BEYOND BINARY: SIM-TO-REAL DEXTEROUS MANIPULATION WITH PHYSICS-GROUNDED CONTACT REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A primary bottleneck in contact-rich manipulation is the difficulty of collecting real-world data. Sim-to-real reinforcement learning offers a scalable alternative, but the simulation-reality gap prevents information-dense modalities like touch from being effectively used. Existing sim-to-real methods often mitigate this gap by simplifying tactile data into coarse low-dimensional features – sacrificing the richness required for complex manipulation. In this work, we introduce **C**enter-**o**f-**P**ressure (**CoP**), an effective tactile representation grounded in physical principles that preserves the dominant normal-force and contact-location information while maintaining robustness for sim-to-real transfer. To support this representation, we propose a sensor calibration scheme based on differentiable dynamics, enabling the estimation of taxel orientations without requiring ground-truth force measurements. We evaluate CoP on two blind, challenging contact-rich manipulation tasks: peg-in-hole insertion and ball balancing. Across both tasks, policies conditioned on CoP transfer to a multi-fingered hand without any task-specific real-world training, and outperform both coarse binary-contact and raw-taxel baselines. Analysis of learned policy states further suggests that CoP-conditioned policies encode task-relevant physical properties, such as object mass, as an emergent byproduct of control.

1 INTRODUCTION

A central challenge in robotics is contact-rich dexterous manipulation. While recent advances in imitation learning from large-scale human demonstrations (Bjorck et al., 2025; Cheng et al., 2025; Zhang et al., 2025) have shown impressive progress, the scalability of these approaches is fundamentally limited by the high cost and human effort required for real-world data collection. Sim-to-real reinforcement learning (RL) offers a compelling alternative through large-scale, autonomous learning in simulation. However, its success has been largely confined to relatively simple tasks (Akkaya et al., 2019; Handa et al., 2023; Chen et al., 2022; Lin et al., 2024b; 2025a), where policies can be learned from low-dimensional features that reliably transfer from simulation to the real world. Among common input modalities, tactile inputs are often avoided or heavily simplified, as accurately simulating tactile sensing is notoriously difficult. Tactile sensor responses depend on complex, high-dimensional, and often unmodeled physical processes, and vary widely due to the lack of standardized sensor designs.

As manipulation tasks become more contact-rich and demand greater precision, control policies must reason over more fine-grained physical interactions – making tactile sensing more essential for policy learning. The lack of an effective tactile representation thus becomes a key bottleneck in extending sim-to-real RL to more challenging settings. At the core of this bottleneck lies a representation trade-off: simplistic tactile features (Miller et al., 2025; Yin et al., 2023; Qi et al., 2023b) enable more robust sim-to-real transfer, while representations closer to raw tactile signals (Higuera et al., 2025; Sharma et al., 2025; Yin et al., 2025; Chen et al., 2026) contain richer contact information necessary for more complex manipulation. To balance this trade-off, we introduce **C**enter-**o**f-**P**ressure (**CoP**), a physics-grounded representation that summarizes local tactile information as a 3D contact force vector and a 3D contact location. CoP provides a middle ground for effective tactile sim-to-real transfer: it is compact enough to align across simulation and hardware, yet expressive enough

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to preserve the contact-force magnitude, direction, and location information needed for contact-rich dexterous control. To support this representation, we further propose a sensor calibration method based on differentiable dynamics, enabling the estimation of CoP without requiring ground-truth force measurements.

Existing evaluations can make it difficult to isolate the role of touch, either because policies rely heavily on vision (Chen et al., 2023b; Heng et al., 2026; Sferrazza et al., 2024) or because the tasks involve relatively simple periodic manipulation (Qi et al., 2023a; Chen et al., 2023a; Yang et al., 2025; Hsieh et al., 2025). To address this, we introduce two challenging *blind* manipulation tasks that explicitly require tactile feedback with minimal visual cues: peg-in-hole insertion and ball balancing. We evaluate our approach through extensive experiments, demonstrating that policies conditioned on CoP achieve successful zero-shot sim-to-real transfer on multi-fingered hands and outperform both simplified and raw tactile baselines. Importantly, we find that learned policies implicitly capture underlying physical properties – such as object mass – highlighting the potential of physically grounded representations for enabling more general and scalable manipulation.

2 Related Work

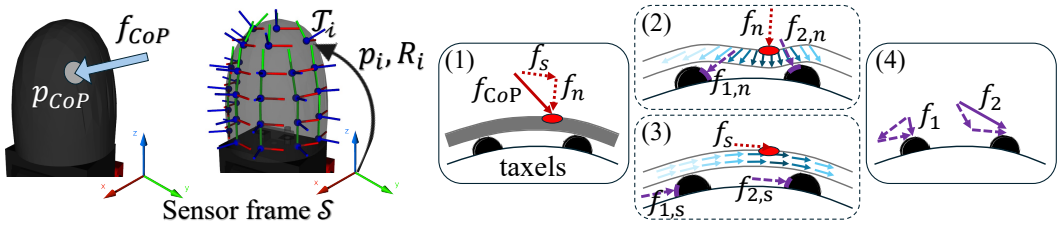
Imitation Learning for Dexterous Manipulation. Recent advances in teleoperation (Cheng et al., 2025; Lin et al., 2025b; Wu et al., 2024; Zhao et al., 2023) and imitation learning (Chi et al., 2025; Li et al., 2024) have enabled remarkable progress in dexterous manipulation. However, teleoperation data remains costly to scale, and achieving high levels of success and robustness requires large real-world demonstration datasets (Zhao et al., 2025; Levine et al., 2018; Lin et al., 2024a), making purely supervised methods expensive for reaching human-level dexterity on complex, dynamic tasks.

Sim-to-Real Transfer of Contact-Informed Policies. Sim-to-real RL has demonstrated success for learning manipulation policies in simulation and transferring onto real-world hardware. Earlier works have used proprioceptive joint control error as an *implicit* representation of external contact (Qi et al., 2023a); however, the resulting performance gains have been found to be limited (Chen et al., 2023c). The recent emergence of diverse tactile sensors has inspired many *explicit* contact representations, primarily simplified contact representations such as binary (Miller et al., 2025; Yin et al., 2023) and discretized ternary signals (Qi et al., 2023b). While these representations can transfer reliably, the coarse discretization discards high-fidelity contact information that is critical for fine-grained contact-rich control. A few recent works have attempted to directly model and simulate the raw tactile sensor behaviors (Yin et al., 2025; Sferrazza et al., 2024; Dang et al., 2026; Akinola et al., 2025), often by training a tactile encoder to compress the raw simulated sensor activations to a latent observation vector for the policy. Although these approaches better capture the complexity of real-world tactile signals, the resulting representations are often sensor-specific, less interpretable, and harder to align with simulator contact quantities. A recent work (Yang et al., 2025) employed a sensor-specific contact representation consisting of force scalar magnitude and contact position in polar coordinates. However, a learned tactile encoder is still used, and the evaluation is limited to only in-hand rotation. In contrast to these learned or sensor-specific representations, CoP adapts the classical principle of *intrinsic contact sensing* (Bicchi et al., 1993) – recovering a resultant contact force and location from force measurements – into a differentiable, taxel-array-based descriptor that aligns directly with the contact quantities exposed by rigid-body simulators.

3 Methodology

3.1 Physics-Aware Contact Representation

We propose **Center-of-Pressure (CoP)** as a physics-aware contact representation. On a round fingertip of a robot hand (Fig. 1a), CoP consists of a 3D force vector ${}^{\mathcal{S}}f_{\text{cop}} \in \mathbb{R}^3$ representing the total contact force acting on the robot link, and a 3D Cartesian contact position ${}^{\mathcal{S}}p_{\text{cop}} \in \mathbb{R}^3$. Both the force vector and the contact point are expressed in the sensor frame $\mathcal{S}$. Most rigid-body simulators such as IsaacSim (NVIDIA) and MuJoCo (Todorov et al., 2012) provide pairwise contact information between simulation bodies, including 3D contact force vector and the contact position expressed in the world frame. Here, it is worth highlighting that we use the term CoP to denote the resultant contact wrench reduced to a single force vector applied at a centroidal contact point. This approximation



(a) **Left**: CoP consists of a contact force vector f_{cop} and contact point p_{cop}, both expressed in $\mathcal{S}$. **Right**: Transformation between taxel frame $\mathcal{T}_i$ and sensor frame $\mathcal{S}$.

(b) (1) A contact force f_{cop} is decomposed into normal f_n and shear f_s components. We model the (2) normal $f_{i,n}$ and (3) shear $f_{i,s}$ forces on each taxel i under surface deformation. (4) Components are combined to obtain the resulting taxel force f_i.

Figure 1: (a) CoP representation. (b) The proposed stress distribution model for XELA uSkin sensors (Tomo et al., 2016).

is intended as a compact local contact descriptor, rather than a complete representation of arbitrary multi-contact pressure distributions which may involve more complex finite-element modeling.

