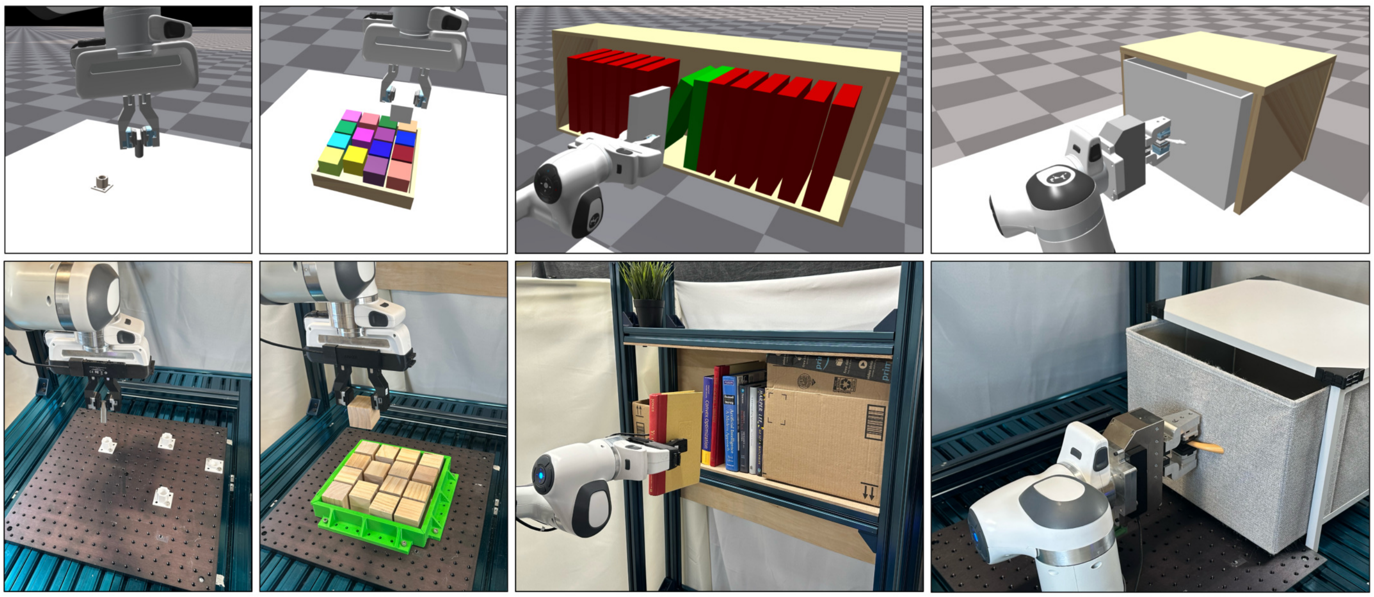
Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

# HydroShear: Hydroelastic Shear Simulation for Tactile Sim-to-Real Reinforcement Learning

**An Dang**[*†] **Jayjun Lee**[*†] **Mustafa Mukadam**[‡] **X. Alice Wu**[‡] **Bernadette Bucher**[†]
**Manikantan Nambi**[‡] **Nima Fazeli**[†‡]
[†]Robotics Department, University of Michigan [‡]Amazon IRG (Industrial Robotics Group)
[*]Equal Contribution

**hydroshear.github.io**



(a) Peg Insertion (b) Bin Packing (c) Book Shelving (d) Drawer Pulling

Fig. 1: **Illustration of our real-world manipulation tasks.** We present **HydroShear**, a non-holonomic hydroelastic model for high-fidelity tactile shear simulation, for training zero-shot sim-to-real tactile reinforcement learning policies. Our model enables a diverse range of contact and force-rich manipulation tasks that require reasoning over extrinsic contacts and slippages through accurate fingertip tactile shear simulation. Note how each task highlights a different challenge in tactile-based contact-rich manipulation. In (a), the robot must reason about in-hand pose uncertainty of the peg during tight insertion. (b) requires reasoning over simultaneous or sequential multi-object contacts through pushing to insert. (c) demands lateral insertion with gravity acting orthogonal to the insertion axis with full contact patches at the fingertips that make object features such as edges less informative for pose inference. (d) drawer opening tests precise and minimal gripper force modulation given a drawer under unknown force perturbations that induce slippage.

*Abstract*—**In this paper, we address the problem of tactile sim-to-real policy transfer for contact-rich tasks. Existing methods primarily focus on vision-based sensors and emphasize image rendering quality while providing overly simplistic models of force and shear. Consequently, these models exhibit a large sim-to-real gap for many dexterous tasks. Here, we present HydroShear, a non-holonomic hydroelastic tactile simulator that advances the state-of-the-art by modeling: a) stick-slip transitions, b) path-dependent force and shear build up, and c) full SE$(3)$ object-sensor interactions. HydroShear extends hydroelastic contact models using Signed Distance Functions (SDFs) to track the displacements of the on-surface points of an indenter during physical interaction with the sensor membrane. Our approach generates physics-based, computationally efficient force fields from arbitrary watertight geometries while remaining agnostic to the underlying physics engine. In experiments with GelSight Minis, HydroShear more faithfully reproduces real tactile shear compared to existing methods. This fidelity enables zero-shot sim-to-real transfer of reinforcement learning policies across four tasks: peg insertion, bin packing, book shelving for insertion, and drawer pulling for fine gripper control under slip. Our method achieves a 93% average success rate, outperforming policies trained on tactile images (34%) and alternative shear simulation methods (58%–61%).**

## I. Introduction

The sim-to-real paradigm shift that has transformed robot locomotion [1, 2, 3, 4] has yet to materialize for contact-rich manipulation. The primary bottleneck is tactile sensing: a modality that is critical to manipulation but difficult to simulate. In fact, the sim-to-real transfer of reinforcement learning (RL) policies trained with tactile data remains challenging due to discrepancies between simulated contact dynamics and the high-fidelity tactile feedback induced by real contact events. Towards addressing this challenge, a dichotomy has emerged in tactile simulation: visual fidelity has outpaced dynamic fidelity for vision-based tactile sensors.

Recent works have successfully simulated RGB images from vision-based sensors, enabling zero-shot policy deployment for some manipulation tasks [5, 6, 7, 8, 9]. However, these images primarily encode contact geometry and fall short of modeling tactile shear [10, 11], which captures the force interactions between the sensor and the object. Existing attempts to simulate shear often struggle to bridge the reality gap [6, 12, 13, 14]. While Finite Element Methods (FEM) are accurate, they are computationally too expensive for scalable Reinforcement Learning (RL) [15, 16]. Conversely, faster penalty-based approaches often approximate shear based merely on instantaneous penetration and velocity, failing to capture the *tactile shadowing* effects [17, 18], where deformation induced by one contact region masks or distorts shear information in neighboring regions, or the path-dependent deformation of elastomers of the tactile sensors [13, 6, 12]. Even recent marker-based methods such as FOTS are limited to tracking object motion in $\mathrm{SE}(2)$, neglecting full $\mathrm{SE}(3)$ complexity required for dexterous manipulation [14].

To address these limitations, we introduce HydroShear, a non-holonomic hydroelastic model for high-fidelity tactile simulation. HydroShear leverages Signed Distance Functions (SDFs) to represent complex geometries and implements a path-dependent contact model that tracks the object's motion history across the sensor membrane during physical interactions. This formulation allows us to simulate realistic shear force fields arising from full $\mathrm{SE}(3)$ motions, effectively capturing complex behaviors like stiction, slippage, and hysteresis. Our approach is computationally efficient via GPU-parallelization while remaining agnostic to the underlying physics engine. We evaluate our method by training RL policies entirely in simulation and comparing the success rate when zero-shot transferring these policies on dense insertion and fine gripper control tasks. We show that HydroShear's force-aware representation enables a 32% increase in task success compared to the best performing baseline.

