating Eq. 1 for finger i , contact onset e_t^i is a one-step pulse that fires when contact is newly established, slip presence b_t^i is a binary threshold on tangential relative velocity $\|v_t^i\|$ at $\|v_{\text{slip}}\| = 5$ mm/s, and slip magnitude $m_t^i = \text{clip}(v_t^i, 0, m_{\text{max}})$ provides a continuous severity signal. This design matches the event-like acoustic transient of first contact while allowing persistent sliding to be captured by the slip channels.

The representation intentionally excludes privileged simulator information such as contact location, surface normal, tangential direction, object identity, and force vectors. This restriction is important because the representation must remain predictable from the physical microphone signals and transferable across objects and tasks. At the same time, it preserves the key feedback required for

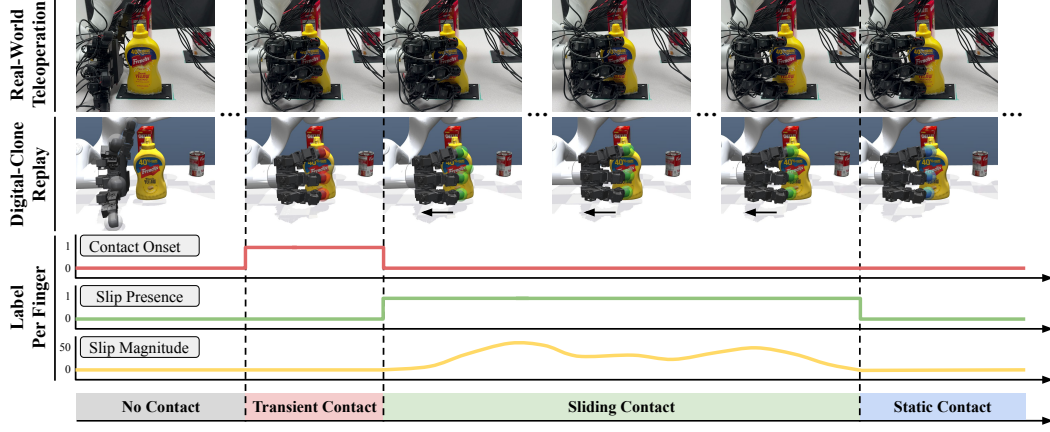


Figure 4: **Vibrotactile data labeling setup.** We replay real-world teleoperation recordings in a calibrated digital clone, where the simulator’s contact solver produces per-finger contact and slip labels for training the tactile estimator without manual annotation.

reactive manipulation: whether a fingertip has just touched the object, whether slip is occurring, and the severity of that slip.

4.3 Tactile Estimator

We assume that contact onset and slip retain the same physical meaning in both simulation and real-world vibro-acoustic sensing. Under this assumption, we train a tactile estimator that maps real microphone audio to the contact and slip labels generated by the digital-clone replay pipeline.

Network. The estimator consists of four independent per-finger subnetworks (Fig. 3, left). For finger i , the input tensor $\mathbf{X}^i \in \mathbb{R}^{n \times M \times T}$ contains multi-channel log-mel spectrograms computed over 200 ms windows from $n = 2$ synchronized microphones, with M mel bins and T temporal frames. We use absolute dB magnitudes and dataset-wide normalization to preserve amplitude information while compensating for microphone gain differences. A learnable microphone-gating layer first suppresses noisy channels, after which the gated spectrograms are processed by a convolutional encoder with frequency-only pooling to preserve temporal resolution for transient events. Temporal convolutions and attention pooling produce per-microphone embeddings, fused into a per-finger representation. Separate heads then predict contact-onset probability, slip-presence probability, and slip magnitude, with the magnitude head additionally conditioned on amplitude statistics.

Training objectives. We train the model with a multi-task objective over contact onset, slip presence, and slip magnitude. Let $y_{\text{onset}}^i, y_{\text{slip}}^i \in \{0, 1\}$ denote the binary contact-onset and slip labels, and let $\|v^{*i}\|$ denote the ground-truth tangential slip magnitude from digital-clone replay. We supervise the binary tasks with class-weighted binary cross-entropy losses $\mathcal{L}_{\text{onset}}^i = \text{BCE}_{w_{+}^{\text{onset}}}(p_{\text{onset}}^i, y_{\text{onset}}^i)$ and $\mathcal{L}_{\text{slip}}^i = \text{BCE}_{w_{+}^{\text{slip}}}(p_{\text{slip}}^i, y_{\text{slip}}^i)$, where $w_{+}^{\text{onset}} > 1$ compensates for sparse contact-onset events and $w_{+}^{\text{slip}} < 1$ balances the predominance of slip-positive frames. Slip magnitude is supervised with a Huber loss applied only on frames labeled as slipping:

$$\mathcal{L}_{\text{mag}}^i = \mathbf{1}[\|v^{*i}\| > \|v_{\text{slip}}\|] \cdot \text{Huber}_{\delta}(\|\hat{v}^i\|, \|v^{*i}\|),$$

where $\|v_{\text{slip}}\| = 5 \text{ mm/s}$ is the slip threshold used during label generation and $\|\hat{v}^i\|$ is the predicted slip magnitude. The overall training objective averages losses across the four fingers:

$$\mathcal{L} = \frac{1}{4} \sum_{i=1}^4 (\lambda_{\text{onset}} \mathcal{L}_{\text{onset}}^i + \lambda_{\text{slip}} \mathcal{L}_{\text{slip}}^i + \lambda_{\text{mag}} \mathcal{L}_{\text{mag}}^i).$$

