

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. For the rigid gripper, we sensorize a $13.8 \text{ mm} \times 22.3 \text{ mm}$ area on each fingertip.

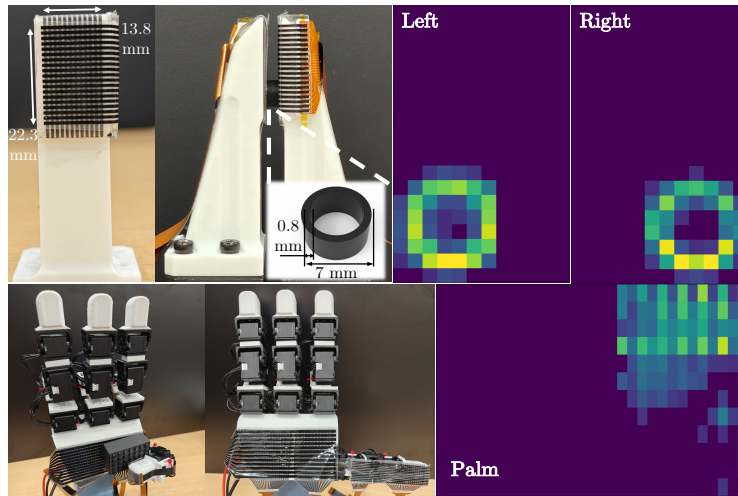


Figure 7: DexSkin application and example readings for a (top) flat parallel jaw gripper (228 taxels each) and (bottom) LEAP hand (372 taxels each for palm and thumb). For the LEAP hand, we visualize the palm taxels when holding the studded rectangular plastic block shown in the bottom left.

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 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 want to emphasize 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 157.8 kPa (2.5 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 157.8 kPa (2.5 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.

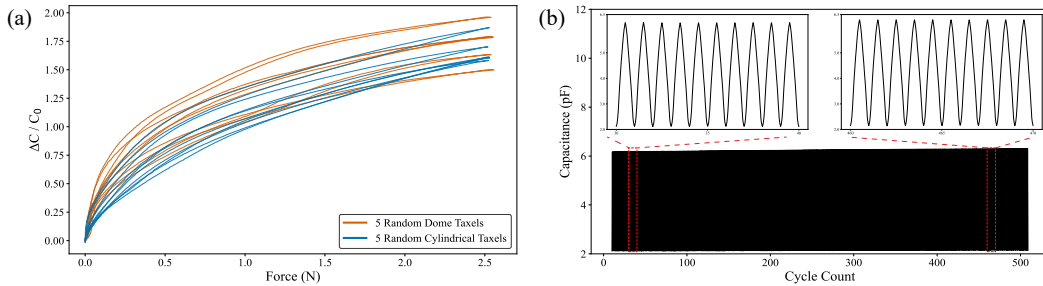


Figure 8: Characterization of the cross-taxel uniformity and cyclic stability of DexSkin. (a) **Normalized capacitance change $\Delta C/C_0$ versus applied force for 10 randomly selected taxels** measured during loading and unloading up to 157.8 kPa (2.5 N). Note the small hysteresis and high consistency across both dome and cylindrical taxels. (b) **Single-taxel cyclic loading up to 157.8 kPa (2.5 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 segments. 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 379 Pa (6 mN) and up to 157.8 kPa (2.5 N), beyond which the sleeve structure yields to provide cushioning. The load sensing range that soft sleeve DexSkin exhibits is comparable to prior sensors such as Gelsight (~ 200 kPa) [48] and DIGIT Pinki (~ 80 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 631.3 kPa (10.0 N).

