Self-supervised perception for tactile skin covered dexterous hands

Akash Sharma[1,2], **Carolina Higuera**[1,3], **Chaithanya Krishna Bodduluri**[1], **Zixi Liu**[1], **Taosha Fan**[1], **Tess Hellebrekers**[1], **Mike Lambeta**[1], **Byron Boots**[3], **Michael Kaess**[2], **Tingfan Wu**[1], **Francois Robert Hogan**[1], **Mustafa Mukadam**[1]

[1]FAIR at Meta, [2]Carnegie Mellon University, [3]University of Washington,

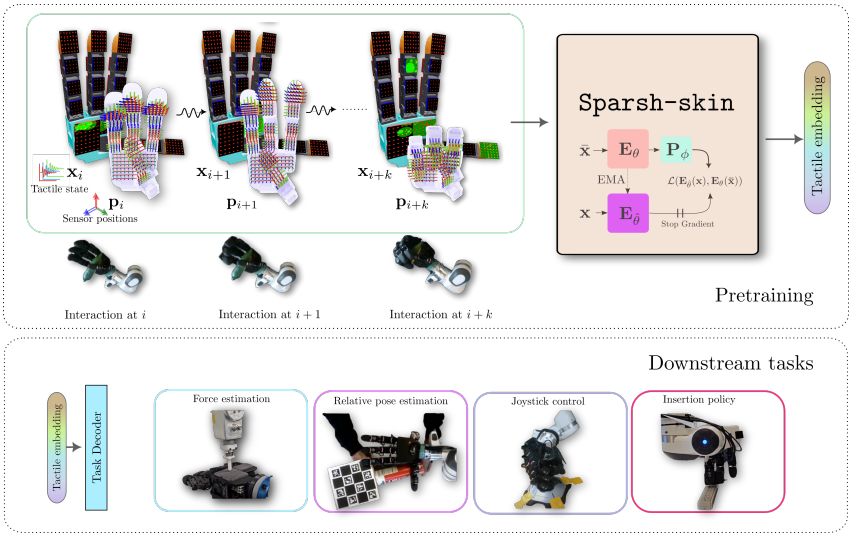


Figure 1: `Sparsh-skin` is an approach to learn general representations for magnetic tactile skins covering dexterous robot hands. `Sparsh-skin` is trained via self-supervision on a large pretraining dataset (~ 4 hours) containing diverse atomic in-hand interactions. It takes as input a brief history of tactile observations $\mathbf{x}_i$ and 3D sensor positions $\mathbf{p}_i$ to produce performant full-hand contextual representations. `Sparsh-skin` representations are general purpose and can be used in a variety of contact-rich downstream tasks.

Abstract: We present `Sparsh-skin`, a pre-trained encoder for magnetic skin sensors distributed across the fingertips, phalanges, and palm of a dexterous robot hand. Magnetic tactile skins offer a flexible form factor for hand-wide coverage with fast response times, in contrast to vision-based tactile sensors that are restricted to the fingertips and limited by bandwidth. Full hand tactile perception is crucial for robot dexterity. However, a lack of general-purpose models, challenges with interpreting magnetic flux and calibration have limited the adoption of these sensors. `Sparsh-skin`, given a history of kinematic and tactile sensing across a hand, outputs a latent tactile embedding that can be used in any downstream task. The encoder is self-supervised via self-distillation on a variety of unlabeled hand-object interactions using an Allegro hand sensorized with Xela uSkin. In experiments across several benchmark tasks, from state estimation to policy learning, we find that pretrained `Sparsh-skin` representations are both sample efficient in learning downstream tasks and improve task performance by over 41% compared to prior work and over 56% compared to end-to-end learning.

Keywords: Magnetic-skin, representation learning, Self-supervised learning

1 Introduction

Touch is inconspicuous, but plays a crucial role in dexterous manipulation, like when playing the guitar or plugging a cord into a socket when vision is impaired. The robotics community has

leveraged touch to enhance robot learning [1, 2, 3, 4], but has so far largely limited their attention to vision-based tactile sensing. Recently, sensors such as the DIGIT [5], GelSight [6], GelSlim [7] and others [2, 8], have gained popularity due to their high-resolution output, human-interpretable signals, and accessibility. Capturing touch as images is attractive, as advances in computer vision can be leveraged with minimal friction. Nevertheless, these sensors are slow compared to human skin's touch receptors, come in bulky form factors precluding large area sensing, and are often custom-designed for specific manipulators [2], making reproducibility a challenge.