3.2 Taxel-CoP Mapping

We show how raw tactile sensor readings can be mapped to CoP representation to enable direct sim-to-real transfer of learned policies, assuming readings are obtained from tactile sensors similar to the XELA uSkin (Tomo et al., 2016). Each sensor array consists of N *taxels* in some grid-like arrangement, and each taxel is an individual tactile sensing point providing 3-axis force measurements. For the i-th taxel, the position and orientation of its local coordinate frame $\mathcal{T}_i$ relative to the sensor frame $\mathcal{S}$ are given by ${}^{\mathcal{S}}p_i \in \mathbb{R}^3$ and $R_i \in \mathbb{SO}(3)$ respectively, as shown in Fig. 1a (right). Each taxel force ${}^{\mathcal{T}_i}f_i \in \mathbb{R}^3$ expressed in its local taxel frame $\mathcal{T}_i$ includes both normal and shear components. We denote the inward, unit surface normal vector at the i-th taxel as ${}^{\mathcal{S}}\hat{n}_i \in \mathbb{R}^3$, which can be easily obtained as each taxel is arranged such that one of its frame axes is orthogonal to the surface at its position.

Let ${}^{\mathcal{S}}f_{\text{cop}}$ represent an unknown contact force vector on the fingertip. This results in taxel force activations ${}^{\mathcal{T}_i}f_i$, which are transformed into the sensor frame as ${}^{\mathcal{S}}f_i$. We derive a *differentiable bidirectional* mapping between raw taxel forces $\{{}^{\mathcal{T}_i}f_i\}_{i=1\ldots N}$ and the CoP representation $\{{}^{\mathcal{S}}f_{\text{cop}}, {}^{\mathcal{S}}p_{\text{cop}}\}$ under a parametric stress-distribution model. From here, since all vector quantities will be defined and expressed in the sensor frame $\mathcal{S}$, we omit frame labeling for better readability.

Stress Distribution Model. Directly summing or averaging individual taxel forces (Lee et al., 2025) does not account for force spreading through the compliant silicone layer covering the tactile sensors, and can therefore produce biased estimates of both the resultant force and the contact location. Instead, we propose a simple model as illustrated in Fig. 1b. First, the CoP force vector f_{cop} is decomposed into its normal f_n and shear f_s components. Then, we model the internal force distribution by accounting for change in force direction due to deformation and force magnitude decay proportional to distance from the contact point. For each taxel i, this results in *effective* normal $f_{i,n}$ and shear $f_{i,s}$ forces, which are combined into the taxel forces f_i.

Forward Mapping Derivation. We estimate the CoP position p_{cop} as the force-magnitude-weighted average of the individual taxel positions p_i. To eliminate idle sensor noise, we include only the active index set $\mathcal{A} = \{i : \|f_i\| > \epsilon\}$, where ϵ is a threshold value:

$$p_{\text{cop}} = \frac{\sum_{i\in\mathcal{A}} \|f_i\| \, p_i}{\sum_{i\in\mathcal{A}} \|f_i\|}$$

To account for the curved fingertip geometry, we approximate the inward unit surface normal at the CoP position, $\hat{n}_{\text{cop}} \in \mathbb{R}^3$, via inverse distance weighting (Shepard, 1968) of the taxel normals $\hat{n}_i$:

$$\hat{n}_{\text{cop}} = \text{normalize}\left(\sum_{i\in\mathcal{A}} \frac{\alpha_i}{\sum_{j\in\mathcal{A}} \alpha_j} \cdot \hat{n}_i\right), \quad \alpha_i = \frac{1}{\|p_i - p_{\text{cop}}\|}$$

To model internal stress distribution under deformation, we use a "blended" unit vector $\hat{b}_i$ to approximate the direction of the effective normal force $f_{i,n}$. Specifically, we interpolate between

the taxel's local normal $\hat{n}_i$ and the relative unit direction vector $\hat{v}_i = \frac{p_i - p_{\text{cop}}}{\|p_i - p_{\text{cop}}\|}$ from the CoP position to the taxel, $\hat{b}_i = \text{normalize}\,(w_i \hat{n}_i + (1 - w_i)\hat{v}_i)$, where w_i is a Gaussian radial weight $w_i = \exp(-\frac{\|p_i - p_{\text{cop}}\|^2}{2\sigma^2})$, and σ is a spread hyperparameter. We approximate the magnitude of $f_{i,n}$ using the same Gaussian decay model on the magnitude of f_n. For the effective shear force $f_{i,s}$, we define a shear projection matrix $P_{\text{shear}} = I_3 - \hat{n}_{\text{cop}} \hat{n}_{\text{cop}}^\top \in \mathbb{R}^{3\times3}$ which projects f_{cop} onto the surface tangent plane at p_{cop} to approximate the direction of $f_{i,s}$, and use the same Gaussian decay model for the magnitude. Altogether, the relation between f_i and f_{cop} can be compactly written as:

$$f_i = M_i f_{\text{cop}}, \quad M_i = w_i(\hat{b}_i \hat{n}_{\text{cop}}^\top + P_{\text{shear}}) \in \mathbb{R}^{3\times3}$$

This low-parameter model captures the dominant distance-dependent spreading effect of the stress field while remaining differentiable and easy to align across simulation and hardware.

Solving for the CoP Force. To find the unknown f_{cop} from the observed set of active taxel forces $\{f_i\}$, we aggregate the individual taxel equations into a global linear system $Af_{\text{cop}} = b$ and obtain f_{cop} as the closed-form solution to the following regularized least-squares problem:

$$f_{\text{cop}} = (A^\top A + \lambda^2 I)^{-1} A^\top b, \quad A = [M_1^\top, \ldots, M_N^\top]^\top, b = [f_1^\top, \ldots, f_N^\top]^\top$$

With this relatively simple representation, our contact model is not only computationally efficient for policy learning in simulation, but also fully differentiable. This differentiability enables gradient-based learning as presented in Section 3.3. Because the same differentiable model can be evaluated in both directions, it enables practical bidirectional conversion between simulated CoP contacts and hardware taxel readings.

3.3 Sensor Calibration via Differentiable Dynamics

In the taxel-CoP mapping, while the taxel origins ${}^{\mathcal{S}}p_i$ are easily obtainable from the sensor specifications (Tomo et al., 2016), the rotations R_i of the taxel frames $\mathcal{T}_i$ relative to the sensor frame $\mathcal{S}$ are unknown and difficult to manually calibrate due to the complex surface geometry of the fingertip. We therefore estimate the unknown taxel orientations by matching the external wrench inferred from tactile readings to the joint torques required to maintain static equilibrium.

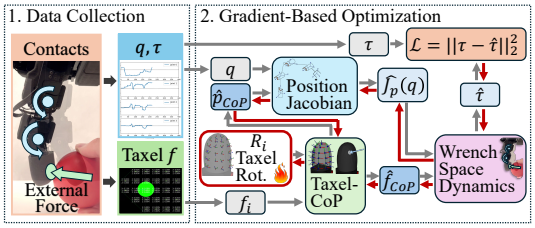


Figure 2: Our proposed differentiable dynamics-based sensor calibration method, consisting of 1) data collection and 2) gradient-based optimization. Red arrows indicate gradient flow during back-propagation.

Data Collection. Consider the kinematic chain from the robot hand's base frame $\mathcal{B}$ to the sensor frame $\mathcal{S}$ of an arbitrary fingertip. To collect data, we first run a stiff PD controller continuously to maintain a fixed set of nominal finger joint positions. Then, we apply random contacts onto the fingertip (see Fig. 2), such that the joint actuators actively apply torques to balance the external force and maintain static equilibrium. At each time step, we record the raw taxel forces ${}^{\mathcal{T}_i}f_i$, applied joint torques $\tau \in \mathbb{R}^4$, and joint angles $q \in \mathbb{R}^4$. To improve identifiability, we apply contacts from diverse fingertip locations and directions, ensuring that the collected dataset excites both normal and shear responses across taxels.

Rotation Parameterization. For choosing how to parameterize the taxel rotations R_i to be learned, we follow recommendations in (Geist et al., 2024) and use the $\mathbb{R}^9$+SVD method, which parameterizes a rotation by an arbitrary 3×3 matrix $P \in \mathbb{R}^{3\times3}$ and projects it to the valid rotation matrix $R \in \mathbb{SO}(3)$ with the least-squares distance to P using its Singular Value Decomposition (SVD):

$$R = \text{SVD}^+(P) = U\text{diag}(1, 1, \det(UV^\top))V^\top, \quad P = U\Sigma V^\top$$

We initialize the learnable rotation parameters as a PyTorch tensor of shape $(N, 3, 3)$ consisting of parameter matrices $P_i \in \mathbb{R}^{3\times3}$ for all N taxels in the sensor array.

