DexSkin: High-Coverage Conformable Robotic Skin for Learning Contact-Rich Manipulation

Suzannah Wistreich[*,1] , **Baiyu Shi**[*,1] , **Stephen Tian**[*,1],
Samuel Clarke[1], **Michael Nath**[1], **Chengyi Xu**[1,2], **Zhenan Bao**[1], **Jiajun Wu**[1]
[1]Stanford University [2]University of Alabama at Birmingham

Abstract: Human skin provides a rich tactile sensing stream, localizing intentional and unintentional contact events over a large and contoured region. Replicating these tactile sensing capabilities for dexterous robotic manipulation systems remains a longstanding challenge. In this work, we take a step towards this goal by introducing DexSkin. DexSkin is a soft, conformable capacitive electronic skin that enables sensitive, localized, and calibratable tactile sensing, and can be tailored to varying geometries. We demonstrate its efficacy for learning downstream robotic manipulation by sensorizing a pair of parallel jaw gripper fingers, providing tactile coverage across almost the entire finger surfaces. We empirically evaluate DexSkin's capabilities in learning challenging manipulation tasks that require sensing coverage across the entire surface of the fingers, such as reorienting objects in hand and wrapping elastic bands around boxes, in a learning-from-demonstration framework. We then show that, critically for data-driven approaches, DexSkin can be calibrated to enable model transfer across sensor instances, and demonstrate its applicability to online reinforcement learning on real robots. Our results highlight DexSkin's suitability and practicality for learning real-world, contact-rich manipulation. Please see our project webpage for videos and visualizations: `https://dex-skin.github.io/`.

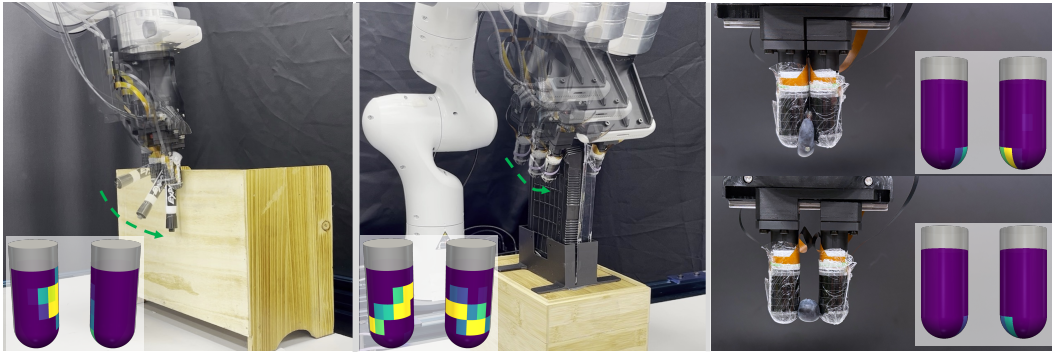**Keywords:** tactile sensing, contact-rich manipulation

Figure 1: **DexSkin sensors enable contact-rich robotic manipulation policies.** We demonstrate the applications of DexSkin in in-hand pen reorientation, box packaging, and berry transporting tasks. In each frame, we visualize tactile readings from DexSkin as heatmaps, where brighter readings correspond to larger forces. DexSkin's high-coverage sensing capabilities enable it to provide contact signals for a variety of robotic manipulation tasks. At the same time, it possesses desirable properties for learning-based manipulation systems.

1 Introduction

Tactile feedback is essential for robust and dexterous manipulation in natural and artificial systems. In humans, mechanoreceptors within the skin provide a rich sensory stream that guides tasks ranging

---

[*]indicates equal contribution

from handling delicate objects to using tools with force [1]. This tactile feedback enables more precise reactive control than can be achieved with human and proprioceptive feedback alone [2].

Emulating this tactile sensitivity in robotic systems has long been a challenge, starting from sensing hardware. Research efforts typically focus on rigid or partially flexible sensors that offer low spatial coverage, limited adaptability, and poor conformability to complex surfaces. In contrast, everyday tasks such as rotating a key in-hand or picking up delicate berries require sensing coverage in multiple contact regions. Similarly, wrapping an elastic band around an object requires a sensor to detect, distinguish, and localize dynamic contact events on all surfaces of a human hand or end effector accurately, applying the appropriate force without allowing the band to slip.

Learning-based approaches offer a general way to harness tactile sensing information. However, data-driven systems pose requirements beyond sensing coverage. Seemingly small distribution shifts in sensor readings from wear or replacement can render previously trained models unusable, thus, output signals must be replicable across sensor hardware instances. Additionally, for real-world online learning, sensors must remain consistent, durable, and precise under repeated stresses during trial-and-error interaction, and output interpretable signals amenable to reward or cost specification.

To address these challenges, we introduce a novel soft tactile skin named **DexSkin** that is particularly suitable for robotic learning applications. DexSkin can be conformably integrated onto robotic end-effectors with unparalleled spatial coverage. It is based on a capacitive mechanism, and features high sensitivity and robustness under repeated interactions. Because each of the dozens of taxels on the skin are individually addressable, DexSkin can localize and characterize simultaneous contacts from distinct regions. It can also be calibrated to provide consistent readings across distinct sensor units, enabling re-use of learned networks. At the same time, it can withstand deformations encountered during typical dexterous tasks such as pinching, twisting, and bending.

In this work, we introduce the DexSkin framework and its fabrication, as well as a representative integration with a soft cylindrical robotic fingertip that sensorizes the distal dome of the finger and 294° of the circumference. Then, we evaluate DexSkin's applicability to robot learning. First, we test whether DexSkin's coverage and tailorability expands the range of learnable manipulation tasks. Then, we evaluate its calibration performance, which are critical for working with and transferring learned tactile models. Finally, we demonstrate its suitability for online robotic learning settings by performing real-world reinforcement learning for a delicate object picking task. The results highlight DexSkin's robust applicability to a wide range of robotic tasks and morphologies, and its particular practicality for robot learning researchers and practitioners.

2 Related Work

Tactile sensing in robotics. Tactile sensors generally fall into three categories: vision-based and optical, magnetic, and electrically addressable.

Vision-based sensors image high-resolution deformations of an elastomer surface using cameras. Various designs have been developed for flat [3, 4, 5, 6] or round and dome-shaped surfaces [7, 8, 9, 10, 11, 12, 13]. Miniaturizing and customizing them has proven challenging [12, 14], and each target form factor requires re-design with extensive optical considerations. Alternatively, DISCO [15] uses photodiodes and LEDs beneath a reflective elastomer, but requires an extensive calibration process to correlate signals with contact locations and magnitudes.

Magnetic-based sensors [16, 17, 18] measure changes in magnetic flux of a magnetized surface due to deformation. Assembling rigid magnetometers onto commercial boards requires lengthy design iterations and is challenging to fit to complex geometries. Further, the sparse point readings struggle to resolve contact imprints or simultaneous contacts, and estimating contact locations and magnitudes demands a data-intensive learning process. In contrast, DexSkin is intrinsically stretchable and conformable to complex geometries. Its individually addressable taxels resolve multiple contacts. Sensors from different batches can be aligned in minutes and calibrating for normal force measurements requires just four loading-unloading cycles per taxel.

Electrically addressable tactile sensors harness resistive, impedance, and capacitive mechanisms. Piezoresistive sensors [19, 20, 21] detect force via pressure-sensitive layers but often have low sensitivity, large hysteresis, and temperature dependence. AC-driven impedance-based sensors (e.g., BioTac [22]) tend to be costly, low-density, and limited in customization. Capacitive sensors [23, 24, 25, 26] measure displacement between electrode plates to resolve force, offering high sensitivity and low power consumption, but often require microfabrication (e.g., photolithography). DexSkin leverages the capacitive mechanism for performance, but can be fabricated on-site via accessible processes, enabling cost-effective same-day customization of sensing pattern and geometry.

Learning robotic manipulation with tactile sensors. Many prior works use tactile sensors toward learning robotic manipulation. Some train in simulation and later transfer to the real world [27, 28, 29, 30]. Since accurately simulating contacts and sensor outputs is difficult, many instead learn directly from real-world data, for instance performing model-free reinforcement learning (RL) [31, 32, 33, 34], learning supervised or self-supervised predictive models [35, 36, 37, 6], or imitating expert demonstrations [38, 34, 39, 40]. Real-world tactile data can be fused with other observation modalities [41] or provide rapid feedback [42]. We operate in this real-world data paradigm. While we do not introduce new learning algorithms in this work, our findings indicate that DexSkin can expand the capabilities and practicality of a variety of learned tactile manipulation systems. Unlike prior works that perform imitation learning that demonstrate prehensile grasping [39], packing [34], and pouring [38] tasks, we focus on tasks that require high-coverage sensing capabilities with minimal blind spots, support for multiple simultaneous contacts, and recovery from perturbation such as in-hand reorientation and manipulating elastic bands. Further, we test the transfer of learned models across DexSkin sensors, a critical practical challenge. Bhirangi et al. [18] narrow inter-sensor gaps with sophisticated manufacturing techniques and hardware design. We instead perform rapid standardization to obtain mappings between sensor taxels, for localized, transferable sensing. In real-world online RL settings, while prior work mostly uses tactile information as policy inputs to optimize task completion objectives [31, 32, 33, 34], DexSkin's readouts are more interpretable compared to optical or magnetic sensors. This allows rewards to be computed directly regarding the contact information itself, for instance, encouraging the gentle grasp of a berry.

3 The DexSkin Framework

In this section, we introduce the main components of the DexSkin framework and compare it to other widely adopted tactile sensors in Table 1. Although DexSkin can be applied to a variety of morphologies, as shown in Appendix Figure 7, we focus on a system that enables a robotic arm to learn tactile-based manipulation policies by sensorizing the fingertip-shaped jaws of a parallel gripper. We first discuss the soft sensor skin itself, its fabrication, then describe its integration with a finger, robotic gripper, and 7-axis arm.