Table 1: **Ablation studies of the VIBEACT tactile estimator.** VIBEACT tactile estimator uses sequential training (ST) with independent encoders (IE) and independent heads (IH). To evaluate the training strategy, we keep the architecture fixed and compare with pretraining only (PT) and joint training (JT). To evaluate architectural choices, we keep the training strategy fixed and compare with variants using a shared encoder with independent heads (SE + IH) and a shared encoder with shared heads (SE + SH). Results are averaged across all four fingers.

Method	Contact Onset F1 $\uparrow$	Slip Presence F1 $\uparrow$	Slip Magnitude MAE (mm/s) $\downarrow$
PT + IE + IH	0.384 ± 0.110	0.781 ± 0.060	6.417 ± 1.399
JT + IE + IH	0.552 ± 0.114	0.923 ± 0.036	4.914 ± 0.792
ST + SE + SH	0.575 ± 0.077	0.891 ± 0.053	4.964 ± 0.838
ST + SE + IH	0.583 ± 0.090	0.907 ± 0.049	4.827 ± 0.728
ST + IE + IH (Full VIBEACT)	0.597 ± 0.101	0.913 ± 0.054	4.736 ± 0.658

4.4 Policy Learning

We train consistent policy architectures across all five tasks in MuJoCo simulation [45] with PPO [10], varying only the scene and task reward. Each task instantiates the same xArm7 and LEAP hand model under per-episode domain randomization of object friction, mass, scale, and pose, so the policy must rely on contact feedback rather than memorized geometry.

At each step, the policy receives a tuple $o_t = (p_t, q_t, z_t)$, where p_t is a fixed-camera point cloud, q_t is proprioception over the hand joints and, where applicable, the arm pose, and $z_t \in \mathbb{R}^{12}$ is the contact-and-slip vector from Sec. 4.2 (Fig. 3, right). A PointNet-style branch [46] maps p_t to a permutation-invariant geometric feature, while separate MLP branches encode q_t and z_t , and the three features are concatenated and passed to symmetric actor and critic heads. Each task combines a dense, task-specific progress term r_t^{task} with a small set of auxiliary terms weighted identically across all tasks, giving $r_t = r_t^{\text{task}} - \lambda_{\text{smooth}} \|a_t - a_{t-1}\|^2 - \lambda_{\text{drop}} d_t + \lambda_{\text{succ}} s_t$, where $d_t, s_t \in \{0, 1\}$ indicate a drop and a successful completion at step t , so the auxiliary terms penalize abrupt action changes, discourage drops, and reward task completion. The progress term r_t^{task} is the negative distance to a goal pose for the grasp, lift, place, and insert tasks, and a signed rotation and upward-displacement rate for the in-hand reposition and climb tasks. We hold λ_{smooth} , λ_{drop} , and λ_{succ} fixed across all ablation conditions, so that only the policy’s tactile observation z_t varies between them.

5 Experiments

We evaluate VIBEACT along three axes. We assess how accurately the tactile estimator recovers contact and slip from real microphone signals (Sec. 5.1), whether the resulting representation improves manipulation policies in simulation and which channels drive the gains (Sec. 5.3), and whether the benefit transfers to a physical robot. Together they test whether a physically grounded tactile representation can bridge real vibrotactile sensing and simulation-based policy learning.

5.1 Tactile Estimator Evaluation

We collect two teleoperated datasets to train and evaluate the tactile estimator before policy deployment. The fixed-object dataset (five hours) contains interactions with objects rigidly mounted to the table, inducing slip purely through robot motion. This scales diverse contact and sliding events but yields slip dynamics unlike dexterous manipulation. The moving-object dataset (under two hours) involves in-hand manipulation of free-moving objects, with slip arising from finger motion, arm motion, and gravity. We pretrain on the fixed-object data to learn general vibro-acoustic patterns, then fine-tune on the moving-object data to match deployment. We evaluate all models on a held-out moving-object split, reporting F1 for binary contact-onset and slip-presence detection and MAE for slip magnitude, averaged over the four fingers.

Table 1 compares training strategies and architectural variants. Pretraining alone underperforms VIBEACT, lowering contact-onset F1 by 35.7% and slip-presence F1 by 14.5% while raising slip-

Table 2: Average task success across tactile observation ablations and random seeds.