Optimization. During training, for each data sample, we first rotate the recorded taxel forces ${}^{\mathcal{T}_i}f_i$ into sensor frame $\mathcal{S}$ using the projected rotations $\hat{R}_i$ based on the current $\hat{P}_i$. Next, the taxel-to-CoP mapping is applied to obtain the estimated CoP force vector ${}^{\mathcal{S}}\hat{f}_{\text{cop}}$ and contact position ${}^{\mathcal{S}}\hat{p}_{\text{cop}}$,

which are transformed into base frame $\mathcal{B}$ via the forward kinematics computed using the recorded joint angles q. Then, we compute the position Jacobian matrix ${}^{\mathcal{B}}\hat{J}_{\text{cop}} \in \mathbb{R}^{3\times 4}$ at the CoP contact point using q and ${}^{\mathcal{B}}\hat{p}_{\text{cop}}$. Under static equilibrium conditions, the following wrench-space relation with all quantities expressed in a common coordinate frame (e.g. base frame) holds (Huang et al., 2011):

$$\tau = -J^\top f + g(q) \approx -J^\top f$$

where f is an external force *acting on* the body, J is the position Jacobian at the contact point, and $g(q)$ is a gravity-compensation term that we assume negligible for simplicity. Using this relation, we compute the expected joint torques $\hat{\tau}$ required to balance the estimated CoP force vector ${}^{\mathcal{B}}\hat{f}_{\text{cop}}$:

$$\hat{\tau} = -{}^{\mathcal{B}}\hat{J}_{\text{cop}}^\top \, {}^{\mathcal{B}}\hat{f}_{\text{cop}}$$

Finally, we compute the MSE loss between the estimated $\hat{\tau}$ and recorded joint torques τ, and back-propagate the gradients to obtain improved estimates of rotation parameters $\hat{P}$. Implementation and training details are included in Appendix A.

3.4 Sim-Real Alignment

Contact Representation. We use the $\mathrm{ContactSensor}$ API in IsaacLab (Mittal et al., 2025) to track contacts between the taxel-covered regions of each fingertip link and the contact object. Because IsaacLab's contact estimates for our fingertip geometries produced unreliable shear components, we use only the surface-normal component of CoP in both simulation and hardware. This choice intentionally sacrifices shear information to maintain sim-to-real robustness. To minimize the sim-to-real gap in contact dynamics, we calibrate the taxel-CoP mapping parameters (see Section 3.2) using simple, predefined movement trajectories to collect data from simulation and hardware. This lightweight calibration is performed once per sensor geometry and does not require task-specific real-world policy learning. Visualizations of the aligned observations are included in Appendix B.

Actuator Dynamics. Because subtle actuator dynamics (e.g. non-uniform friction) are notoriously difficult to analytically model, we employ a Bayesian optimization-based system identification approach inspired by prior real-to-sim approaches (Lin et al., 2025a; He et al., 2025) to align the simulated actuator dynamics with the hardware. Details and visualizations are included in Appendix C.

Sensor Delay. The tactile sensors introduce non-negligible delay from time-of-contact to the policy receiving it as observation, which is critical for highly dynamic tasks. We measure sensor delay via vision-based methods and introduce it into simulation during policy training.

4 Experiments

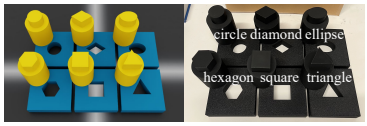
Tasks. We use a 16-DOF Allegro hand equipped with XELA uSkin sensors covering the fingertips, phalanges, and palm of the hand. We evaluate on two challenging contact-rich manipulation tasks: **peg-in-hole insertion** and **ball-balancing**. For both tasks, we train a *blind* policy conditioned only on proprioception (current and commanded joint angles) and contact observations. For *peg-in-hole insertion*, a single policy is trained to generalize across different insertion shapes, and similarly a single policy is trained for different ball types in the **ball-balancing** task. While existing dexterous manipulation tasks often involve only *primary* contacts (interaction between the hand and the manipulated object), our tasks require the policy to infer *secondary* contacts (interaction between the manipulated object and another task-relevant object). Success in these tasks crucially depends on the ability of the policy to implicitly infer the state of secondary contacts through the tactile feedback of primary contacts, making them particularly challenging.

Baselines. We compare our proposed CoP representation (*cop*) against several baselines, including proprioception (*base*), binary contact per sensing array (*binary*), and CoP force magnitude (*mag*). In addition, we include force vector-only (*vec*) and contact position-only (*pos*) as self-ablated baselines, and raw taxel forces (*taxel*). We also include an expert human (*human*) for comparison.

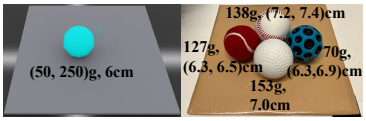
Policy Architecture. A common approach of providing temporal information to the policy is to condition it on a flattened stack of past observations $o_{t-N:t}$, which has demonstrated effectiveness in robot manipulation (Miller et al., 2025; Sferrazza et al., 2024; Qi et al., 2023a; Hsieh et al., 2025). We instead use a recurrent policy architecture to provide temporal context without increasing the

Table 1: Performance comparison of different contact representations for the peg-in-hole insertion task. We report success rate and task completion time (s) across 10 trials for each condition and each insertion shape.

	Circle		Diamond		Ellipse		Hexagon		Square		Triangle		Overall		OOD Init.		Masked	
	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓	sr↑	time↓
base	0.8	5.36 ±0.79	0.2	4.28 ±0.35	0.3	2.85 ±0.11	0.6	2.54 ±0.28	0.4	2.72 ±0.74	0.3	10.36 ±1.31	0.43	4.65 ±2.80	0.17	11.76 ±6.60	-	-
bin	1.0	2.60 ±0.34	0.2	10.82 ±3.54	0.3	2.58 ±0.87	0.8	8.30 ±3.42	0.6	16.73 ±4.03	0.3	26.29 ±0.32	0.53	10.15 ±8.57	0.20	14.40 ±5.83	0.52	14.87 ±17.30
mag	1.0	3.10 ±0.34	0.5	15.94 ±7.88	0.4	3.95 ±1.25	0.8	13.03 ±7.90	0.5	14.08 ±10.53	0.1	23.50 ±0.00	0.55	9.47 ±9.73	0.27	19.33 ±7.00	0.48	11.13 ±16.12
vec	1.0	2.97 ±0.64	0.4	9.32 ±6.54	0.5	6.89 ±1.30	1.0	8.06 ±10.29	0.7	3.43 ±1.98	0.4	19.56 ±2.60	0.67	7.19 ±7.60	0.42	20.79 ±5.07	0.57	13.09 ±14.58
pos	1.0	2.56 ±0.30	0.3	15.59 ±10.96	0.2	27.62 ±6.31	0.5	4.71 ±0.31	0.6	7.68 ±4.28	0.4	24.40 ±1.00	0.50	10.19 ±10.12	0.28	11.96 ±8.01	0.48	8.12 ±10.53
taxel	0.8	9.02 ±9.78	0.2	20.89 ±8.26	0.4	5.57 ±0.17	0.6	2.94 ±0.56	0.6	17.65 ±6.34	0.3	18.08 ±10.36	0.48	10.94 ±9.81	0.27	15.44 ±10.07	0.30	9.09 ±14.33
cop (ours)	**1.0**	**4.22** ±0.97	**0.6**	**14.89** ±7.91	**0.6**	**7.61** ±4.85	**1.0**	**6.06** ±2.99	**0.9**	**13.61** ±6.11	**0.6**	**20.37** ±6.39	**0.78**	**10.34** ±7.62	**0.63**	**15.21** ±6.54	**0.62**	**12.70** ±14.71
human	1.0	0.78 ±0.19	1.0	2.99 ±1.13	1.0	2.49 ±1.32	1.0	1.17 ±0.30	1.0	1.69 ±0.94	1.0	3.07 ±1.19	1.0	2.03 ±1.32	-	-	-	-



(a) Custom insertion task assets.



(b) Simulation and real-world assets for the ball balancing task.

Figure 3: Task assets.

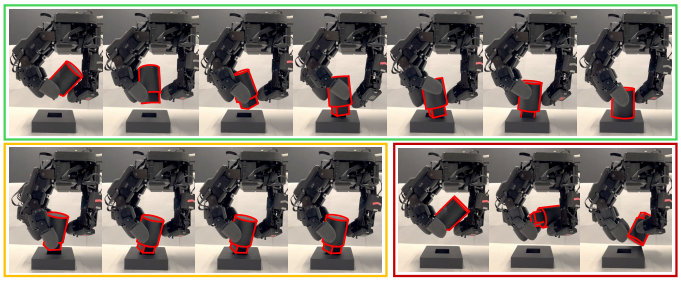


Figure 4: **Top**: The *cop* insertion policy succeeds under OOD initialization. **Bottom**: The *binary* policy fails under external contacts (left) and OOD initialization (right). The peg object is outlined in red.

observation dimension through explicit history stacking. For both tasks, we found that this achieved better sample efficiency and performance over a direct MLP network on stacked history observations (see Appendix D for details).

Direct Sim-to-Real Transfer. We use IsaacLab (Mittal et al., 2025) and train policies using asymmetric actor-critic PPO (Schwarke et al., 2025). The actor observation includes proprioception and the contact representation. The critic additionally receives privileged information such as object states and other task-relevant information. Both networks share the same recurrent architecture. Here, we note that prior works often employ teacher-student distillation (Qi et al., 2023b; Yin et al., 2025; Chen et al., 2026; Qi et al., 2023a) because the tactile observations available in simulation and hardware are not directly aligned. In contrast, our aligned CoP representation enables direct sim-to-real transfer. During training, we systematically introduce domain randomization to robustify the policy against real-world variations. Full training details, including observations, actions, rewards, resets, domain randomization, and PPO hyperparameters are included in Appendix E.