Magnetic skin-based sensors such as uSkin (Xela) [9, 10], ReSkin [11], and others [12, 13], offer an alternative for tactile feedback. They provide fast response times (~ 100 Hz), lower dimensionality and flexible form factors that can be adapted to complex embodiments, such as multifinger robot hands providing richer states for dexterous manipulation. Despite their potential, the widespread use of these sensors is primarily limited by their complexity: these sensors are difficult to interpret, difficult to model due to hysteresis and other factors, and are primarily hindered by a lack of infrastructure.

Self-supervised learning of general touch representations offers a potential solution: it can learn priors from unlabeled data, making subsequent learning on specific tasks (downstream learning) sample efficient. However, while previous research [14, 15], has applied self-supervision for tactile learning, these approaches often use techniques from computer vision such as treating temporal signals as images [14] and employing masked reconstruction objectives [15, 16] that may be ill-suited for signals like noisy magnetic flux.

We present `Sparsh-skin`, a pre-trained tactile encoder model trained using self-supervised learning (SSL) for magnetic skin-like sensors covering a multifinger robot hand (see Figure 1). `Sparsh-skin` directly learns in-hand contact priors from tactile history and hand configuration using a classification objective. Our tactile encoder simplifies downstream task use, by introducing standardized magnetic time-series data, and reducing the need for real-world labeled data, which is difficult to collect and oftentimes infeasible. For instance, we do not yet have hardware to annotate spatially distributed ground truth force fields. By combining our representation learning algorithm, tactile signal tokenization, and a fully-sensorized multi-fingered hand, we achieve state-of-the-art tactile representations for magnetic-skin sensors, outperforming end-to-end training by $\sim 56.37\%$ and prior works by $\sim 41.04\%$ on average in both performance and sample efficiency for downstream tasks. We aim to open source our code, datasets, and models. The main contributions of our work are:

1. `Sparsh-skin`: a general purpose tactile representation model, trained via self-distillation for magnetic-skin based tactile sensors.
2. A revisit of tokenization, masking and the learning algorithm choices for temporal magnetic tactile signals which improves downstream task performance by over 41%.
3. A dataset containing 4 hours of random play-data of the Allegro robot hand sensorized with the Xela tactile sensors, labeled datasets, metrics, and task design that cover relevant problems in tactile perception to evaluate learned representations.

2 Related work

2.1 Tactile sensing

Tactile sensors can be broadly categorized into vision-based (e.g. DIGIT [5], GelSight [6], GelSlim [7] and others [2, 8]), pressure-based (e.g. force sensitive resistors), impedance-based (e.g. BioTac [17]), and magnetic-based (e.g. uSkin (Xela) [9, 10], ReSkin [11], and others [12, 13]) sensors. Vision-based sensors commonly used in robot manipulation capture finger-object-environment interactions as images [5, 6]. However, their bulky form factor, low-frequency feedback and high bandwidth requirement limit their application in tasks that require large areas coverage. Impedance-based tactile sensors offer high temporal resolution, but are difficult to interpret, and currently do not provide full-hand coverage solutions either. Pressure-based sensors can offer a wide coverage area, but lacks capabilities in shear force sensing. Magnetic tactile sensors, on the other hand, provide a thin skin-like alternative with options such as ReSkin [11], AnySkin [12], and Xela [9, 10] being popular choices. They provide low-dimensional but high-frequency signals. However, when these sensor pads are distributed on all contact interfaces of a robot hand, the total output is high-dimensional.

These sensors primarily use hall-effect sensing for force measurement. Xela [10] in particular works by transducing displacements of permanent magnets embedded in an elastomer arranged in a grid pattern to magnetic flux changes, essentially capturing 3-axis shear and normal forces. ReSkin and AnySkin [11, 12] magnetize the entire elastomer layer continuously, instead of using discrete magnets. This sensing modality has been explored for various contact-rich applications, including planar pushing [18], surface material classification [19], grasp stability [20], and policy learning tasks [21, 22].