Task	Prop + PC (No Tactile) (%)	+ Contact Onset (%)	+ Slip Presence (%)	+ Slip Magnitude (Full VIBEACT) (%)
Box Climb	46.7	0.0	48.3	50.0
Nut Rotation	28.5	28.5	8.4	44.0
Peg in Hole	6.5	13.5	15.0	30.0
Cube Rotation	6.0	0.0	2.0	57.0
Can Climb	60.0	0.0	33.0	76.0

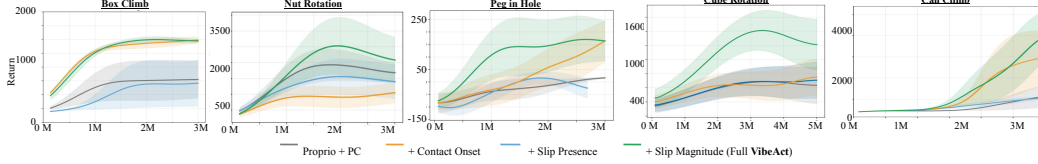


Figure 5: Training curves for the VIBEACT policies and baselines.

magnitude MAE by 35.5%. This suggests a large domain gap between fixed-object and in-hand slip. Joint training on both datasets narrows this gap and marginally exceeds VIBEACT on slip-presence F1, but still underperforms overall, reducing contact-onset F1 by 7.5% and increasing slip-magnitude MAE by 3.8%. We attribute this to the differing temporal structure: contact onset is a sparse transient requiring precise temporal alignment, while slip presence is temporally persistent and transfers more readily across domains. VIBEACT’s pretrain-then-fine-tune strategy thus preserves transient sensitivity while retaining strong slip estimation.

We further evaluate parameter sharing across fingers. Sharing the encoder while keeping independent prediction heads reduces contact-onset F1 by 2.3% and raises slip-magnitude MAE by 1.9% relative to VIBEACT. Sharing both the encoder and prediction heads further compounds the degradation, reducing contact-onset F1 by 3.7% and increasing MAE by 4.8%. These results suggest that each fingertip exhibits distinct contact geometry and vibration propagation characteristics that benefit from dedicated feature extraction and prediction heads.

5.2 Tasks

We evaluate VIBEACT on five contact-rich tasks spanning regrasping, in-hand reorientation, and insertion (Fig. 6). **Box Climb** and **Can Climb** fix the arm and require the hand to walk its fingers along a held YCB object under randomized gravity-relative orientations [47]. **Peg in Hole** starts from a pregrasped cylinder and requires sideways insertion, focusing on post-contact alignment and force control. **Cube Rotation** reorients a held cube purely in-hand, while **Nut Rotation** rotates a hex nut through coordinated hand-arm motion. Together, these tasks stress distinct failure modes of contact-rich control, including grip loss during gaiting, slip during insertion, and stalled rotation, where reactive tactile feedback should help. We deploy Box Climb, Can Climb, and Nut Rotation on hardware, sampling tasks that require both in-hand dexterity and hand-arm coordination. Task rewards are in the appendix.

5.3 Task Performance

We evaluate VIBEACT against a proprioception and point cloud baseline (Prop+PC) with no tactile input, and ablate the tactile representation by incrementally adding contact onset, slip presence, and slip magnitude channels. We report the results in Table 2 based on 100 trials trained across 3 random seeds, and the policy training curves are visualized in Fig. 5.

The full VIBEACT achieves the highest success rate across all five tasks. The gains are largest on tasks requiring reactive fingertip control: tactile feedback lifts Cube Rotation and Peg in Hole from near-failing baselines by roughly +51 and +24 points, and improves Can Climb and Nut Rotation by

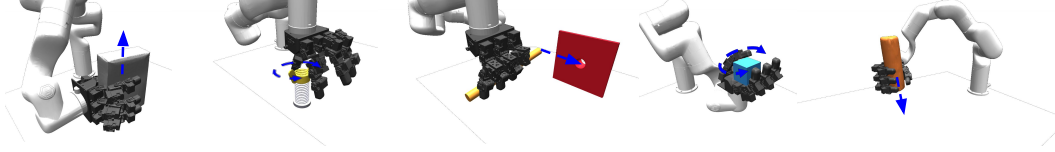


Figure 6: **Tasks.** We evaluate VIBEACT on five contact-rich manipulation tasks: **Box Climb** and **Can Climb** require the hand to walk its fingers along a held YCB object; **Peg in Hole** requires sideways insertion after a scripted pregrasp; **Cube Rotation** requires repeated finger gaiting to rotate the cube, while **Nut Rotation** requires hand-arm coordination to succeed. The blue arrow indicates the task objective.

about +16 and +15 points. Box Climb gains only a few points, suggesting that coarser point-cloud feedback is already largely sufficient for finger-gaiting along larger objects.

The ablations reveal that slip magnitude is a critical channel where adding it alone drives the largest jump in performance across all tasks. Contact onset gives inconsistent gains in isolation and in some tasks hurts performance, likely because sparse onset pulses alone carry limited information for sustained reactive control, while slip presence offers intermediate benefit. Together, these results indicate that the continuous slip magnitude signal is the primary source of VIBEACT’s advantage, providing the graded feedback needed for contact-rich manipulation.

We deploy the simulation-trained policies directly on the physical platform, replacing the simulator-derived tactile channel with the trained estimator (Table 3). VIBEACT improves over Prop+PC on every deployed task. These results indicate that the contact-and-slip representation, learned entirely from real microphone recordings and used as a drop-in observation channel, retains its benefit under real sensing and actuation noise.

Table 3: **Real-world deployment.** *Prop+PC*: proprioception + point cloud baseline.