4.1 Peg-in-Hole Insertion

Task Setup. We generate six pairs of custom peg and hole assets with the same cylindrical handle and a variety of insertion head shapes, as shown in Fig. 3a (see Appendix F for asset details). The task requires the hand to maintain grasp of the peg and fully insert it into the hole object which is fixed to the table. At each reset, we fully randomize the yaw orientation of the peg, and also introduce small randomizations in its position from manual reset noise. We conduct 10 trials for each observation and each shape, and record the success rate (sr) and task completion time (time), which is averaged over successful trials only.

Real-World Performance. As summarized in Table 1, *cop* achieves the highest overall success rate and outperforms all baselines on most insertion shapes. Higher-fidelity contact representations (e.g. *vec*, *cop*) generally lead to more adaptive and persistent policies that achieve higher success rates (Fig. 4, top) but result in longer task completion times than simplified representations (e.g. *base* and *bin*). On the other hand, simplified representations achieve fast initial insertions only in certain yaw-initializations and fail to adapt to peg-hole contacts and OOD initializations (Fig. 4, bottom), suggesting that these representations do not provide sufficient tactile information for recovery once

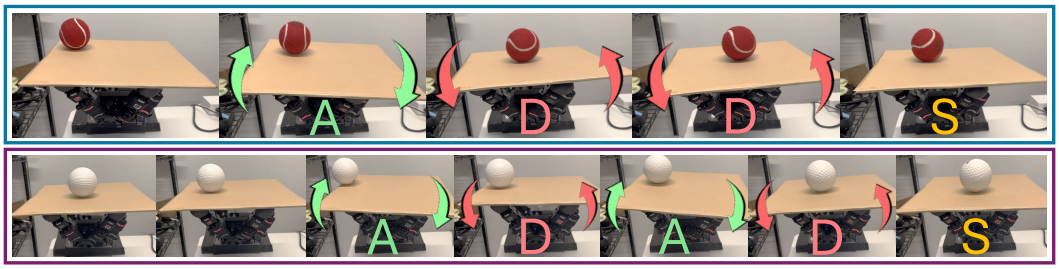


Figure 5: Examples of emergent movement patterns of the *cop* policy to center and balance the ball on the plate. "A", "D" and "S" indicate accelerate, decelerate, and stabilize, respectively.

the initial insertion attempt fails. The ablations using force-only (*vec*) and position-only (*pos*) observations indicate that both components contribute to task performance, though the force channel is the stronger of the two. Moreover, *taxel* performs worse than nearly all other baselines, possibly due to a combination of imperfect tactile simulation, higher dimensionality, and sensor-specific mismatch. This finding resonates with prior works in tactile simulation for sim-to-real policy transfer, which attempt to alleviate this problem by either training an additional tactile encoder (Yin et al., 2025) or relying on complex modeling and precise real-world data calibration (Dang et al., 2026; Su et al., 2026). Finally, human performance significantly surpasses the best performing robot policy in both metrics. One possible explanation is that humans combine tactile cues with high-level geometric reasoning and exploratory strategies, whereas the robot policy relies on reactive contact feedback learned from simulation.

Robustness Against Out-of-Distribution Initialization. We repeated the experiment under out-of-distribution peg pose initialization (see Fig. 4). As summarized in Table 1 (OOD Init.), while all policies exhibit longer task completion times as more movements are required to shift the peg downwards towards the hole, *cop* maintains the least reduction in success rate while demonstrating the emergent capabilities of persistent and robust in-hand object translation and re-orientation to achieve peg-hole alignment for successful insertion (Fig. 4, top).

Robustness Against Masked Sensor. We repeated the experiment while randomly masking 40% of the raw hardware taxel forces at each time step. As shown in Table 1 (Masked), the higher-fidelity contact representations generally suffer from larger performance degradation than the simplified representations, as the former are more sensitive to precise individual taxel forces.

4.2 Ball Balancing

Task Setup. A lightweight (50g) square plate is supported by the four fingertips, and the hand is required to balance a ball placed on the plate and keep it centered as best as possible. As shown in Fig. 3b, the policy is trained in simulation using a smooth sphere and evaluated in the real world on a variety of four balls with different mass, size, friction, and surface texture, resulting in significantly different rolling behaviors outside of the training distribution. Moreover, this task is particularly challenging for several other reasons. First, we observed that subtle actuator dynamics such as non-uniform friction and gear backlash are difficult to precisely simulate, which can significantly degrade performance. Second, one of the fingertips does not have its taxel-covered region in contact with the plate and hence lacks explicit contact information. Finally, 3/4 of the fingers are close to singularity configurations, making it challenging to achieve rapid fingertip movements. We conduct 10 trials for each observation and each ball type, and record the time-to-fall (TTF).

Real-World Performance. As shown in Table 2, precise force information is crucial for this task, since only *cop*, *vec*, and *taxel* policies, which are conditioned on numerical force values, successfully learned the task in simulation (Fig. 13b). Moreover, the comparable real-world performance between *cop* and *vec* policies suggests that, on a curved fingertip where the normal direction already encodes contact location, the explicit position channel adds little for this task. Fig. 5 shows two distinct emergent movement patterns, including a more aggressive single-step accelerate-decelerate maneuver (top) and a slower two-step centering process (bottom). Interestingly, while the robot policy struggles more with the smoother and faster-rolling hockey ball, the human expert struggles the most with the dented moon ball which exhibits nonlinear, unpredictable rolling. As suggested by prior research, one possible explanation is that humans rely more heavily on predictive extrapola-

Table 2: Performance comparison of different contact representations for the ball balancing task. We report time-to-fall (s) across 10 trials for each condition and each ball type.

	Tennis ball↑	Baseball↑	Moon ball↑	Hockey ball↑	Overall↑
base	$1.38_{\pm 0.17}$	$1.42_{\pm 0.15}$	$1.50_{\pm 0.22}$	$1.24_{\pm 0.16}$	$1.38_{\pm 0.21}$
bin	$2.20_{\pm 1.02}$	$2.22_{\pm 1.44}$	$1.78_{\pm 0.71}$	$1.75_{\pm 0.45}$	$1.99_{\pm 1.03}$
mag	$2.83_{\pm 0.97}$	$2.17_{\pm 0.48}$	$2.35_{\pm 0.36}$	$2.25_{\pm 0.90}$	$2.40_{\pm 0.79}$
vec	$5.59_{\pm 3.52}$	$3.27_{\pm 0.41}$	$4.59_{\pm 1.11}$	$2.80_{\pm 0.91}$	$4.52_{\pm 2.93}$
pos	$1.63_{\pm 0.15}$	$1.59_{\pm 0.30}$	$1.70_{\pm 0.16}$	$1.26_{\pm 0.14}$	$1.55_{\pm 0.27}$
taxel	$1.38_{\pm 0.33}$	$1.73_{\pm 0.31}$	$1.61_{\pm 0.34}$	$1.22_{\pm 0.10}$	$1.49_{\pm 0.36}$
cop (ours)	$\mathbf{5.07}_{\pm \mathbf{2.13}}$	$\mathbf{4.77}_{\pm \mathbf{1.55}}$	$\mathbf{4.50}_{\pm \mathbf{1.29}}$	$\mathbf{3.06}_{\pm \mathbf{0.64}}$	$\mathbf{4.60}_{\pm \mathbf{2.19}}$
human	$11.29_{\pm 6.55}$	$9.41_{\pm 4.46}$	$5.96_{\pm 0.78}$	$10.82_{\pm 5.18}$	$9.37_{\pm 5.32}$

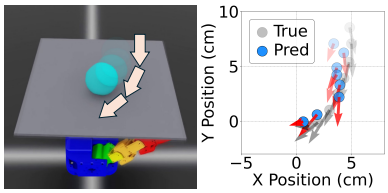


Figure 6: Predicted and actual ball state (xy-plane position & velocity) in a simulated rollout.

tion using a *linear* model (de Langhe et al., 2017; Tsutsui et al., 2021). In contrast, the robot policy may be more reactive towards the immediate state rather than extrapolating future information for control.

Object State Prediction. To understand how a trained policy leverages the explicit CoP information for control, we analyze the 256-dim latent output of the recurrent layer of the policy network. We first collected a training set of 1000 5s trajectories (100 samples each), and performed *linear probing* to predict the ball state consisting of position $p \in \mathbb{R}^2$ and velocity $v \in \mathbb{R}^2$ in the xy-plane. Then, we collected a test set of 100 5s trajectories, and compute the RMSE and r^2 scores between the true states and model predictions. As shown in Table 3, the policy's latent representation captures the ball's position well along x ($r^2 = 0.76$) and moderately along y ($r^2 = 0.62$). However, velocity prediction is noticeably weaker, which indicates that while the policy leverages contact information to track ball position, it may not encode motion dynamics as precisely, likely due to the inherent noise in contact-based state estimation. Fig. 6 illustrates the predicted and actual ball states in an example policy rollout trajectory.