## II. Related Work

**Tactile Simulation.** Tactile sensing requires simulation of physical contact interactions between the sensor and object. While tactile shear forces can be accurately simulated using FEM [15, 19, 16], it is computationally expensive, limiting scalability for training policies at scale. Recent work on FEM-based tactile simulation improves simulation speed via a superposition method to approximate FEM [5]. A new accelerated and parallelized physics simulator, DiffTaichi [20], also enabled further accelerated FEM-based tactile simulation [21].

Physical approximations of contact interactions have substantially higher computation speed than FEM-based approximations at the cost of fidelity, so this line of tactile simulation research focuses on preserving speed while increasing the accuracy of the simulation. Current simulations perform these physical approximations with rigid body contact models [22], soft contact, penalty-based models [6], and hydroelastic models [23, 24, 25]. However, these physical approximations struggle to bridge the sim-to-real gap because they do not model tactile shadowing effects such as sensor inaccuracies or signal degradation. Other physical approximations of contact used in sim-to-real transfer [13, 26] bypass this problem by normalizing the shear field at the cost of losing magnitudinal information. A recent simulator, FOTS [14], uses learning based methods to model tactile shadowing and demonstrates effective sim-to-real transfer. However, FOTS can only simulate $\mathrm{SE}(2)$ motion and cannot simulate shear induced by $\mathrm{SE}(3)$ motions such as rolling.

Our method, HydroShear, models physical interactions with a hydroelastic model. To match our simulation of tactile shear with real-world marker displacement fields, we track the full $\mathrm{SE}(3)$ motion of SDF-based complex geometries relative to the sensor elastomer, enabling zero-shot sim-to-real transfer. Our formulation is GPU-parallelizable, enabling efficient large-scale policy training in simulation. We also provide a GPU-parallelizable implementation of FOTS [14], whose original implementation was not GPU-parallelized, thereby enabling enables efficient large-scale policy training using FOTS.

**Tactile Sim-to-Real Transfer and RL.** Sim-to-real transfer of RL policies trained with tactile data remains challenging due to discrepancies between simulated contact dynamics and the high-fidelity tactile feedback induced by real contact events. For tasks involving contact solely between the hand and the object, discretizing or normalizing tactile signals can be sufficient to bridge the sim-to-real gap [12, 13]. However, for more general contact-rich manipulation tasks such coarse representations no longer suffice. Fine-grained tactile feedback is critical for executing tasks with hand-object-environment interactions with both intrinsic and extrinsic contacts such as peg insertion. Prior work demonstrates successful sim-to-real transfer of RL policies trained on more accurate simulated tactile data for peg insertion [6, 14]. A differential simulation built on top of TacSL's [6] contact force estimation model shows results on a broader set of tasks [21] to ensure generality of the sim-to-real results. We perform sim-to-real transfer of RL policies trained on simulated tactile data from TacSL [6], FOTS [14], and HydroShear (ours) on multiple tasks with hand-object-environment interactions (peg insertion, bin packing, book shelving, and drawer pulling). We find that HydroShear substantially outperforms these baselines [6, 14] in zero-shot sim-to-real transfer.

## III. Methodology

Our overarching goal is to train tactile-driven manipulation policies in simulation then sim-to-real transfer these policies as in Fig. 2. To achieve this, we present HydroShear, which

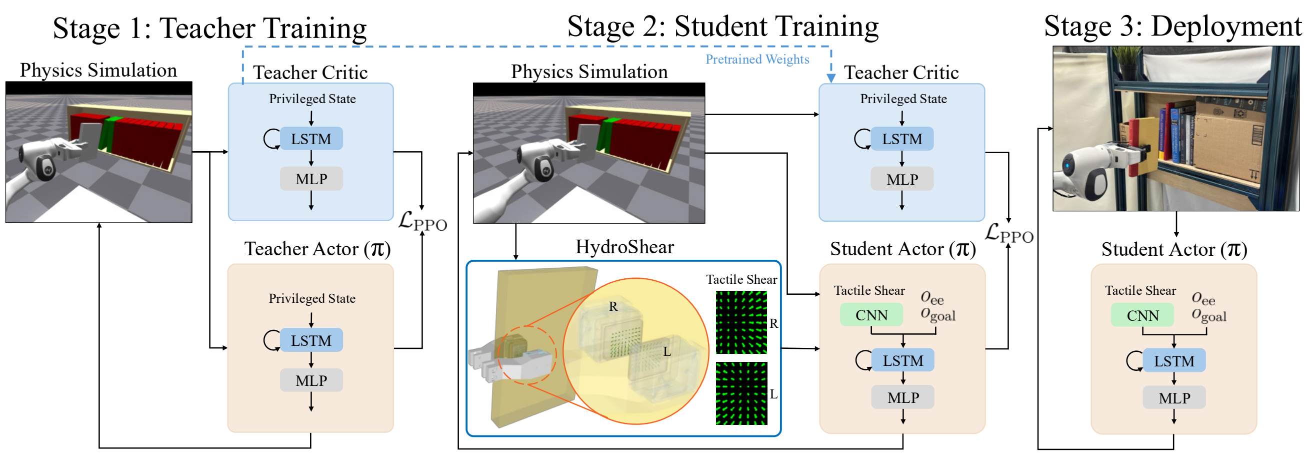



Fig. 2: **Illustration of the teacher-student RL training using Asymmetric Actor-Critic Distillation (AACD)** [6]. Stage 1: a teacher actor-critic is trained with access to privileged states such as contact forces and object poses. Stage 2: the critic is initialized with the expert critic pretrained from Stage 1 and the student actor that takes high-dimensional inputs (end-effector (EE) pose, relative EE-goal pose, left and right tactile shear) is trained from scratch. The actor-critic is optimized with the PPO RL objective and uses encoder-LSTM-MLP networks (see Appx. C). Stage 3: we deploy the student actor in the real world.

simulates tactile shear fields induced by the physical interaction between the soft *elastomer* of tactile sensors on the robot and *indenter* objects. This shear field serves as a proxy for the forces transmitted between the robot and the object.

Formally, we define a list of $N$ tactile grid query points (marker positions) on the elastomer membrane $\{\mathbf{p}_i\}_{i=1}^N$, where $\mathbf{p}_i \in \mathbb{R}^3$, and their associated 2D tactile shear vectors $\{\mathbf{s}_i\}_{i=1}^N$, where $\mathbf{s}_i = (d_x, d_y) \in \mathbb{R}^2$. While the sensor membrane will undergo a 3D deformation when it is interacted with, our goal is to derive the 2D projection $\mathbf{M}$ of this deformation onto the sensor image plane. $\mathbf{M}$ maps the query points and a sequence of indenter poses in the elastomer frame $\{{}^E\mathbf{X}_t^I\}_{t=0}^T \in \mathrm{SE}(3)$ to the marker displacement field:

$$\mathbf{M} : \left((x, y), \{{}^E\mathbf{X}_t^I\}_{t=0}^T\right) \rightarrow \mathbf{s}_i \tag{1}$$

where $(x, y) \in \mathbb{R}^2$ represent the coordinates of the tactile query point $\mathbf{p}_i$ on the 2D sensor plane. More concretely, for a vision-based tactile sensor, each image is a collection of pixels that are samples of $(x, y)$ with the corresponding shear $(d_x, d_y)$. $\mathbf{M}$ can be decomposed into dilation and shear fields:

$$\mathbf{M}_t((x, y), {}^E\mathbf{X}_{0:t}^I) = \mathbf{M}_t^d((x, y)) + \mathbf{M}_t^s((x, y), {}^E\mathbf{X}_{0:t}^I) \tag{2}$$

where $\mathbf{M}_t$ is the total marker displacement field at time $t$, $\mathbf{M}_t^d$ is the dilation field, and $\mathbf{M}_t^s$ is the shear field. Dilation is due to penetration along sensor normal. Shear is due to object's tangential translation or $\mathrm{SO}(3)$ rotation while in contact and depends on the indenter pose ${}^E\mathbf{X}_{0:t}^I$ history.

Next, we detail the computation of dilation and shear vector fields (Fig. 3), followed by the system identification procedure used to calibrate the model for a tactile sensor. Finally, we outline the integration of our model into a RL framework to train and sim-to-real transfer tactile policies.