Task	Prop+PC	VIBEACT
Box Climb	4/20	12/20
Can Climb	11/20	19/20
Nut Rotation	1/20	8/20

6 Conclusion

We present VIBEACT, a framework for leveraging vibrotactile sensing in contact-rich dexterous manipulation without requiring audio simulation. By defining a compact physical contact and slip representation that can be computed from both real piezoelectric microphones signals and simulated contact dynamics, VIBEACT decouples the sensor problem from the control problem. A tactile estimator trained on real microphone signals with digital-clone-derived labels bridges the two domains, enabling policies trained entirely in simulation to exploit tactile feedback at deployment. Across five contact-rich manipulation tasks, VIBEACT consistently outperforms a proprioception and point cloud baseline, with the largest gains on tasks requiring sustained reactive control such as in-hand rotation and peg insertion. These results suggest that physically grounded intermediate representations are a practical and effective route for incorporating high-bandwidth tactile sensing into sim-to-real policy learning.

7 Limitations

VIBEACT’s compact contact and slip representation discards information in the raw vibration signal, such as contact location, surface texture, and interaction dynamics beyond slip magnitude; richer representations could help but would demand closer sim-to-real acoustic alignment. The tactile estimator is also tied to a fixed hardware configuration and may need recalibration if microphone placement, finger material, or object properties change. The digital-clone labeling, while annotation-free, depends on accurate object pose tracking, limiting use in unstructured settings. Finally, the policy treats the tactile representation as a flat vector, leaving room for architectures that model the spatial and temporal structure of per-finger contact events.

Acknowledgments

This work was supported by Samsung Research America and NSF Graduate Research Fellowship under Grant No. DGE2140739.

References

- [1] Y. C. Nakamura, D. M. Troniak, A. Rodriguez, M. T. Mason, and N. S. Pollard. The complexities of grasping in the wild. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 233–240. IEEE, 2017.
- [2] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.
- [3] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
- [4] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.
- [5] Y. Mao, B. P. Duisterhof, M. Lee, and J. Ichnowski. Hearing the slide: Acoustic-guided constraint learning for fast non-prehensile transport. In *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, pages 1127–1133. IEEE, 2025.
- [6] U. Yoo, Z. Lopez, J. Ichnowski, and J. Oh. Poe: Acoustic soft robotic proprioception for omnidirectional end-effectors. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14980–14987. IEEE, 2024.
- [7] M. Lee, U. Yoo, J. Oh, J. Ichnowski, G. Kantor, and O. Kroemer. Sonicboom: Contact localization using array of microphones. *IEEE Robotics and Automation Letters*, 2025.
- [8] U. Yoo, Y. Mao, J. Oh, and J. Ichnowski. A-slip: Acoustic sensing for continuous in-hand slip estimation, 2026. URL <https://arxiv.org/abs/2604.08528>.
- [9] S. Clarke, N. Heravi, M. Rau, R. Gao, J. Wu, D. James, and J. Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *Conference on Robot Learning*, pages 662–673. PMLR, 2022.
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [11] Y. Niu, Z. Fang, B. Chen, S. Zhou, R. Senthilkumaran, H. Zhang, B. Chen, C. Qiu, H. E. Tseng, J. Francis, et al. Learning versatile humanoid manipulation with touch dreaming. *arXiv preprint arXiv:2604.13015*, 2026.
- [12] F. Liu, C. Li, Y. Qin, J. Xu, P. Abbeel, and R. Chen. Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface. *arXiv preprint arXiv:2504.06156*, 2025.
- [13] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 304–311. IEEE, 2015.
- [14] S. Dong, W. Yuan, and E. H. Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–144. IEEE, 2017.

- [15] R. Li and E. H. Adelson. Sensing and recognizing surface textures using a gelsight sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
- [16] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes. Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2722–2727. IEEE, 2018.
- [17] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021.
- [18] A. Alspach, K. Hashimoto, N. Kuppaswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 597–604. IEEE, 2019.
- [19] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10241–10247. IEEE, 2022.
- [20] M. Oller, M. P. i Lisbona, D. Berenson, and N. Fazeli. Manipulation via membranes: High-resolution and highly deformable tactile sensing and control. In *Conference on Robot Learning*, pages 1850–1859. PMLR, 2023.
- [21] C. Lin, B. Huo, M. Yu, E. Ruppel, B. Chen, J. Francis, and D. Zhao. Lighttact: A visual-tactile fingertip sensor for deformation-independent contact sensing. *arXiv preprint arXiv:2512.20591*, 2025.
- [22] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *CoRL*, 2021.
- [23] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. URL <https://arxiv.org/abs/2409.08276>.
- [24] T. Hellebrekers, N. Chang, K. Chin, M. J. Ford, O. Kroemer, and C. Majidi. Soft magnetic tactile skin for continuous force and location estimation using neural networks. *IEEE Robotics and Automation Letters*, 5(3):3892–3898, 2020. doi:10.1109/LRA.2020.2983707.
- [25] T. P. Tomo, A. Schmitz, W. K. Wong, H. Kristanto, S. Somlor, J. Hwang, L. Jamone, and S. Sugano. Covering a robot fingertip with uskin: A soft electronic skin with distributed 3-axis force sensitive elements for robot hands. *IEEE Robotics and Automation Letters*, 3(1):124–131, 2018. doi:10.1109/LRA.2017.2734965.
- [26] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. In *8th Annual Conference on Robot Learning*, 2024.
- [27] X. Liu, W. Yang, F. Meng, and T. Sun. Material recognition using robotic hand with capacitive tactile sensor array and machine learning. *IEEE Transactions on Instrumentation and Measurement*, 73:1–9, 2024. doi:10.1109/TIM.2024.3383886.
- [28] S. Wistreich, B. Shi, S. Tian, S. Clarke, M. Nath, C. Xu, Z. Bao, and J. Wu. Dexskin: High-coverage conformable robotic skin for learning contact-rich manipulation. *arXiv preprint arXiv:2509.18830*, 2025.
- [29] S. Lu and H. Culbertson. Active acoustic sensing for robot manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3161–3168. IEEE, 2023.