Implicit Mass Identification. We further examine whether a trained policy's latent representation implicitly captures dynamical properties of the ball such as its mass. We choose 3 different ball masses (50g, 150g, 250g) and collect 100 5s trajectories for each mass. Then, we perform PCA to transform all 256-dim latent embeddings into their first two principal components.

Table 3: Accuracy of ball state prediction via linear-probing of the policy latents.

	x pos	y pos	x vel	y vel
RMSE (m)	0.013	0.019	0.059	0.065
r^2 score	0.76	0.62	0.23	0.15

Fig. 7 reveals that as trajectories evolve over time, the latent embeddings naturally reorganize into distinct clusters corresponding to different mass values, as seen through the gradually increasing Silhouette Coefficient (SC). This suggests that conditioning on CoP enables the recurrent policy state to organize around task-relevant physical properties, such as object mass, even without explicit supervision. We note this latent analysis is conducted in simulation, where ground-truth ball state and mass are available for probing.

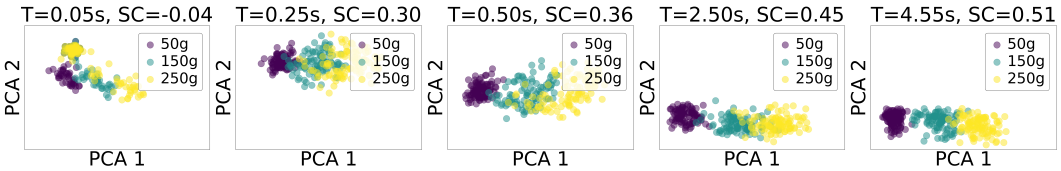


Figure 7: Visualization of latent-embedding clusters over the course of trajectories: clusters separate by object mass (50/150/250 g) as the policy accumulates contact history.

5 Conclusion

We propose Center-of-Pressure (CoP), a physics-aware contact representation that reduces the tactile sim-to-real gap by aligning real taxel readings with contact quantities available in simulation. We demonstrate its effectiveness through systematic evaluations on dynamic "blind" manipulation tasks without visual input, and provide insight into the emergent latent representations of the learned policy conditioned on the contact representation. These results suggest that physically grounded intermediate tactile representations are a promising path toward scalable sim-to-real learning for contact-rich dexterous manipulation.

6 LIMITATIONS & FUTURE WORK

(A) Fidelity vs. Transferability. CoP intentionally abstracts raw taxel readings into contact force and location information that is more transferable across simulation and hardware. This improves robustness under imperfect tactile simulation, but also discards some sensor-specific details that may be useful for more complex manipulation. Though harder to calibrate and less likely to generalize across sensors, richer raw tactile representations may still offer higher performance when paired with accurate sensor-specific models.

(B) Sim-Real Contact Discrepancies. Our current implementation restricts CoP force vectors to the surface-normal direction because simulated shear force estimates were unreliable. This reflects an important limitation of current contact simulation. In addition, the simulator used in our experiments reports only task-object contacts, whereas real tactile sensors respond to all contacts, including self-collisions and environmental interactions. This mismatch may introduce out-of-distribution tactile observations in more diverse settings.

(C) Scope and Future Directions. We focus on fixed-base dexterous hands with XELA uSkin sensors to isolate the effect of tactile representation. Extending CoP to arm-hand systems, full-hand tactile coverage, and other tactile sensor types remains important future work. Beyond sim-to-real RL, we also hope to integrate CoP as a physically grounded tactile modality for imitation learning and sample-efficient real-world RL.

AI USE STATEMENT

In this work, we used generative AI tools for research ideation and execution, and to aid in polishing the writing. We have not used generative AI tools for retrieval and discovery of related work, for drafting sections of the paper, nor for generating synthetic datasets. Using AI for proving mathematical claims is not applicable to this work. We have reviewed all AI-assisted work, including checking LLM-generated research ideas for potential plagiarism through manual literature surveys, and manually verifying and testing LLM-generated code for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Iretiayo Akinola, Jie Xu, Jan Carius, Dieter Fox, and Yashraj Narang. Tacsl: A library for visuotactile sensor simulation and learning. *IEEE Transactions on Robotics*, 2025.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Antonio Bicchi, J Kenneth Salisbury, and David L Brock. Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249–262, 1993.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Elie Chelly, Andrea Cherubini, Philippe Fraisse, Faïz Ben Amar, and Mahdi Khoramshahi. Tactile-based force estimation for interaction control with robot fingers. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15800–15807. IEEE, 2025.

Rosy Chen, Mustafa Mukadam, Michael Kaess, Tingfan Wu, Francois R Hogan, Jitendra Malik, and Akash Sharma. Ptld: Sim-to-real privileged tactile latent distillation for dexterous manipulation. *arXiv preprint arXiv:2603.04531*, 2026.

Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pp. 297–307. PMLR, 2022.

Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84): eadc9244, 2023a.

Yizhou Chen, Mark Van der Merwe, Andrea Sipos, and Nima Fazeli. Visuo-tactile transformers for manipulation. In *Conference on Robot Learning*, pp. 2026–2040. PMLR, 2023b.

Yuanpei Chen, Chen Wang, Li Fei-Fei, and Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In *Conference on Robot Learning*, pp. 3809–3829. PMLR, 2023c.

Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *Conference on Robot Learning*, pp. 2729–2749. PMLR, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Hojung Choi, Jun En Low, Tae Myung Huh, Seongheon Hong, Gabriela A Uribe, Kenneth AW Hoffmann, Julia Di, Tony G Chen, Andrew A Stanley, and Mark R Cutkosky. Coinft: A coin-sized, capacitive 6-axis force torque sensor for robotic applications. *arXiv preprint arXiv:2503.19225*, 2025.

An Dang, Jayjun Lee, Mustafa Mukadam, X Alice Wu, Bernadette Bucher, Manikantan Nambi, and Nima Fazeli. Hydroshear: Hydroelastic shear simulation for tactile sim-to-real reinforcement learning. *arXiv preprint arXiv:2603.00446*, 2026.

B Bart de Langhe, Stefano Puntoni, and Richard P Larrick. Linear thinking in a nonlinear world. *Harvard Business Review*, 95(3):130–139, 2017.

Kipp Freud, Yijiong Lin, and Nathan F Lepora. Simshear: Sim-to-real shear-based tactile servoing. In *9th Annual Conference on Robot Learning*, 2025.

Andreas René Geist, Jonas Frey, Mikel Zhobro, Anna Levina, and Georg Martius. Learning with 3d rotations, a hitchhiker's guide to so (3). In *International Conference on Machine Learning*, pp. 15331–15350. PMLR, 2024.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5977–5984. IEEE, 2023.

Tairan He, Zi Wang, Haoru Xue, Qingwei Ben, Zhengyi Luo, Wenli Xiao, Ye Yuan, Xingye Da, Fernando Castañeda, Shankar Sastry, et al. Viral: Visual sim-to-real at scale for humanoid loco-manipulation. *arXiv preprint arXiv:2511.15200*, 2025.

Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation. In *Robotics: Science and Systems*, 2026.

Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, et al. Tactile beyond pixels: Multisensory touch representations for robot manipulation. In *Conference on Robot Learning*, pp. 105–123. PMLR, 2025.

Elvis Hsieh, Wen-Han Hsieh, Yen-Jen Wang, Toru Lin, Jitendra Malik, Koushil Sreenath, and Haozhi Qi. Learning dexterous manipulation skills from imperfect simulations. *arXiv preprint arXiv:2512.02011*, 2025.

Tian Huang, HT Liu, and DG Chetwynd. Generalized jacobian analysis of lower mobility manipulators. *Mechanism and Machine Theory*, 46(6):831–844, 2011.

Yitaek Kim, Jeeseop Kim, Albert H Li, Aaron D Ames, and Christoffer Sloth. Robust adaptive safe robotic grasping with tactile sensing. In *2025 European Control Conference (ECC)*, pp. 2531–2538. IEEE, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Haegu Lee, Yitaek Kim, Victor Melbye Staven, and Christoffer Sloth. Trajectory optimization for in-hand manipulation with tactile force control. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 21773–21779. IEEE, 2025.

Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

Xuanlin Li, Tong Zhao, Xinghao Zhu, Jiuguang Wang, Tao Pang, and Kuan Fang. Planning-guided diffusion policy learning for generalizable contact-rich bimanual manipulation. *arXiv preprint arXiv:2412.02676*, 2024.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2024a.

Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. In *Conference on Robot Learning*, pp. 5220–5235. PMLR, 2024b.

Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. In *Conference on Robot Learning*, pp. 4926–4940. PMLR, 2025a.

Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5637–5643. IEEE, 2025b.

Elle Miller, Trevor McInroe, David Abel, Oisin Mac Aodha, and Sethu Vijayakumar. Enhancing tactile-based reinforcement learning for robotic control. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

NVIDIA. Isaac sim. URL `https://github.com/isaac-sim/IsaacSim`.

Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pp. 1722–1732. PMLR, 2023a.

Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pp. 2549–2564. PMLR, 2023b.

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. Rsl-rl: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9698–9705. IEEE, 2024.