### A. Dilation

We model the dilation vector field by aggregating the influence of all tactile grid query points $\{\mathbf{p}_i\}_{i=1}^N$ from Fig. 3(b) that are in contact, as shown in Fig. 3(c). We define a tactile grid point $\mathbf{p}_i = \begin{bmatrix}\mathbf{p}_i^{xy} & p_i^z\end{bmatrix}^T$ to be in-contact if $\phi_I(\mathbf{p}_i) < 0$ where $\phi_I : (x, y, z) \in \mathbb{R}^3 \rightarrow \mathbb{R}$ is the signed distance function of the indenter $I$ as shown in Fig. 3(c). Next, we denote the sequence of tactile grid point indices in contact as $C = \{c_i \in \mathbb{N} \mid \phi_I(\mathbf{p}_{c_i}) < 0\}$. With this, we can model the dilation vector field as also seen in FOTS [14]:

$$\mathbf{M}_t^d = \sum_{i=1}^{|C|} -\phi_I(\mathbf{p}_{c_i}) \cdot \mathbf{v}_{c_i} \cdot \exp\{-\lambda_d \|\mathbf{v}_{c_i}\|_2^2\} \tag{3}$$

$$\mathbf{v}_{c_i} = \begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{p}_{c_i}^{xy} \tag{4}$$

where $\exp\{-\lambda_d \|\mathbf{v}_{c_i}\|_2^2\}$ dissipates the effect of the contact point on the queried tactile 2D grid coordinate depending on how far away the point is to model the tactile shadowing effects in the real-world caused by elastomer deformations.

### B. Shear

We introduce a hydroelastic model that extends the hydrostatic pressure field contact model [23, 25] to track forceful interactions between the indenter and the sensor's compliant elastomer. Specifically, we track contact forces on a set of indenter on-surface points $\{\mathbf{o}_{j,t}\}_{j=1}^M$ in the elastomer frame where $M$ is the total number of points in Fig. 3(a). The coordinates of $\mathbf{o}_{j,t}$ are calculated by transforming the local indenter surface points in indenter frame $\bar{\mathbf{o}}_j$ with the indenter pose ${}^E\mathbf{X}_t^I$ via $\mathbf{o}_{j,t} = {}^E\mathbf{X}_t^I \bar{\mathbf{o}}_j$. To account for the viscoelasticity of the elastomer membrane, we recursively track the contact forces $\{\tilde{\mathbf{f}}_{j,t}\}_{j=1}^M$ by taking the indenter object pose path in elastomer frame $({}^E\mathbf{X}_t^I, {}^E\mathbf{X}_{t-1}^I)$ and converting the displacement of the in-contact indenter surface points into forces. We compute this using the viscoelastic stiffness parameters of the elastomer $(E, K, A_j)$ where $E$ is the stiffness along the indenter's normal direction, $K$ is the stiffness tangential to the sensor normal, and $A_j$ is the surface area associated to the indenter

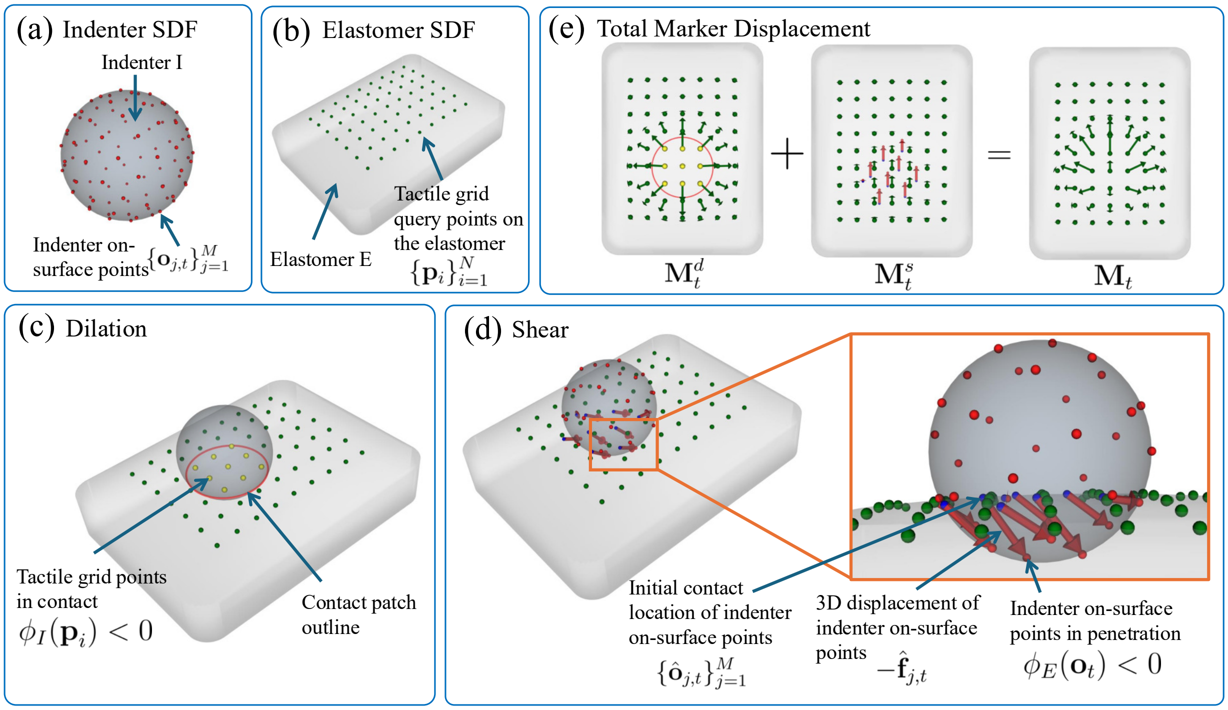



Fig. 3: **Illustration of HydroShear and its pipeline for tactile shear simulation.** HydroShear simulates the tactile shear feedback that arises from the physical interaction between the indenter $I$ in (a) and the sensor elastomer $E$ in (b). The goal is to compute the marker displacement fields across the tactile grid query points on the elastomer $\{\mathbf{p}_i\}_{i=1}^{N}$ from (b). HydroShear computes the dilation displacement field $\mathbf{M}_t^d$ in (c) and the shear displacement field $\mathbf{M}_t^s$ in (d) to get the total marker displacement field $\mathbf{M}_t$ in (e). The dilation field is computed by identifying the tactile grid points that are in contact with the indenter SDF ($\phi_I(\mathbf{p}_i) < 0$). Here, the red circle represents the outline of the contact patch. For the shear field, we take the history of indenter poses in the elastomer frame ${}^{E}\mathbf{X}_{0:t}^{I}$ to track the 3D displacement $-\hat{\mathbf{f}}_{j,t}$ of the indenter on-surface points $\{\mathbf{o}_{j,t}\}_{j=1}^{M}$, represented by the red arrows in (d), which connects the initial contact location of indenter on-surface points ($\{\hat{\mathbf{o}}_{j,t}\}_{j=1}^{M}$) to the current position of the on-surface indenter points while in-penetration to the elastomer SDF ($\phi_E(\mathbf{o}_t) < 0$).

surface points. We model our recursive formulation as:

$$\tilde{\mathbf{f}}_{j,t} = F(\tilde{\mathbf{f}}_{j,t-1}, {}^{E}\mathbf{X}_t^I, {}^{E}\mathbf{X}_{t-1}^I, \bar{\mathbf{o}}_j; E, K, A_j, \mu) \quad (5)$$

where $\mu$ denotes the friction coefficient for the forceful interaction between the indenter and elastomer. For our recursive formulation, we assume in the base case $t = 0$ that the indenter and elastomer are not in contact, thus $\tilde{\mathbf{f}}_{j,0} = 0$ for all $j$.

