

OpenTac: An Open Ecosystem for Customizable, Scalable, and Transferable 3D Tactile Sensing

Ariel Slepyan^{1,2}, Geeta Chandra Raju Bethala¹, Jiafeng Gu¹, Anagha Rajendra Dangle¹, Yinzhe Zhou¹, Hongyuan Fu¹, Jade Sim¹, Tuhin Basu¹, Xinyi Yin¹, Md Faizal Karim¹, Gregory Hager^{†1}, Homanga Bharadhwaj^{†1}, Anand Bhattad^{†1}, Krishna Murthy Jatavallabhula^{†1}, Nitish V. Thakor^{†1}

¹ Johns Hopkins University, ² Massachusetts Institute of Technology

[†]Equal advising

opentac.github.io

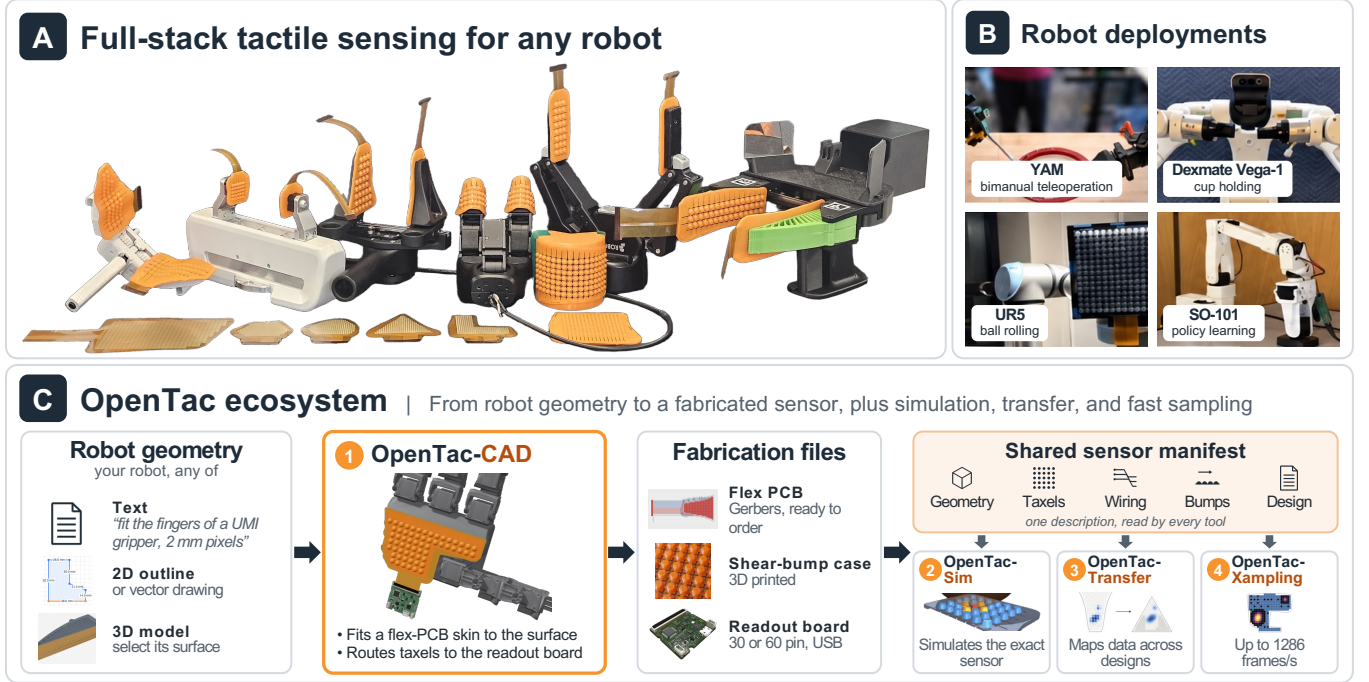


Figure 1. The OpenTac ecosystem. (A) Custom tactile skins that sense both normal and shear force, built for robot hands, grippers, arms, and feet. (B) Robot deployments on 4 real-world platforms. (C) ① OpenTac-CAD compiles a text description, 2D drawing, or 3D CAD surface into a fabrication-ready flex PCB, a shear-bump case, and a machine-readable sensor manifest. The manifest is shared by ② OpenTac-Sim, which instantiates the corresponding sensor geometry in simulation; ③ OpenTac-Transfer, which maps tactile observations between sensor designs; and ④ OpenTac-Xampling, which performs adaptive partial sampling at up to kilohertz rates.

Abstract

We present OpenTac, a novel sensor and a full-stack ecosystem for tactile skins that are highly customizable, scalable, and readily deployed in downstream robot control and learning applications. Our core contribution is twofold. First, we develop a scalable resistive tactile skin with a structured deformable overlay that converts tangential loading into spatially varying pressure. Differential responses across neighboring ‘taxels’ support decoding of local shear vectors, and the common-mode response encodes normal force. Second, we provide a highly customizable sensor design tool that is able to automatically generate fabrication files for these tactile skins from 2D outlines and developable 3D surfaces, including from abstract descriptions such as text (e.g., “design a sensor for a size 9 shoe insole”). We develop an ecosystem around the OpenTac sensor, including the CAD design tool, a sensor simulator, cross-sensor transfer, and fast/adaptive data acquisition. We demonstrate the broad applicability of the OpenTac-CAD sensor design tool by designing and fabricating more than 30 sensors across a variety of robot embodiments. We characterize the three-dimensional tactile fields produced by OpenTac with a commercial, high-precision force/torque sensor. We demonstrate the broad applicability of 3D tactile sensing across contact-rich feedback control and manipulation policy learning on four real-world robot embodiments. Additionally, adaptive acquisition enables kilohertz-rate partial readouts, exceeding full-array raster scanning on the same sensor by $19.2\times$ at 16 sampled taxels per update for a 1024-taxel skin. We provide all hardware, software, and result videos are at opentac.github.io

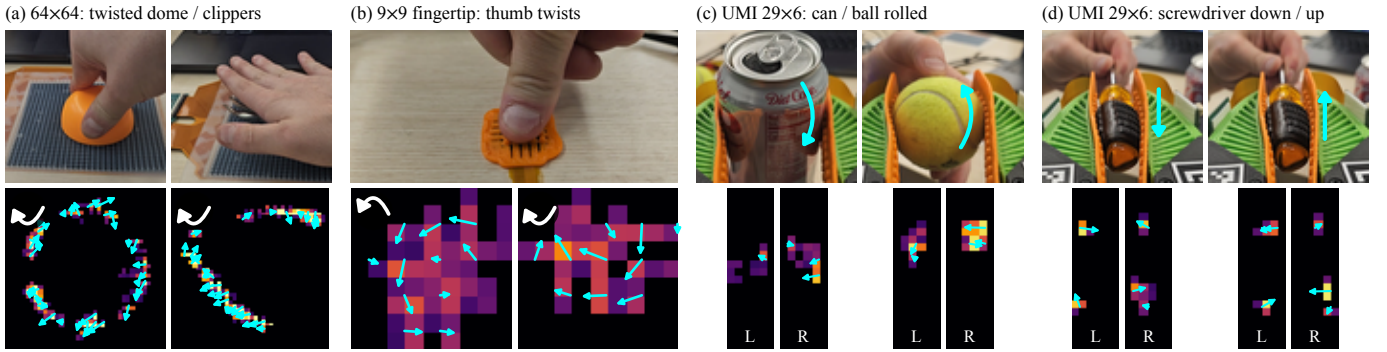


Figure 2. Tactile samples recorded with OpenTac skins. Heatmaps indicate normal load; cyan arrows denote per-bump shear; white arcs denote rotation estimates. (a) 64×64 skin, twisting a dome and clippers. (b) 9×9 fingertip, opposite thumb twists. (c) UMI gripper, 29×6 finger skins: can and ball rocked in the grasp. (d) Extrinsic contact: screwdriver moved down and up.

1 Introduction

As robots expand to perform tasks in more general, varied, and dynamic environments, their ability to manipulate quickly, reliably, and safely will increasingly depend on directly sensing and reacting to contact—that is, on tactile sensing. However, sensors must first be customized to fit the end-effector or surface of interest. As a result, research on the use of tactile sensing is fundamentally limited by customizability: researchers must either adapt their work to sensors manufactured in fixed geometries, or they must possess sufficient design and fabrication expertise to develop custom sensors in-house. These constraints are particularly limiting when instrumenting custom-designed robot geometries or larger, irregular surfaces.

In this paper, we focus on *conformal tactile skins*, as they integrate distributed sensing elements into thin, flexible surfaces that can be wrapped around robot geometries, enabling fully sensorized robot bodies. However, existing tactile skins, such as FlexiTac [1], provide only normal-force (1D) estimates and require manual redesign for each new geometry. We present a sensor design that can be easily customized to fit many robot geometries, scaled in sensing area or resolution, and used to estimate normal and shear forces (3D) over local groups of sensing elements.

Several efforts have partially addressed sensor customizability and made tactile skins more capable and accessible for the robot control and learning communities [1–3]. AnySkin provides replaceable sensing surfaces but relies on a predefined electronic layout [2], while eFlesh automates the design of a deformable sensor body but relies on separately supplied magnetometer electronics [3]. In addition to the challenges of customizing the entire sensor, existing sensors are sampled at rates ranging between 30 and 100 Hz. Such raster scanning-based acquisition can limit the ability to resolve and respond to contacts lasting only a few milliseconds. Beyond hardware, changes in sensor geometry must also be accommodated by the software used to acquire, simulate, and process tactile observations.