Low-cost, high-performance, conformable DexSkin sensors. Electrically addressable tactile sensors remain rare in robotics [39, 43] due to challenges in sensitivity, hysteresis, cost, and integration. We introduce DexSkin, a soft stretchable capacitive sheet that can be fabricated for under $10 per pair (at 1,000 units) with easily accessible fabrication tools while maintaining high sensitivity, commercial
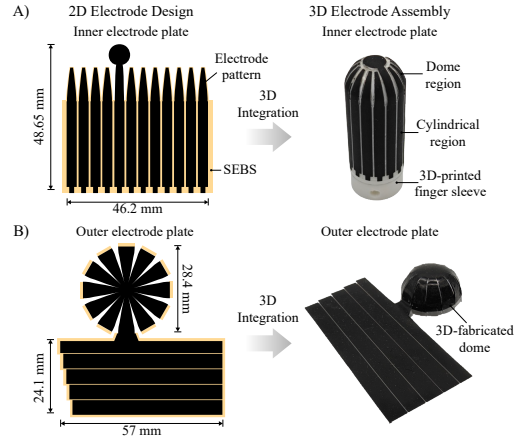

Figure 2: **Designs of electrode patterns.** The 2D pattern design of the inner plate (top left) and outer plate (bottom left) electrodes on the soft SEBS substrate and its appearance after its vertical traces have been conformed onto the dome and cylindrical circumference of the finger sleeve (top right and bottom right). Note that SEBS is transparent and can be hard to distinguish in the post-assembly photo (right).

off-the-shelf cyclic drift and hysteresis performance (TekScan WB201), and a broad pressure sensing range (up to 702kPa). It provides almost full coverage for a robotic finger while offering calibratable and individually addressable taxels that inherently resolve simultaneous contacts.

DexSkin adopts a compliant parallel plate grid configuration where a highly deformable dielectric layer is sandwiched by two soft electrode plates, forming individually addressable capacitive sensing taxels at every electrode intersection. We tailor the sensor to envelop a fingertiplike shape despite being a single continuous piece. It conforms to the curved sides of the sleeve and also to shapes with non-zero Gaussian curvature that challenge traditional sheetlike taxel-based sensors. The hemispherical dome of the finger adopts a flower petal-inspired design as the outer plate, as shown in Figure 2(b). It uses protruding columns, which narrow toward the tip, as the inner plate. When the petals' edges are sealed, it becomes a snugly fit-

Sensor	Area (mm²) ↑	Force res. (N) ↓	Spatial res. (mm) ↓
DexSkin (ours)	1944–6471+	0.086†	≤ 0.60*
ReSkin [17]	400+	< 0.2	1.00
BioTac [22]	484	0.26	1.40
DISCO [15]	6107	< 0.35	≥ 0.7
GelTip [8]	3142	N/A	≥ 3.33
DIGIT [6]	304	0.006	0.150
DIGIT360 [13]	2340	0.001	0.007

Table 1: Comparison of DexSkin framework to state-of-the-art tactile sensors. DexSkin enables the largest sensing area coverage, the finest spatial and force resolution among selected non-vision based systems. †RMSE of force estimation over $0-2.5\,\mathrm{N}$ normal loads. *Estimated from pitch size, machine resolution dependent. More details are in Appendix A.3.

ting hemispherical outer plate that forms 12 taxels with each column underneath. The cylindrical body features 48 taxels, located at the intersections of the black electrode crossbar structure formed by wrapping the row-wise outer plate around the column-wise inner plate (see Figure 2(a)) providing an angular coverage of 294° around the finger.

Same-day tailorable sensor fabrication. The fabrication process begins with computer-aided electrode pattern design, providing full control over taxel layout and size. We incorporate conductive wire traces and capacitive plate patterns into a styrene-ethylene-butylene-styrene (SEBS) substrate, resulting in a 240µm-thick thin film. We seal the edges of the flower-petal outer plate using drips of SEBS solution, to form the hemispherical structure shown in Figure 3(b). We screen-print silver paste adhesive (MG Chemicals 8331) through paper shadow masks to connect sensor traces to FFCs. For the dielectrics, we spincoat the SEBS solution onto a reusable micro-structured PDMS mold (Dow Sylgard™ 184) to produce the 20µm-thick dielectric layer.
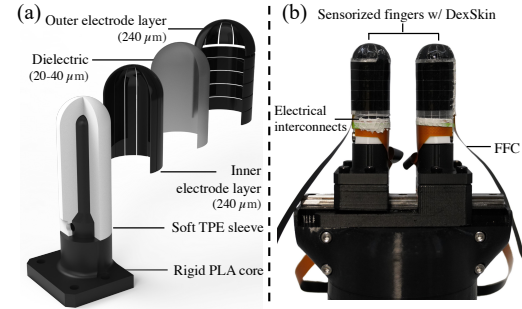

Figure 3: **Robotic gripper finger assembly sensorized with DexSkin.** (a) Exploded section-view diagram of the finger assembly. (b) Two fully assembled sensorized gripper fingers on the SSG-48 gripper used in our experiments.

We assemble the sensor by first attaching the inner plate to the soft sleeve using double-sided tape. We then wrap the dielectric around the inner plate and cap it on top with the hemispherical plate.

We conducted experiments characterizing DexSkin's sensing range, crosstalk, cross-taxel uniformity, robustness, hysteresis, and cyclic stability. Please see Appendix A.3 for the details and results. *We are committed to open-sourcing the detailed fabrication instructions and materials for DexSkin.*

Integrated Robotic Setup. We integrate two DexSkin-sensorized fingers into a tabletop manipulation system with a 7-axis Franka Emika Panda arm, shown in Appendix Figure 15.

While rigid gripper fingers may offer better sensor performance by causing most of the force-induced deformation to occur between its electrode plates, soft materials provide better cushioning for object interactions. Balancing compliance and sensor performance, we design each finger with three layers (Figure 3): the sensor skin, a deformable outer sleeve, and a rigid inner core for structural integrity. We 3D-print the sleeve and inner core using NinjaFlex TPU and PLA respectively.

For precise and reactive finger control, we mount the DexSkin sensors to a Source Robotics SSG-48 gripper. A complete fabricated gripper assembly is shown in Figure 3(b). For tasks that use visual input, we mount a RealSense D415 RGB-D camera to the robot end-effector's wrist.
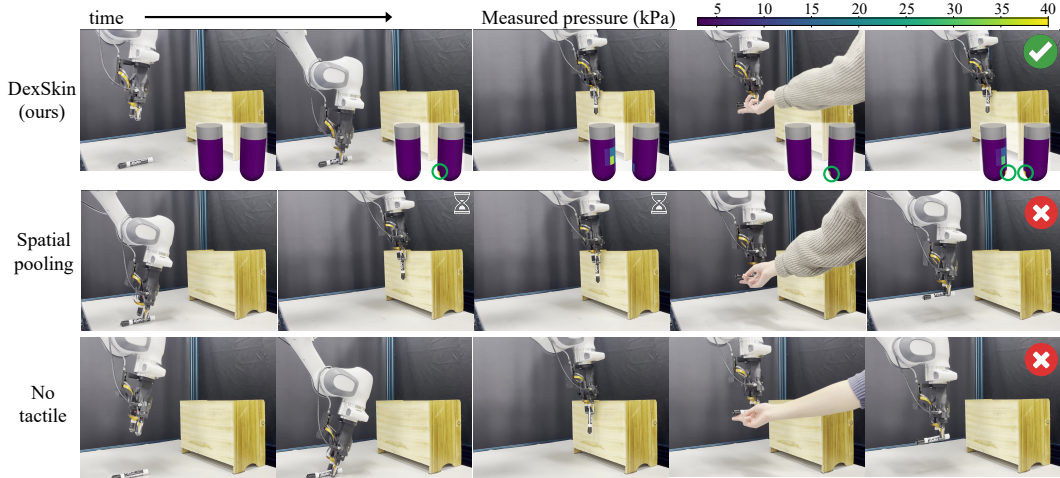
Figure 4: Pen reorientation task rollouts. *(Top)* **The DexSkin policy successfully reorients the pen (cols. 1-3) and reacts to human perturbation (col. 4) by repeating the motion (col. 5).** We visualize DexSkin readings as if facing the robot's front. Note the readings at the sensor's domed tip when the pen is grasped (green circle, col. 2) and the impression when the pen is reoriented to vertical (cols. 3 & 5). *(Middle)* **With spatially pooled tactile information, the robot fails to respond to perturbation and stalls excessively (cols. 2-3)**. *(Bottom)* **The no tactile policy is unable to detect perturbation**. *Best viewed in supplementary video.*

4 Evaluating DexSkin for Learning Robotic Manipulation

We evaluate DexSkin for learning robotic manipulation from the following perspectives:

1. Can DexSkin's coverage and tailorability enable robots to learn a range of manipulation tasks?
2. Can calibrating DexSkin allow learned policies to be transferred across sensor instances?
3. Is DexSkin suitable for applications to learning robot behaviors *online*?

4.1 Learning Manipulation with Expanded Coverage and Tailorability

To test whether the coverage and resolution of DexSkin can expand the manipulation capabilities of learned robotic policies, we consider two representative tasks. For each task, we learn policies via imitation learning from 50 expert human demonstrations by teleoperating the setup from Section 3 using GELLO [44] and training diffusion policies [45, 46]. The tasks are as follows:

- **In-hand pen reorientation.** The robot must pick up the pen and push it against a flat face of a nearby cabinet to reorient it in-hand, using only tactile and proprioceptive feedback. Further, it must be robust to disturbances (e.g. human perturbation) of the pen at inference time. This task requires the robot to use tactile information to estimate the in-hand pose of the pen.
- **Packaging a box with a suitable elastic band.** Humans use the dorsal side of their fingertips to perform everyday tasks like flipping switches or removing batteries from remotes. To evaluate similar capabilities, we task the robot with securing the lid of a container with a rubber band. The robot is randomly provided with an intact band or a perforated band that will snap if used. In the latter case, the robot must retrieve a replacement. This is challenging because the robot must adjust its behavior based on forces exerted by the unknown band on the dorsal sides of the fingers.