To calculate $\tilde{\mathbf{f}}_{j,t}$ through the force tracking function $F$, our hydroelastic force tracker first takes the displacement of the indenter surface points at time $t$ and $t-1$ and only tracks the portion of this displacement $\mathbf{o}_{j,t} - \mathbf{o}_{j,t-1}$ that is in contact with the elastomer. We approximately determine this portion using the elastomer signed distance function $\phi_E : (x, y, z) \in \mathbb{R}^3 \rightarrow \mathbb{R}$ by calculating a fraction $\alpha_{j,t}$ of the SDF difference $\phi_E(\mathbf{o}_{j,t}) - \phi_E(\mathbf{o}_{j,t-1})$ that is in-contact:

$$\alpha_{j,t} = \frac{\mathrm{ReLU}(-\phi_E(\mathbf{o}_{j,t})) - \mathrm{ReLU}(-\phi_E(\mathbf{o}_{j,t-1}))}{\phi_E(\mathbf{o}_{j,t}) - \phi_E(\mathbf{o}_{j,t-1})} \quad (6)$$

$$\mathbf{d}_{j,t} = \alpha_{j,t}^{d}(\mathbf{o}_{j,t} - \mathbf{o}_{j,t-1}) \quad (7)$$

where $\mathbf{d}_{j,t}$ is the indenter surface point displacement scaled by how much of the displacement is in-contact. We introduce the ReLU into Eq. 6 to compactly represent 3 cases that can occur when tracking indenter surface point displacements. When $\phi_E(\mathbf{o}_{j,t})$ and $\phi_E(\mathbf{o}_{j,t-1})$ are both negative, the indenter surface point displacement is fully in contact with the elastomer and $\alpha_{j,t} = 1$. When both are positive, indenter surface point displacements are fully out of contact with the elastomer and $\alpha_{j,t} = 0$. When they have opposite signs, we will get an $\alpha_{j,t}$ such that $0 < \alpha_{j,t} < 1$ because only a part of the indenter surface point displacement will be inside the elastomer.

Using the viscoelastic properties $(E, K)$ of the elastomer, we convert the tracked in-contact indenter surface point displacements $\mathbf{d}_{j,t}$ and convert them into forces $\mathbf{f}_{j,t}$. We decompose $\mathbf{d}_{j,t}$ into a normal component $d_{j,t}^n$ and tangential component $\mathbf{d}_{j,t}^{xy}$. We compute $d_{j,t}^n$ by projecting $\mathbf{d}_{j,t}$ onto the indenter object normal $\hat{\mathbf{n}}_j$ at the indenter surface point $\mathbf{o}_{j,t}$ with $d_{j,t}^n = \langle \mathbf{d}_{j,t}, \hat{\mathbf{n}}_j \rangle$. By subtracting the normal component $d_{j,t}^n \hat{\mathbf{n}}_j$ from the original in-contact displacement $\mathbf{d}_{j,t}$, we recover the tangential component $\mathbf{d}_{j,t}^{xy} = \mathbf{d}_{j,t} - d_{j,t}^n \hat{\mathbf{n}}_j$. With this in mind, we linearly scale the displacements into forces:

$$f_{j,t}^n = \tilde{f}_{j,t-1}^n + E A_j d_{j,t}^n \quad (8)$$

$$\mathbf{f}_{j,t}^{xy} = \tilde{\mathbf{f}}_{j,t-1}^{xy} + K A_j \mathbf{d}_{j,t}^{xy} \quad (9)$$

Next, we impose that normal forces be lower bounded by zero (no adhesion) and enforce Coulomb friction by clipping the magnitude of the tracked tangential friction force $\mathbf{f}_{j,t}^{xy}$ by the normal force scaled by the coefficient of friction:

$$\bar{f}^n_{j,t} = \text{ReLU}(f^n_{j,t}) \tag{10}$$

$$\bar{\mathbf{f}}^{xy}_{j,t} = \min\{1, \frac{\mu \bar{f}^n_{j,t}}{\|\mathbf{f}^{xy}_{j,t}\|_2}\}\mathbf{f}^{xy}_{j,t} \tag{11}$$

where $\mu$ is the friction coefficient between the elastomer and indenter. $\mu$ determines how much tangential force $\mathbf{f}^{xy}_{j,t}$ is required before sliding occurs. Finally, the contact forces are reset if the indenter surface point breaks contact with the elastomer:

$$\tilde{\mathbf{f}}_{j,t} = H(-\phi_E(\mathbf{o}_{j,t})) \cdot (\bar{f}^n_{j,t}\hat{\mathbf{n}}_j + \bar{\mathbf{f}}^{xy}_{j,t}) \tag{12}$$

where $H$ is the Heaviside function.

As part of the shear vector field computation, we take the contact forces $\tilde{\mathbf{f}}_{j,t}$ on the indenter surface and project it onto the elastomer surface. Specifically, for every indenter surface point $\mathbf{o}_{j,t}$ that is in contact with the elastomer, we want to find the corresponding point on the elastomer surface $\hat{\mathbf{o}}_{j,t}$ that the indenter point is in contact with. This way, each contact force $\tilde{\mathbf{f}}_{j,t}$ has a corresponding point on the elastomer surface that it is being applied on. When the interaction between the indenter and elastomer is sticking, $\hat{\mathbf{o}}_{j,t}$ does not change. However, when the interaction is slipping, $\hat{\mathbf{o}}_{j,t}$ shifts to represent the change in the point of contact on the elastomer surface.

In order to model these effects, we formulate projection as finding a set of displacements $\{\hat{\mathbf{f}}_{j,t}\}^M_{j=1}$ that are applied onto the indenter surface points $\{\mathbf{o}_j\}^M_{j=1}$ to acquire $\{\hat{\mathbf{o}}_{j,t}\}^M_{j=1}$:

$$\hat{\mathbf{o}}_{j,t} = \mathbf{o}_{j,t} + \hat{\mathbf{f}}_{j,t} \tag{13}$$

We illustrate the displacements $-\hat{\mathbf{f}}_{j,t}$ by the red arrows on $\mathbf{M}^s_t$ in Fig. 3(e). To compute $\hat{\mathbf{f}}_{j,t}$ such that it provides slippage and provides tracked indenter surface displacement while in contact, we formulate an indenter displacement tracker based on the Eqn. 5 but the elastomer stiffness parameters and the surface area parameter $A_j$ for this tracker are set to 1:

$$\hat{\mathbf{f}}_{j,t} = F(\hat{\mathbf{f}}_{j,t-1}, {}^E\mathbf{X}^I_t, {}^E\mathbf{X}^I_{t-1}, \bar{\mathbf{o}}_j; 1, 1, 1, \hat{\mu}) \tag{14}$$

where $\hat{\mu}$ provides the slippage effect for the projected indenter points on the elastomer surface $\hat{\mathbf{o}}_{j,t}$.

To calculate the shear vector field, we would need both the hydroelastic contact force tracker and the projection displacement tracker. However, we show that only one tracker is needed and the other can be recovered given a few assumptions. If we set $E = K$, $\mu = \hat{\mu}$, and $A_j = A$ for the hydroelastic contact force tracker where $A$ is a constant that assumes the surface area associated with each indenter surface point is uniform, then we can recover the contact forces from just the displacements by scaling the $\hat{\mathbf{f}}_{j,t}$ by KA as follows:

$$\tilde{\mathbf{f}}_{j,t} = KA \cdot \hat{\mathbf{f}}_{j,t} \tag{15}$$

After computing $\hat{\mathbf{f}}_{j,t}$ and $\tilde{\mathbf{f}}_{j,t}$, we calculate the shear vector field using the aggregated effects of each contact force $\tilde{\mathbf{f}}_{j,t}$ at the projected indenter surface point $\hat{\mathbf{o}}_{j,t} = \begin{bmatrix}\hat{\mathbf{o}}^{xy}_{j,t} & \hat{o}^z_{j,t}\end{bmatrix}^T$ on each of tactile 2D grid query coordinate. We denote the set of indenter on-surface points that are in contact with the elastomer as $K_t = \{k_i \mid \phi_E(\mathbf{o}_{k_i,t}) < 0\}$ in Fig. 3(d). Now, we calculate the shear vector field as follows:

$$\mathbf{M}^s_t = \sum_{i=1}^{|K_t|} -\phi_E(\mathbf{o}_{k_i,t}) \cdot -\tilde{\mathbf{f}}^{xy}_{k_i,t} \cdot \exp\{-\lambda_s \|\mathbf{v}_{k_i}\|^2_2\} \tag{16}$$

$$\mathbf{v}_{k_i} = \begin{bmatrix} x \\ y \end{bmatrix} - \hat{\mathbf{o}}^{xy}_{k_i,t} \tag{17}$$

where we weigh the effect each indenter surface point $\mathbf{o}_{j,t}$ has on the vector field by its penetration depth $-\phi_E(\mathbf{o}_{k_i,t})$ and dissipate the effects of the surface point on a queried tactile 2D grid coordinate $(x, y)$ based on distance using $\exp\{-\lambda_s\|\mathbf{v}_{k_i}\|^2_2\}$. In total, our method takes in 4 parameters $(\lambda_d, \lambda_s, K, \mu)$. Next, we present how to perform system identification to simulate realistic tactile shear fields.