To address these limitations, we introduce **OpenTac**, an open sensor design environment and software ecosystem for customizable, scalable tactile sensing (Fig. 1). OpenTac supports the design of low-cost piezoresistive tactile skins that estimate *both* local normal and shear forces with millimeter-scale spatial resolution over planar and developable surfaces. Our sensor combines a flexible printed circuit board and a piezoresistive film with a structured elastomeric overlay that redistributes lateral pressure across neighboring sensing elements, allowing local shear to be decoded while retaining normal-force information. The sensing layout, electrode geometry, and mechanical overlay can be customized to the surface geometry and force range of interest.

To support this degree of customizability, OpenTac includes a full-stack ecosystem of design, simulation, and interoperability tools. **OpenTac-CAD** is a design tool that automatically generates sensor fabrication files from a desired geometry. Users can provide a 2D outline, select a developable surface on a 3D robot model, or specify an abstract verbal description, such as “*design a star-shaped sensor that fits inside a circle with a diameter 5 cm.*” In addition, OpenTac-CAD produces a machine-readable *manifest* file describing the sensor’s geometry, electrical addressing, and mechanical configuration. **OpenTac-Sim** provides tools to simulate the sensors and sensor-object interactions for robot manipulation tasks across any sensor design described by the manifest file. **OpenTac-Transfer** enables cross-sensor transfer by mapping calibrated tactile observations across sensor layouts using their physical coordinates and local surface frames. These components allow sensors with different shapes, resolutions, and electrical layouts to share acquisition and processing tools and provide a common interface for downstream control and learning.

Scaling tactile coverage also increases the number of measurements required to read the sensor. To address this, we also provide **OpenTac-Xampling**, an adaptive acquisition method which operates across different sensor outlines using adaptive partial scans. For example, on a 32×32 skin, adaptive acquisition of sparse contact events reaches 1286 Hz, more than a $19\times$ increase over full-array scans (67 Hz) achieved by conventional acquisition [4].

Table 1. Representative tactile sensing systems. Auto-design lists automatically generated components. Shear denotes sensitivity to tangential force. Est. cost is per sensor unit (or sq. cm for skin). [†] indicates cost is for just the sensor. Readout board costs vary based on form.

Sensor	Shear	Form	Auto-design	Rate (Hz)	Est. cost (\$/unit)
GelSight [5]	✓	Finger	✗	30	65 – 400
DIGIT [6, 7]	✓(with markers)	Finger	✗	60	55 – 103
GelSlim 4.0 [8]	✓	Finger	✗	30-60	73 – 123
TacTip [9]	✓	Finger	✗	30–120 ^b	67 – 169
uSkin [10]	✓	Patch	✗	100 ^c	10 – 25
ReSkin [11]	✓	Skin	✗	400	5.5 – 10.5
AnySkin [2]	✓	Skin	Fingertip + Moulds	100	2 – 10
eFlesh [3]	✓	3D body	Sensor surface	100	25 – 50
iCub skin [12]	✗	Skin	✗	25	2.5 – 7
FlexiTac [1]	✗	Skin	✗	100	≈ 1.95 [†]
Sun et al. [13]	✓	Skin	✗	N/A	2 – 10
OpenTac	✓	Skin	full-sensor	67–1286 (adaptive)	≈ 1.95 [†]

We evaluate all components of the OpenTac ecosystem. We use OpenTac-CAD to generate fabrication files for more than 30 sensor layouts (including robot fingers, grippers, palms, and feet), fabricate the corresponding sensors, and deploy them on four real robot embodiments. We further evaluate the 3D force estimates against a commercial force/torque sensor. Finally, we demonstrate that our sensor’s normal and shear force estimates enable contact-rich feedback control and manipulation policy learning.

To summarize, our main contributions are:

- **OpenTac: a full-stack ecosystem** for automated sensor design, fabrication, simulation, cross-sensor observation mapping, and kilohertz-rate adaptive acquisition.
- A customizable tactile skin that estimates **normal and shear forces** using a simple, scalable construction.
- Evaluations spanning over 30 fabricated layouts, sensor characterization, high-speed sensing, and real-world robot control and learning, demonstrating the utility of the resulting tactile observations for contact-rich manipulation.

All OpenTac hardware, software, and firmware is available under permissive licenses, enabling replication, modification, and redistribution for research and commercial use.

2 Related Work

Tactile sensing hardware typically involves optical, magnetic, capacitive, and/or resistive transduction to recover contact. GelSight provides high-resolution optical contact sensing [5], while uSkin integrates distributed three-axis force elements [10]. GelSlim uses marker displacements and inverse finite-element modeling to estimate dense contact forces [14], while Yan *et al.* use a magnetized film to decouple normal force and one shear component [15]. Mechanical structures can also encode shear into differential pressure: Sun *et al.* place a bump over four piezoresistive cells and infer load di-

rection from their relative responses [13]. OpenTac builds on this principle using a single-FPCB (flexible PCB) sensing architecture designed for automated customization and shared readout design that specifies a unified cross-sensor interface.

Early work on **customizing and scaling tactile skins** used modular or severable elements to cover complex robot surfaces [16, 17], while later systems such as the iCub skin, RobotSweater, and conformal tactile textiles target large-area conformable coverage [12, 18, 19]. ReSkin and AnySkin decouple the replaceable magnetic interface from the electronics [2, 11]. eFlesh automates magnetic microstructures [3], and FlexiTac provides an open flexible pressure array [1]. Computational design tools have also produced personalized FPCB tactile gloves [20] and surface-conforming capacitive sensors for existing 3D objects [21]. Lim *et al.* use laser-patterned flexible circuits to integrate capacitive pressure and bending sensors with temperature and proximity modules in customizable multimodal skins [22]. OpenTac instead starts from robot geometry and jointly generates the sensing layout, routing, connectorization, mechanical structure, and associated metadata, enabling robot-specific tactile skins.

Tactile simulation and manipulation. TACTO [23] and Taxim [24] focus on simulating the imaging process inherent in vision-based tactile sensors. DiffTactile [25], TacEx [26], and Taccel [27] additionally implement sensor surface deformation mechanics for robot manipulation. HydroShear further models shear and stick-slip effects for sim-to-real learning [28]. Tactile feedback has been used in fine manipulation and shear-based control [29, 30]. Different to prior work, OpenTac-Sim implements a simulation framework specifically for OpenTac sensors using the compiled sensor geometry and bump layout to generate normal and tangential deformation observations.

Acquisition and cross-sensor representations. High-rate tactile sensing has used event-driven hardware [31, 32]

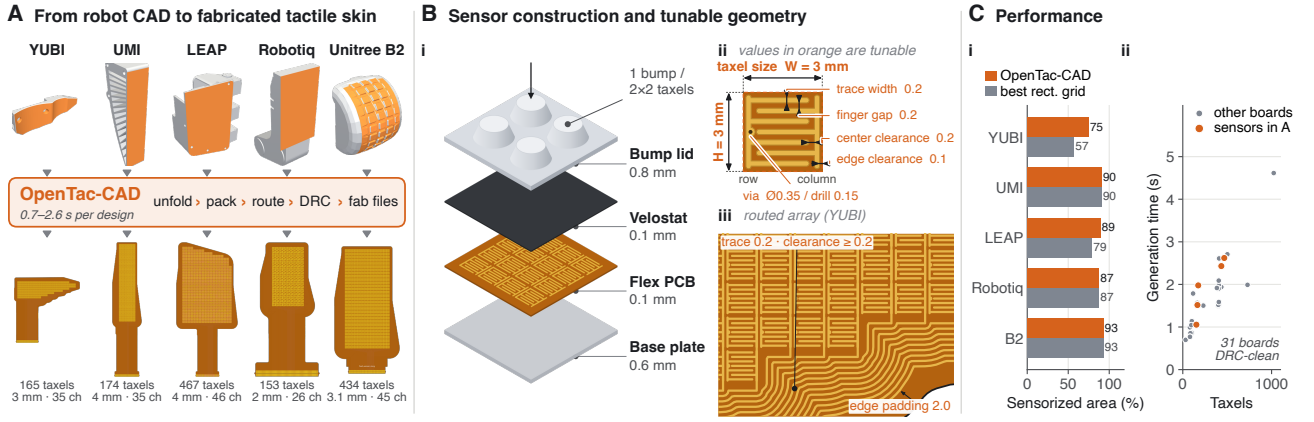


Figure 3. OpenTac-CAD creates manufacturable tactile skins for 2D/3D shapes. (A) Selected surfaces (orange) on YUBI, UMI, LEAP Hand, Robotiq 2F-85, and Unitree B2 models and the generated FPCBs; labels give taxel count, pitch, and row-plus-column readout channels. (B) (i) Sensor stack: a printed bump lid places one bump over each 2×2 taxel group above the piezoresistive film and two-layer FPCB. (ii) One generated taxel and (iii) a routed corner of the YUBI board; values in orange are compiler inputs, shown at the YUBI settings, that can be changed to generate a different sensor. (C) Sensorized area versus the largest inscribed rectangular array at the same pitch, and generation time versus taxel count for all 31 fabricated boards, all with zero DRC violations.

and adaptive subsampling of dense arrays [33]. OpenTac-Xampling applies adaptive sampling across arbitrary compiled layouts using manifest-defined valid regions and neighborhoods. Cross-sensor tactile representation learning combines aligned synthetic tactile data with real-world multimodal data to learn sensor-agnostic visuo-tactile representations [34–36]. OpenTac-Transfer instead maps calibrated contact fields through physical coordinates and local frames, without needing paired cross-sensor data.

3 The OpenTac Ecosystem

OpenTac introduces a full-stack ecosystem for piezoresistive tactile skins in robot control and learning applications. The OpenTac tactile sensor comprises a structured deformable overlay, which redistributes tangential loading across a small grid of underlying taxels. This enables the estimation of both normal and shear forces over the piezoresistive film.