Comparative evaluation. To test our hypothesis, we compare the following settings:

- *DexSkin (ours)*: Uses the full 120 taxels of DexSkin readings, proprioception, and wrist camera RGB images. Note wrist camera images are provided for the box task only, not the pen task.
- *No tactile*: A baseline identical to *DexSkin (ours)* but excluding all tactile information.
- Ablated policies trained with spatially aggregated tactile readings for the pen task (*DexSkin (spatial pooling)*), emulating force-sensitive resistors or load cells that cannot detect multiple localized contacts, and a separate ablation using readings from only inner sensor columns between the gripper fingers for the box task (*DexSkin (inner cols. only)*), a proxy for sensors with flat or restricted sensing surfaces. These test the importance of DexSkin's resolution and coverage.
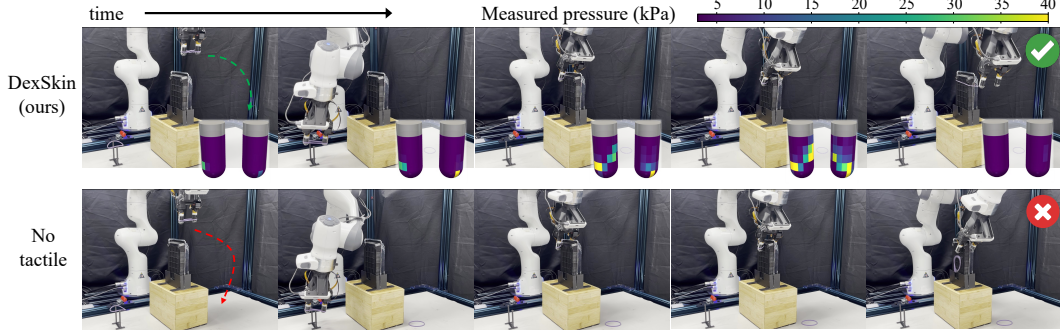
Figure 5: Examples of box packaging task rollouts. *(Top)* **With an initially provided perforated band, the DexSkin policy correctly detects and discards the perforated band based on the tactile reading (col. 1), retrieves the backup (col. 2), and wraps it around the box (cols. 3-5).** We visualize DexSkin sensor readings as if the viewer is facing the robot, looking at the gripper. *(Bottom)* **With an initially provided *un*perforated band, the policy without tactile information unnecessarily replaces the band (cols. 1-2) and then fails to package the box as the band slips off the gripper fingers (cols. 3-5).** *Best viewed in supplementary video.*

- *DIGIT*: Uses a pair of DIGIT [6] sensors as gripper fingers and as input to the policy, comparing our system to existing and commercially available sensors.

Pen task results. We test two settings: the first is the same as in training; in the second, if the robot reorients the pen successfully, a human experimenter perturbs the pen, rotating it within the robot's gripper to be again parallel to the ground. The results are shown in Table 2 and Figure 4. Without perturbation, three out of four policies solve the task 19/20 times. The DexSkin (spatial pooling) policy achieves moderate but poorer performance. When perturbed, the policy with full DexSkin information maintains its success rate, and is the only policy that can recover from perturbation.

Qualitatively, we find that the policy without tactile information repeats the same trajectory regardless of external perturbation, per expectation, making it effective only in the unperturbed setting. While one may anticipate that the policy with spatially pooled tactile information may struggle to estimate in-hand object pose and deal with perturbations, it does not succeed reliably in the unperturbed setting either, likely because it sometimes has insufficient information to determine whether the pen has already been reoriented or not. In its failure cases, this policy often does not fully reorient the pen. For DIGIT, while the sensor provides rich signal within its sensitive region, we find surprisingly that many contacts with the pen are outside this narrow region, even though demonstrations include natural motions with approximately antipodal grasps.

Box packaging results. The results are shown in Table 2 and Figure 5. Only the policy with full DexSkin information (ours) can consistently perform the box packaging task with both non-perforated and perforated initial bands. For the first stage – selecting the correct band for the packaging task – all other policies adopt a singular strategy, either discarding or directly using the initially provided band every time, regardless of its suitability for packaging. Because the bands are difficult to visually distinguish, we hypothesize that each policy commits to an arbitrary trajectory mode from the data. However, the policy with full DexSkin information can successfully select the appropriate band based on the tension perceived by the tactile skin.

Beyond allowing the policy to discern the suitability of a given elastic band, access to full DexSkin information improves the robot's ability to dynamically wrap the band around the box. This can be seen in the fraction of rollouts that succeed in the band selection stage but fail in completing the packaging task by wrapping the band. While only 32% and 70% of rollouts for the *No tactile* and *DexSkin (inner cols. only)* policies complete the second stage given success in the first stage, the policy with complete DexSkin readings does so 86% of the time.

4.2 Calibration and Model Transfer Across Sensor Instances

The physical mechanisms and intricate fabrication procedures of tactile sensors make it challenging to manufacture hardware that produces identical sensing outputs given the same stimuli. For exam-

Model	Pen reorientation		Box: Non-Perforated Band		Box: Perforated Band	
	No perturb	Perturb	Select	Wrap	Select	Wrap
No tactile	**19/20**	0/20	0/20	0/20	**19/20**	6/20
DIGIT [6]	**19/20**	0/20	**20/20***	**20/20***	0/20	0/20
DexSkin (ablated)	12/20	0/20	**19/20**	14/20	1/20	0/20
DexSkin (ours)	**19/20**	**19/20**	18/20	**17/20**	**19/20**	**15/20**

Table 2: Pen reorientation and box packaging success rates. Policies with full DexSkin information achieve the most consistent performance across tasks requiring high sensing coverage. For box packaging, we report success in two cumulative stages for evaluation only: (1) determining the correct elastic band to use and (2) physically wrapping the band around the box. *Placing the band around DIGIT's bulkier geometry elongates the band more, making wrapping it around the box easier.

ple, optical sensor readings vary based on the elastomer interface's texture, while outputs of both magnetic and resistive sensors rely on consistent filler loading and composite thickness [18, 47]. Yet, learned models that ingest tactile information are highly sensitive to the distribution of sensor signals present in the training data. This presents a major practical bottleneck for using tactile sensing in robot learning. Because sensors must eventually be replaced, significant distribution shifts from existing training data often rendering existing models unusable.

In this section, we present a calibration procedure for DexSkin, and evaluate whether it can alleviate these challenges to enable downstream policy transfer to a different sensor hardware instance.

Calibration procedure. We develop two calibration protocols. The first houses DexSkin with a custom 3D-printed airtight PLA chamber with a thin, soft Ecoflex 00-50 inner membrane, and ramps the relative internal pressure linearly from 0psi to 6psi, imposing uniform stress across the entire sensor surface. Fitting each taxel's response to an exponential curve yields both a forward mapping ($\Delta C/C_0 \rightarrow$ pressure) and an inverse mapping that when combined can align new sensor data with a legacy sensor's output. The second involves mounting DexSkin on a motorized linear stage with a force gauge and recording three loading-unloading cycles for each taxel to map sensor outputs to normal force. Details on calibration procedures and performance are in Appendix A.5.

Policy transfer experiment. We revisit the pen reorientation task with perturbation from Section 4.1. We start with a policy trained on 50 demonstrations from one pair of sensors, denoted *Source sensors*, and test its performance when transferred to the same sensors but swapped between left and right fingers (*Swapped*) and transferred to replacement sensors fabricated in a different batch (*Replaced*). We test policy performance with and without calibrating sensor outputs.

The results are presented in Table 3. DexSkin exhibits reasonable policy transfer to target sensor configurations even without explicit calibration, and achieves an additional performance improvement after a short calibration procedure. In contrast, a policy trained on DIGIT sensors cannot transfer at all to a setting where gels have been swapped from their assignments during data collection, due to changes in visual appearance of

Sensor configuration	Stage 1	Stage 2
DexSkin		
Source sensors	20/20	20/20
Swapped (no calib.)	17/20	12/20
Swapped (calib.)	18/20	16/20
Replaced (no calib.)	13/20	5/20
Replaced (calib.)	18/20	14/20
DIGIT		
Source sensors	20/20	0/20
Swapped	0/20	0/20
Swapped (diff. img)	20/20	0/20
Replaced (diff. img)	15/20	0/20

Table 3: Pen reorientation policy performance when transferred across sensor hardware. We report successes across two stages: (1) successfully reorienting the pen the first time and (2) detecting and fixing human perturbation.

the tactile readings. Performance improves if the policy ingests difference images relative to contactless readings, but still underperforms DexSkin's calibration pipeline for replaced sensors.

4.3 Real-World Online Robot Learning with DexSkin

Beyond applications in imitation learning, tactile sensing is a promising source of signals for robots that learn by interacting with their environments, for instance via reinforcement learning (RL). How-

ever, many tactile sensors lack the durability for repeated trial-and-error rollouts in the real world, and further, it is often unclear how to define reward functions based on sensor outputs from optical or magnetic mechanisms without collecting extensive datasets for supervision or calibration.

In this section, we validate DexSkin's suitability for online learning applications, by studying the task of delicate object grasping via real-world residual reinforcement learning. Specifically, the robot's task is to grasp and transport a fragile blueberry without damaging it. This is challenging: episodes are over 200 timesteps long, and a single overly forceful action can crush the delicate berry.

Given a policy trained via imitation on data from a gripper without a tactile sensor, we sensorize the gripper with DexSkin, and then learn a neural network residual policy that parameterizes a modification of the frozen base policy's output. The reward function penalizes tactile sensor readings above a set threshold, representing undesirably strong contacts, ℓ_2 action magnitude, and failed grasps or object transports. For real-world sample efficiency, we use soft actor-critic [48, 49] to learn the residual policy. We do not advocate for a particular RL training procedure, but rather aim to study the capabilities of DexSkin in such frameworks. Further experimental details are in Appendix B.2.