### C. Calibration

We calibrate our model parameters one at a time by isolating the effects of each parameter in the real-world data. Our model contains 4 parameters in total $(\lambda_d, \lambda_s, K, \mu)$. These parameters are calibrated individually in the order they are presented. First we calibrate $\lambda_d$ by collecting tactile shear data $\mathbf{Y}^d_i$ in the real world where the indenter presses down on various positions on the elastomer and formulate the following least-squares optimization problem:

$$\lambda^\star_d = \arg\min_{\lambda_d} \sum_{i=1}^{n} \|\mathbf{Y}^d_i - \mathbf{M}^d_t((x_i, y_i); \lambda_d)\|^2_2$$

Next we calibrate $\lambda_s$ using tactile shear $\mathbf{Y}^s_i$, but we scale $\mathbf{Y}^s_i$ to be in the same range of vector magnitudes as $\mathbf{M}_t(x, y; \lambda^\star_d, \lambda_s, K, \mu)$. We collect $\mathbf{Y}^s_i$ by dilating the elastomer with the indenter and then translating while the indenter is in contact. This requires us to use the full model and previously calibrated parameters to estimate the vector field as follows:

$$\lambda^\star_s = \arg\min_{\lambda_s} \sum_{i=1}^{n} \|\hat{\mathbf{Y}}^s_i - \mathbf{M}_t((x_i, y_i); \lambda^\star_d, \lambda_s, 1, 1e5)\|^2_2$$

where $\hat{\mathbf{Y}}^s_i$ is the rescaled real shear vector field. Afterwards, we calibrate $K$ but without the rescaling trick:

$$K^\star = \arg\min_{K} \sum_{i=1}^{n} \|\mathbf{Y}^s_i - \mathbf{M}_t((x_i, y_i); \lambda^\star_d, \lambda^\star_s, K, 1e5)\|^2_2$$

Finally, we calibrate $\mu$ by collecting data of when the indenter is slipping on the elastomer and solve the following problem:

$$\mu^\star = \arg\min_{\mu} \sum_{i=1}^{n} \|\mathbf{Y}^\mu_i - \mathbf{M}_t((x_i, y_i); \lambda^\star_d, \lambda^\star_s, K^\star, \mu)\|^2_2$$

In summary, we calibrate the dilation dissipation parameter $\lambda_d$, tangential shear dissipation parameter $\lambda_s$, tangential stiffness parameter $K$, and friction coefficient $\mu$. We solve individual single-variable optimization problems to improve the conditioning of the calibration procedure and avoid directly solving the highly nonlinear joint optimization, for which reliable convergence to good optima is difficult to guarantee.

### D. Sim-to-Real Reinforcement Learning

We formulate our contact-rich manipulation tasks with tactile feedback as infinite-horizon discrete-time Partially-Observable Markov Decision Processes defined by the tuple $\mathcal{M} := (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho_0, \gamma)$ with state space $\mathcal{S}$, the observation space $\mathcal{O}$, action space $\mathcal{A}$, reward $\mathcal{R}(s, a)$, transition probability $\mathcal{P}(s'|s, a)$, initial state distribution $\rho_0$, and discounting factor $\gamma \in [0, 1)$. The robot seeks to learn a parameterized policy $\pi_\theta(\cdot|s)$ to maximize the expected discounted return $\mathcal{J} := \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s\right]$. The only distinction between simulation and real is access to the state space.

**Teacher-Student Distillation:** We train HydroShear RL policies entirely in Isaac Gym [27] simulation with parallelized environments using PPO [28] and zero-shot deploy in the real-world without any modification or fine-tuning. We list the task reward functions in Appx. F. As outlined in Fig. 2, we first train a teacher actor-critic using privileged state information. We find that naively training RL agents with contact penalties results in suboptimal policy that avoids contact with the goal, thus preventing successful task completion. To prevent such collapse, we design a teacher training curriculum where we first start training with no contact penalty until initial task success. Then, we finetune with the contact penalty term in the next stage. Next, we follow asymmetric actor-critic distillation (AACD) [6] where we use the privileged teacher critic to train a student policy actor from scratch. Note that all students for each task are trained with the same teacher critic checkpoint trained on the contact penalty curriculum. For a more detailed descriptions of the training pipeline, refer to Appx. D. Our student policies observe their proprioceptive states, goal pose, and tactile feedback with no access to privileged information such as in-hand object poses or contact forces.

## IV. Experiments and Results

We evaluate HydroShear along two axes: (1) its ability to model diverse modes of real-world shear induced by full SE(3) indenter motion with respect to the elastomer, and (2) its ability to zero-shot sim-to-real transfer RL policies. To this end, we first describe the real-world calibration setup used to collect a dataset of tactile shear field for system identification. We then assess HydroShear's shear simulation accuracy using a separate validation dataset with unseen contact configurations in a digital twin environment, comparing against TacSL [6] and FOTS [14]. Next, we test our sim-to-real policies across four contact rich tasks, consisting of three dense insertion tasks: (1) Peg Insertion, (2) Bin Packing, (3) Book Shelving, and (4) Drawer Pulling that specifically requires precise gripper control and attention to slippage. Additionally, we conduct a speed comparison experiment between FOTS and HydroShear in Tab. XI

### A. Calibration and Real-world Shear Evaluation

We calibrate our method's parameters (Sec. III-C) by collecting controlled marker displacement data designed to isolate the contribution of each parameter $(\lambda_d, \lambda_s, K, \mu)$. Fig. 4 shows our experimental setup, were we use a 7-DoF KUKA MED LBR robot equipped with a 35-mm spherical indenter mounted at the tool center point (TCP) to calibrate the sensor. Here, we use the GelSight Mini sensor that we rigidly mount to the table. During calibration, we assume access to the full poses of both the tactile sensor and the sphere indenter.

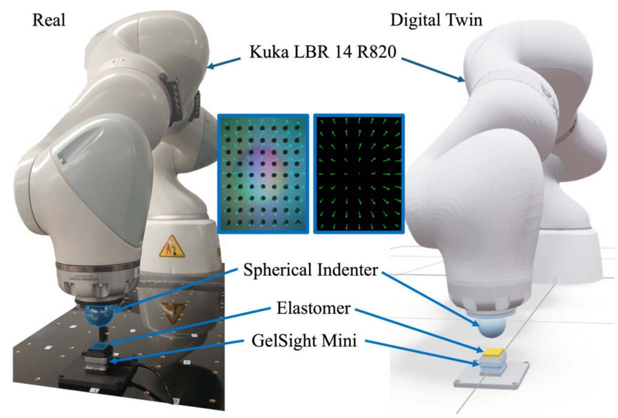



Fig. 4: **Digital twin calibration setup:** In the real-world, a KUKA robot arm equipped with a sphere indenter to dilate and shear the elastomer of the GelSight Mini vision-based tactile sensor mounted on the table. We replicate the same motion in a digital twin of the real-world setup to run baseline and our algorithms to calibrate the tactile simulation model.