All OpenTac components use a shared sensor description: the *manifest*. The ecosystem comprises four distinct modules:

1. OpenTac-CAD, which designs a sensor given a 2D outline or a developable 3D shape, producing the fabrication files and the manifest;
2. OpenTac-Sim, a simulation toolkit built on MuJoCo to simulate sensor readings;
3. OpenTac-Transfer, a module to enable cross-sensor geometry transfer; and
4. OpenTac-Xampling, an adaptive readout module enabling kilohertz-rate sampling for sparse contact acquisition.

Together, these enable OpenTac to be directly deployed across highly versatile traditional, learning-based, and interactive robotics applications.

3.1 OpenTac: Normal and Shear Sensor design

We choose piezoresistive sensing for its low material cost, simple readout circuitry, and compatibility with standard flexible-PCB (fPCB) fabrication. The OpenTac sensor stack (Fig. 3B) comprises three primary components: a single fPCB (bottom), a piezoresistive film (middle), and a printed mechanical cover (top). At each row-column intersection on the fPCB, interdigitated copper fingers form one taxel. Any applied pressure on the sensor decreases the local resistance on the piezoresistive film and is converted by a readout circuit into an output voltage to estimate normal force. The top surface of the sensor is a deformable bump strip, based on [37], which is where sensor-object interactions occur. Each bump sits atop a small window (e.g., 2×2) of taxels. For three-axis force sensing, each elastomeric bump covers a local group of taxels, with the number and arrangement of taxels determined by the sensor geometry. For a 2×2 configuration, let the baseline-subtracted signals be z_{uv} for $u, v \in \{-1, +1\}$, with signs denoting each taxel’s offset from the bump center. Three-axis tactile readings are then:

$$h_z = \sum_{u,v} z_{uv}, \quad h_x = \sum_{u,v} u z_{uv}, \quad h_y = \sum_{u,v} v z_{uv}. \quad (1)$$

Normal loading produces a near-uniform pressure distribution over all taxels underneath a bump, resulting in a common-mode response h_z . Tangential loading tilts the bump, resulting in a pressure, and by extension, resistance differential across the underlying taxels, producing differential responses h_x and h_y . Optionally, we may use an affine map to calibrate sensor readings $[h_x, h_y, h_z]$ to force units $[F_x, F_y, F_z]$. The manifest defines the grouping and signs for each compiled array. This design choice enables a high degree of customization, by leveraging automated CAD and circuit design principles to parametrize sensor density, electrode layout, and mechanical transduction, while retaining the sensing, assembly, and readout processes.

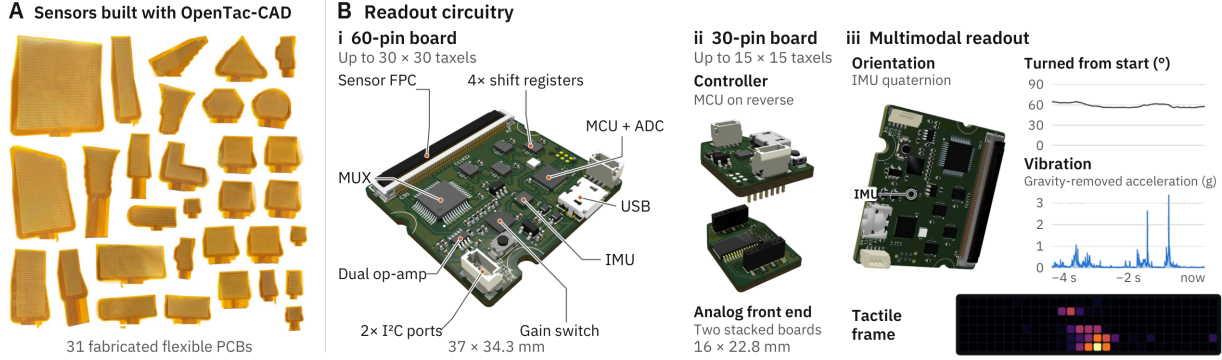


Figure 4. **A.** 31 sensors in various form factors designed with OpenTac-CAD and fabricated. All sensors were tested and work. **B.** We provide two **readout board** designs, varying in the size and the number of sensor wires they support. Each board contains filtering and readout circuitry, an IMU, and a microcontroller. It can be plugged directly into a USB port to stream data.

3.2 OpenTac-CAD: Automatic sensor design

To deploy OpenTac sensors on a new robot, surface, or end-effector, we need to first custom-design a sensor that fits the surface. With existing sensors, customization is challenging, necessitating expertise across electronic circuit design, electrode spacing, cable routing, mechanical design and modeling, and more. To address this, and to enable high degree of customizability, we build OpenTac-CAD. Given a 2D or 3D shape, our compiler deterministically generates the board geometry. Users can customize sensing regions, taxel footprint, pitch, electrode and clearance rules, cable, and connector.

Sensor outline extraction. OpenTac-CAD supports sensor design for a user-drawn 2D input region, a 2D outline imported from a vector drawing (DXF), or a 3D mesh (STL) where users specify the surface(s) to be sensorized. In the case of a 3D mesh, surfaces are first unfolded to 2D triangle by triangle across shared edges. Since each triangle’s flat shape follows exactly from its edge lengths, developable surfaces such as cylinders, cones, and faceted parts flatten without distortion, and doubly curved selections that cannot be flattened are detected and reported. We additionally support an agentic outline extraction mode, where a large language model processes an input text prompt and designs a 2D outline. Extracted outlines from this step are passed along to the next step for wiring.

Taxel size, pitch, and placement. The default size across most OpenTac sensors is 4 mm taxels at 4.2 mm centre-to-centre spacing (also referred to as the *pitch*). In our tests, we fabricated sensors ranging from 4 mm to 1 mm taxel sizes. Reducing the pitch increases the number of taxels within a given area and provides finer spatial sampling; increasing taxel size increases the signal measured. Sensor width and height can be independently configured if needed. Once taxel size and pitch are determined, we populate taxels all along the area to be sensorized, accounting for any additional free space required to route the wires originating from the taxels. We also allow users to specify additional design constraints, such as additional boundary space on the sensor edge, or the length and shape of the sensor *tail* (i.e., the side of the

sensor where the wires are all routed to, and connected to the readout board).

Readout boards. We provide two readout board designs: a 60-pin board for sensors with up to 30 rows and 30 columns, and a compact 30-pin pair for up to 15 rows and 15 columns. Both use the same firmware codebase and USB interface, with the sensor-specific layout supplied through its manifest. One half of the connector drives up to 32 lines through shift registers; the other senses up to 32 lines through a multiplexer and an inverting amplifier that holds the sensed line at virtual ground to suppress crosstalk. The compiler assigns one electrode set to each connector half and allocates pins outward from the center. This enables skins of varying sizes to be plugged into the same readout board.

Contour-following routing. Rows and columns lie on opposite faces of the fPCB, so routing each face is a planar single-layer problem. Every net follows a span of the sensor-array contour, split into hops around concavities. Spans on a face must be nested or disjoint, so unavoidable crossings are detected a priori. Routes are processed from the shortest to the longest span. Each route follows a clearance-offset contour around the array and previously routed copper, avoiding an expensive routing search. Every route is checked for clearance before acceptance. This estimates the required routing band for each sensor outline.

Sensor Manifest The manifest describes the sensor’s mechanical and electronic construction, and enables interoperability across all tools in the OpenTac ecosystem. For every taxel i , it stores the physical center $\mathbf{p}_i$, row-column scan indices (r_i, c_i) , and footprint parameters, together with the sensing boundary, manufactured outline, connector pin map, and the recipe that generated the board. Bump strips additionally store each taxel’s mechanical group $b(i)$. Separating the geometric information from raster scan ordering is important for irregular skins, because the wires are optimized for routing, and that order could be different from that of the physical coordinates. Fig. 4 shows 31 sensors that were designed entirely in OpenTac-CAD and fabricated using a flexPCB printing service.

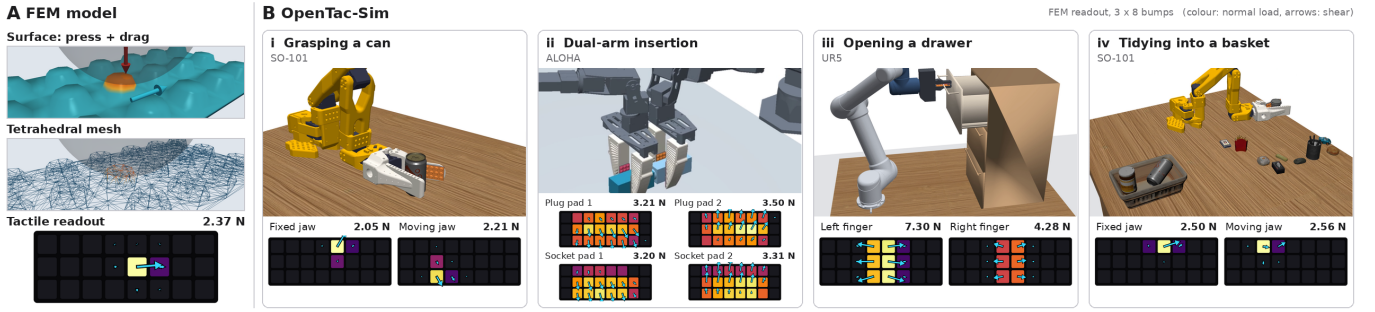


Figure 5. OpenTac-Sim enables the simulation of OpenTac sensors, ensuring that normal and shear forces are accurately modeled. We use an FEM model that reads the sensor manifest created by OpenTac-CAD and supports the simulation of sensors of arbitrary shapes and resolutions. **A.** FEM model uses a tetrahedral mesh to simulate bump deformation and computes normal and shear forces using Eq. 1. **B.** OpenTac-Sim includes support for multiple robots from the MuJoCo menagerie, across several contact-rich tasks.