- [30] S. Rupavatharam, C. Escobedo, D. Lee, C. Prepscius, L. Jackel, R. Howard, and V. Isler. Sonicfinger: Pre-touch and contact detection tactile sensor for reactive pregrasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12556–12562, 2023. doi:10.1109/ICRA48891.2023.10161074.
- [31] X. Yi, Y. Xing, Z. Manchester, and N. Fazeli. Sound of touch: Active acoustic tactile sensing via string vibrations. *arXiv preprint arXiv:2602.16846*, 2026.
- [32] K. Zhang, D.-G. Kim, E. T. Chang, H.-H. Liang, Z. He, K. Lampo, P. Wu, I. Kymissis, and M. Ciocarlie. Vibecheck: Using active acoustic tactile sensing for contact-rich manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12278–12285. IEEE, 2025.
- [33] D. Gandhi, A. Gupta, and L. Pinto. Swoosh! rattle! thump!—actions that sound. *arXiv preprint arXiv:2007.01851*, 2020.
- [34] J. Liu and B. Chen. Sonicsense: Object perception from in-hand acoustic vibration. In *Conference on Robot Learning*, pages 4332–4353. PMLR, 2025.
- [35] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer. Learning audio feedback for estimating amount and flow of granular material. In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 529–550. PMLR, 29–31 Oct 2018. URL <https://proceedings.mlr.press/v87/clarke18a.html>.
- [36] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer. Leveraging multimodal haptic sensory data for robust cutting. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 409–416. IEEE, 2019.
- [37] J. Mejia, V. Dean, T. Hellebrekers, and A. Gupta. Hearing touch: Audio-visual pretraining for contact-rich manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6912–6919. IEEE, 2024.
- [38] A. Thankaraj and L. Pinto. That sounds right: Auditory self-supervision for dynamic robot manipulation. In *Conference on Robot Learning*, pages 1036–1049. PMLR, 2023.
- [39] M. Du, O. Y. Lee, S. Nair, and C. Finn. Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning. *arXiv preprint arXiv:2205.14850*, 2022.
- [40] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song. Maniwav: Learning robot manipulation from in-the-wild audio-visual data. *arXiv preprint arXiv:2406.19464*, 2024.
- [41] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In *Conference on Robot Learning*, pages 1368–1378. PMLR, 2023.
- [42] H. Qi, B. Yi, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. From simple to complex skills: The case of in-hand object reorientation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14291–14298. IEEE, 2025.
- [43] E. Xing, V. Luk, and J. Oh. Stabilizing reinforcement learning in differentiable multiphysics simulation. In *International Conference on Learning Representations*, volume 2025, pages 91165–91198, 2025.
- [44] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.
- [45] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

- [46] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [47] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.
- [48] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [49] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [50] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

A Tactile Estimator Training

A.1 Network Architecture

Table S.1 lists the layer dimensions of each per-finger subnetwork of VIBEACT tactile estimator. The network takes per-microphone log-mel spectrogram inputs of shape $(B, 1, 19, 64)$, computed at 48 kHz with $n_{\text{fft}}=2048$, hop length = 512, and 64 mel bins with $f_{\text{min}}=500$ Hz. The spectrogram encoder and temporal block process each microphone independently to produce two 128-dimensional embeddings per finger. Cross-microphone fusion concatenates these embeddings and projects them back to a shared 128-dimensional representation. For slip magnitude estimation, the decoder additionally receives four amplitude scalars per microphone (8 per finger). We apply dropout with probability $p=0.2$ after each convolution block. Contact onset and slip presence outputs represent sigmoid probabilities, while the magnitude head predicts slip speed in mm/s.