Akash Sharma, Carolina Higuera, Chaithanya Krishna Bodduluri, Zixi Liu, Taosha Fan, Tess Hellebrekers, Mike Lambeta, Byron Boots, Michael Kaess, Tingfan Wu, et al. Self-supervised perception for tactile skin covered dexterous hands. In *Conference on Robot Learning*, pp. 2311–2328. PMLR, 2025.

Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pp. 517–524, 1968.

Lei Su, Zhijie Peng, Renyuan Ren, Shengping Mao, Juan Du, Kaifeng Zhang, and Xuezhou Zhu. Tacmap: Bridging the tactile sim-to-real gap via geometry-consistent penetration depth map. *arXiv preprint arXiv:2602.21625*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Tito Pradhono Tomo, Wai Keat Wong, Alexander Schmitz, Harris Kristanto, Alexandre Sarazin, Lorenzo Jamone, Sophon Somlor, and Shigeki Sugano. A modular, distributed, soft, 3-axis sensor system for robot hands. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 454–460. IEEE, 2016.

Kazushi Tsutsui, Keisuke Fujii, Kazutoshi Kudo, and Kazuya Takeda. Flexible prediction of opponent motion with internal representation in interception behavior. *Biological Cybernetics*, 115(5): 473–485, 2021.

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12156–12163. IEEE, 2024.

Max Yang, Alex Church, Yijiong Lin, Christopher J Ford, Haoran Li, Efi Psomopoulou, David AW Barton, Nathan F Lepora, et al. Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch. In *Conference on Robot Learning*, pp. 4727–4747. PMLR, 2025.

Jessica Yin, Haozhi Qi, Jitendra Malik, James Pikul, Mark Yim, and Tess Hellebrekers. Learning in-hand translation using tactile skin with shear and normal force sensing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5850–5856. IEEE, 2025.

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

Di Zhang, Chengbo Yuan, Chuan Wen, Hai Zhang, Junqiao Zhao, and Yang Gao. Kinedex: Learning tactile-informed visuomotor policies via kinesthetic teaching for dexterous manipulation. In *Conference on Robot Learning*, pp. 4123–4138. PMLR, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *Conference on Robot Learning*, pp. 1910–1924. PMLR, 2025.

Sheng Zhong, Thomas Power, Ashwin Gupta, and Peter Mitrano. PyTorch Kinematics, February 2024.

APPENDIX

A TAXEL ORIENTATION LEARNING - IMPLEMENTATION & TRAINING

We implement the forward pass calculations based on (Zhong et al., 2024) which provides differentiable implementations of forward kinematics and Jacobian matrix calculations. A dataset of 2400 samples (2 minutes at 20Hz control rate) containing random object-fingertip contacts was collected. We used Adam (Kingma & Ba, 2014) with a learning rate of 0.1 and performed batch gradient descent for 100 steps. An animation visualizing the training process is provided in the supplementary video.

Prior works mostly use high-precision force sensors to acquire ground-truth data for sensor calibration (Yang et al., 2025; Dang et al., 2026; Freud et al., 2025; Choi et al., 2025; Chelly et al., 2025; Kim et al., 2025). Instead, our method leverages the differentiability of robot dynamics and our proposed taxel-CoP mapping. Moreover, the framework can in theory learn any differentiable mapping function (e.g. neural network) from raw sensor readings to the CoP representation.

B CONTACT OBSERVATION ALIGNMENT

The proposed bidirectional taxel-CoP mapping enables us to both convert raw taxel activations into the CoP contact representation and compute taxel forces from a CoP contact interaction. Fig. 8 visualizes the computed CoP force vector and contact position from raw taxel forces recorded on hardware during real-world contact interactions with an external object. Fig. 9 visualizes the CoP contact information provided by the simulator and the computed raw taxel force vectors using the taxel-CoP mapping, where both show the forces on the middle fingertip as an example.

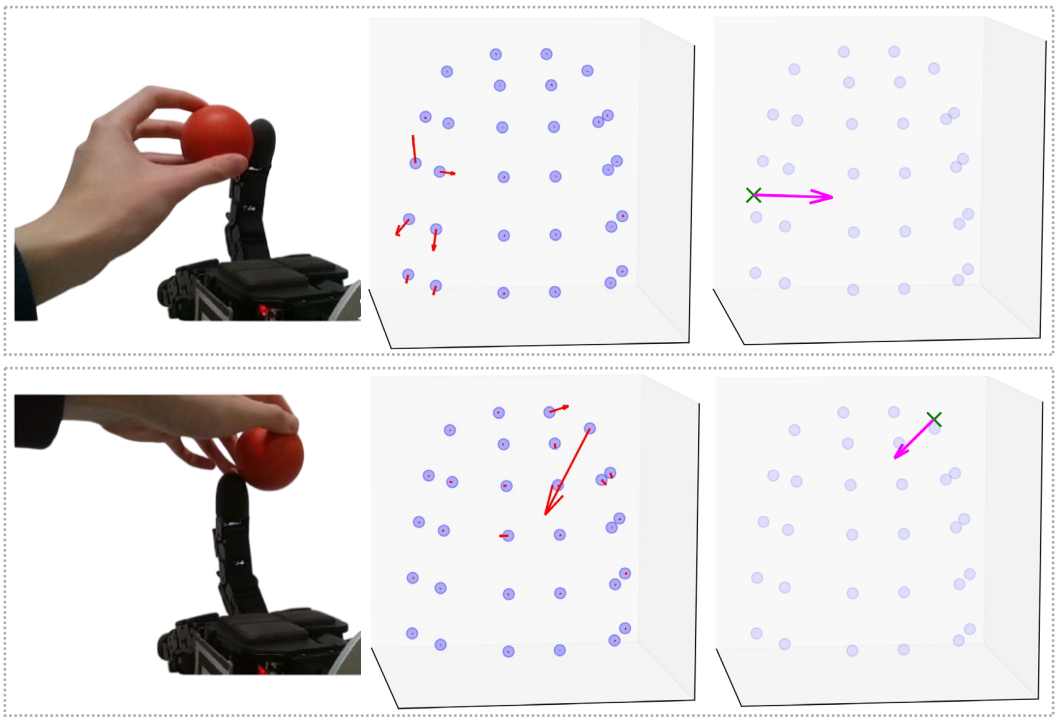


Figure 8: Two example visualizations of the computed CoP contact representations (green cross & pink arrow) from raw taxel forces (red arrows) on hardware. In each example, the middle subplot shows the raw taxel force vectors, and the right subplot shows the computed CoP force vector and contact position.

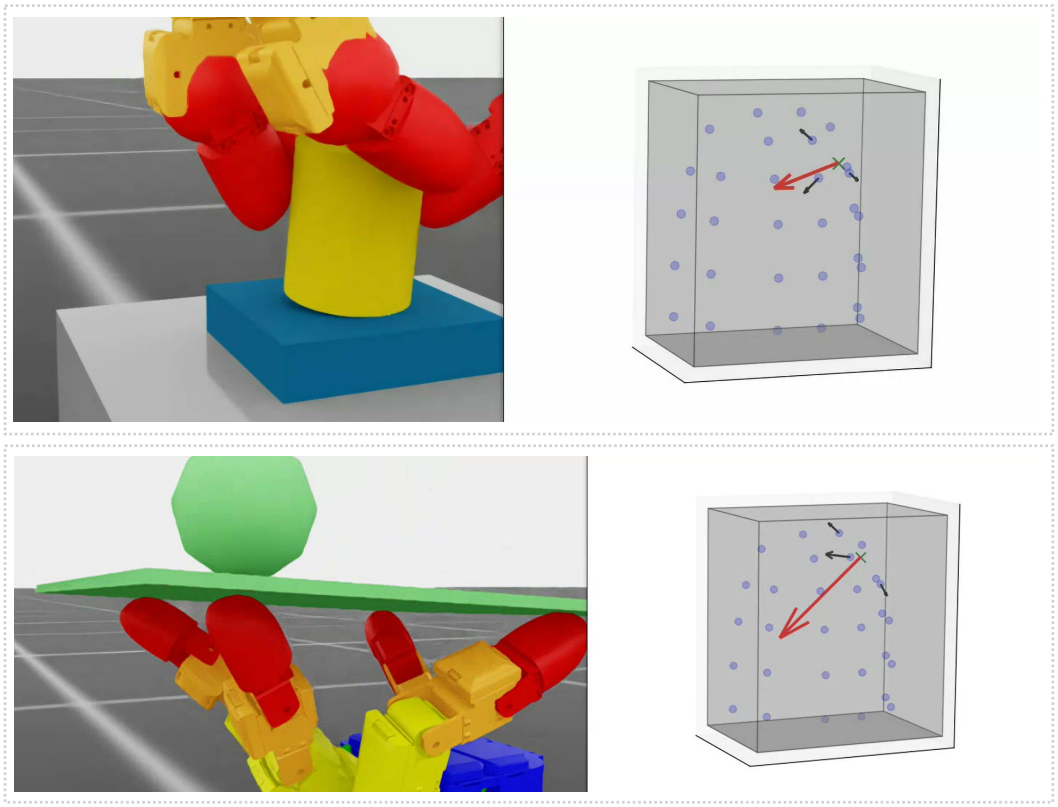


Figure 9: Visualizations of the computed taxel force vectors (black arrows) from contacts on the *middle fingertip* given in CoP representation (green cross & red arrow) from the simulator, for the insertion (left) and ball-balancing (right) tasks. Grey boxes represent the taxel-covered regions which are used to filter contacts during simulation.