To calibrate $\lambda_d$, we collect 10 samples in which the sphere indenter presses vertically into the elastomer at randomly sampled surface locations, and select the value of $\lambda_d$ that best matches the resulting tactile shear response. To calibrate $\lambda_s$ and $K$, we collect 10 additional samples in which the indenter applies combined normal and tangential motions, shearing the elastomer at random locations and directions, and jointly optimize $\lambda_s$ and $K$ to best fit the observed tactile shear. Each sample contains the actual tactile shear field measured as a marker displacement field and the indenter trajectory that induced the shear field. Finally, to estimate the friction coefficient $\mu$, we collect 10 samples of controlled slip motions while maintaining contact with the elastomer and select $\mu$ to best match the resulting output. To convert the predicted shear displacement from meters to pixels, we use the GelSight Mini camera resolution, yielding a scale factor of $1000/0.065$ [px/m].

**Baselines:** We evaluate the performance of our proposed model by comparing it against prior methods that simulate tactile shear and support sim-to-real policy transfer. Specifically, we benchmark against two representative baselines: TacSL [6] and FOTS [14]. TacSL implements a penalty-based force field [13] in Isaac Gym, leveraging the object signed distance field (SDF) and the object's relative velocity with respect to the elastomer to estimate contact forces at tactile grid query points on the elastomer $\{\mathbf{p}_i\}_{i=1}^N$ from Fig. 3(b). FOTS [14] decomposes the marker displacement field into three components that track object motion in SE(2) (see Appx. A). For a fair comparison, we follow each method's original calibration procedure and use the same real-world calibration dataset to estimate their model parameters.

| Model | Dilation | | Shear | | Twist | | Roll | |
|---|---|---|---|---|---|---|---|---|
| | Shear Type | | | | | | | |
| | RMSE ↓ | CS ↑ | RMSE | CS | RMSE | CS | RMSE | CS |
| TacSL | 2.187 | 0.323 | 4.262 | 0.126 | 4.022 | 0.157 | 3.861 | 0.137 |
| FOTS | 1.333 | **0.976** | 2.146 | 0.930 | 3.596 | 0.683 | 2.841 | 0.561 |
| Ours | **1.000** | **0.976** | **1.719** | **0.951** | **2.021** | **0.974** | **1.576** | **0.904** |

TABLE I: **Calibration Results.** Per-taxel shear RMSE [px] and cosine similarity (denoted as CS) between the predicted and ground-truth shear vectors are reported for each shear type.

**Results:** We evaluate our method against the baselines by comparing the simulated tactile shear field from the digital twin with real-world measurements of marker displacements with Gelsight Minis. We collect 10 samples each for dilation, translational shear, torsional shear (twist), and rolling shear, resulting in a total of 40 test samples. Tab. I reports the per-taxel Root Mean Squared Error (RMSE) and mean cosine similarity (CS) between the predicted and ground-truth shear vector fields, averaged over all taxels, with qualitative samples shown in Fig. 5. Our method achieves consistently lower error and higher directional alignment across all shear types, indicating a more faithful representation of elastomer shear behavior. TacSL exhibits increased error due to the absence of tactile shadowing in its formulation, which limits its ability to accurately capture shear patterns. FOTS explicitly accounts for tactile shadowing and attains comparable performance for in-plane object motions. However, its accuracy degrades under tilt motions, which involve out-of-plane components of $\mathrm{SE}(3)$ that are not explicitly modeled in the FOTS formulation. More broadly, FOTS represents contact through relative object motion expressed in an object-centric frame rather than an SDF-based geometric representation. In contrast, our method leverages object SDFs and on-surface contact points to resolve higher-resolution contact motion and interaction paths, enabling more precise tactile shear simulation, particularly for complex and out-of-plane interactions. Finally, Fig. 6 illustrates HydroShear's ability to simulate complex object geometries involving multiple simultaneous contact patches.

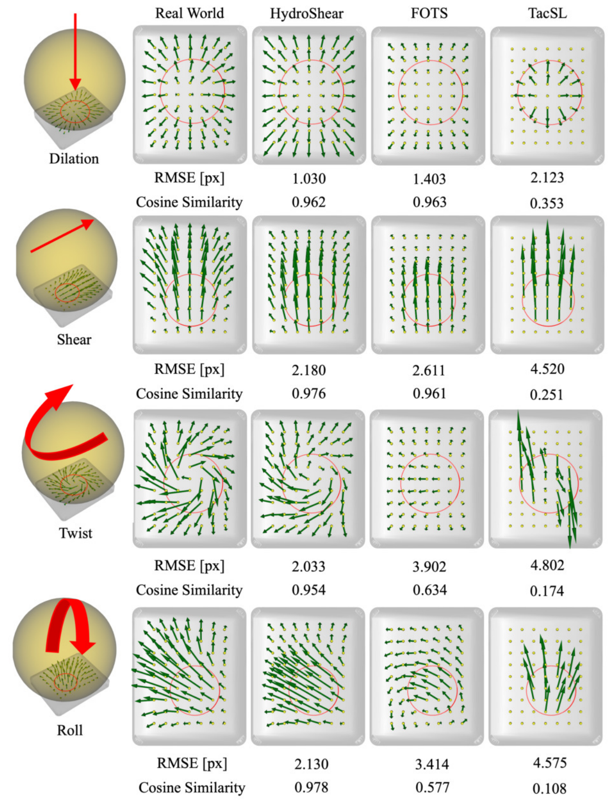



Fig. 5: **Samples from calibration.** We showcase samples from representative motions such as dilation, translational shear, rotational shear (twist) and from rolling motion of the sphere. The red circle indicates the contact patch on the soft elastomer from the rigid sphere. The green arrows represent the shear vector field that are not to real-world scale but consistent across samples for visualization purposes. We list the RMSE and cosine similarity between the real-world ground truth sample and the simulated ones respectively as reference values to Tab. I.

### B. Zero-shot Sim-to-Real RL

**Real-world Setup:** We conduct our sim-to-real experiments using a 7-DoF Franka Emika Panda robot equipped with a pair of GelSight Mini vision-based tactile sensors mounted. We mount the sensors on the default Franka hand gripper for Peg Insertion, Bin Packing, and Book Shelving tasks and on a Weiss WSG-50 gripper for Drawer Pulling task for more precise gripper control (Fig. 1). We compute the real-world shear from GelSight Minis using marker displacements.

The four real-world tasks in Fig. 1 are designed to highlight distinct use cases of high-fidelity tactile shear simulation for sim-to-real policy learning without any visual feedback.

**1. Peg Insertion:** This task is designed to introduce in-hand object pose uncertainty during force-rich insertion. The task goal is to insert a cylindrical peg, grasped at varying in-hand poses, into a socket mounted on the tabletop as shown in Fig. 1(a). For the real-world rollouts and testing, we tested each policy over 3 different goals with 10 different in-hand orientations in the range of $[-30, 30]$ degrees per goal.

**2. Bin Packing:** This task is designed to handle multi-object interactions during an insertion. The goal is to insert a grasped wooden cube block into a specified target location within a 4 by 4 bin populated with other wooden cubes. The cubes inside the bin are arranged in a grid of rows and columns, with orientations aligned to the bin and with uniform spacing between neighboring cubes, as shown in Fig. 1(b). To increase the task difficulty, cubes adjacent to the goal are squished either vertically or horizontally, partially or heavily occluding the goal (see Appx. F), using either one or two cubes.