Residual RL w/ DexSkin rewards (ours)   **Random residual policy**   **Base imitation learning policy**
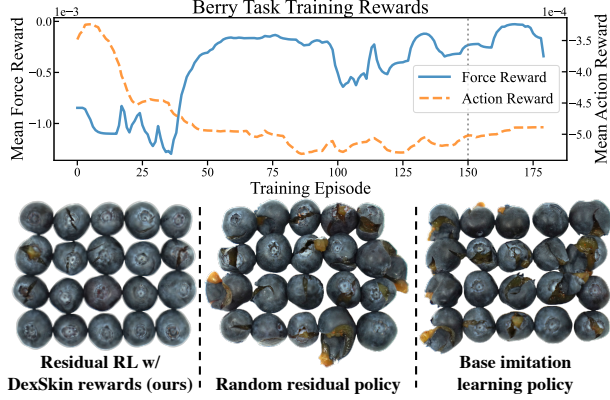
Figure 6: (Top) Residual policy learning curves. Over the course of training, the policy learns to minimize DexSkin-output forces that exceed the specified threshold (maximizing *force reward*), and trade off the action norm cost of larger residual actions (decreasing *action reward*). We train with a faux berry until episode 150, and real berries for the last 30 episodes. (Bottom) Berries after grasp and transport by each policy. The finetuned residual policy causes significantly less damage.

Figure 6 shows force and action rewards throughout the learning procedure, indicating that the residual policy learns to adjust the base policy's actions to produce gentler grasping behavior. Qualitatively, the base imitation learning policy (without tactile information) causes the robot to grasp the berries firmly and crush them – likely as haptic feedback is absent for demo collection. With the combined residual and base policy, the robot can grasp berries delicately enough to avoid breaking them. This is supported quantitatively by our experimental evaluation, shown in Table 4.

This study illustrates how DexSkin can not only be used to sensorize and adapt policies for agents trained initially without tactile information, but also that it enables sensible reward definitions without learned classifiers or calibration.

	Avg. Pressure ↓ (Artificial)	Avg. Pressure ↓ (Real)	Intact (Real) ↑
Base IL (no tac.)	14.5kPa	3.36kPa	20%
Random resid.	6.17kPa	3.64kPa	10%
Resid. RL (ours)	**1.53kPa**	**1.92kPa**	**60%**

Table 4: Evaluation of the final residual RL policy compared to base IL and uniformly random residual policy baselines. We report mean normal pressure applied per DexSkin taxel across time over evaluation trajectories on the artificial berry used for training and real blueberries, and the percentage of **fully intact** real blueberries after transport for each policy. We perform 20 evaluation trials per policy.

5 Conclusion

In this paper, we introduced the DexSkin framework, beginning with our novel elastic tactile sensor that conforms to different geometries, offering repeatable, localized, and high-coverage sensing. We equip a robotic gripper with DexSkin and investigate its suitability for robotic learning applications. Through our experiments, we find that DexSkin can enable imitation learning of improved policies for contact-rich manipulation tasks requiring high-coverage and localized sensing, enable learned model transfer across hardware instances via calibration, and provide feedback for real-world online reinforcement learning. We hope DexSkin is a step toward tactile sensors that can rival human skin in coverage and sensitivity, while remaining practical and suitable for data-driven systems.

6 Limitations

In this work, we instantiate DexSkin on a pair of parallel jaw gripper fingers and conduct all experiments on this platform, which may not reflect its performance for different robot morphologies. Due to DexSkin's customizability, sensorizing robots with morphologies like dexterous hands and covering even non-end-effector regions is an exciting and realizable next step.

Additionally, during policy learning, we treat DexSkin readings naïvely as a one-dimensional feature vector. Future work may investigate how policy architectures can instead take advantage of spatial correlations in the DexSkin data to learn more efficiently and robustly. On the sensor development, despite significant improvement on coverage, a blind spot of $66°$ measured in angular coverage remains. Furthermore, an electrical limitation exists, as ensuring a common ground is critical to maintaining signal integrity, particularly when the sensors are installed onto unshielded, non-metal grippers. The current open-sourced PCB design accommodates this with external jumper wires, which represents a less robust solution. To overcome this limitation, dedicated sensor shielding layers will be implemented in future iterations.

Acknowledgments

References

[1] R. S. Johansson and J. R. Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 2009.

[2] V. G. Macefield and R. S. Johansson. Control of grip force during restraint of an object held between finger and thumb: responses of muscle and joint afferents from the digits. *Experimental brain research*, 1996.

[3] M. K. Johnson, F. Cole, A. Raj, and E. H. Adelson. Microgeometry capture using an elastomeric sensor. *ACM Transactions on Graphics (TOG)*, 2011.

[4] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[5] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez. GelSlim: A High-Resolution, Compact, Robust, and Calibrated Tactile-sensing Finger. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[6] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. 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 (RA-L)*, 2020.

[7] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *IEEE International Conference on Soft Robotics (RoboSoft)*, 2019.

[8] D. F. Gomes, Z. Lin, and S. Luo. Geltip: A finger-shaped optical tactile sensor for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[9] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake. Soft-bubble grippers for robust and perceptive manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[10] W. K. Do and M. Kennedy. Densetact: Optical tactile sensor for dense shape reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[11] W. K. Do, B. Jurewicz, and M. Kennedy. Densetact 2.0: Optical tactile sensor for shape and force reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[12] W. K. Do, A. K. Dhawan, M. Kitzmann, and M. Kennedy. Densetact-mini: An optical tactile sensor for grasping multi-scale objects from flat surfaces. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[13] M. Lambeta, T. Wu, A. Sengul, V. R. Most, N. Black, K. Sawyer, R. Mercado, H. Qi, A. Sohn, B. Taylor, et al. Digitizing touch with an artificial multimodal fingertip. *arXiv:2411.02479*, 2024.

[14] J. Di, Z. Dugonjic, W. Fu, T. Wu, R. Mercado, K. Sawyer, V. R. Most, G. Kammerer, S. Speidel, R. E. Fan, et al. Using fiber optic bundles to miniaturize vision-based tactile sensors. *IEEE Transactions on Robotics*, 2024.

[15] P. Piacenza, K. Behrman, B. Schifferer, I. Kymissis, and M. Ciocarlie. A sensorized multicurved robot finger with data-driven touch sensing via overlapping light signals. *IEEE/ASME Transactions on Mechatronics*, 2020.

[16] T. P. Tomo, M. Regoli, A. Schmitz, L. Natale, H. Kristanto, S. Somlor, L. Jamone, G. Metta, and S. Sugano. A new silicone structure for uSkin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot iCub. *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[17] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *Conference on Robot Learning (CoRL)*, 2021.

[18] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch. *arXiv:2409.08276*, 2024.

[19] Z. Ding, Y.-Y. Tsai, W. W. Lee, and B. Huang. Sim-to-real transfer for robotic manipulation with tactile sensory. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[20] Y. Luo, Y. Li, P. Sharma, W. Shou, K. Wu, M. Foshey, B. Li, T. Palacios, A. Torralba, and W. Matusik. Learning human–environment interactions using conformal tactile textiles. *Nature Electronics*, 2021.

[21] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, and W. Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 2019.

[22] R. S. J. Nicholas Wettels, Veronica J. Santos and G. E. Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 2008.

[23] N. Bai, L. Wang, Q. Wang, J. Deng, Y. Wang, P. Lu, J. Huang, G. Li, Y. Zhang, J. Yang, K. Xie, X. Zhao, and C. Guo. Graded intrafillable architecture-based iontronic pressure sensor with ultra-broad-range high sensitivity. *Nature Communications*, 2020.

[24] Z. Luo, J. Chen, Z. Zhu, L. Li, Y. Su, W. Tang, O. M. Omisore, L. Wang, and H. Li. High-resolution and high-sensitivity flexible capacitive pressure sensors enhanced by a transferable electrode array and a micropillar–pvdf film. *ACS Applied Materials & Interfaces*, 2021.

[25] J. Yang, S. Luo, X. Zhou, J. Li, J. Fu, W. Yang, and D. Wei. Flexible, tunable, and ultrasensitive capacitive pressure sensor with microconformal graphene electrodes. *ACS Applied Materials & Interfaces*, 2019.

[26] Y.-C. Huang, Y. Liu, M. Chao, H.-C. Cheng, Q. He, H. Wu, C. Wang, C.-Y. Lin, Y. Huang, and X. Duan. Sensitive pressure sensors based on conductive microstructured air-gap gates and two-dimensional semiconductor transistors. *Nature Electronics*, 2020.

[27] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General In-hand Object Rotation with Vision and Touch. In *Conference on Robot Learning (CoRL)*, 2023.

[28] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, J. Ortiz, and M. Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. *Science Robotics*, 2024.

[29] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang. Robot Synesthesia: In-Hand Manipulation with Visuotactile Sensing. *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[30] J. Xu, Y. Jia, D. Yang, P. Meng, X. Zhu, Z. Guo, S. Song, and M. Ciocarlie. Tactile-based object retrieval from granular media. *arXiv:2402.04536*, 2023.

[31] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[32] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[33] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[34] K. Yu, Y. Han, Q. Wang, V. Saxena, D. Xu, and Y. Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[35] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In *Conference on Robot Learning (CoRL)*, 2017.

[36] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More Than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch. *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[37] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine. Manipulation by feel: Touch-based control with deep predictive models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[38] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, Hear, and Feel: Smart Sensory Fusion for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[39] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. *Conference on Robot Learning (CoRL)*, 2024.

[40] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv:2404.16823*, 2024.

[41] J. Zhao, N. Kuppuswamy, S. Feng, B. Burchfiel, and E. Adelson. Polytouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[42] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Robotics: Science and Systems (RSS)*, 2025.

[43] S. H. Huang, M. Zambelli, J. Kay, M. F. Martins, Y. Tassa, P. M. Pilarski, and R. Hadsell. Learning gentle object manipulation with curiosity-driven deep reinforcement learning. *arXiv:1903.08542*, 2019.

[44] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[45] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[46] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[47] S. Stassi, V. Cauda, G. Canavese, and C. F. Pirri. Flexible tactile sensing based on piezoresistive composites: A review. *Sensors*, 2014.

[48] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

[49] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 2021.

[50] W. Yuan, S. Dong, and E. H. Adelson. GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force. *Sensors*, 2017.