Module	Layers	Output
Per-mic spectrogram encoder	Conv2d ($1 \rightarrow 32, 3 \times 3$) + BN + ReLU + MaxPool (1×2) + Dropout	$(B, 32, 19, 32)$
	Conv2d ($32 \rightarrow 64, 3 \times 3$) + BN + ReLU + MaxPool (1×2) + Dropout	$(B, 64, 19, 16)$
	Conv2d ($64 \rightarrow 128, 3 \times 3$) + BN + ReLU + MaxPool (1×2) + Dropout	$(B, 128, 19, 8)$
	AdaptiveAvgPool2d over mel axis, squeeze	$(B, 128, 19)$
Temporal convolution + attention pooling	Conv1d ($128 \rightarrow 256, k=5$) + BN + ReLU + Dropout	$(B, 256, 19)$
	Conv1d ($256 \rightarrow 128, k=5$) + BN + ReLU + Dropout	$(B, 128, 19)$
	Conv1d ($128 \rightarrow 1, k=1$) score + softmax over T , weighted sum	$(B, 128)$
Cross-microphone fusion	concat 2 mics $\rightarrow$ Linear($256 \rightarrow 128$) + ReLU	$(B, 128)$
Contact onset decoder	Linear ($128 \rightarrow 1$) + Sigmoid	$(B, 1)$
Slip presence decoder	Linear ($128 \rightarrow 1$) + Sigmoid	$(B, 1)$
Slip magnitude decoder	concat[spec (128), amp (8)] $\rightarrow$ Linear ($136 \rightarrow 1$)	$(B, 1)$

Table S.1: **Estimator network layer dimensions.** Each fingertip uses an independent subnetwork with identical architecture and parameters. This table lists tensor dimensions and layer configurations for a single per-finger network.

A.2 Training Parameters

We train all models, including the final deployment model and all ablation variants, with AdamW [48] with cosine learning-rate decay. We pretrain on the large-scale stationary-object dataset for 100 epochs with an initial learning rate of 3×10^{-4} . For models with sequential training, we then fine-tune for 100 epochs with an initial learning rate of 3×10^{-5} . We set $\lambda_{\text{onset}} = 1.0$, $\lambda_{\text{slip}} = 1.0$, and $\lambda_{\text{mag}} = 0.1$, and choose $w_+^{\text{on}} = 30$, $w_+^{\text{sl}} = 0.5$, and $\delta = 5$ mm/s. During inference, we classify contact onset and slip presence as positive when their predicted probabilities exceed 0.5.

B Reactive Policy Training

B.1 Network Architecture

The same network architecture is used for every task and every tactile variant, where only the action head’s output dimension changes with the task action space (Table S.4).

The action vector is sampled from a diagonal Gaussian $\mathcal{N}(\mu_\theta(o_t), \text{diag}(e^{\log \sigma}))$ and clipped to $[-1, 1]$ before being passed to the environment. $\log\text{-}\sigma$ is initialized to zero (so $\sigma_{\text{init}} = 1$). The network has $\sim 378\text{K}$ trainable parameters for the 23-D arm+hand action space used by the climb tasks; the encoder accounts for $\sim 11\%$ of these, and the two MLP heads $\sim 89\%$.

B.2 PPO Hyperparameters

We use Stable Baselines 3’s PPO [10] with the settings in Table S.2. The same values are used for every task and tactile variant.

Hyperparameter	Value
Parallel environments	24 (SubprocVecEnv)
Rollout length per env	128
Minibatch size	768
Update epochs per rollout	10
Total environment steps	5×10^6
Optimizer	Adam [49]
Learning rate	3×10^{-4} (constant)
Discount γ	0.99
GAE λ [50]	0.95
Clip range ϵ	0.2
Value loss coefficient c_{vf}	0.5
Entropy coefficient c_{ent}	0.01
Target KL (early stop)	0.02
Gradient norm clip	0.5 (SB3 default)
Episode length	400 RL steps, fixed-length
RL substeps per action	15 MuJoCo steps
Physics timestep	0.001 s (1 kHz)
Effective control rate	≈ 66.7 Hz

Table S.2: **PPO and simulation hyperparameters.** Shared across all tasks and tactile variants.

Episodes are fixed-length (400 RL steps ≈ 6 s of simulated time), with success and drop signals reported in the info dict rather than as termination triggers. This keeps the rollout horizon constant across conditions so per-step reward and success-rate curves are directly comparable.

B.3 Tactile Encoding

All variants share the 12-D per-finger layout (4 fingertips $\times$ 3 features per tip). Let $c_t^i \in \{0, 1\}$ be the binary in-contact flag, v_t^i the maximum tangential relative velocity over active contacts on finger i , and $\Delta c_t^i = c_t^i \wedge \neg c_{t-1}^i$ the rising-edge contact flag. The three feature variants then differ as follows.

Legacy [b, m, e]. Computed at every step:

$$\begin{aligned}
 b_t^i &= \mathbf{1}[c_t^i \wedge v_t^i > v_{\text{slip}}], \\
 m_t^i &= \text{clip}(v_t^i, 0, m_{\text{max}}), \\
 e_t^i &= \mathbf{1}[\text{TTL}_t^i > 0],
 \end{aligned}$$

where $v_{\text{slip}} = 5$ mm/s, m_{max} matches the slip-magnitude clip of the real perception model, and TTL^i is a 3-step rising-edge pulse that requires 2 consecutive off-steps before a new onset can fire. This is the canonical representation Sec. 4.2 defines and the perception model outputs.