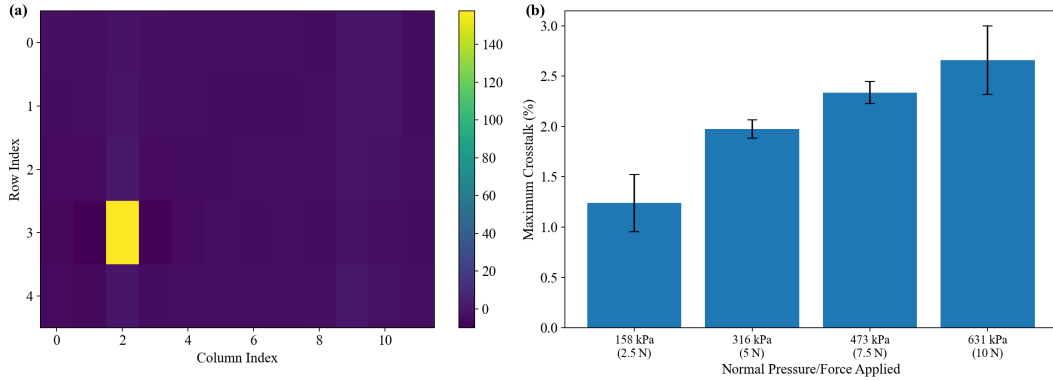


Figure 9: Evaluation of DexSkin’s crosstalk under a localized normal load. (a) Heatmap of the sensed normal pressure across DexSkin’s full 60 taxels when a single taxel is subjected to a localized normal load measuring 157.8 kPa (2.5 N). (b) Quantitative evaluation of the maximum crosstalk (in percentage) observed across the remaining 59 taxels with four different normal pressures applied on the loaded taxel. Crosstalk remains below 3% under maximum 631.3 kPa (10.0 N) 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 126.3 kPa (2.0 N) to 631.3 kPa (10.0 N) on a randomly selected taxel. We read out the sensor outputs using our custom PCB, and all raw values were converted to normal pressure using the calibration function described in Section A.5. We were able to obtain 1019 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 157.8 kPa (2.5 N). The visualized response is highly localized and exhibits minimal activations in the surrounding taxels.

To quantify crosstalk, we define the maximum 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:

$$\text{Crosstalk (\%)} = \left(\frac{\max_{j \neq i} P_j}{P_i} \right) \times 100\%$$

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 maximum crosstalk increases with the magnitude of normal loads but remains below 3% for all four loading conditions shown. Expanding the analysis to all 1019 pairs acquired, we obtain a mean crosstalk of $1.89\% \pm 0.008\%$. This highlights the minimal amount of interference and couplings

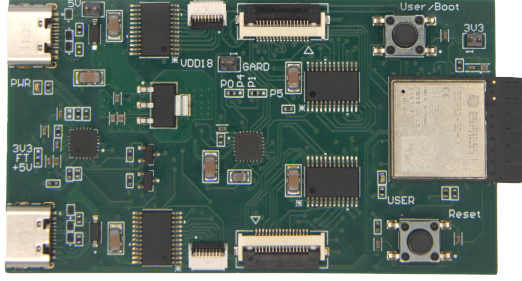


Figure 10: Custom readout PCB used by DexSkin.

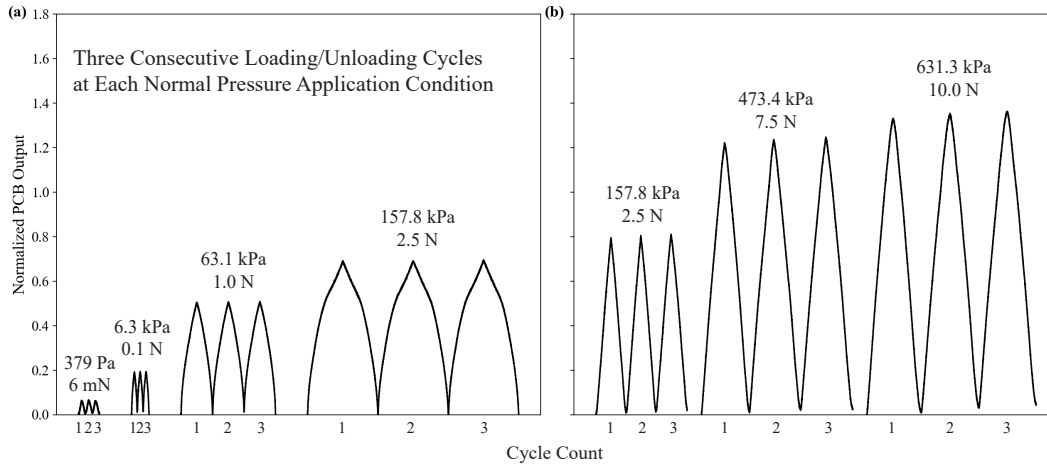


Figure 11: Normalized PCB output from a randomly selected taxel during three consecutive loading and unloading cycles under different normal pressure application conditions for two sleeve types. (a) DexSkin with the low-fill TPE soft sleeve: data shown for 379 Pa (6 mN), 6.3 kPa (0.1 N), 63.1 kPa (1.0 N), and 157.8 kPa (2.5 N). (b) DexSkin with a more rigid 100% infill TPE sleeve: data shown for 157.8 kPa (2.5 N), 473.4 kPa (7.5 N), and 631.3 kPa (10.0 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 mm but is reduced to 0.002 mm for the 6 mN condition to prevent force overshoot.