3.3 OpenTac-Sim: Simulating OpenTac sensors

OpenTac-Sim provides a set of accurate simulation methods for OpenTac sensors. The bump-covered sensors that measure both normal and shear forces are simulated using an FEM model, while smooth sensor skins that only measure normal force are simulated using a reduced foam-contact model. Sensor geometry and output formats are directly instantiated from the sensor manifest, enabling OpenTac-Sim to support the simulation of sensors of arbitrary shapes and resolutions.

Sensor model: The manifest file describes sensor geometry and is used in the initialization of multiple tactile *pads*. Each pad is converted into a tetrahedral mesh and is simulated as a linear-elastic FEM body in MuJoCo, with its base pinned onto the sensor base plate (or a robot gripper, as applicable). Upon physical contact with an object, the tactile pads deform based on the contact geometry and the load (Fig. 5A). We retain the original surface mesh for collision evaluation. At each bump, we compute normal and shear forces in the local tactile pad frame and interpolate them to taxel resolution.

Robot and task simulation: We support the simulation of multiple robots from MuJoCo menagerie. New robots and tasks can be easily instantiated. The current release of the simulator only supports stable simulation of sensor-object interactions with rigid objects.

Tactile readout: To simulate the effects that the sensor stack and circuitry would have on taxel readings, we also model interaction between the tactile pads, the base, and the electrode footprints. Starting from the FEM state, we compute the elastic force exerted by each tactile pad on the base, map it onto the electrode footprints, and apply our normal and shear force estimation method from Eq. 1. We run an automated calibration step per sensor to remove loads originating from the pad’s own scaled mass under gravity and arm acceleration.

Smooth sensor simulation: The simulation also supports smooth sensor skins using a reduced foam-contact model. We model the foam as a Winkler layer, $p = k(\delta - z)_+$, solve δ to match the applied normal force, smooth the resulting

pressure field with a Gaussian of width σ , and integrate it over the manifest taxels before applying the target calibration.

3.4 OpenTac-Transfer: Cross-sensor transfer

OpenTac-Transfer maps calibrated normal-contact fields between skins in physical coordinates. With source and target taxel cells A_i and B_j , contact-frame alignment T , and source calibration C_s ,

$$M_{ji} = \frac{|T(A_i) \cap B_j|}{|A_i|}, \quad \hat{F}_j = \sum_i M_{ji} C_s(s_i), \quad (2)$$

and target counts are predicted through the inverse target calibration, $\hat{s}_j = C_t^{-1}(\hat{F}_j)$, with its noise floor and saturation applied. Force is conserved when the target covers the contact. To compute this mapping, we only need the calibration map (an affine transform) per-sensor, and for the contact alignment to be known.

3.5 OpenTac-Xampling: Adaptive Acquisition

Piezoresistive sensor arrays are commonly read via *raster scanning*. We read taxels left-to-right and top-to-bottom, by energizing one row and column at a time. For N taxels, this raster strategy costs N samples even when contact occupies only a small region. This results in two unintended consequences. First, readout rates slow down with increase in the area or resolution of a tactile skin. Second, fast contacts that typically occur in under 10 milliseconds go undetected by the sensor. To mitigate these, OpenTac-Xampling assigns a budget of $M \ll N$ samples per frame, using the binary-quadtrees adaptive subsampling of [33]. We allocate one quarter of this per-frame sampling budget to search over the full mask, and the remainder to follow active regions detected in the previous frame. This leverages the idea that contacts tend to be sparse, local, and temporally coherent. The approach works on any kind of sensor layout (rectangular, concave, curved-part, etc.). In the worst-case (i.e., when every single taxel experiences contact), this gracefully degrades to have identical complexity as raster scanning.

A AI-assisted sensor design from text descriptions

"Tactile skin for the fingertip pad of an Allegro Hand, cable running down toward the palm."

look up Allegro URDF: 19.6 × 27.5 mm finger, R12 mm tip
assume tip cap is doubly curved: palmar face + 7 mm of cap
trade off 2.0 mm pitch: 40-pin connector > finger width → 2.2 mm



B More examples

"A pressure-sensing sole for the foot of a Unitree G1 humanoid robot."



246 taxels · 43 ch · 74×213 mm

"Tactile skin for the palm of a Shadow Dexteros Hand, cable exiting at the wrist."



192 taxels · 33 ch · 62×91 mm

"A seat-pressure sensor for a road bike saddle, cable at the back."



254 taxels · 49 ch · 139×259 mm

"A sensor shaped like the state of Massachusetts, 120 mm wide, cable on the southern edge."



115 taxels · 34 ch · 124×78 mm

"A sensor shaped like a gecko silhouette, about 90 mm nose to tail, cable at the tail."



146 taxels · 37 ch · 62×111 mm

"A five-pointed star sensor, 60 mm tip to tip, with the cable leaving between the two lower points."



168 taxels · 43 ch · 64×78 mm

Figure 6. Tactile skins from one-sentence language descriptions. A coding agent (Claude Opus 5) generated each board design from the quoted prompt using OpenTac-CAD and web search. (A) Logged decisions for an Allegro Hand fingertip: a dimension looked up online, an assumption about the tip geometry, and a pitch change forced by connector width. (B) Further designs; the bottom row used outline-growth mode at the operator’s request. Labels give taxel count, routed row-plus-column nets, and board size. These examples evaluate automated design generation, not fabrication.

4 Experimental Evaluation

We experimentally evaluate every component of the OpenTac ecosystem, including the compiler, sensor characterization, acquisition, transfer, and applicability to contact-rich control and manipulation policy learning.

The key experimental findings are that:

1. OpenTac-CAD auto-generates designs that result in higher taxel coverage, without any design violations
2. OpenTac sensors provide accurate shear and normal force estimates, measured against reference commercial 3-axis force sensors
3. OpenTac-Sim faithfully simulates the sensors, as quantified by aligned real-and-simulated sensor data
4. OpenTac computes accurate shear for downstream contact-rich control and manipulation policies
5. OpenTac-Transfer enables transferring normal forces across sensor designs
6. OpenTac-Xampling reads out at kilohertz-rates, measuring fast contacts that raster scanning methods miss

4.1 Evaluating auto-generated robot skin designs

We compiled and fabricated 31 boards spanning robot-derived and synthetic outlines, taxel counts, pitches, and electrode gaps. Generation was run three times per design on an Apple M4 Pro laptop; all reported boards completed with

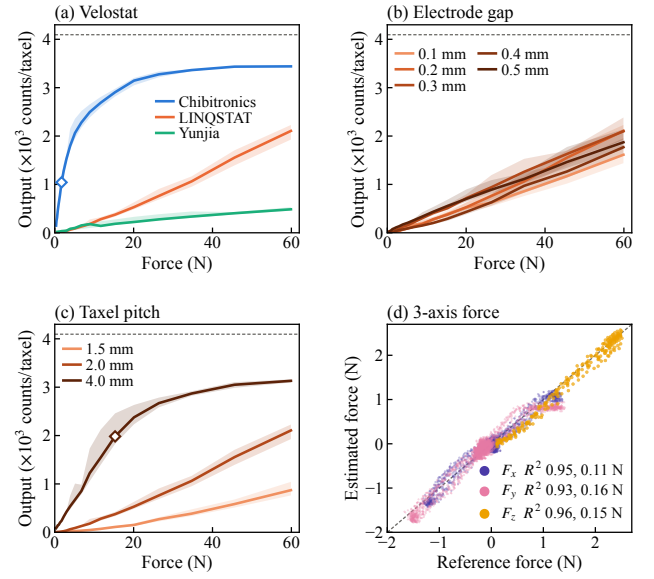


Figure 7. Sensor calibration. (a–c) Per-taxel response to normal loading against a load cell: median and interquartile range per force bin; diamonds mark saturation onset (5% of pressed taxels at full scale); dashed lines mark the 12-bit ADC limit. (a) Three films (2.0 mm pitch, 0.2 mm gap). (b) Electrode gaps of 0.1–0.5 mm (LINQSTAT, 2.0 mm pitch). (c) Pitches of 1.5, 2.0, and 4.0 mm (LINQSTAT, 0.2 mm gap). (d) In-sample three-axis calibration of a binned 4 × 4 patch against an OptoForce reference; the dashed line is $y = x$, and legends give R^2 and RMSE.

zero design-rule violations, and every fabricated board was functional. Figure 3A highlights five embodiments: a YUBI finger pad, UMI soft-gripper finger, LEAP Hand palm, Robotiq 2F-85 finger pad, and Unitree B2 foot. They contain 153–467 taxels at 2–4 mm pitch and require only 35–46 electrical channels (an $R \times C$ matrix needs $R + C$ lines).

To quantify the benefit of shape-aware packing, we compare the active taxel-cell area with the largest rectangular grid of the same pitch that fits inside each target region. OpenTac-CAD covers 75%, 90%, 89%, 87%, and 93% of the YUBI, UMI, LEAP, Robotiq, and B2 regions, respectively (Fig. 3C). The rectangular baseline covers 57%, 90%, 79%, 87%, and 93%. As expected, both methods agree on nearly rectangular regions; the compiler gains most on tapered or irregular ones, improving YUBI and LEAP by 18 and 10 percentage points. Across the five highlighted designs, compilation from outline to fabrication files takes 0.7–2.6 s.