2.2 Tactile representation learning

Representation learning for vision-based tactile sensors has recently gained significant attention. Since the sensor outputs are images, techniques from computer vision [16, 23] have been extended to tactile sensors. This is motivated by a move beyond task-specific encoders to pretrained encoders that promise generalization, with prior work leveraging mask-autoencoders (MAE) [24, 25], contrastive learning [26, 27, 28, 29], and state-of-the-art methods like self-distillation and joint-embedding predictive architectures [3] to learn tactile representations.

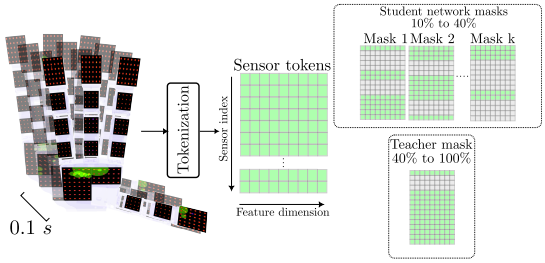


Figure 2: Illustration of Xela signal corruption via masking for SSL prediction task: Once a 100(ms) window of tactile measurements and sensor positions are tokenized, block masking is applied to corrupt the signal, . For each data sample, the student network receives k different masks, each randomly retaining 10% to 40% of the data denoted $\bar{z}_i$. The teacher network, in contrast receives 1-2 masks each retaining 40% to 100% of the data denoted z_i^*.

Research on learning representations for magnetic-based sensors remains relatively underexplored. Since these sensors produce low-dimensional signals, the consensus view is that representation learning is likely unnecessary. However, as we highlight in our work, these signals are indeed high-dimensional due to the complexities of full hand sensing, dynamic tactile signals and hand poses, and magnetic sensor physical properties. This high dimensionality means they benefit from large-scale pretraining to compress information into semantically rich representations that enhance downstream task performance. Recently, HyperTaxel [21] applied contrastive learning to learn representations for the Xela sensor for the task of surface recognition but it did not show whether these representations capture contact dynamics. Similarly [14, 15] propose representation learning with self-supervised methods such as BYOL [30] and MAE [16]. While the idea of representation learning is promising, the choice of meaningful image augmentations without data corruption, is unclear for BYOL. Furthermore, by treating instantaneous tactile measurements as images, these methods discard temporal information and may therefore be suboptimal for tactile tasks that rely on contact dynamics.

3 `Sparsh-skin`: self-supervised representations for tactile skins

`Sparsh-skin` is a self-supervised modeling approach to learn from random-play data, generalizable tactile features for dexterous hands equipped with magnetic-skin tactile sensors.

3.1 Preliminaries

Self-distillation for representation learning Self-distillation [31, 32, 30] is a powerful paradigm in self-supervised learning involving a pair of identical neural networks, termed the student $\mathbf{E}_\theta$ and teacher network $\mathbf{E}_{\hat{\theta}}$. The student network receives a corrupted version of a data signal $\bar{\mathbf{x}}$ that is to be encoded, while the teacher network receives privileged information about the same data sample $\mathbf{x}$. Then, the student network is tasked with predicting through a small predictor network $\mathbf{P}_\phi$, the data representation that the teacher produced. To prevent the teacher from producing degenerate representations – for instance, a constant representation for all data – the teacher weights are not updated via back-propagation, but only through an exponential moving average (EMA) of the student

weights. Specifically, the following objective is optimized:

$$\arg\min_{\theta,\phi} \left\| \mathbf{P}_\phi(\mathbf{E}_\theta(\bar{x})) - \text{sg}(\mathbf{E}_{\hat{\theta}}(\mathbf{x})) \right\| \tag{1}$$

where sg indicates stop gradient, and $\hat{\theta} \triangleq \text{EMA}(\theta)$. Since the teacher network is an exponential moving average (EMA) of the student work, knowledge is *self-distilled* through the representation prediction task.