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

561 A.4 Readout circuitry

562 We measure pressure-induced capacitance changes with an inexpensive (costs USD \$18.4 at quan-
 563 tities of 1000) custom readout PCB shown in Figure 10, measuring 72.6 mm x 40.6 mm. A single
 564 PCB is capable of handling inputs from all 120 taxels across both fingers, sufficient for a parallel jaw
 565 gripper setup. The onboard ESP32-S3 microcontroller scans through each taxel sequentially using
 566 multiplexers while communicating with the onboard capacitance-to-digital chip to clock discharging
 567 for capacitance measurement. The PCB also provides active and passive shielding to minimize the
 568 effects of electromagnetic interference (EMI). During our experiments, each taxel is measured four
 569 times and averaged for accuracy. The resulting serial data stream is transmitted to the computer at
 570 30 Hz.

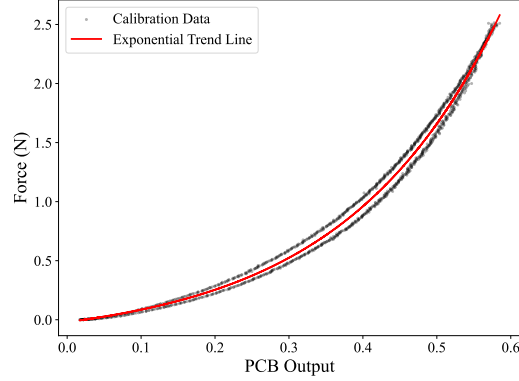


Figure 12: Representative scatter plot visualizing the collected force estimation dataset with 3 loading-unloading cycles with normal force applied up to 2.5 N (157.8 kPa) on a randomly selected taxel. Each of the gray dots represents a force-sensor output pair obtained during the calibration process, 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 0.9934, indicating an excellent fit.

571 A.5 Sensor Calibration with Readout PCB

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

577 A.5.1 Comprehensive Normal Force Calibration

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

590 A.5.2 Force Estimation Performance of Calibrated-Taxels

591 To evaluate force estimation errors, we randomly selected four taxels, calibrated them following pro-
 592 cedures outlined in the previous section, and obtained the four fitting parameters for its exponential
 593 trend lines. The exponential calibration curve for one of the taxels is visualized in Figure 12. Then
 594 we used the vertical stage setup to apply normal loads up to 2.5 N (157.8 kPa) onto each of the four
 595 taxels until we have obtained more than 6500 pairs of ground truth normal force values and PCB
 596 readings for each of them. For each resulting evaluation dataset, we feed the PCB readings into the
 597 fitted exponential trend line for force estimation, the results of which for one of the four taxels are
 598 plotted with the ground truth force values in Figure 13. Quantitatively, we collected a total of 29930
 599 force-PCB value pairs across taxels and report (0.089 ± 0.004) N as the root mean square error for
 600 estimated forces.

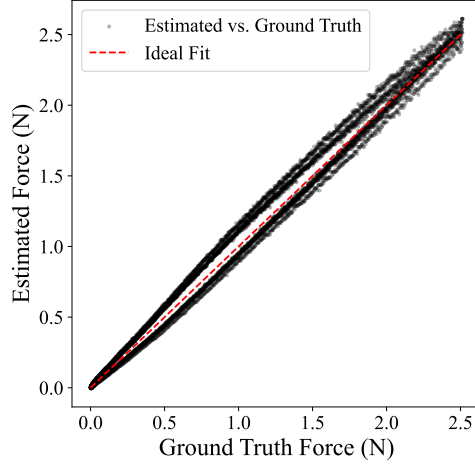


Figure 13: Representative scatter plot of estimated vs. ground truth force for one of the randomly selected taxels. We visualize the force estimation results against 7031 collected ground truth normal forces up to 2.5 N (157.8 kPa). Each of the gray dots represents an estimated force-ground truth force pairs and the red line depicted is the ideal fit line representing perfect estimation.

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

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

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

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

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

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

620 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-
 621 load output for the source sensor respectively, a_2, b_2, d_2 are fitting parameters for the target sensor,
 622 and $C_2, C_{0,2}$ are the current output and no-load output for the target sensor respectively.

623 To evaluate the effectiveness of the pressure calibration, we subject two DexSkin fingers to uniform
 624 18.7 kPa pressure loads and obtained the fitting parameters. Then we converted the recorded source
 625 sensor outputs using the formula shown above. Under a uniform load of 18.7 kPa, the outputs for
 626 the source sensor, the target sensor, and the source sensor calibrated to emulate the target sensor are