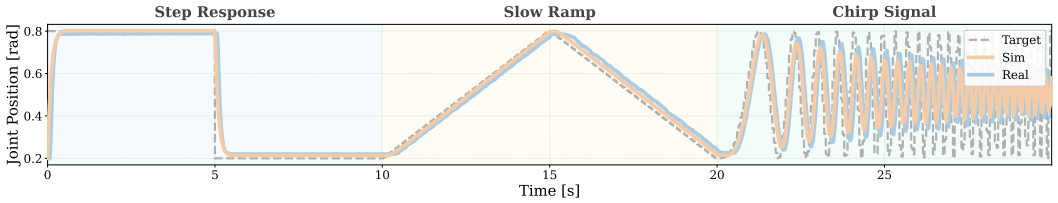
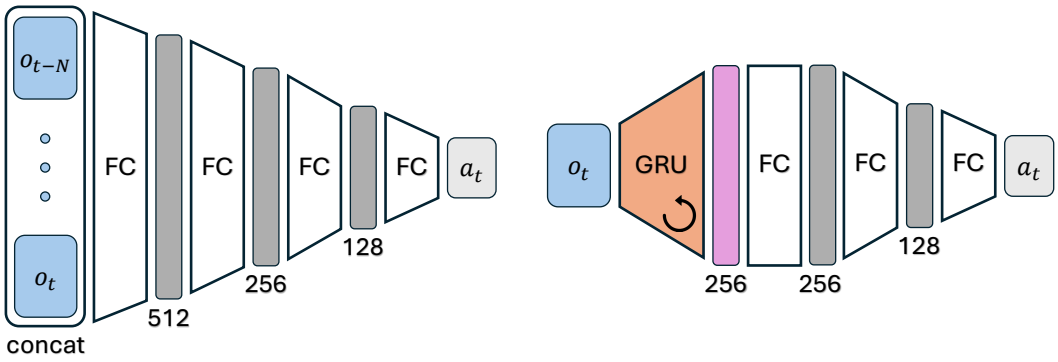


Figure 10: Actuator dynamics are aligned across simulation and the real world by executing a sequence of varying inputs to probe the system.

C System Identification of Actuator Dynamics

In IsaacSim, the key configurable actuator parameters include stiffness, damping, and joint friction. To find the set of simulation parameters that results in actuation behaviors closely-aligned to the real hardware, we designed a series of trajectory sequences to probe the dynamic responses of each actuator, including step inputs, slow linearly-varying ramp inputs, and chirp inputs with increasing frequency, as shown in Fig. 10. Then, we choose a fixed set of actuator parameters, apply onto the real robot, and execute the trajectory sequences and record the trajectory data consisting of joint angles at all time steps. Next, we begin an automatic tuning process in simulation via a Bayesian optimization loop. At each iteration, the current sampled joint parameters are applied to the simulated robot, the same trajectory sequences are executed, and the trajectory data is recorded. We then compute the MSE loss between the sim and real trajectories and use it as the minimization objective for the Bayesian optimizer to sample the parameters in the next iteration. Fig. 10 shows an example of trajectory alignment after optimizing for the actuator parameters of a given joint.



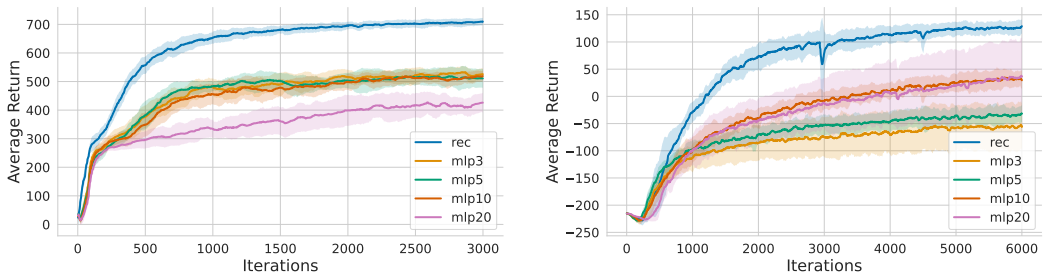
(a) The past N-step observations are concatenated and passed as input to the MLP policy network.

(b) The observation from only the current step is passed as input to the recurrent policy network.

Figure 11: Comparison of policy network architectures, including (a) direct MLP architecture consisting of only fully-connected (FC) layers and (b) recurrent architecture consisting of a recurrent (GRU) layer followed by FC layers.

D Policy Network Architecture Comparison

We analyze the performance of our recurrent policy architecture consisting of a single recurrent layer (GRU) which outputs a 256-dim latent that is passed to two linear layers with sizes $[256, 128]$ and ELU activation, and compare it against a direct MLP with three linear layers with sizes $[512, 256, 128]$ and ELU activation (see Fig. 11). Moreover, to examine whether providing longer temporal context via stacking more history observations to a MLP policy improves its performance, we train several policies conditioned on different numbers ($N = \{3, 5, 10, 20\}$) of history steps of concatenated observation vectors. The PPO hyperparameters and training configurations are kept identical for the recurrent and MLP policies to ensure fairness (see Appendix E.5). As shown in Fig. 12, the performance gains saturate for MLP policies conditioned on longer observation histories. For both tasks, we found that the recurrent policy results in better sample efficiency and convergence quality than the MLP policy. One possible underlying factor for this result is that for incorporating longer-horizon temporal information, the recurrent layer maintains the same observation space dimension, whereas the direct MLP network requires concatenating larger numbers of past observation vectors and thus drastically increases the observation space dimension. Another possible factor may be that the recurrent layer's update mechanism fuses the current observations with its internal hidden state, which may act as a "regularizer" and effectively smooths out noisy observations.



(a) Peg-in-hole insertion.

(b) Ball balancing.

Figure 12: Comparing the training performance of our recurrent policy network against MLP networks with different history lengths of stacked input observation vectors as input. Shaded area represents the standard deviation over 5 random seeds.

E RL CONFIGURATION & TRAINING DETAILS

E.1 ACTOR & CRITIC OBSERVATIONS

We employ PPO with asymmetric actor and critic observations for policy training. At each time step, the policy observation includes the current joint angles q_t, previous actions a_{t-1}, and the contact representation. For contact representations such as *cop* and *taxel* which are stored as higher-order tensors, we flatten them into a single vector and concatenate it to the proprioception vector. The critic observation includes privileged task-relevant information in addition to the actor observations, and are summarized in Table 4.

Table 4: Summary of privileged critic observations for both tasks.

Peg-in-Hole Insertion		Ball Balancing	
observation	quantity	observation	quantity
peg position	$p_{\text{peg}} \in \mathbb{R}^3$	plate position	$p_{\text{plate}} \in \mathbb{R}^3$
peg rotation	$R_{\text{peg}} \in \mathbb{R}^4$	plate rotation	$R_{\text{plate}} \in \mathbb{R}^4$
goal vector	$r_{\text{goal}} = p_{\text{goal}} - p_{\text{peg}} \in \mathbb{R}^3$	plate linear velocity	$v_{\text{plate}} \in \mathbb{R}^3$
goal reached	$\mathbb{1}(\lVert r_{\text{goal}} \rVert \leqslant \epsilon)$	plate angular velocity	$\omega_{\text{plate}} \in \mathbb{R}^3$
		ball position	$p_{\text{ball}} \in \mathbb{R}^3$
		ball linear velocity	$v_{\text{ball}} \in \mathbb{R}^3$
		goal vector	$r_{\text{goal}} = p_{\text{ball}} - p_{\text{plate}} \in \mathbb{R}^3$

E.2 POLICY ACTION SPACE

At each time step, the policy network outputs 16-dim actions $a \in \mathbb{R}^{16}$, which correspond to target joint position *increments* across all 16 DOFs. We first clip the actions to the range $[-1, 1]^{16}$, and apply an action scale of 0.03 for peg-in-hole insertion and 0.05 for ball balancing. Then, we apply an exponential moving average (EMA) with $\alpha = 0.5$ to the scaled target joint position increments, and add it to the previously commanded joint target positions to obtain the current joint targets which are then tracked by the PD controller. For the PD controller gains, we use $P = 3.0, D = 0.1$ for the insertion task for more compliant control, and $P = 6.0, D = 0.15$ for the balancing task for more reactive control.

E.3 REWARDS

The reward terms for each task consists of task progress-based reward terms and regularizing penalty terms, and are summarized in Table 5.

Table 5: Summary of reward terms for both tasks.