Design generation from natural language. We pair OpenTac-CAD with a VLM agent (Claude Opus 5) to take in a natural language description specifying arbitrary design requirements and constraints. The VLM agent then constructs the sensing outline, optionally web-searching as needed, and selects taxel size, pitch, and board mode, the same inputs a user enters in the interface. This design is then passed to the OpenTac-CAD compiler, which handles all subsequent steps, including electrode and clearance rules, packing, readout-aware wiring, routing, and design-rule checks. The agent

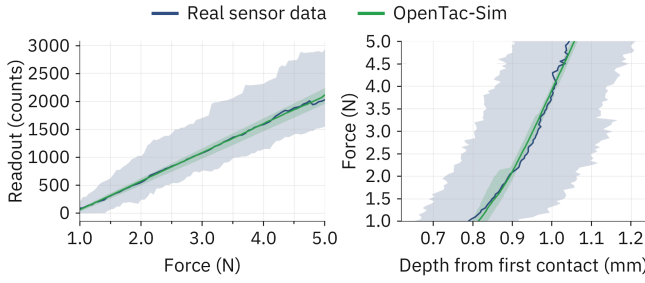


Figure 8. OpenTac-Sim faithfully simulates our shear sensor. We collect data by pressing the bump strip on top of a calibrated sensor with a force-controlled UR5e robot with forces ranging from 1 N to 5 N. We reproduce the forces in simulator and compare alignment between simulation vs real-world force readings and contact depths.

can call the compiler repeatedly, for example to sweep pitch, before selecting a design. Using OpenTac-CAD and web search, all nine agent sessions produced DRC-clean layouts (76–254 taxels, median 6.5 min), each recording dimensional sources and design assumptions. (Fig. 6).

4.2 Shear, material, and geometry Characterization

We conduct multiple studies to characterize the sensor and answer two questions:

1. How do each of the design choices (piezoresistive material, electrode geometry) impact force response?
2. How accurate are the normal and shear force estimates from OpenTac compared to reference 3-axis force sensors?

The experiments all involve pressing a square sensor array pressed against a load cell, while varying one design variable at a time. The results are shown in Fig. 7 a–c.

Choice of piezoresistive film: We compare 3 commercially available piezoresistive films – Chibitronics, LINQSTAT, and Yunjia. Each film exhibits different sensitivity and saturation behavior. At 2.0 mm pitch and 0.2 mm gap, Chibitronics saturates at low force, LINQSTAT does not reach the saturation criterion through 60 N, and Yunjia produces the lowest ADC response over the tested force range.

Electrode geometry: Electrode gap has a modest effect on response, whereas larger pitch increases the per-taxel signal. Taxel pitch therefore affects both spatial resolution and sensitivity. For instance, 2 mm taxel pitch with a Chibitronics piezoresistive has a 60 N force range, but increasing the taxel pitch to 4 mm saturates the sensor earlier (at about 20 N).

Force estimation accuracy: We conduct this experiment with our *default* configuration, which is a 4×4 bump strip array using a LINQSTAT piezoresistive sheet with a 4.0 mm taxel pitch and 0.2 mm electrode gap. The sensor is first calibrated with a reference force measured by the load cell. We then apply a range of reference forces across the X, Y, and Z axes on the bumps and log sensor measurements and reference forces measured by a load cell. We measure correlation between OpenTac and the reference force sensor

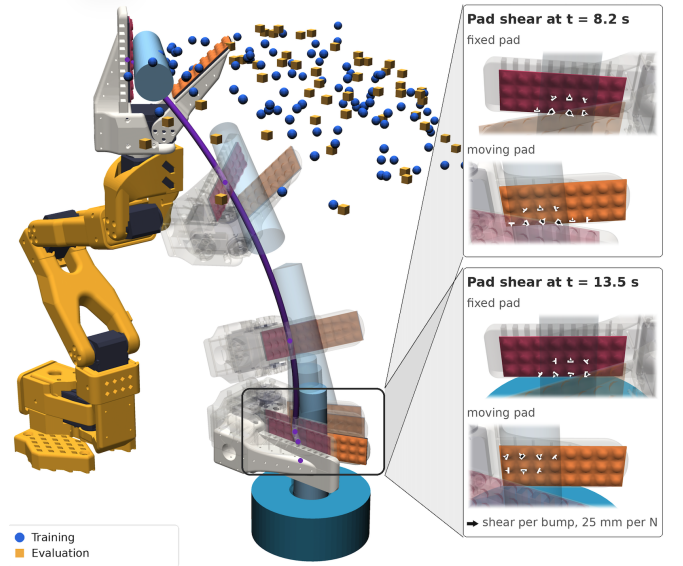


Figure 9. SO-101 peg insertion with deformable tactile pads. Blue spheres and orange cubes mark training ($n = 100$) and held-out evaluation ($n = 50$) start poses, respectively. The purple curve shows an expert trajectory, with translucent gripper configurations at multiple timestamps; the dashed gray curve shows an alternative demonstration. Inset shows close-ups of the pads during insertion. Arrows show per-bump tangential displacement.

by computing R^2 . The estimation method in Eq. 1 achieves R^2 of 0.95, 0.93, and 0.96 and RMSE of 0.11, 0.16, and 0.15 N for F_x , F_y , and F_z against OptoForce (Fig. 7d).

4.3 Shear sensor simulation

To validate whether the simulation faithfully models a real OpenTac shear sensor, we collected sensor measurements by pressing the bumps with a force-controlled UR5e robot end-effector. We vary the forces from 1 N and 5 N in increments of 0.5 N. We then repeat the experiment in simulation, where we press down a bump with exactly the same sequence of forces as we did in the real-world. We calibrate the real-world sensor so we can map the sensor readings to force units. We then compute two correlation plots, shown in Fig. 8. We notice that simulated sensor measurements are extremely well correlated with the mean value of real sensor data. Measurements from real sensor exhibit signal-dependent noise that generally increases with the force value. We also find a strong correlation between the contact depth and force across real-world and OpenTac. We tested the viability of the simulator for manipulation policy learning by training a peg insertion policy in the simulator, and show example qualitative results in Fig. 9.

4.4 Tactile Policy Learning

We evaluate OpenTac for real-world policy learning on two contact-rich manipulation tasks on two robot platforms: an SO-101 robot arm and a bimanual YAM robot.

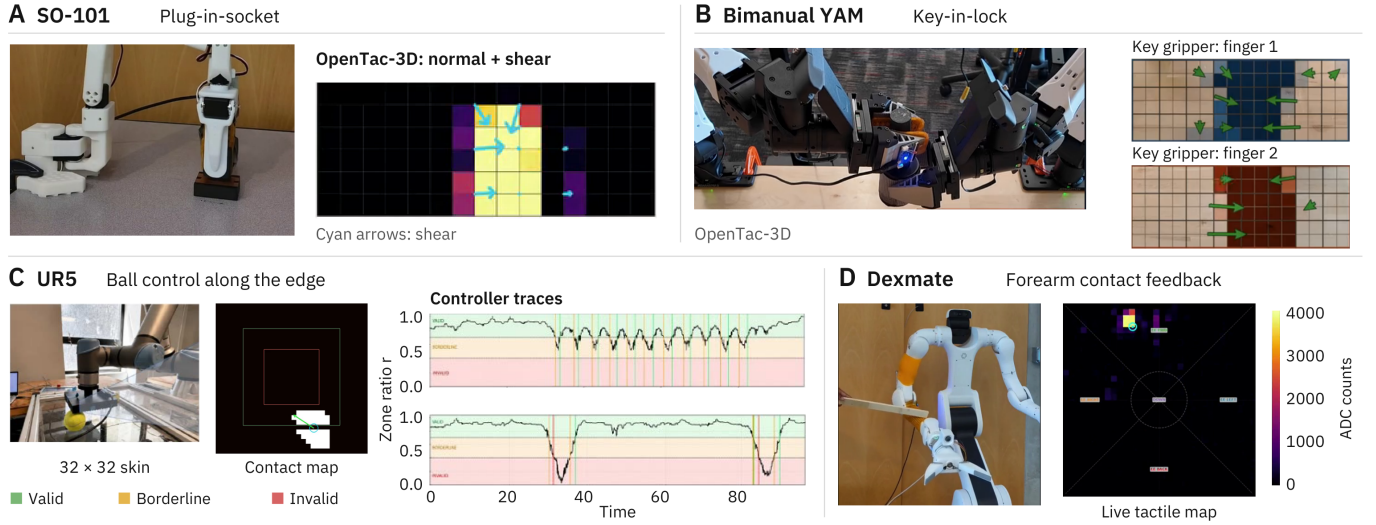


Figure 10. Robot deployments. OpenTac sensors deployed on four physical robot platforms to support policy learning, contact-rich control, and interactive applications. **A.** *Plug-in-socket insertion* on an SO-101 robot arm where shear sensing is central to task success. **B.** *Key-in-lock insertion* using a custom-designed cylindrical locking mechanism. **C.** *Rolling a ball* using a shear-controller on the UR5e robot. **D.** *Interactive impedance control* on the Dexmate Vega-1 Pro robot. Tactile skin on the robot’s forearm senses and moves in the direction of a push.