Dense (used for the production simulation policies). Each fingertip’s three slots are replaced by exponential moving averages with three different time constants:

$$\begin{aligned}
 \text{ema_short_mag}_t^i &= \alpha_s v_t^i + (1 - \alpha_s) \text{ema_short_mag}_{t-1}^i, & \alpha_s &= 0.4, \\
 \text{ema_long_mag}_t^i &= \alpha_\ell v_t^i + (1 - \alpha_\ell) \text{ema_long_mag}_{t-1}^i, & \alpha_\ell &= 0.05, \\
 \text{ema_contact}_t^i &= \alpha_c c_t^i + (1 - \alpha_c) \text{ema_contact}_{t-1}^i, & \alpha_c &= 0.2.
 \end{aligned}$$

The two slip time scales let the policy approximate slip onset and decay through their difference, and the contact EMA provides a continuous estimate of contact recency in place of the noisy single-step flag.

Dense + smoother. Identical to dense, except v_t^i is first passed through a four-stage causal filter before the EMAs are taken:

1. **Hard cap:** values exceeding 0.05 m/s (a physically-impossible rigid-replay artifact in our digital-clone pipeline) are replaced by the recent-valid median.
2. **Causal local-median fill:** capped samples are replaced by the median of the last 5 valid samples.
3. **Causal sliding-window median:** 5-step window over the filled stream.
4. **Causal one-pole IIR:** low-pass with $\alpha = 0.15$.

This matches the post-processing applied to real microphone-derived slip estimates in our perception pipeline, so that the simulated and real tactile streams have the same noise characteristics and temporal smoothing at training time.

B.4 Domain Randomization

Table S.3 lists the per-episode randomization applied during training. Object scale randomization was attempted but disabled after we observed visible misalignment between the visual mesh and the primitive-box collider when the scale changed; the visual mesh’s MuJoCo asset has no scale attribute, so only the collider was being resized.

Axis	Distribution	Notes
Object friction	$\mu \sim \mathcal{U}(0.5, 1.5) \mu_0$	Per-task nominal μ_0 ; band narrowed to $\mathcal{U}(0.8, 1.2) \mu_0$ for peg-in-hole, where the nominal is intentionally inflated.
Object mass	$m \sim \mathcal{U}(0.7, 1.3) m_0$	Baked into a freshly compiled MJCF each reset so inertia rescales.
Object initial xy	± 1.5 cm on each axis	Climb tasks; nut family randomizes the assembly xy.
Object initial yaw	$\pm 10^\circ$	Climb tasks.
Wrist orientation	$\pm 90^\circ$ about wrist-local z (per episode)	Climb tasks; resets gravity-relative grasp pose.
Wrist position	± 4 cm xy, $\mathcal{U}(-1, 6)$ cm z	Climb tasks; injects pose diversity for the PC.
Camera extrinsics	± 5 cm pos, $\pm 10^\circ$ rot	Per-episode jitter of the scene depth camera.
Depth pixel noise	$\sigma \sim \mathcal{U}(2, 12)$ mm	Per-pixel Gaussian on the back-projected depth.
Depth pixel dropout	$\mathcal{U}(5\%, 30\%)$	Random pixel masking before point unprojection.
PC lateral jitter	$\mathcal{U}(1, 5)$ mm	In-plane noise on each kept pixel before unprojection.
Hand servo noise	$\sigma = 0.01$ rad per step	Gaussian noise on the 16-D hand ctrl setpoint, clipped to joint limits.

Table S.3: **Per-episode domain randomization.** Climb tasks additionally re-randomize the wrist orientation and position after each scripted pre-grasp; the cube-rotation environment adds per-episode cube-COM jitter $\mathcal{U}(-5, 5)$ mm on top of the shared friction/mass DR.

Pre-grasp procedure (climb tasks). The scripted pre-grasp moves the wrist through 6 phases (home settle, translate to standoff at safe- z , rotate + descend, drive in to grasp pose, pre-close fingers, force-thresholded close, lift) and then applies the per-episode wrist tilt and position jitter. The pre-grasp pose was hand-authored using an interactive grasp-definition tool and stored as `env_config/<task>_grasp.json`; for the cracker box, the grasp sits ~ 3 cm below the cracker’s top edge so the lowest finger has ~ 13 cm of climb headroom. The expensive Phases 1–6 run once per worker; subsequent episodes restore the post-lift snapshot and re-roll only the per-episode wrist randomization.

Task	Action	Arm	Pre-grasp	Episode reward signal
cracker_climb	23-D (7 arm + 16 hand)	frozen (action scale 0)	scripted side grasp + lift; near-top, hand-tuned	climb progress, finger advance, drop, success
peg_climb	23-D (7 arm + 16 hand)	frozen	scripted side grasp + lift, hand-tuned	same shape as cracker_climb
peg_in_hole	19-D (3 EE + 16 hand)	relative xyz delta on EE site, quat locked	authored grasp snap + lift+tilt	insertion depth, drop, success
hex_nut_fingers	19-D (same as nut_fingers)	same	same	same as nut_fingers but on a hex nut
cube_rotation	16-D (hand only)	fully frozen at authored qpos	none (cube spawned in hand)	forward-direction in-hand rotation, drop

Table S.4: **Per-task action space and high-level reward structure.** “EE” denotes the LEAP palm site that mink IK tracks. The cracker and peg climb tasks freeze the arm so the policy must walk the fingers down the held object to climb it.