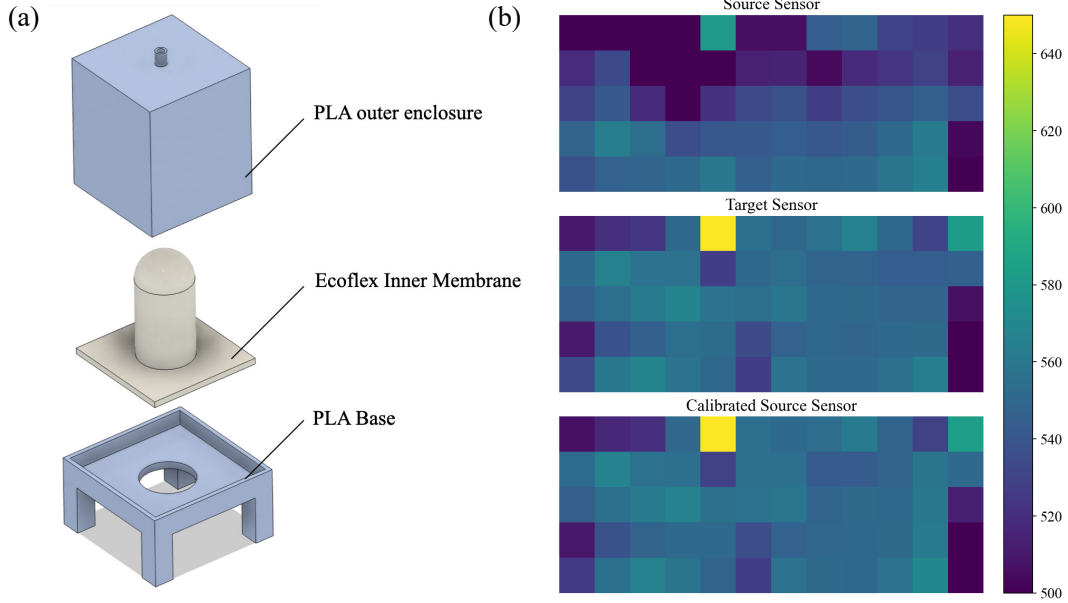


Figure 14: The pneumatic normal pressure calibration setup for DexSkin. (a) Exploded view of the airtight chamber that houses DexSkin during calibration and applies stress over the entire sensing region when pressurized. (b) Example heatmaps showing a set of raw values from the source sensor, raw values 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.

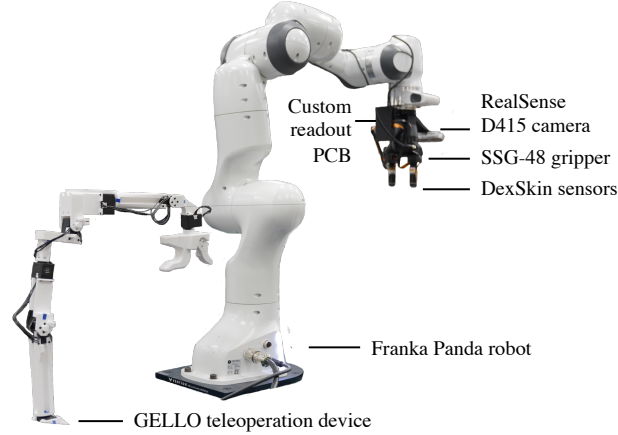


Figure 15: Hardware setup for manipulation and teleoperated demonstration collection with DexSkin. Our system consists of a Franka Panda robot, GELLO teleoperation device [42], SSG-48 gripper with DexSkin-equipped fingers, RealSense D415 RGB-D camera, and custom DexSkin readout circuitry.

647 B.1.2 Policy Training and Inference

648 To train imitation learning policies for the pen and box tasks in Section 4.1 and 4.2, as well as
 649 the base berry picking policy in Section 4.3, we use diffusion policies [43, 44]. We use the open-
 650 source implementation by the original authors and default hyperparameters from the official release,
 651 specifically the `unet_hybrid_image` configuration specified in the GitHub repository. We summa-
 652 rize 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.

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

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

At inference time, we perform 8 diffusion policy denoising steps for computational efficiency, as recommended by Chi et al. [43]. 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 demonstrations **with an unsensorized gripper** and train diffusion policies following the same configuration as the box packaging task in Section 3 (including using wrist camera inputs), but with one key difference: while the box packaging policies used DexSkin tactile sensor readings as inputs, our base policy for online learning does use tactile information as input.

Then, we use SAC [46, 47] 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 **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.

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

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

718 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,

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

719 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.

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

728 Table 6 shows detailed hyperparameters for SAC.

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 [47] 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.

729 B.2.2 Evaluation Details

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

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

738 To compute the pressure value comparisons in Table 4, we first filter out any raw sensor values below
739 0.005 to select only potentially active taxels, then apply the mapping of sensor values to pressure
740 values (kPa) obtained following the calibration procedure from Section A.5, and finally compute
741 the mean of the mapped pressure values over all timesteps in the 20 evaluation trajectories in each
742 setting.

743 B.3 Details on Calibration and Model Transfer Experiments

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

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

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

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