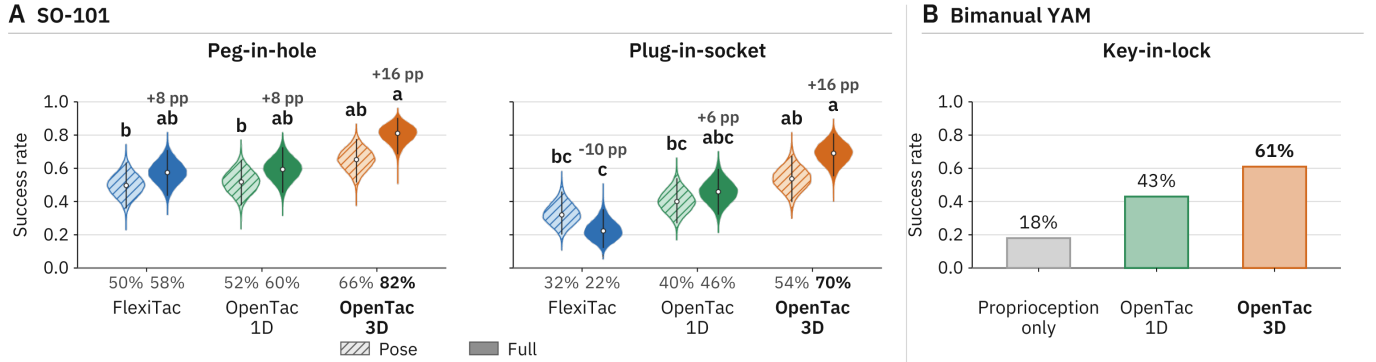


Figure 11. Tactile policy learning: **A.** *Peg-in-hole* and *Plug-in-socket* insertion on an SO-101 robot arm. Success rates are reported over 50 trials for each condition. “Pose” uses joint angles only; “full” adds tactile observations; neither uses vision. OpenTac-1D senses normal pressure; OpenTac-3D adds shear. We evaluate FlexiTac [1] as a baseline normal force sensor. To account for each sensor resulting in slightly different grip friction (e.g., due to bump strips) and demonstration distributions, we train and evaluate a proprioception-only variant for each sensor as a baseline. We use compact letter display (CLD), based on statistical significance testing protocols from [38]. Two violins that do not share a letter are (statistically) significantly different ($> 90\%$ confidence interval). **B.** *Key-in-lock* insertion task on bimanual YAM robot arms. Success rates are reported across 25 trials. Across all tasks, shear-sensing variants (OpenTac-3D) achieve higher success rates.

SO-101 robot arm: On the SO-101 robot arm, we evaluated the *peg-in-hole* from LeFlexiTac [39] and *plug insertion* from TacOBench [40]. In the first, the robot inserts a peg already held in its gripper into a slot. Plug insertion is a harder task and requires sub-centimeter alignment of the electrical contacts with the socket. We compare FlexiTac [1], OpenTac-1D (normal-force only; no bump strip), and OpenTac-3D (normal and shear forces; with the bump strip). The two OpenTac conditions use the same skin and readout board and differ only in the cover. For each sensor, we train an action chunking transformer (ACT) that takes tactile signals as input in addition to joint angles (referred to as “full”) and a variant that uses joint angles only (referred to as “pose”). See Fig. 11. Each policy is trained on 100 demonstrations and evaluated over 50 trials. We also train a per-sensor

proprioception-only baseline to account for any variation in sensor hardware and in demonstrator trajectory distribution. In particular, adding bumps changes contact mechanics as well as sensitivity to shear.

OpenTac-3D-full achieves the highest observed success rate on both tasks. On peg-in-hole, tactile observations improve success from 66% to 82%, compared with 50% to 58% for FlexiTac and 52% to 60% for OpenTac-1D. In plug insertion, OpenTac-3D-full succeeds in 70% of trials against 54% for its pose-only baseline(+16 percentage points). Normal force may not distinguish alignment with the socket from contact against its surrounding surface. Neither normal-only tactile policy improves on its corresponding proprioception-only baseline in these trials. The higher success rate with OpenTac-3D suggests that shear is essential for the plug

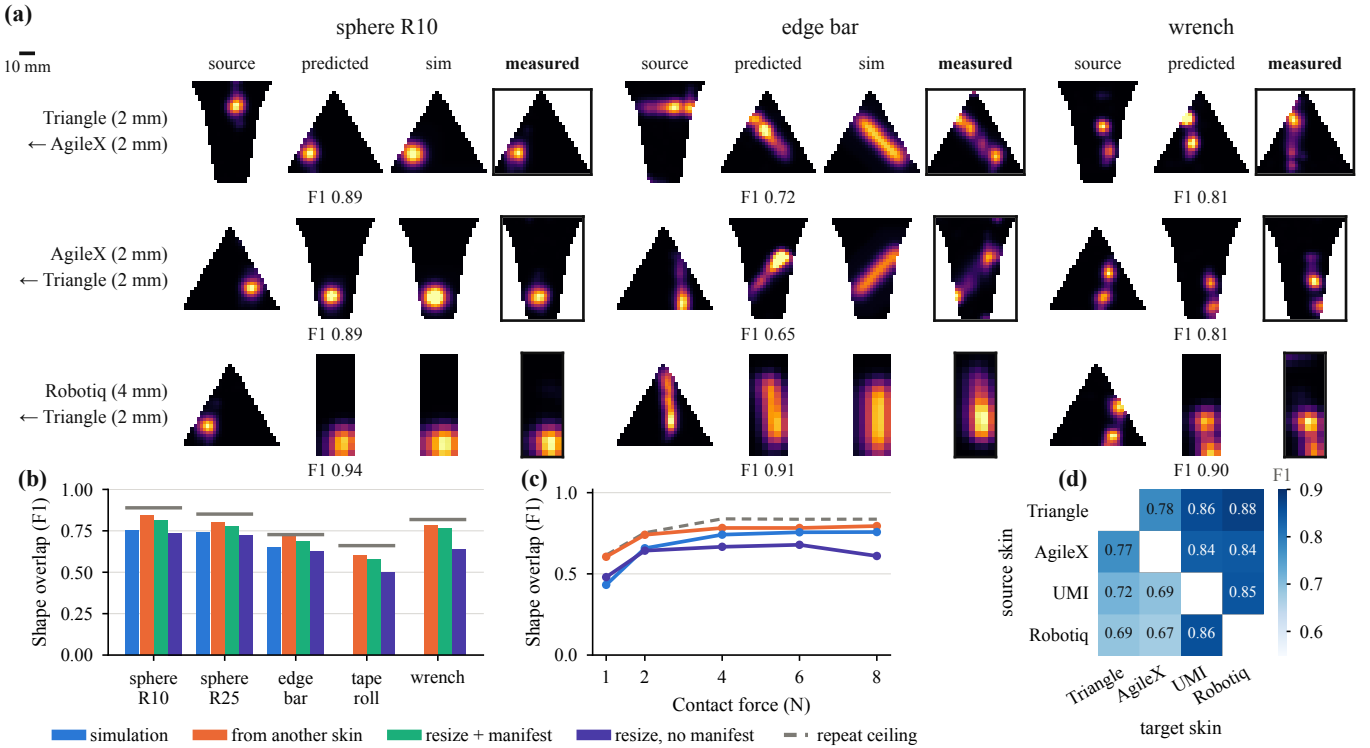


Figure 12. Predicting tactile contact across skins and from a reduced foam model. (a) Median-performing presses at ≥ 4 N; rows are target $\leftarrow$ source at a shared physical scale (bar: 10 mm). Predicted, simulated, and measured panels share the target's peak scale; sources use their own. (b) Median shape F1 by object; grey marks same-sensor repeatability. (c) F1 versus calibrated force. (d) Transfer F1 for each ordered skin pair on co-visible support. All predictors receive contact pose. Simulation covers printed indenters only; Robotiq simulation exemplars are in-sample and excluded from aggregates.

insertion task, as also independently reported in [40]. Additionally, as shown in Fig. 10 (top-left), the normal force maps are nearly identical regardless of whether the plug is in the air or inserted in the socket. It is only the shear estimates that vary significantly and provide cues for task success.

Bimanual YAM robots: We evaluate a *key-in-lock* insertion task on bimanual YAM robot arms. We design a cylindrical locking mechanism where a *key* with an extruded *slot* needs to be twisted and inserted into a *lock*. The key will only fit into the lock at a specific configuration. Like the previous experiments, no vision is assumed available. For this task, one of the arms holds the lock and the other attempts to twist or rotate and insert the key. Success rates over 25 trials are reported in Fig. 11. This is an extremely challenging task for behavioral cloning methods that solely rely on proprioception. It is evident from the low success rates (18%) for the 'proprioception-only' variant. Shear sensing is pivotal here, enabling the detection of whether the key is in the slot, on the edge, or not in contact with the lock at all. The 'OpenTac 3D' variant that uses shear improves success rate to 62%, when compared to the 'OpenTac 1D' variant that uses only normal force and achieves a success rate of 43%.

4.5 Contact-rich control tasks

We evaluate OpenTac on two contact-rich control tasks, comparing shear feedback against normal force alone.

Ball manipulation. A UR5 with a 32×32 OpenTac sensor on its end-effector rolls a ball along the end-effector's outer edges without ball-position feedback (Fig. 10, UR5 panel), regulating toward a target shear vector along the edge. The shear controller recovers from the "Borderline" to the "Valid" region more reliably than normal force alone.

Bimanual grasp stability under impulsive loading. A Dexmate Vega-1 Pro holds a paper cup between two OpenTac probes under proprioceptive, wrist force/torque, normal-pressure, or aggregate-shear feedback, with identical trajectories, gains, and a 2.5 N nominal grip; the shear controller raises this target before slip. Table 2 reports trials that keep the cup for 40 s (*Held*) and net cup displacement (*Slip*).