Pre-grasp procedure (peg-in-hole, nut family). A hand-authored arm+hand pose is loaded from `env_config/<task>_grasp.json` and applied as a hard snap to the simulator state, followed by a few mocap+IK steps to settle contacts. Gravity compensation is applied to the held object during the snap window to prevent it from falling before the fingers re-engage.

B.5 Reward Functions

We summarize the dominant terms for each task; full coefficients are documented in code.

Climb tasks (cracker, peg). The shaped reward combines five terms:

- **Climb progress:** $+500 \cdot \Delta z_t^{\text{best}}$ for new positive increments of the held-object world- z (one-sided, ratcheting on best-so-far).
- **Per-finger advance:** $+1000 \cdot \Delta a_t^{i,\text{best}}$ per fingertip i for new descent on the object body axis (one-sided).
- **Hold:** $+0.02 \cdot \min(n_{\text{tips}}, 5)$ per step, where n_{tips} is the number of fingertips in contact.
- **Drop:** -50 one-shot if the held object falls more than 15 cm below its initial pose.
- **Success:** $+100$ one-shot at $\Delta z_t \geq 0.12$ m (cracker) or 0.10 m (peg) with ≥ 2 fingertips still in contact.

A small action-magnitude penalty ($-5 \times 10^{-4} \|a_t\|^2$) and action-smoothness penalty ($-5 \times 10^{-3} \|a_t - a_{t-1}\|^2$) are also applied; these are shared across all tasks.

Peg-in-hole. Insertion-depth shaping with a $+500$ terminal bonus on success past 80% of the hole depth, -200 on a drop, and an arm slow-start linear ramp on the wrist xyz delta cap (3 mm/step ramping to 5 mm/step over the first ~ 1.2 s of an episode).

Nut and nut_fingers. The nut env rewards continuous clockwise spin of the hinge ($+8 \times (-\dot{\theta}_{\text{hinge}})_+$), per-tip contact at $+1$ per fingertip per step, fingers staying near the canonical grip pose ($-0.1 \|q_t^{\text{hand}} - q_*^{\text{hand}}\|^2$), a -15 drop penalty, and a -50 wrist-ceiling penalty per metre. The fingers variant additionally rewards a ratchet on best-so-far CW revolutions ($+200 \cdot \Delta R_t^{\text{best}}$), a saturating continuous spin bonus ($+5 \cdot \min(R_t, 1)$), per-finger advance ($+500$ per CW rev per finger), and a hold-with-freshness bonus.

Cube rotation. A finite-difference rotation reward in the forward direction ($+1.25 \cdot \dot{\theta}_z$ around the target axis) and a -10 object-fallen penalty.

C Sim-to-Real Perspective Alignment

For sim-to-real policy transfer, we align the point clouds in simulation and the real world using ICP and simulation camera view refinement (Fig. 7).

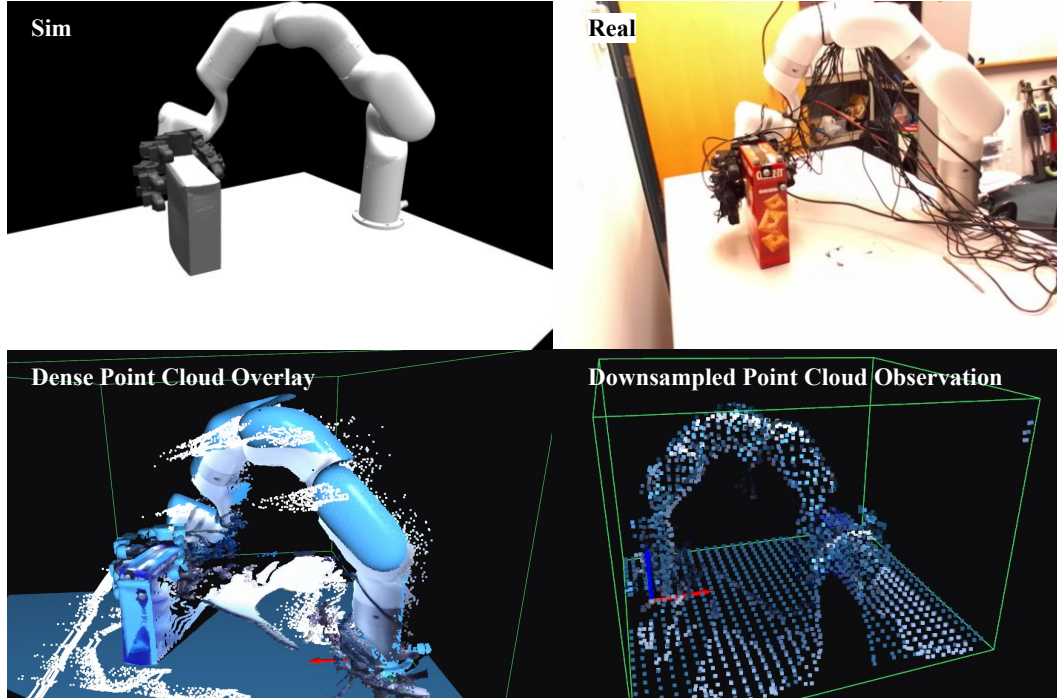


Figure 7: Sim-to-real perspective alignment.