Robot setup and pretraining data Our setup consists of the Allegro hand sensorized with Xela uSkin, attached to a Franka Panda robot arm. The Allegro hand is equipped with 18 Xela uSkin sensing pads, consisting of 4 curved fingertip sensors with 30 individual sensors, 11 4x4 grid sensors pads attached to the finger phalanges, and 3 4x6 sensing pads attached to the palm, resulting in a total of 368 individual sensors. We collected a dataset of the hand performing various atomic manipulation actions with 14 household objects and toys, including squeeze, slide, rotation, pick-and-drop, circumduction, pressing, wiping, and articulation. Using a VR-based teleoperation system with Meta Quest 3, which builds on the inverse kinematics-based re-targeting method proposed in [33], we recorded 11 sequences (approximately 2 minutes each) for each object, totaling around 4 hours of varied interactions. The dataset includes top/left camera views, Xela signals, and robot and hand joint states, covering a range of rigid and deformable objects with diverse tactile properties (see Figure 4).

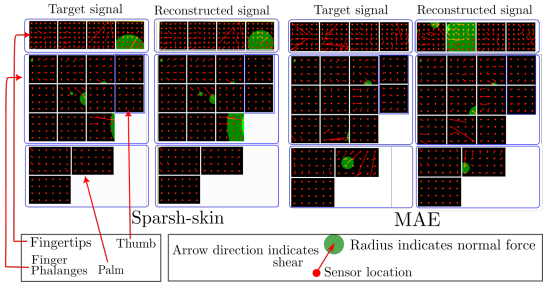


Figure 3: Visualization of reconstructions from the reconstruction online probe. When compared to MAE, `Sparsh-skin` reconstructs signals effectively. Specifically, note that the normal forces and directions are better preserved by `Sparsh-skin`. Here, we visualize a single frame from a 0.1s tactile window.

3.2 Architecture

`Sparsh-skin` uses a Transformer [34] as the student and teacher network for self-distillation.

Sensor tokenization We perform baseline subtraction on Xela signals to account for their uncalibrated nature and consistent biases. A single baseline signal is collected with the Allegro hand in a resting configuration (palm up and flat) and used for all downstream tasks, unlike prior work [11, 22], which collects a new baseline signal per training sequence. We also resample Xela signals to a consistent 100Hz frequency. as the sensor data rate fluctuates between 80Hz to 100Hz, unlike prior work [14] that subsamples data to match modalities at lower frequencies.

We note that for representation learning, tactile data can be temporally correlated, and instantaneous signals cannot provide context for contact changes, therefore we choose to learn representations for chunks of 100ms of data. First, inputs to `Sparsh-skin` are formatted corresponding to a brief history of 0.1 seconds of the sensor signal $x_{1:10} \in \mathbb{R}^{10\times368\times3}$ concatenated with the history of sensor position $p_{1:10} \in \mathbb{R}^{10\times368\times3}$ computed from the forward kinematics of the Allegro hand. Inputs are then tokenized through a linear projection f_{linear} to the dimension d of the representation $z_i = f_{\text{linear}}(x_{1:10}|p_{1:10}) \in \mathbb{R}^{368\times d}$. Finally, a learnable token is added to each sensor according to the three types of Xela sensing pads (see 3.1) on the Allegro hand. We do not add additional positional embedding and instead rely on the sensor position to provide 3D positional information to the transformer network.

SSL prediction task Although cropping and resizing images is a common technique for signal corruption in the image domain, applying this method to magnetic flux readings alters the shear profile. Therefore to avoid any untoward data augmentation that changes the semantic meaning of the signal, we use block masking [35] to corrupt signals that are input to the encoding networks. Specifically, input data is masked after sensor tokenization in a cross-taxel manner i.e., given tokenized data from 368 sensors, we mask sensor data from local contiguous blocks including sensors even from neighboring sensor islands by removing those sensors from the input (see Figure 2). The masked

sensor tokens are subsequently transformed through the student and teacher network as $\mathbf{E}_\theta(\bar{z}_i)$ and $\mathbf{E}_{\hat{\theta}}(z_i^*)$ respectively.