Reward Term	Expression	Weight
Peg-in-Hole Insertion		
goal distance reward	$\exp(-0.5(\lVert \mathrm{r_{goal}} \rVert / 0.015))$	1.0
goal reached reward (given once)	$\mathbb{1}(\lVert r_{\text{goal}} \rVert \leqslant \epsilon)$	400.0
good contact reward	$\sum_{i=1}^{N_{\text{sensors}}} \mathbb{1}(\lVert f_i \rVert \geqslant 1.0)$	0.25
rotation deviation (from z-axis) penalty	$1 - \exp(-0.5(\lVert \mathrm{R_{peg,z}} \rVert / 0.6)^2)$	-1.0
hand DOF pos deviation penalty	$1 - \exp(-0.5(\lVert \mathrm{q} - \mathrm{q_{default}} \rVert / 0.2)^2)$	-1.0
Ball Balancing		
goal distance reward	$\exp(-\lVert \mathrm{r_{goal}} \rVert / 0.05)$	1.0
plate contact reward	$\sum_{i=1}^{N_{\text{sensors}}} \mathbb{1}(\lVert f_i \rVert > 0)$	0.2
ball-plate relative linear velocity penalty	$\tanh(\lVert \mathrm{v_{ball}} - \mathrm{v_{plate}} \rVert / 0.3)$	-1.0
plate position deviation penalty	$\tanh(\lVert \mathrm{p_{plate}} - \mathrm{p_{plate,default}} \rVert / 0.15)$	-1.0
plate yaw penalty	$\tanh(\lVert \mathrm{R_{plate,z}} \rVert / 0.5)$	-1.0
ball fallen penalty	$\mathbb{1}(p_{\text{ball,z}} \leqslant 0.2)$	-200.0
action diff penalty	$\tanh(\lVert \mathrm{a_t} - \mathrm{a_{t-1}} \rVert^2 / 16.0)$	-1.0

E.4 Domain Randomization

We employ systematic domain randomization to object physical properties including friction and mass, object states during resets, and observation noise and delay to make the policy robust against noisy real-world variations. Domain randomization setups for both tasks are summarized in Table 6.

Table 6: Domain randomization setup including both task-specific and general configurations.

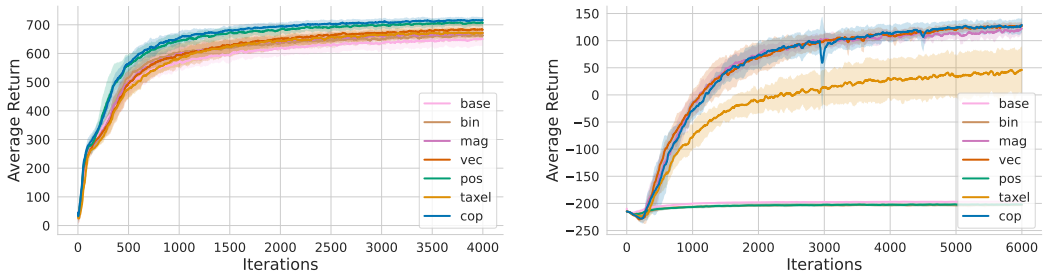
Peg-in-Hole Insertion	
Peg: Mass (kg)	$[0.03, 0.04]$
Peg: Friction (static & dynamic)	$[0.2, 0.4]; [0.1, 0.2]$
Peg: Initial Roll & Pitch (rad)	$+\mathcal{U}(-0.2, 0.2)$
Peg: Initial Yaw (rad)	$+\mathcal{U}(0, 2\pi)$
Hole: Friction (static & dynamic)	$[0.3, 0.5]; [0.1, 0.3]$
Hand: Friction (static & dynamic)	$[0.5, 0.7]; [0.3, 0.5]$
Hand: Initial Position (cm)	$+\mathcal{U}(-0.5, 0.5)$
Hand: Initial Roll & Pitch (rad)	$+\mathcal{U}(-0.02, 0.02)$
Hand: Initial DOF pos (rad)	$+\mathcal{U}(-0.05, 0.05)$
Ball Balancing	
Ball: Mass (kg)	$[0.05, 0.25]$
Ball: Friction (static & dynamic)	$[0.01, 0.02]; [0.0, 0.01]$
Ball: Initial Position (cm)	$+\mathcal{U}(-5.0, 5.0)$
Plate: Mass (kg)	$[0.035, 0.055]$
Plate: Friction (static & dynamic)	$[0.01, 0.02]; [0.0, 0.01]$
Plate: Initial Position (cm)	$+\mathcal{U}(-0.5, 0.5)$
Hand: Friction (static & dynamic)	$[1.9, 2.0]; [1.8, 1.9]$
Hand: Initial Roll & Pitch (rad)	$+\mathcal{U}(-0.02, 0.02)$
Hand: Initial DOF pos (rad)	$+\mathcal{U}(-0.05, 0.05)$
General	
PD Controller: P Gain	$\times\mathcal{U}(0.8, 1.2)$
PD Controller: D Gain	$\times\mathcal{U}(0.7, 1.3)$
Joint Pos Obs Noise (rad)	$+\mathcal{U}(-0.1, 0.1)$
Force Vector Obs Noise: Prob	0.2
Force Vector Obs Noise: Rotation (rad)	$+\mathcal{U}(-0.1, 0.1)$
Force Vector Obs Noise: Magnitude	$\times\mathcal{U}(0.9, 1.1)$
Contact Pos Obs Noise Prob	0.2
Contact Pos Obs Noise (cm)	$+\mathcal{U}(-0.1, 0.1)$
Joint Pos Obs Delay (s)	0.05
Contact Obs Delay (s)	$[0.05, 0.1]$

E.5 PPO Configuration & Training Results

We use the Proximal Policy Optimization (PPO) algorithm to learn RL policies. To ensure stable learning across varying feature scales, we apply online observation normalization to both the actor and critic inputs. The training uses an adaptive learning rate schedule starting at 5.0×10^{-4} with a target KL divergence of 0.016. During each iteration, we collect 64 steps for the insertion task and 16 steps for the balancing task per environment and perform 5 epochs of updates over 4 mini-batches. We use discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, clipping range of 0.2, and entropy coefficient of 0.005 to encourage exploration.

Fig. 13 shows the training results for both tasks. In the peg-in-hole insertion task, the *cop* policy on average attains the highest rewards during training across all other observations, though the difference is fairly marginal. For this task, the high exploration noise during training may have led to successful insertions through action stochasticity, thus artificially inflating the rewards of less robust

policies. In contrast, as shown in the real-world transfer results, during inference, the cop observation enables the policy to achieve more consistent and robust performance than other observations. For the ball balancing task, the policies conditioned on contact representation which lack explicit force information, including *base*, *bin*, and *pos*, completely fail to learn the task. Consistent with the real-world results, we observe that the *taxel* observation results in worse performance than more compact contact representations such as *mag* and *cop*, which is likely due to the high dimensionality and complexity of the raw taxel forces.



(a) Peg-in-hole insertion.

(b) Ball balancing.

Figure 13: Comparison of training performance across various contact representations for policy observation. Shaded area represents the standard deviation over 5 random seeds.

F Insertion Task Asset Details

Fig. 14 details the dimensions of the insertion peg assets across the six different shapes – circle, diamond, ellipse, hexagon, square, and triangle. For the corresponding hole assets, the sizes of the hole are increased by 10% in the x and y axes, resulting in 10% error tolerance for insertion. For the sizes of the assets used in this experiment, this corresponds to approximately 1.5 mm of radial tolerance. This was motivated by the observation that in simulation, using zero or lower tolerance induces "jamming" behaviors between the head of the insertion peg and the hole during the insertion process, likely due to mesh approximations and potential numerical errors in the physics engine's collision resolution.

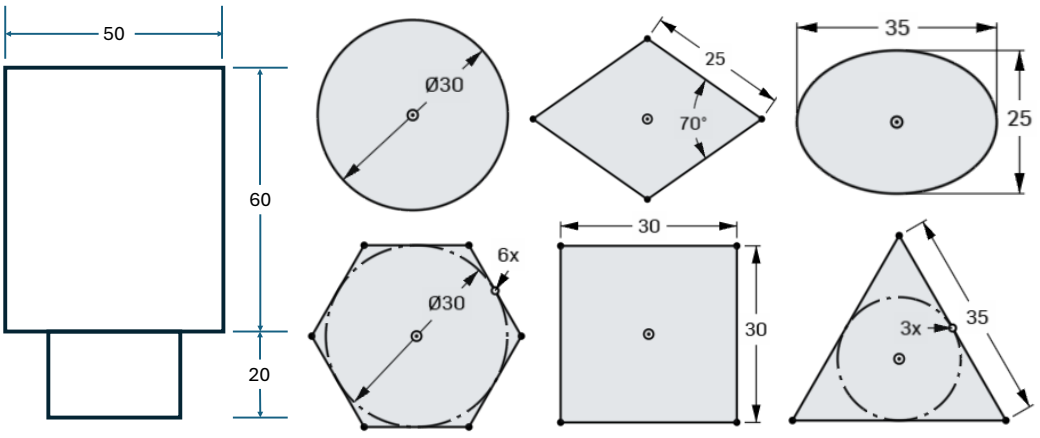


Figure 14: **Left**: The dimensions of the insertion peg which are shared across all six shapes. **Right**: The dimensions of each insertion shape. All lengths are given in millimeters (mm).