In addition to the following information, please see the supplementary video and project website (https://dex-skin.github.io/) for a visual walkthrough of the framework hardware, a demonstration of DexSkin's tactile sensing capabilities, and example videos of rollouts from each policy on each experiment.

A   Additional Details on DexSkin Framework

A.1   Additional Sensorized Morphologies

Unlike other sensors limited by the form factors of commercial sensing elements, DexSkin allows for complete customization of the size, distribution, and layout of taxels. Its thin-sheet form also seamlessly integrates with existing designs.

To showcase DexSkin's tailorability, Figure 7 shows applications of DexSkin to additional morphologies: a LEAP robotic hand and a rigid parallel jaw gripper with planar grasping surfaces. For the LEAP hand, we sensorize the palm and thumb links. We omit the visualization for the thumb links here because the grasp configuration produced unclear contacts in that region. For the rigid gripper, we sensorize a $13.8\,\mathrm{mm} \times 22.3\,\mathrm{mm}$ area on each fingertip.
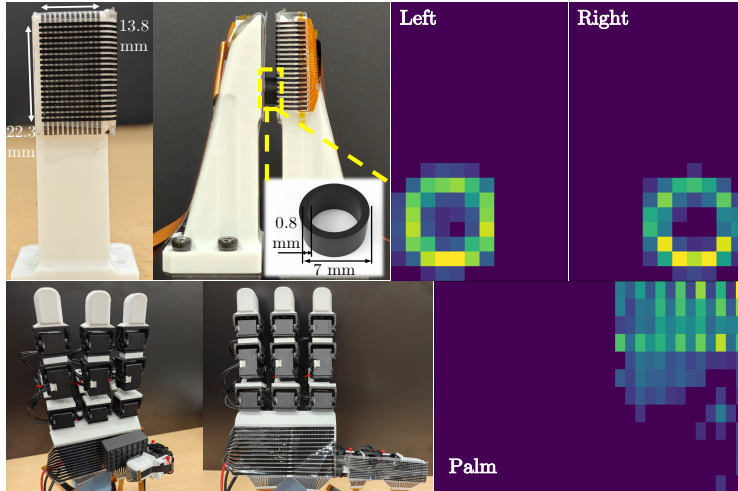

Figure 7: DexSkin application and example readings for a (**top**) flat parallel jaw gripper (228 taxels each) and (**bottom**) LEAP hand (372 taxels each for palm and thumb). For the LEAP hand, we visualize the palm taxels when holding the studded rectangular plastic block shown in the bottom left.

A.2   Sensor Characterization Setup

We used a custom-built programmable pressure station to characterize the force-induced responses of our DexSkin under repetitive loading and unloading cycles. The pressure station features a motorized vertical stage (Newmark Systems, NVS-12), a force gauge (Mark-10, M5-10), and either a precision LCR meter (Agilent E4980A) or our customized PCB for readout. DexSkin is mounted on a 3D-printed rigid PLA core fixed to the stage with tape.

Depending on the taxel to be tested, DexSkin is either laid flat to orient one column of the cylindrical taxels normally upwards, or tilted to position a single dome taxel oriented normally upwards. The force gauge is fixed, while the stage moves in constant steps of $0.01\,\mathrm{mm}$ to gradually increase and decrease the normal force applied as programmed. Cured 10:1 PDMS applicators (Dow Sylgard™ 184) matching the sizes of the tested taxels are placed between the force gauge probe and the surface of DexSkin. They compensate for the rigidity and non-curving nature of the force probe, enabling conformal contacts with soft DexSkin.

Measurements from the force gauge and the LCR meter are automatically synchronized with a custom LabView script at logging time. Measurements from the PCB are synchronized with the force gauge by aligning key timestamps.

A.3 Sensor Characterization Experiments

To measure the physical characteristics of DexSkin, we conducted key experiments to showcase its cross-taxel uniformity, low hysteresis, robustness, cyclic stability, broad sensing range, and minimal crosstalk. While we report the forces applied in the following sections, we would like to note that they are in terms of normal force per taxel and cannot be directly compared with other setups without accounting for differences in the size of force applicators. Therefore, in the following discussions, we normalize the forces to report normal pressure loads.

A.3.1 DexSkin's Cross-taxel Uniformity, Hysteresis and Cyclic Stability

We evaluated the sensor's cross-taxel uniformity by randomly selecting ten taxels (five on the dome and five on the cylindrical body) and measuring their responses subject to normal pressure up to $702.1\,\text{kPa}$ ($11.12\,\text{N}$), as visualized in Figure 8(a). All ten taxels, though designed with different sizes, exhibit consistent logarithmic force-capacitance (normalized) responses in the entire sensing range. Additionally, the hysteresis performance, indicated by the maximum height of the gap between the loading and unloading curves, averages $6.52\% \pm 1.58\%$ across ten selected taxels, is significantly lower than the 17.8% reported for the STAG glove [21] and comparable to commercial flexible force sensors (Tekscan FlexiForce WB201).

We then evaluated the long-term cyclic stability and robustness of DexSkin by subjecting a single taxel from the sensor dome to 500 cyclic loading and unloading cycles of up to $702.1\,\text{kPa}$ ($11.12\,\text{N}$) normal loading, over the course of eight hours. As depicted in Figure 8(b), the smooth and minimally tilted response over 500 cycles shown at the bottom indicates minimal drift over time, and quantitative analysis reveals a peak drift of 2.09% and a zero drift of 1.72%. Zoomed-in views of the beginning and ending cycles further confirm consistent loading and unloading behaviors. This level of consistency and stability of DexSkin, attainable without the need for individual calibration or drift compensation, enables us to directly input sensor responses into our policy learning pipeline.
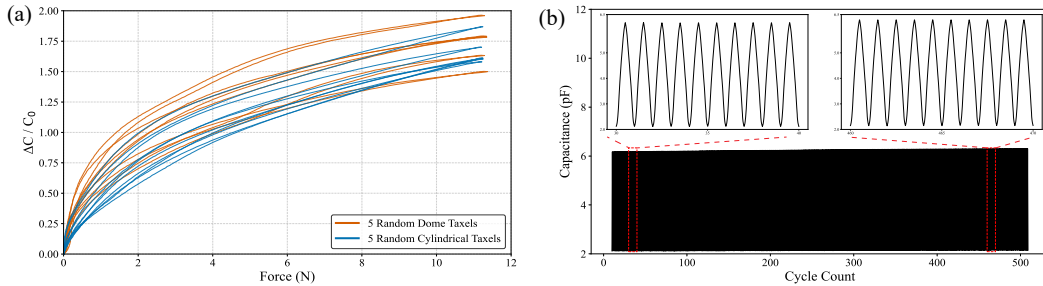

Figure 8: Characterization of the cross-taxel uniformity and cyclic stability of DexSkin. (a) **Normalized capacitance change $\Delta C/C_0$ versus applied force for 10 randomly selected taxels** measured during loading and unloading up to $702.1\,\text{kPa}$ ($11.12\,\text{N}$). Note the small hysteresis and high consistency across both dome and cylindrical taxels. (b) **Single-taxel cyclic loading up to** $702.1\,\text{kPa}$ ($11.12\,\text{N}$) **for 500 cycles.** The red rectangles (top) highlight 10 cycles (30th. to 40th., 460th. to 470th.) near the beginning and the end of the test, with zoomed-in view of consistent loading and unloading behavior for each highlighted segment. The black rectangle (bottom) depicts the taxel's changes in capacitance across all 500 cycles, with smooth and minimally-tilted top and bottom edges. This indicates minimal drift in both the maximum and the baseline responses.

A.3.2 DexSkin's Broad Sensing Range

To support diverse use cases, we evaluated our DexSkin under two sleeve configurations: a soft low-infill TPE sleeve for compliant, low-load tasks and a more rigid sleeve with 100% TPE infill for high-

load applications. We conducted cyclic loading tests using our custom-designed PCB for readout, in the same configuration as deployed on the real robotic system, to characterize each configuration's response to normal pressure loads. We used the vertical-stage based sensor characterization setup discussed in Section A.2 to apply cyclic normal pressure loadings on a randomly selected taxel.

As shown in Figure 11, both configurations demonstrate stable and repeatable responses across three loading-unloading cycles, aligning well with the stability observed using an LCR meter. The soft sleeve variant of DexSkin responds reliably to normal pressure loads as small as $1.7\,\mathrm{kPa}$ ($27\,\mathrm{mN}$) and up to $702.1\,\mathrm{kPa}$ ($11.12\,\mathrm{N}$), beyond which the sleeve structure yields to provide cushioning. The loading range of soft sleeve DexSkin, characterized up to $702.1\,\mathrm{kPa}$, exceed prior sensors such as Gelsight ($\sim 200\,\mathrm{kPa}$) [50] and DIGIT Pinki ($\sim 80\,\mathrm{kPa}$) [14]. The variant of DexSkin with a rigid sleeve, in contrast, withstands higher loads without significant deformations and stably responds to normal pressure loads beyond those achieved by the soft sleeve counterpart, up to $2527.4\,\mathrm{kPa}$ ($40.03\,\mathrm{N}$).
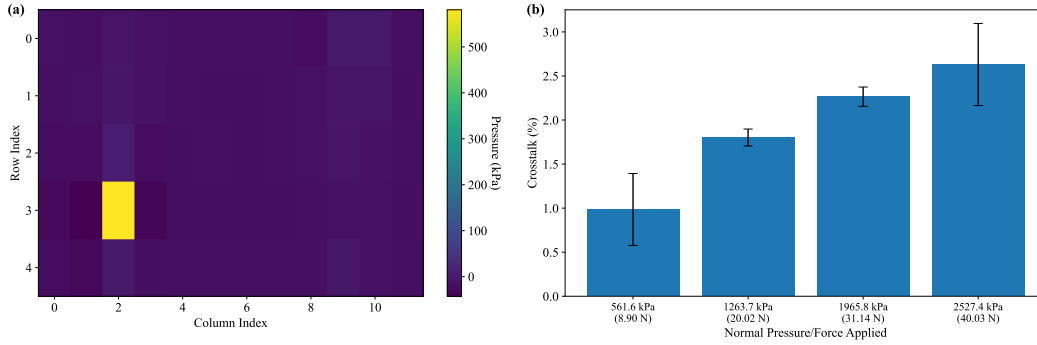

Figure 9: Evaluation of DexSkin's crosstalk under a localized normal load. (a) Heatmap of the sensed normal pressure across DexSkin's full 60 taxels when a single taxel is subjected to a localized normal load measuring $561.6\,\mathrm{kPa}$ ($8.90\,\mathrm{N}$). (b) Quantitative evaluation of the maximum crosstalk (in percentage) observed across the remaining 59 taxels with four different normal pressures applied on the loaded taxel. The average crosstalk remains below 3% at the four sampled loads, demonstrating minimal interference and couplings between closely-positioned taxels.