**3. Book Shelving:** This task tests lateral insertion where gravity is no longer exerting forces along the axis of insertion. Note the grasped book is larger than the fingertip and therefore the robot cannot infer the object pose solely from touch. The goal is to insert the book, grasped at a consistent in-hand pose, into a target location on a bookshelf where neighboring

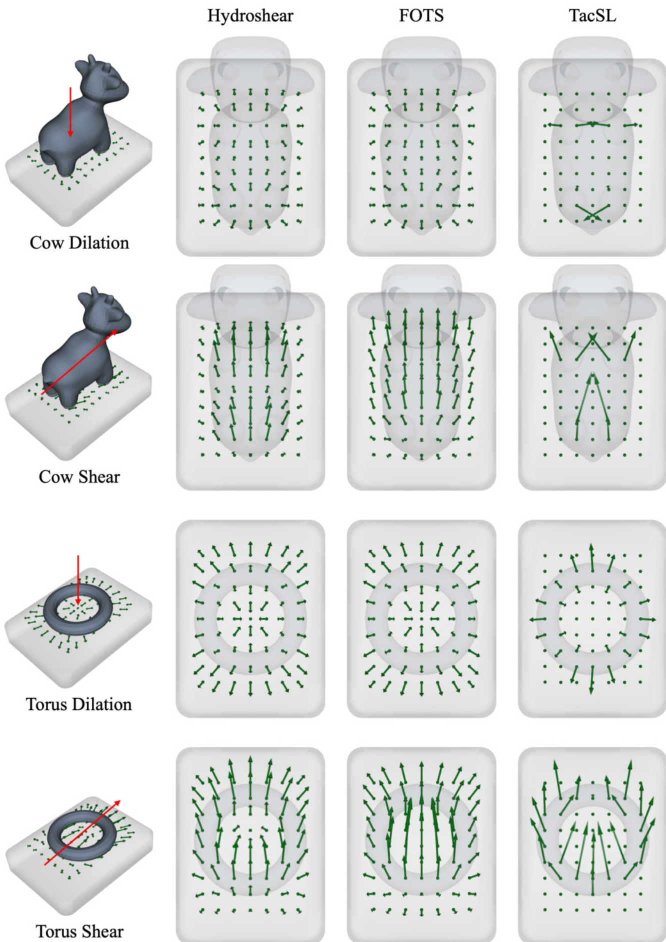


Fig. 6: **Illustration of shear simulation with complex geometries.** We showcase HydroShear's ability to simulate marker displacement fields from interactions between the sensor elastomer and complex geometries such as a cow with four legs and a ring torus. These geometries introduce complex configurations by the shape of the contact patch or by the number of it. We compare against the baseline models and find that with HydroShear's SDF-based formulation and with tactile shadowing effects, we can effectively simulate the shear feedback resulting from complex contact geometries compared to other methods.

books occlude the goal via parallel translation or tilting (see Appx. F), as in Fig. 1(c). We further evaluate performance across books of varying sizes to test more diverse settings.

**4. Drawer Pulling:** This task is designed to test gripper width modulation and reaction to slippage by learning the continuous control of the gripper position. The robot starts by grasping the drawer handle at an initial gripper width that is sufficient to pull the drawer but allows slippage under external disturbances or additional load on the drawer. The task goal is to modulate the gripper width only if an external force is sensed or slippage is detected, thereby preventing further slippage while opening the drawer. To challenge the grasp stability, we apply multiple force perturbations to the drawer at varying timings during pulling. In our evaluation, we introduce three different weight loads and vary the perturbation position and duration. The drawer handle leaving the grasp is considered a failure.

**Baselines:** We compare the zero-shot sim-to-real performance of tactile RL policies each trained with different tactile simulation framework against our method HydroShear. As outlined in Sec. III-D, we first trained a teacher actor-critic for each task through the contact penalty curriculum as shown in Fig. 8. Next, we train one student policy per task and per tactile simulation framework using AACD with using the same teacher checkpoint for each task, as illustrated in Fig. 2. The network architectures (see Appx. C) are identical across tasks and policies, taking the same set of observations (EE pose, relative EE-goal pose, and tactile feedback), with the only difference being the tactile simulation framework.

Here, we highlight the key distinctions between the different baseline student actors with their tactile feedback. (1) TacSL Gray: We train a tactile image-based policy and use grayscale tactile image as we found it to work well empirically. (2) TacSL Shear: We train a TacSL shear-based policy where we normalize the tactile shear into a unit vector, per-taxel, to mitigate for the lack of real-world tactile shadow effects, a strategy adopted by prior works [13, 6, 26]. Next, we train two shear-based FOTS policies: (3) Original FOTS: following the original FOTS formulation but *reimplemented by us for parallelization*, and (4) a second *improved, parallelized* variant that additionally computes marker displacements relative to the initial contact patch center between the object and elastomer, rather than the object center under $\mathrm{SE}(2)$ motion that the original FOTS proposed (see Appx. A). We show that these modifications improve performance significantly and provide a much stronger baseline to compare against. Finally, we train (5) our HydroShear-based policies. The RL training curves for these student policies for each task can be found in Fig. 9-12.

**Gravity Effect on Shear:** For displacement-based shear algorithms like FOTS and HydroShear, we also add in a gravity effect in simulation to compensate for the object's gravity pulling down on the grasped object and hence the elastomer while in-grasp. We discuss this in detail in Appx. J.

**Results:** The experimental results are summarized in Tab. II. We report success rates over episode rollouts and the time-to-success for successful episodes in Tab. IX in Appx. Fig. 18 shows representative sim-to-real rollout keyframes across all tasks along with tactile shear feedback. Across all four real-world tasks, we observe a clear relationship between the fidelity of the simulated tactile shear representation and zero-shot sim-to-real performance in Tab. II.

TacSL Gray performs moderately on Peg Insertion and Bin Packing, contrasted with substantially poorer transfer on Book Shelving and Drawer Pulling. TacSL Gray is an image-based tactile policy, and such policies perform reasonably well when contact patches are small and localized. For Book Shelving, however, full fingertip contact patches hinder textures and the lack of explicit shear direction, accumulation, and slip cues in Drawer Pulling cause the performance to degrade sharply.

TacSL Shear achieves strong performance across most tasks

| **Model** | Peg Insertion | Bin Packing | Book Shelving | Drawer Pulling | Total |
|---|---|---|---|---|---|
| TacSL Gray | 16/30 | 16/30 | 6/30 | 3/30 | 41/120 |
| TacSL Shear | 19/30 | 4/30 | 23/30 | 24/30 | 70/120 |
| FOTS | 1/30 | 5/30 | 20/30 | 15/30 | 41/120 |
| FOTS (Reimpl.) | 20/30 | 24/30 | 26/30 | 3/30 | 73/120 |
| HydroShear | **25/30** | **29/30** | **28/30** | **30/30** | **112/120** |

TABLE II: **Zero-shot Sim-to-Real Policy Evaluation.** We test 5 different policies that is trained with different tactile simulation frameworks for each task: (1) TacSL tactile grayscale image; (2) TacSL tactile normalized shear; (3) Original FOTS tactile shear with *our* parallelization; (4) Our *reimplemention* of FOTS with improvements; (5) HydroShear tactile shear.

but degrades notably on Bin Packing. This limitation arises from its reliance on per-taxel shear normalization, which discards critical information about shear magnitude and the spatial structure of contact geometries with tactile shadows. While this normalization is effective for Peg Insertion, where it preserves contact geometry and in-hand pose, and for Drawer Pulling, where the formulation naturally captures slippage, it becomes detrimental for Bin Packing. In this task, large shear and torsional interactions induce broader contact patches and global elastomer deformations producing normalized shear fields that span a much wider patch than what the student policy was trained to expect, leading to reduced performance.

The original FOTS implementation performs poorly on the Peg Insertion and Bin Packing tasks because the object frame center is defined much further away from the contact patch center between the elastomer and grasped objects. Conversely, it performs well on Book Shelving and Drawer Pulling because the object frame center is much closer to the contact patch center between the elastomer and grasped object. FOTS exposes a sim-to-real gap when the object frame is poorly aligned with the grasp or when contact occurs far from the reference origin. In these cases, the resulting tactile signals fail to reflect the local contact dynamics experienced by the sensor.

FOTS (Reimpl.) substantially improves transfer for tasks with relatively stable contacts, but the performance degrades for Drawer Pulling. Re-centering shear around an initial contact patch enables this improved transfer for stable contacts, but when a single reference patch is no longer well-defined in Drawer Pulling, the performance degrades. The assumption of the contact patch location breaks down under frequent contact transitions, intermittent slip, or external perturbations.