4.6 Cross-Sensor Transfer

We evaluate four LINQSTAT skins without bumps—Triangle (409 taxels) and AgileX (502) at 2 mm pitch, and UMI (174) and Robotiq (96) at 4 mm—pressing five objects at 95 locations and scoring contacts at 1–8 N using each skin's calibration. Every predictor receives the contact pose. We compare calibrated transfer with raw-count area averaging over manifest geometry ("resize + manifest"), bilinear array resizing ("resize, no manifest"), and the foam model of Sec. 3.3, scoring shape F1 between the predicted and measured active taxels after a one-pitch blur and a threshold at 25% of each field's peak. Transfer reaches a median F1

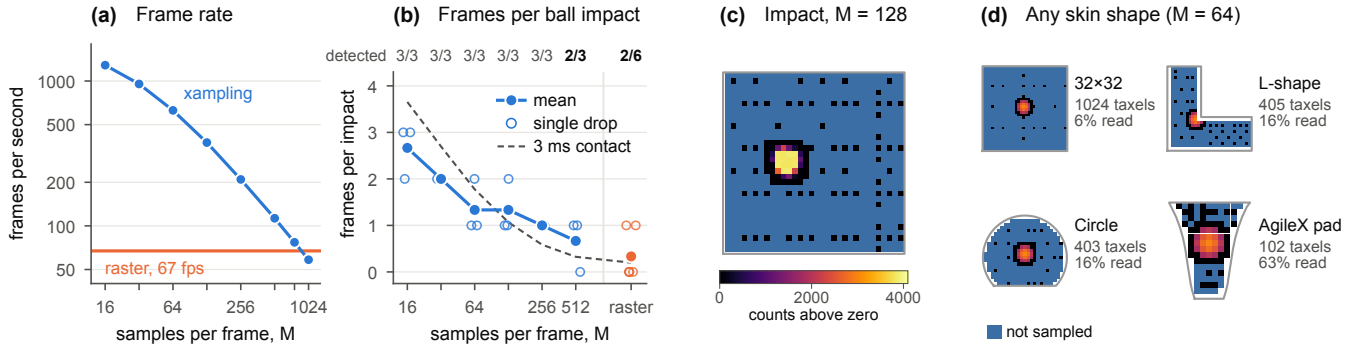


Figure 13. Adaptive partial scanning. (a) Frame rate versus per-frame sampling budget M on a 32×32 skin; full raster scanning reads all 1024 taxels at 67 frames/s. (b) Frames capturing a tennis-ball impact from a 30 cm drop (50% full-scale threshold). Open circles are trials and filled circles means; the dashed curve gives the frames spanned by a 3 ms contact at each frame rate; detection counts appear above. (c) Representative impact at $M = 128$. (d) Firmware sampling patterns at $M = 64$ on four skin shapes (not to scale). Color is ADC counts above baseline; black taxels were sampled without contact, and blue taxels were not sampled.

Table 2. Hold stability on a Dexmate Vega-1 Pro. Twenty trials per condition, Slip is reported as mean $\pm$ std.; Held counts trials reaching 40s. — dropped, so slip is undefined.

Feedback	Held $\uparrow$	Slip (cm) $\downarrow$
No load		
Proprioception only	3	—
F/T sensor	20	0.15 ± 0.08
OpenTac-1D	20	0.2 ± 0.1
OpenTac-3D	20	0.11 ± 0.06
Load		
Proprioception only	0	—
F/T sensor	9	5.5 ± 1.3
OpenTac-1D	9	5.8 ± 2.1
OpenTac-3D	16	2.6 ± 1.2

of 0.750 over 5,290 predictions, versus 0.714 and 0.634 for the two resizing baselines; on support covered by both skins it reaches 0.788, close to the 0.800 same-skin repeatability. Coarser 4-mm targets score higher than 2-mm targets (0.84–0.88 vs. 0.67–0.78; Fig. 12d), and the foam model reaches 0.727 on the three skins excluded from its fit. These results cover pose-conditioned normal-contact shape, not force magnitude or sim-to-real policy transfer.

4.7 High-Speed Adaptive Sampling

On a 32×32 skin, raster scanning reads all 1024 taxels at 67 frames/s, while shrinking the adaptive budget raises the rate monotonically to 1286 frames/s at $M = 16$, a $19.2 \times$ increase (Fig. 13a). To evaluate whether partial scanning resolves short-duration contacts, we drop a tennis ball from 30 cm and count frames above 50% of ADC full scale: raster scanning detects only two of six impacts, each in a single frame, whereas budgets $M = 16$ –256 detect all three tested impacts per condition and the fastest settings record several frames of the roughly 3 ms event (Fig. 13b); at $M = 512$, two of three impacts are detected as partial scanning approaches the raster limit. The same firmware runs

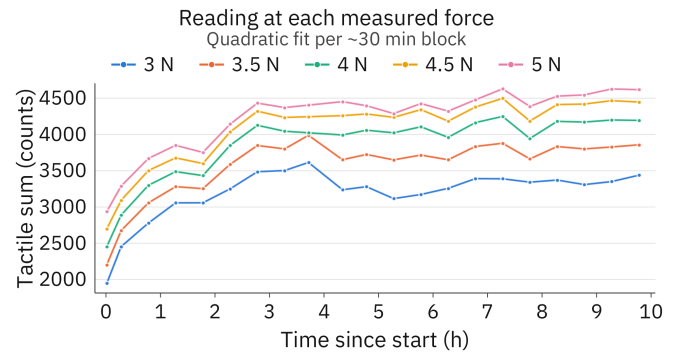


Figure 14. Sensor endurance testing: We continuously indent a bump on a sensor with a robot end effector for 10 hours. Every 30 minutes, we log sensor readings for forces in the range 3 N - 5 N. We find that after the first 2 hours of interaction, the readings stabilize, and maintain a clear relative separation throughout the 10-hour load-cycle test.

unchanged on square, L-shaped, circular, and gripper-pad manifests (Fig. 13d), reading 6% of the 1024-taxel square and 63% of the 102-taxel pad per frame at $M = 64$, without a hand-coded raster or neighborhood map.

4.8 Endurance (duty cycle) testing

We also carried out endurance testing on OpenTac sensors. Using a UR5e robot with an indenter on its end-effector, we constantly press on top of a single bump for 10 hours at a stretch. At every 30-minute interval, we exert forces ranging from 3 N - 5 N (in increments of 0.5 N), and plot the sensor readouts in Fig. 14. We notice a gradual increase in the force response after the first 2 hours of use, upon which the sensor readings stabilize. However, throughout the 10-hour test window, we observe that the force readings are distinctly separable in relative magnitudes. Therefore, either a *warmup* phase for the sensor at the start of a usage session, or using relative sensor measurements as opposed to absolute ADC readings is recommended for consistent results.

5 Conclusion

We have presented OpenTac, an open ecosystem spanning robot surface geometry, fabricated tactile skins, manifest-driven simulation, cross-sensor transfer, and high-speed acquisition. Several limitations remain. The flexible PCBs are not stretchable, and mesh-derived layouts require surfaces that can be flattened into developable patches. Piezoresistive sensing introduces hysteresis, drift, and temperature dependence. Adaptive acquisition trades spatial completeness for update rate; larger contact regions require larger sampling budgets. The simulated signals are deformation proxies, and transfer is evaluated on normal-contact shape rather than force accuracy or sim-to-real policy performance. We release all hardware, firmware, software, and data.

Acknowledgement

The authors acknowledge in-kind support from Dexmate Inc. and Toyota Research Institute for robot equipment used in the project. KMJ, HB, AB, and GH were supported by a DexMate research grant program. AS thanks Analog Devices, Inc. for financial support. Addison Johnson designed the lock-key fixtures used in the bimanual policy learning experiments. Reyhaneh Behraves and Chuhan Zhang participated in early brainstorming meetings. Junjun Chen helped cure silicone for bump designs that were used in early prototypes. Jonathan Yao contributed helpful initial discussions on a broad range of topics involving mechanical fabrication. AS also thanks his postdoctoral advisor, Pulkit Agrawal, for helpful discussions.