A.3.3 DexSkin's Minimal Crosstalk

We evaluated the level of crosstalk using the same vertical-stage based sensor characterization setup as described in Section A.2 to apply normal loads ranging from $280.8\,\mathrm{kPa}$ ($4.45\,\mathrm{N}$) to $2808.1\,\mathrm{kPa}$ ($44.48\,\mathrm{N}$) on a randomly selected taxel in 100% infill TPE sleeve DexSkin. We read out the sensor outputs using our custom PCB in normal pressure applied. We were able to obtain 1435 force-sensor value pairs for crosstalk evaluations.

Figure 9(a) shows a representative heatmap of the sensed normal pressure when a single taxel encounters a normal load of $561.6\,\mathrm{kPa}$ ($8.90\,\mathrm{N}$). The visualized response is highly localized and exhibits minimal activations in the surrounding taxels.

To quantify crosstalk, we define the crosstalk (in %) as the ratio between the highest pressure sensed by any of the 59 unloaded taxels and the pressure sensed by the loaded taxel:

$$\mathrm{Crosstalk}\ (\%) = \left(\frac{\max\limits_{j \neq i} P_j}{P_i} \right) \times 100\%$$

where P_i denotes the pressure sensed by the loaded taxel. This formulation of crosstalk reflects the magnitude of ghost contacts at regions experiencing zero normal loads. As seen in Figure 9(b), the average crosstalk increases with the magnitude of normal loads but remains below 3% for all four loading conditions shown. Expanding the analysis to all 1435 pairs acquired, we obtain a mean crosstalk of $1.48\% \pm 1.07\%$. This highlights the minimal amount of interference and couplings and
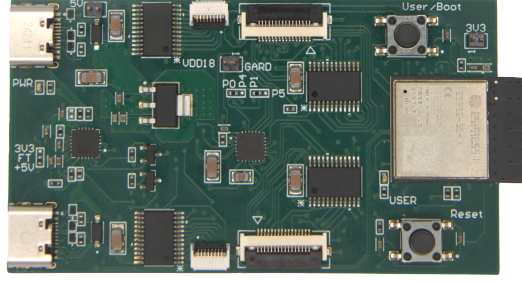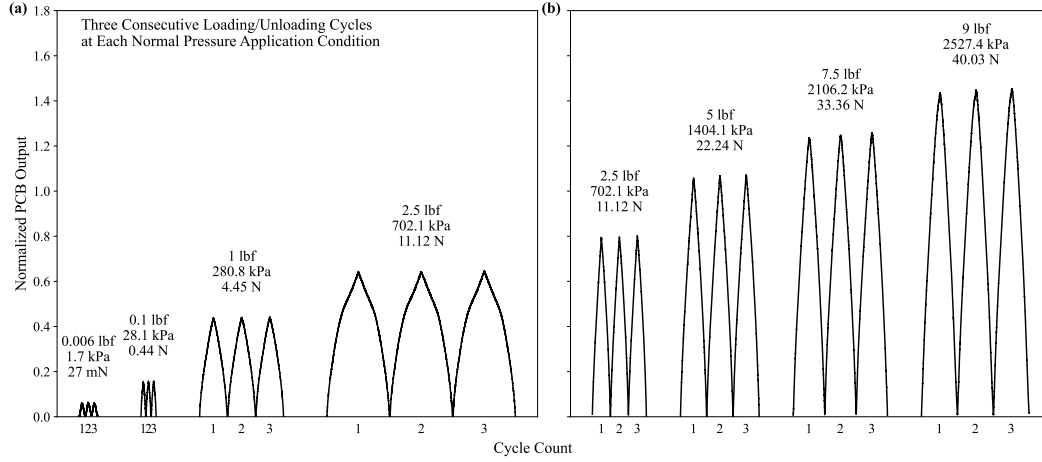
Figure 10: Custom readout PCB used by DexSkin.

Figure 11: Normalized PCB output from a randomly selected taxel during three consecutive loading and unloading cycles under different normal pressure application conditions for two sleeve types. (a) DexSkin with the low-fill TPE soft sleeve: data shown for $1.7\,\mathrm{kPa}$ ($27\,\mathrm{mN}$), $28.1\,\mathrm{kPa}$ ($0.44\,\mathrm{N}$), $280.8\,\mathrm{kPa}$ ($4.45\,\mathrm{N}$), and $702.1\,\mathrm{kPa}$ ($11.12\,\mathrm{N}$). (b) DexSkin with a more rigid 100% infill TPE sleeve: data shown for $702.1\,\mathrm{kPa}$ ($11.12\,\mathrm{N}$), $1404.1\,\mathrm{kPa}$ ($22.24\,\mathrm{N}$), $2106.2\,\mathrm{kPa}$ ($33.36\,\mathrm{N}$), and $2527.4\,\mathrm{kPa}$ ($40.03\,\mathrm{N}$). Taxel readings are normalized with respect to the initial zero-load values. The y-axis, representing timestamps, has been relabeled as cycle counts to better highlight the three consecutive cycles performed in each condition. The vertical stage step size defaults to $0.01\,\mathrm{mm}$ but is reduced to $0.002\,\mathrm{mm}$ for the $1.7\,\mathrm{kPa}$ ($27\,\mathrm{mN}$) condition to prevent force overshoot.

the individually addressable characteristic of taxels in DexSkin, which is required for accurately localizing contact events, particularly for densely-packed taxel arrays.

A.4 Readout circuitry

We measure pressure-induced capacitance changes with an inexpensive (costs USD $18.4 at quantities of 1000) custom readout PCB shown in Figure 10, measuring $72.6\,\mathrm{mm}$ x $40.6\,\mathrm{mm}$. A single PCB is capable of handling inputs from all 120 taxels across both fingers, sufficient for a parallel jaw gripper setup. The onboard ESP32-S3 microcontroller scans through each taxel sequentially using multiplexers while communicating with the onboard capacitance-to-digital chip to clock discharging for capacitance measurement. The PCB also provides active and passive shielding options to minimize the effects of electromagnetic interference (EMI). During our experiments, each taxel is measured four times and averaged for accuracy. The resulting serial data stream is transmitted to the computer at $30\,\mathrm{Hz}$.
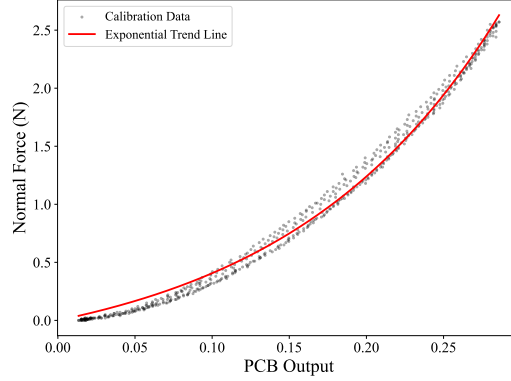
Figure 12: Representative scatter plot visualizing the collected force estimation dataset with 4 loading-unloading cycles with normal force applied up to $2.5\,\mathrm{N}$ ($157.8\,\mathrm{kPa}$) on a randomly selected taxel. Each of the gray dots represents a force-sensor output pair obtained during the calibration process, and the red line depicted is the exponential trend line obtained by fitting onto the calibration dataset composed of gray dots. The R^2 value for the trend line is 0.9929, indicating an excellent fit.

A.5 Sensor Calibration with Readout PCB

DexSkin exhibits consistency and low hysteresis, which are both crucial for reliable calibrations that accurately map measured sensor outputs into normal force and pressure values. In the following sections, we describe a calibration method that builds an accurate mapping from the PCB outputs to the normal pressure and force experienced. We then showcase the method by calibrating five randomly selected taxels and evaluating their force estimation error post-calibration.

A.5.1 Comprehensive Normal Force Calibration

For calibrating DexSkin's output to normal forces, we employ the same vertical-stage-based sensor characterization setup introduced in Section A.2 with max normal load set to $157.8\,\mathrm{kPa}$ ($2.5\,\mathrm{N}$), step size set to $0.01\,\mathrm{mm}$, and number of loading-unloading cycles set to 3. DexSkin taxels consistently follow an exponential pattern between the PCB readout and the force applied; thus we select the exponential function $a \cdot \left(e^{b \cdot (x+d)} - e^{b \cdot d}\right)$ where a, b, and d are the three fitting parameters to describe the relationship. After four loading/unloading cycles, we obtain the dataset required for our parameter fitting process. We first detect the peaks of these cycles and align the peak timestamps between the PCB readings and the applied force before performing downsampling on the former to match the sampling frequency of the force gauge. The resulting downsampled dataset is then used to fit the exponential trend line and derive the three fitting parameters. This completes the calibration of one taxel and takes approximately 3 minutes. The entire calibration process for all 60 taxels on DexSkin takes around 3 hours.