In contrast, HydroShear achieves robust zero-shot transfer across each tasks by explicitly modeling path-dependent, SE(3) contact interactions by tracking on-surface indenter points. By tracking how shear accumulates, dissipates, and transitions through stick–slip events, the simulated tactile feedback remains consistent with real sensor observations even as contact configurations evolve. This enables a single policy architecture and training pipeline to generalize across dense insertion, multi-object interactions, full-contact manipulation, and slip-sensitive force modulation.

We provide a qualitative comparison of different tactile shear feedbacks during policy rollouts in sim for each task in Fig. 14-17 to compare against real rollouts in Fig. 18.

Ablation on Contact Penalty: We empirically find that teacher checkpoints finetuned with the contact penalty curriculum significantly improve downstream student policy behavior during sim-to-real transfer. For example, peg insertion policies trained without any contact penalty make overly aggressive contact with the socket and repeatedly react in the same manner, damaging the soft elastomer membrane of the GelSight Mini sensors. In contrast, using the contact penalty curriculum yields safer tactile policies that make controlled contact with the environment and exhibit emergent goal-searching behavior.

## V. Discussion and Limitations

HydroShear enables a powerful recipe for sim-to-real tactile policy transfer. While effective, our method depends on whether the underlying physics engine allows for penetrating contact simulation. In this work, we use Isaac Gym TacSL's Kelvin-Voigt implementation for compliant contact. Future extensions could incorporate physics engines that natively support hydroelastic contacts.

HydroShear uses SDFs to represent both the indenter and elastomer. Thus, using higher SDF resolutions can result in higher shear simulation accuracy at the cost of slowing down the simulation due to the increased number of points to track during indenter-elastomer penetration. In addition, SDFs are particularly effective for rigid-bodies. However, deformable objects may require an alternative representation that will affect the algorithms used to track points and compute forces.

HydroShear assumes that the elastomer of the vision-based tactile sensor is flat. Future work could extend HydroShear to curved elastomers on dexterous hand fingertips.

HydroShear is effective for vision-based tactile sensors that rely on deformation cues. While extension to other tactile sensors is possible, their evaluation is outside the scope of the current work.

## Acknowledgments

We would like to thank Miquel Oller and James Lorenz for their help with the real-world experimental setup. This material is based upon work supported by the National Science Foundation under Grant No. 2337870 & 2220876.


## References


[1] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

[2] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *International Conference on Robotics and Automation (ICRA)*, pages 2811–2817, 2021.

[3] Yinuo Wang and Xiaowen Tao. Locomamba: Vision-driven locomotion via end-to-end deep reinforcement learning with mamba. *Advanced Engineering Informatics*, 70:104230, 2026. ISSN 1474-0346. doi: https://doi.org/10.1016/j.aei.2025.104230. URL https://www.sciencedirect.com/science/article/pii/S1474034625011231.

[4] Yunlong Song, Kim, Sangbae, and Davide Scaramuzza. Learning quadruped locomotion using differentiable simulation. In *Conference on Robot Learning (CoRL)*, 2025.

[5] Zilin Si and Wenzhen Yuan. Taxim: An Example-Based Simulation Model for GelSight Tactile Sensors. *IEEE Robotics Autom. Lett.*, 7(2):2361–2368, 2022. doi: 10.1109/LRA.2022.3142412.

[6] Iretiayo Akinola, Jie Xu, Jan Carius, Dieter Fox, and Yashraj S. Narang. TacSL: A Library for Visuotactile Sensor Simulation and Learning. *CoRR*, abs/2408.06506, 2024.

[7] Duc Huy Nguyen, Tim Schneider, Guillaume Duret, Alap Kshirsagar, Boris Belousov, and Jan Peters. TacEx: GelSight Tactile Simulation in Isaac Sim – Combining Soft-Body and Visuotactile Simulators. 2024. URL https://arxiv.org/abs/2411.04776.

[8] Daniel Fernandes Gomes, Paolo Paoletti, and Shan Luo. Generation of gelsight tactile images for sim2real learning. *IEEE Robotics and Automation Letters*, 6(2):4177–4184, 2021.

[9] Daniel Fernandes Gomes, Paolo Paoletti, and Shan Luo. Beyond Flat GelSight Sensors: Simulation of Optical Tactile Sensors of Complex Morphologies for Sim2Real Learning. *Robotics: Science and Systems (RSS)*, 2023.

[10] Kipp Freud, Yijiong Lin, and Nathan F. Lepora. Simshear: Sim-to-real shear-based tactile servoing. *Conference on Robot Learning*, 2025.

[11] Christopher J. Ford, Haoran Li, Manuel G. Catalano, Matteo Bianchi, Efi Psomopoulou, and Nathan F. Lepora. Shear-based grasp control for multifingered underactuated tactile robotic hands. *IEEE Transactions on Robotics*, 41:3113–3128, 2025. doi: 10.1109/TRO.2025.3563046.

[12] 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)*, pages 5850–5856, 2025.

[13] Jie Xu, Sangwoon Kim, Tao Chen, Alberto Rodriguez Garcia, Pulkit Agrawal, Wojciech Matusik, and Shinjiro Sueda. Efficient Tactile Simulation with Differentiability for Robotic Manipulation. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 1488–1498. PMLR, 2022.

[14] Yongqiang Zhao, Kun Qian, Boyi Duan, and Shan Luo. FOTS: A Fast Optical Tactile Simulator for Sim2Real Learning of Tactile-motor Robot Manipulation Skills. *IEEE Robotics and Automation Letters*, 2024.

[15] Yashraj S Narang, Karl Van Wyk, Arsalan Mousavian, and Dieter Fox. Interpreting and predicting tactile signals via a physics-based and data-driven framework. *arXiv preprint arXiv:2006.03777*, 2020.

[16] Zihan Ding, Nathan F Lepora, and Edward Johns. Sim-to-real transfer for optical tactile sensing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1639–1645. IEEE, 2020.

[17] Youngsun Wi, Jayjun Lee, Miquel Oller, and Nima Fazeli. Neural Inverse Source Problem. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4371–4391. PMLR, 06–09 Nov 2025.

[18] Miquel Oller, Dmitry Berenson, and Nima Fazeli. TactileVAD: Geometric Aliasing-Aware Dynamics for High-Resolution Tactile Control. In *7th Annual Conference on Robot Learning*, 2023.

[19] Yashraj Narang, Balakumar Sundaralingam, Miles Macklin, Arsalan Mousavian, and Dieter Fox. Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6444–6451. IEEE, 2021.

[20] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *ICLR*, 2020.

[21] Zilin Si, Gu Zhang, Qingwei Ben, Branden Romero, Zhou Xian, Chao Liu, and Chuang Gan. DIFFTACTILE: A Physics-based Differentiable Tactile Simulator for Contact-rich Robotic Manipulation. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. TACTO: A Fast, Flexible, and Open-Source Simulator for High-Resolution Vision-Based Tactile Sensors. *IEEE Robotics Autom. Lett.*, 7(2):3930–3937, 2022. doi: 10.1109/LRA.2022.3146945.

[23] Ryan Elandt, Evan Drumwright, Michael Sherman, and

Andy Ruina. A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8238–8245, 2019. doi: 10.1109/IROS40897.2019.8968548.

[24] Miquel Oller, An Dang, and Nima Fazeli. Hydrosoft: Non-Holonomic Hydroelastic Models for Compliant Tactile Manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2026.

[25] Joseph Masterjohn, Damrong Guoy, John Shepherd, and Alejandro Castro. Velocity level approximation of pressure field contact patches. *IEEE Robotics and Automation Letters*, 7(4):11593–11600, 2022.

[26] Jayjun Lee and Nima Fazeli. ViTaSCOPE: Visuo-tactile Implicit Representation for In-hand Pose and Extrinsic Contact Estimation. In *Proceedings of Robotics: Science and Systems*, 2025.

[27] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.