References

- [1] Binghao Huang and Yunzhu Li. FlexiTac: A low-cost, open-source, scalable tactile sensing solution for robotic systems. arXiv:2604.28156, 2026.
- [2] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Erciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. AnySkin: Plug-and-play skin sensing for robotic touch. In *ICRA*, 2025.
- [3] Venkatesh Pattabiraman et al. eFlesh: Highly customizable magnetic touch sensing using cut-cell microstructures. arXiv:2506.09994, 2025.
- [4] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, 2019. doi: 10.1038/s41586-019-1234-z.
- [5] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gel-Sight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 2017.
- [6] Mike Lambeta et al. Digitizing touch with an artificial multimodal fingertip. In *arXiv*, 2024.
- [7] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. doi: 10.1109/LRA.2020.2977257.
- [8] Andrea Spos, William van den Bogert, and Nima Fazeli. GelSlim 4.0: Focusing on touch and reproducibility. arXiv:2409.19770, 2024.
- [9] Nathan F. Lepora. Soft biomimetic optical tactile sensing with the TacTip: A review. *IEEE Sens. J.*, 2021.
- [10] Tito Pradhono Tomo, Alexander Schmitz, Wai Keat Wong, Harris Kristanto, Sophon Somlor, Jinsun Hwang, Lorenzo Jamone, and Shigeki Sugano. Covering a robot fingertip with uSkin: A soft electronic skin with distributed 3-Axis force sensitive elements for robot hands. *RA-L*, 2018.
- [11] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. ReSkin: Versatile, replaceable, lasting tactile skins. In *CoRL*, 2022.
- [12] Perla Maiolino, Marco Maggiali, Giorgio Cannata, Giorgio Metta, and Lorenzo Natale. A flexible and robust large scale capacitive tactile system for robots. *IEEE Sens. J.*, 2013.
- [13] Xuguang Sun, Jianhai Sun, Tong Li, Shuaikang Zheng, Chunkai Wang, Wenshuo Tan, Jingong Zhang, Chang Liu, Tianjun Ma, Zhimei Qi, Chunxiu Liu, and Ning Xue. Flexible tactile electronic skin sensor with 3D force detection based on porous CNTs/PDMS nanocomposites. *Nano-Micro Lett.*, 2019.
- [14] Daolin Ma, Elliott Donlon, Siyuan Dong, and Alberto Rodriguez. Dense tactile force estimation using GelSlim and inverse FEM. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424. IEEE, 2019. doi: 10.1109/ICRA.2019.8794113.
- [15] Youcan Yan, Zhe Hu, Zhengbao Yang, Wenzhen Yuan, Chaoyang Song, Jia Pan, and Yajing Shen. Soft magnetic skin for super-resolution tactile sensing with force self-decoupling. *Science Robotics*, 6(51):eabc8801, 2021. doi: 10.1126/scirobotics.abc8801.
- [16] Yoshiyuki Ohmura, Yasuo Kuniyoshi, and Akihiko Nagakubo. Conformable and scalable tactile sensor skin for curved surfaces. In *ICRA*, 2006.
- [17] Philipp Mittendorf and Gordon Cheng. Humanoid multimodal tactile-sensing modules. *IEEE Trans. on Robotics*, 2011.
- [18] Zilin Si, Tianhong Catherine Yu, Katrene Morozov, James McCann, and Wenzhen Yuan. Robotsweater: Scalable, generalizable, and customizable machine-knitted tactile skins for robots. In *ICRA*, 2023.
- [19] Yiyue Luo, Yunzhu Li, Pratyusha Sharma, Wan Shou, Kui Wu, Michael Foshey, Beichen Li, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. Learning human-environment interactions using conformal tactile textiles. *Nature Electronics*, 4(3):193–201, 2021. doi: 10.1038/s41928-021-00558-0.
- [20] Devin Murphy, Yichen Li, Crystal Elaine Owens, Layla Stanton, Paul Pu Liang, Yiyue Luo, Antonio Torralba, and Wojciech Matusik. Fits like a flex-glove: Automatic design of personalized FPCB-based tactile sensing gloves. In *CHI EA*, 2025.
- [21] Andela Ilic, Junpeng Gao, Zhipeng Li, Yijing Jiang, Rachel Schuchert, Manuel Meier, Philipp Herholz, and Christian Holz. Retrofitting existing 3D objects with surface-conforming capacitive sensing. In *SIGGRAPH*, 2026.

- [22] Hakhyun Lim, Jungrak Choi, Chankyu Han, Dabin Kim, Hanbit Jin, Minki Kim, Yunjeong Kim, Jinhyeok Yang, Saerom Seo, Jaehoon Jung, Hunpyo Ju, Chan-Hwa Hong, Dongyoung Lee, Junseong Ahn, and Hye Jin Kim. Scalable in-situ fabrication of multimodal electronic skin for intelligent robotics and interactive systems. *npj Flexible Electronics*, 10: 34, 2026. doi: 10.1038/s41528-026-00538-4. URL <https://www.nature.com/articles/s41528-026-00538-4>.
- [23] 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. *RA-L*, 2022.
- [24] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *RA-L*, 2022.
- [25] 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 *ICLR*, 2024.
- [26] 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. In *CoRL Workshop*, 2024.
- [27] Yuyang Li, Wenxin Du, Chang Yu, Puhao Li, Zihang Zhao, Tengyu Liu, Chenfanfu Jiang, Yixin Zhu, and Siyuan Huang. Taccel: Scaling up vision-based tactile robotics via high-performance GPU simulation. In *NeurIPS*, 2025.
- [28] An Thanh 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. In *RSS*, 2026.
- [29] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. In *CoRL*, 2025.
- [30] Kipp Freud, Yijiong Lin, and Nathan F. Lepora. SimShear: Sim-to-real shear-based tactile servoing. In *CoRL*, 2025.
- [31] Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin CK Tee, and Harold Soh. Event-driven visual-tactile sensing and learning for robots. *RSS*, 2020.
- [32] Niklas Funk, Erik Helmut, Georgia Chalvatzaki, Roberto Calandra, and Jan Peters. Evetac: An event-based optical tactile sensor for robotic manipulation. *IEEE Trans. on Robotics*, 2024.
- [33] Ariel Slepian, Dian Li, Hongjun Cai, Ryan McGovern, Aidan Aug, Sriramana Sankar, Trac Tran, and Nitish Thakor. Adaptive compressive tactile subsampling: Enabling high spatiotemporal resolution in scalable robotic skin. arXiv:2410.13847, 2024.
- [34] Carolina Higuera, Akash Sharma, Chaithanya Krishna Boduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *CoRL*, 2024.
- [35] Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *ICLR*, 2025.
- [36] Yan Zhang, Zheng Wang, Pengpeng Zeng, Xing Xu, Jingkuan Song, and Heng Tao Shen. Cross-tactile sensor representation learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 159122–159137. PMLR, 2026. URL <https://proceedings.mlr.press/v306/zhang26gh.html>.
- [37] Yuanxiang Zhang, Jiantao Zeng, Yong Wang, and Guoquan Jiang. Flexible three-dimensional force tactile sensor based on velostat piezoresistive films. *Micromachines*, 2024.
- [38] TRI LBM Team, Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, Naveen Kuppaswamy, Kuan-Hui Lee, Katherine Liu, Dale McConachie, Ian McMahon, Haruki Nishimura, Calder Phillips-Grafflin, Charles Richter, Paarth Shah, Krishnan Srinivasan, Blake Wulfe, Chen Xu, Mengchao Zhang, Alex Alspach, Maya Angeles, Kushal Arora, Vitor Campagnolo Guizilini, Alejandro Castro, Dian Chen, Ting-Sheng Chu, Sam Creasey, Sean Curtis, Richard Denitto, Emma Dixon, Eric Dusel, Matthew Ferreira, Aimee Goncalves, Grant Gould, Damrong Guoy, Swati Gupta, Xuchen Han, Kyle Hatch, Brendan Hathaway, Allison Henry, Hillel Hochshtein, Phoebe Horgan, Shun Iwase, Donovan Jackson, Siddharth Karamcheti, Sedrick Keh, Joseph Masterjohn, Jean Mercat, Patrick Miller, Paul Mitiguy, Tony Nguyen, Jeremy Nimmer, Yuki Noguchi, Reko Ong, Aykut Onol, Owen Pfannenstiehl, Richard Poyner, Leticia Priebe Mendes Rocha, Gordon Richardson, Christopher Rodriguez, Derick Seale, Michael Sherman, Mariah Smith-Jones, David Tago, Pavel Tokmakov, Matthew Tran, Basile Van Hoorick, Igor Vasiljevic, Sergey Zakharov, Mark Zolotas, Rares Ambrus, Kerri Fetzer-Borelli, Benjamin Burchfiel, Hadas Kress-Gazit, Siyuan Feng, Stacie Ford, and Russ Tedrake. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 2026.
- [39] Naian Tao, Yifan He, Wesley Maa, Binghao Huang, and Yunzhu Li. LeFlexiTac: Giving robots a sense of touch. *Columbia University RoboPIL Blog*, 2026.
- [40] Anya Zorin, Zilin Si, Myungsun Park, Junsung Park, Alexiy Buynitsky, Sachin Bhadang, Taejun Park, Sohee John Yoon, Yong-Lae Park, Oliver Kroemer, Zeynep Temel, Michael T. Tolley, Sha Yi, and Xiaolong Wang. Taco: Benchmarking tactile sensors for object manipulation. *arxiv*, 2026.

Contribution Statements

Ariel Slepian conceived and initiated the OpenTac project and led its overall development. AS developed the core tactile sensing architecture, designed and fabricated the sensor hardware and readout electronics, and developed the acquisition firmware. AS developed OpenTac-CAD, OpenTac-Transfer, and OpenTac-Xampling, and co-developed OpenTac-Sim. AS established the initial working system and coordinated the broader collaborative effort. AS conceived and designed experiments, prepared figures, and led the writing and preparation of the manuscript.

Geeta Chandra Raju Bethala was a core contributor to OpenTac-CAD, alongside AS. GCRB also ran robot experiments on the DexMate Vega Pro and produced several qualitative results for this work.

Jiafeng Gu led the development of OpenTac-Sim and produced multiple quantitative and qualitative results to assess the efficacy and accuracy of OpenTac-Sim.

Anagha Rajendra Dangle coordinated sensor design, placement, accuracy, and ensured experiment design consistency across multiple experiments for this work, ranging from policy learning to characterization to contact-rich control.

Yinzhe Zhou and **Tuhin Basu** led the sensor characterization and calibration experiments. YZ and TB also con-

ducted load-cycle testing experiments. TB additionally led the contact-rich control experiments.

Hongyuan Fu and **Xinyi Yin** led the SO-101 policy learning experiments. XY instrumented sensors, designed/fabricated test fixtures, and HF collected data and trained and evaluated policies. **Md Faizal Karim** led the bimanual manipulation experiments using the YAM robots. **Jade Sim** designed and fabricated a tactile glove using OpenTac sensors.

Homanga Bharadhwaj helped shape and advise the policy learning experiments, participated in the weekly discussions and planning for the project's experiments, and participated in paper writing and designing visuals.

Anand Bhattad led weekly project discussions and shaped the vision for the OpenTac-Sim. AB was also heavily involved in sensor characterization and policy learning experiments, and wrote/edited significant portions of this manuscript.

Krishna Murthy Jatavallabhula was involved in *nearly every* experiment run for the paper. KMJ advised on experiment design, overall framing/scope, and wrote/edited significant portions of this manuscript and release materials.

Gregory Hager and **Nitish V. Thakor** provided research supervision, scientific guidance, and strategic advice throughout the project. They shaped the project's research direction, experimental scope, and overall presentation.