A.5.2 Force Estimation Performance of Calibrated-Taxels

To evaluate force estimation errors, we randomly selected five taxels, calibrated them following procedures outlined in the previous section, and obtained the three fitting parameters for its exponential trend lines. The exponential calibration curve for one of the taxels is visualized in Figure 12. Then we used the vertical stage setup to apply normal loads up to $2.5\,\mathrm{N}$ ($157.8\,\mathrm{kPa}$) onto each of the five taxels until we have obtained more than 6500 pairs of ground truth normal force values and PCB readings for each of them. For each resulting evaluation dataset, we feed the PCB readings into the fitted exponential trend line for force estimation, the results of which for one of the five taxels are plotted with the ground truth force values in Figure 13. Quantitatively, we collected a total of 39304 force-PCB value pairs across taxels and report $(0.086 \pm 0.021)\,\mathrm{N}$ as the root mean square error for estimated forces.
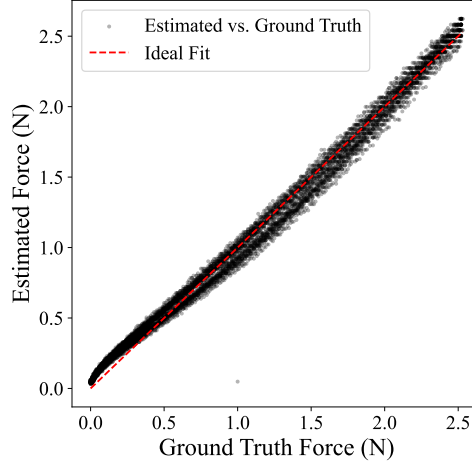
Figure 13: Representative scatter plot of estimated vs. ground truth force for one of the randomly selected taxels. We visualize the force estimation results against 7367 collected ground truth normal forces up to $2.5\,\text{N}$ ($157.8\,\text{kPa}$). Each of the gray dots represents an estimated force-ground truth force pairs and the red line depicted is the ideal fit line representing perfect estimation.

A.5.3 Three-Minute Pneumatic Pressure Calibration for Policy Transfer

In real-world robotic deployments, tactile sensors inevitably suffer from physical damage and wear, eventually failing. Yet due to fabrication variations, replacing a failed unit often results in distinct force responses which invalidates previously collected data and trained neural networks.

To address this, we developed a pneumatic calibration fixture, shown in Figure 14(a). It is comprised of a custom 3D-printed PLA chamber with an air outlet on top, a thin and soft Ecoflex 00-50 inner membrane adhered to the bottom surface of the chamber, and a 3D-printed PLA base with leg support tailored for DexSkin's height. We adopt a breadboard setup where the onboard ESP32-C3 microcontroller drives a 2N7000 MOSFET in low-side configuration to modulate a 6V DC air pump for inflation and reads the chamber pressure from a parallel-connected gas pressure sensor (Honeywell ABP) for feedback. When pressurized, the internal membrane deforms and applies a uniform stress over the surface of the DexSkin sensor. The system is capable of ramping up the relative chamber pressure up to $6psi$ ($41.4\,\text{kPa}$).

Similarly to Section A.5.1, with the pressure and corresponding sensor values collected, one can fit an exponential function $a \cdot e^{b \cdot x} + d$ where a, b, d are the three fitting parameters. For the pen reorientation model transfer evaluation, we capped the maximum inflating pressure at $18.7\,\text{kPa}$.

With the three fitting parameters obtained for both the source sensor and the target sensor, we adopt the following formula to calculate the estimated target sensor outputs using outputs from the source sensor:

$$C_2 = \frac{C_{0,2}}{b_2} \ln\left(\frac{a_1 \, \exp\left(b_1 \, \frac{C_1 - C_{0,1}}{C_{0,1}}\right) + d_1 - d_2}{a_2}\right) + C_{0,2}$$

where a_1, b_1, d_1 are fitting parameters for the source sensor, $C_1, C_{0,1}$ are the current output and no-load output for the source sensor respectively, a_2, b_2, d_2 are fitting parameters for the target sensor, and $C_2, C_{0,2}$ are the current output and no-load output for the target sensor respectively.

To evaluate the effectiveness of the pressure calibration, we subject two DexSkin fingers to $18.7\,\text{kPa}$ pressure loads and obtained the fitting parameters. Then we converted the recorded source sensor outputs using the formula shown above. Under a relatively uniform load of $18.7\,\text{kPa}$, the outputs for the source sensor, the target sensor, and the source sensor calibrated to emulate the target sensor are
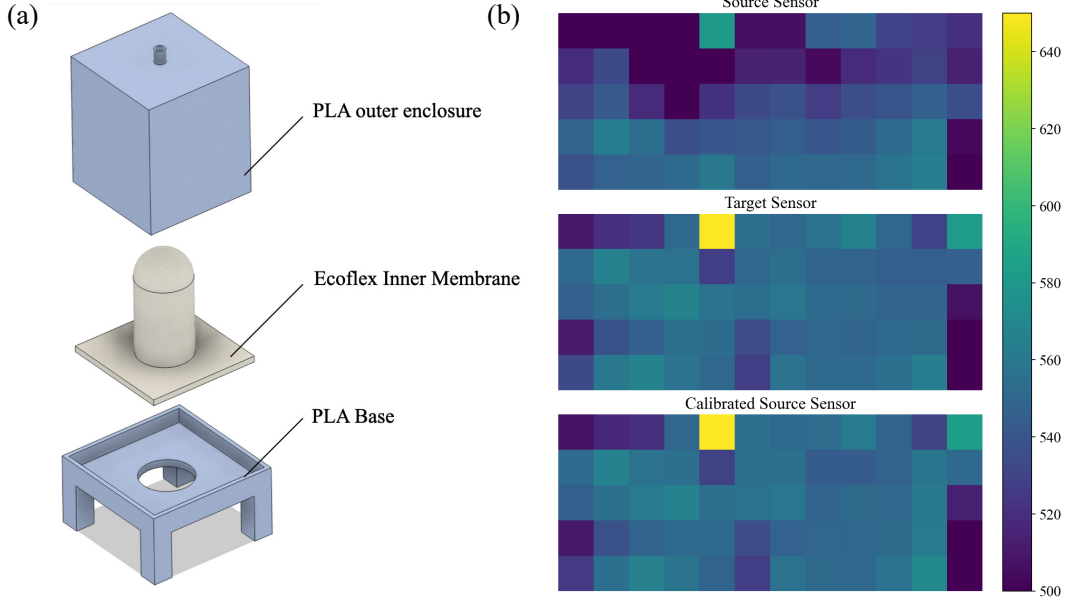
19

Figure 14: The pneumatic normal pressure calibration setup for DexSkin. (a) Exploded view of the airtight chamber that houses DexSkin during calibration and applies stress over the entire sensing region when pressurized. (b) Example heatmaps showing a set of raw values from the source sensor, raw PCB values from the target sensor to be transferred to, and the source sensor values after pneumatic pressure calibration when subject to the same 18.7 kPa normal pressure load. Ideally, the target sensor and calibrated source sensor readings should match.

visualized in Figure 14(b). The heatmaps of the target sensor and the calibrated source sensor are nearly indistinguishable while the source sensor's map remains noticeably different. Quantitatively, the calibration brings the source sensor's outputs much closer to the target sensor's: the Structural Similarity Index Measure (SSIM) increases from 0.3344 to 0.9442, and the mean squared error decreases from 1085.1 to 13.9. The effectiveness of this calibration procedure can be seen not only in these direct comparisons of sensor output values but also by downstream improvements in policy model transfer performance (Section 4.2).

B Additional Details on Robotic Experiments

Figure 15 shows an overview of the entire robotic hardware system, including the sensorized gripper, wrist camera, robot, and teleoperation device.

B.1 Details on Imitation Learning for Manipulation with Expanded Coverage and Tailorability

B.1.1 Task Definitions

Here we provide additional details about the physical definitions for each task:

- Pen reorientation: We use the same standard-size Expo pen when collecting all training episodes and when performing evaluation. The starting position of the pen on the table is essentially fixed (with allowances for small 1-2 cm variations) at the beginning of each episode. We also define a fixed target final location for the pen on the surface of the table.

- Box packaging: We mount a plastic sushi box in a fixed location on the table. We also 3D-print a small holder to hold the extra elastic band. All bands that we use are of an identical #19 size.
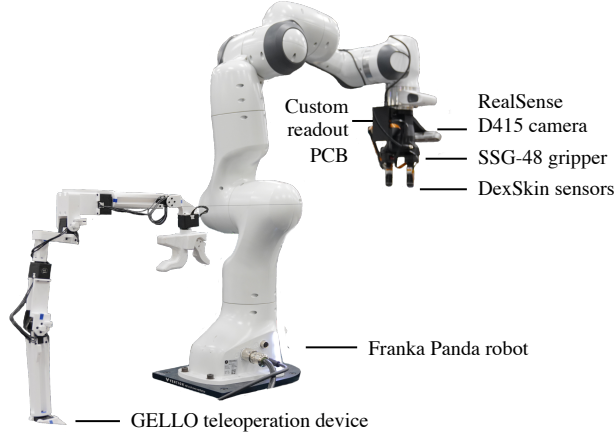
20

Figure 15: Hardware setup for manipulation and teleoperated demonstration collection with DexSkin. Our system consists of a Franka Panda robot, GELLO teleoperation device [44], SSG-48 gripper with DexSkin-equipped fingers, RealSense D415 RGB-D camera, and custom DexSkin readout circuitry.

B.1.2 Policy Training and Inference

To train imitation learning policies for the pen and box tasks in Section 4.1 and 4.2, as well as the base berry picking policy in Section 4.3, we use diffusion policies [45, 46]. We use the open-source implementation by the original authors and default hyperparameters from the official release, specifically the unet_hybrid_image configuration specified in the GitHub repository. We summarize these hyperparameters in Table 5.

Hyperparameter	Value
Model Configuration	
Horizon (H)	16
Observation Steps (n_{obs})	2
Inference Steps (n_{action})	8
Crop Shape	[202, 202]
Kernel Size	5
Optimizer (AdamW)	
Learning Rate (η)	1e-4
Betas (β_1, β_2)	(0.95, 0.999)
Epsilon (ϵ)	1e-8
Weight Decay	1e-6

Table 5: Diffusion policy training and inference hyperparameters.

For policy configurations where DexSkin observations are provided as input, the DexSkin observations are processed as low-dimensional observations of dimension 120, which are ingested by the policy in the same way as the robot proprioceptive state and gripper state, that is, using fully-connected (linear) layers. We perform random crops on the RGB camera observations, which is a standard recommendation for training image-based diffusion policies.

Training is carried out on a mixture of NVIDIA TITAN RTX, 3090, and A5000 GPUs. Training policies for the pen reorientation task takes approximately 12 hours, completing 581k global steps. The policies for the box packaging task trained for approximately 34 hours, completing 554k global steps.

At inference time, we perform 8 diffusion policy denoising steps for computational efficiency, as recommended by Chi et al. [45]. We execute all 16 actions output by the policy before performing another policy forward pass, rolling out the policy actions at 20Hz.

We perform all policy inference on an NVIDIA RTX A4000 GPU.

B.1.3   Evaluation Details

When conducting comparative evaluations for a set of policies, we programmatically randomly shuffle the order in which the trials for each policy will be evaluated. This order is unknown to the experimental evaluator to eliminate potential bias. We perform experimental trials for all policies for a particular experimental configuration (e.g. no perturbation vs. with perturbation for the pen reorientation task) consecutively before moving to the next configuration.

We define success conditions for each task as follows:

- Pen reorientation: A trajectory is considered successful if the pen reaches an orientation that is within $30°$ of perpendicular to the ground, and then hovers or rests its cap over a predefined target location. In the experimental setting where the human experimenter perturbs the pen, the trajectory is deemed successful only if the robot reorients the pen to within $30°$ of perpendicular to the ground initially and returns it to this pose after the human perturbs the pen one time to its starting orientation.

  In the calibration experiments described in Section 4.2, we report more granular notions of success: solving stage 1 involves completing a successful initial reorientation of the pen (prior to perturbation), and stage 2 requires detecting and fixing the human perturbation.

- Box packaging: A trajectory is considered successful if both of the following conditions are true: (a) the robot discards the band it initially is holding if it is perforated and does not discard it otherwise, and (b) the trajectory ends with the rubber band wrapped around the box and lid, with the rubber band not touching any part of the robot. We also report performance on the "Select Band" subtask, which corresponds to satisfying condition (a).

B.2   Details on Real-World Online Learning Experiments

B.2.1   Reinforcement Learning Training Details

To train the base imitation learning policy ("base policy"), we collect 50 task demonstrations **on a robot equipped with an unsensorized gripper**. Using this data, we train a diffusion policy following almost an identical configuration as the box packaging task in Section 4.1, including the use of wrist camera inputs. However, the base imitation policy training has one key difference: while the experiments in Section 4.1 used DexSkin tactile sensor readings as inputs, our base policy for online learning **does not** use tactile information as input.

Then, we use SAC [48, 49] to train a residual policy. The residual policy takes as input the same proprioceptive information as the base policy, but **does not** use visual (wrist camera image) inputs and instead **does** use 120 taxels of DexSkin tactile information. Excluding visual inputs from the residual policy input keeps the residual policy lightweight and quicker to learn, and avoids potential visual distribution shifts from faux to real berries. In addition to the proprioceptive states from the robot, the residual policy receives the base policy's predicted action a_b at a given step as input, such that it has access to the action it is modifying.

Just as in the imitation learning experiments, the base policy computes a sequence of 16 actions, which are all executed (with residual actions applied) before the base policy is queried again.

The residual policy's actions $a_r \in \mathcal{A_R} = [0.8, 1.2]$ (where $\mathcal{A_R}$ is the residual policy's action space) represent scaling modifications to the base policy's gripper action. That is, the action executed on the robot is modified for the gripper joint to instead be $a = \min(\max(a_b * a_r, 0), 1)$. This scaling formulation allows the residual policy to have a high level of control when the gripper is mostly

closed, while reducing high-amplitude oscillations when the gripper is mostly open (e.g. when the robot is simply approaching the object) that cause exploration to become very challenging.

We apply exponential moving average smoothing to the final gripper actions with $\alpha = 0.3$ to reduce high frequency jittering that makes exploration challenging and can potentially damage hardware.

During training, we train for 130 episodes, approximately equivalent to 42k environment steps. An episode ends after the robot attempts or successfully deposits a berry into the basket, returning its end effector to above a prespecified height, or reaches a time limit of 425 steps (approximately 30 seconds) without doing so. For the first 100 episodes, we train using a faux berry that has a visual appearance and geometry similar to that of a real berry. For the last 30 episodes, to fine-tune the model for better performance on real blueberries, we remove all transitions from the replay buffer and then resume training using real blueberries.

The reward function has three terms as follows:

- **Large force penalty**: Unlike some optical or magnetic tactile sensors, DexSkin readings are immediately localized and readily interpretable, and can thus be used to compute rewards such as this large force penalty, which equals the amount by which each DexSkin taxel reading exceeds a specified threshold. We first filter out very large tactile values (> 0.35) that are caused by unintentional contacts with the surface of the plate that the berry starts on. That is, using $t \in \mathbb{R}^{120}$ to represent the DexSkin outputs,

$$r_{force} = \| \max(0, t - t_{thresh}) \|_2^2$$

  where $t_{thresh} = 0.1$.

- **Residual policy action regularizer**: This is an ℓ_2 penalty that discourages modifications to the base policy that do not result in larger rewards. Specifically, we penalize the amount by which the residual action changes the gripper action in gripper width space:

$$r_{action} = -\|(1 - a_r) * a_b\|_2$$

- **Task failure penalty**: A poor residual policy can cause the combined policy to entirely fail to solve the task, for example, by failing to grasp the berry, dropping it in the middle of transport, or causing the policy to continuously stall in midair. In these cases, the human operator manually assigns a large negative reward only at the last timestep of $r_{failure} = -10$. Otherwise, $r_{failure} = 0$. Note that the failure reward is not assigned when the robot applies too much force and squishes or breaks the berry; it is only assigned when the picking or transporting motions fail.

The total reward is $r = r_{force} + 0.01 * r_{action} + r_{failure}$.

Table 6 shows detailed hyperparameters for SAC.

B.2.2 Evaluation Details

During evaluation, we evaluate each policy over 20 trials. To be consistent and minimize potential bias, we perform comparative evaluations of policies in one sitting and programmatically randomize the order in which the trial for each policy will be evaluated. This order is unknown to the experimental evaluator.

For Table 4, we use strict criteria to determine if a berry remains "intact": A berry with **any visible tear or cut whatsoever** is considered **not intact**. Our residual policy keeps 60% of the berries fully intact, but visual inspection shows that even those it does not keep intact are considerably less damaged than the ones handled by the baselines (Figure 6).

To compute the pressure value comparisons in Table 4, we first filter out any raw sensor values below 0.005 to select only potentially active taxels, then apply the mapping of sensor values to pressure values (kPa) obtained following the calibration procedure from Section A.5, and finally compute the mean of the mapped pressure values over all timesteps in the 20 evaluation trajectories in each setting.

Hyperparameter	Value
Learning rate	3e-4
Batch size	256
(Polyak) parameter τ	0.005
Discount factor γ	0.99
Num gradient steps per env step	5
Entropy coefficient	learned
Target network updates per env step	1
Policy/critic architecture	MLP w/ hidden dimensions [256, 256]

Table 6: Hyperparameters for real-world reinforcement learning with SAC. We largely keep to Stable Baselines3 [49] defaults, and any unlisted hyperparameters take on their SB3 default values. One notable difference is that we use 5 gradient steps per environment step as opposed to the default of 1, which we found to improve sample efficiency.

B.3 Details on Calibration and Model Transfer Experiments

For this experiment, the task definition and setup are identical to that of the pen reorientation task in Section B.1.1. We start by collecting a dataset of 50 demonstrations for the task, in the same fashion as in Section B.1.1.

For the baseline experimental configuration, we verify that the policies achieve strong success rates with the sensors with which the training data was collected (source sensors).

We then physically exchange the sensors that are attached to the left and right sides of the gripper to the right and left respectively (as our sensors are designed to be interchangeable), simulating a rearrangement of the sensors. Deploying the policy directly with this configuration is the "Swapped (no calib.)" setting. We also calibrate the sensors using values computed as in Section A.5 to map all readings first into pressure values, and then into the space of source sensor readings. Note that this does not require retraining the policy. The setting with the same policy model, but providing remapped tactile values as inputs, is denoted "Swapped (calib)". We conduct blind, order-randomized evaluations between the two "Swapped" settings. Unfortunately, we cannot do so with "Source sensors" because running evaluation on that setting requires physically adjusting the hardware, making it impossible to evaluate blindly.

We repeat this experiment with a set of physically different sensors that were fabricated using the same procedure, but in different batches. This simulates a replacement of the sensors, or deployment on an entirely different robotic setup. These experiments are denoted "Replaced (no calib.)" and "Replaced (calib.)".

For the DIGIT comparison, we test the policies from Section 4.1 first by physically swapping the **gel cartridge** between left and right DIGIT sensors. The visual appearance of the gel can vary from cartridge to cartridge as a result of variations in manufacturing. This setting is denoted "Swapped" in Table 3. Replacement of gel cartridges is a common maintenance operation that must be performed as the surfaces of DIGIT sensors accumulate wear over time. While the DIGIT policy in the target sensor setting (swapped gels) does not complete the task in any of the 20 trials, it exhibits reasonable behaviors such as reaching for and often grasping the pen. However, it shows a variety of failure modes such as mispositioning the gripper before attempting a grasp, and being unable to push the pen against the cabinet in an appropriate way to reorient it. It is likely that the differences in the visual appearances of the gels led to these errors that ultimately cause task failure.

One technique for reducing a policy's sensitivity to visual appearance is to train policies computed on difference images. We tested a setting where the policy's tactile input is a difference between the current tactile image observation and the image from the first step in the trajectory, which is assumed to be captured while the sensor is not deformed. This yields significantly improved performance in the swapped sensor setting ("Swapped (diff. image)" in Table 3).

We finally test the performance of the DIGIT difference image policies on entirely physically distinct sensors ordered from Gelsight Inc. (different housing, image sensor, gel) to simulate sensor replacement, denoted ("Replaced (diff. image)").