For the prediction task, we use classification by defining a set of prototype classes as in [36, 31], which is robust to sensor noise compared to masked auto-reconstruction. The sensor tokens after transformation are converted into prototype logits through a classification head f_{class} as $\bar{p}_i$ and p_i^* respectively for the student and teacher networks. We use both the class token and the patch level cross entropy objective between the student and teacher logit predictions to enforce local-to-global correspondence learning in the sensor representation. Additional details about the model architecture, MAE reconstruction comparisons and training hyper-parameters are in the Appendix.

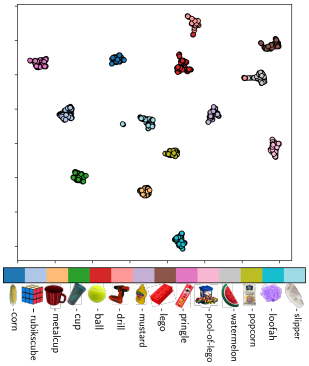


Figure 4: UMAP visualization of representations colored by object in robot hand.

Online Probes Unlike supervised learning (SL), where model performance is easily monitored through training and validation losses, in self-supervised learning (SSL), prediction task losses do not directly convey downstream task performance. In fact, in the presence of an EMA teacher network, which acts as a moving target, the prediction task loss can increase in tandem with the predictions of the teacher network. Therefore, we rely on online probes to monitor downstream performance. During training, we evaluate the tactile representation for a) reconstruction and b) the ability to identify objects used in play data.

Figure 3 provides a qualitative visualization of the reconstruction performance obtained by the decoder using representations computed by the student network $\mathbf{E}_\theta(\bar{z})$. Here, we find that `Sparsh-skin` trained using the MAE reconstruction objective (identical tokenization) scheme, is significantly inferior at reconstruction compared to `Sparsh-skin` trained via self-distillation. In terms of object classification performance, we achieve approximately 95% accuracy across 14 classes, while both BYOL (treating tactile signal as images) [14] and MAE (using tactile and proprioception history) [16] are limited to $\sim 81\%$ accuracy. Additionally, Figure 4 presents a UMAP [37] visualization of the representations, where sequences from each object are mapped to distinct, non-overlapping clusters.

Implementation Details Our method is designed for the Xela sensor but can be extended to any skin sensor with 3-axis time-series output signals. `Sparsh-skin` is trained for 500 epochs on 8 Nvidia A100 GPUs with a batch size of 64, using the AdamW optimizer, and linear warmup followed by cosine schedule as the learning rate scheduler. Downstream tasks are trained with task-labeled data on 1 Nvidia A100 / 4090 GPU. Furthermore, `Sparsh-skin` supports realtime inference with an inference time of $\sim$7ms. Please refer to the appendix for additional training details.

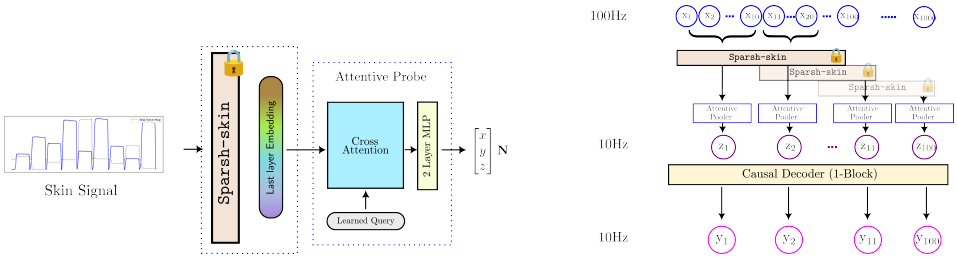
4 Experiments

In this section, we assess the ability of `Sparsh-skin` to comprehend tactile properties, enhance perception, and enable policy learning for manipulation through four downstream tasks spanning tasks studied in the tactile sensing literature: namely **(1)** Force estimation, **(2)** Joystick state estimation, **(3)** Pose estimation, and **(4)** Policy learning via the plug insertion task.

4.1 Evaluation protocol

Downstream task decoders. The tasks we consider are of two types: a) requiring instantaneous prediction, and b) requiring temporal reasoning over tactile data. For tasks such as force estimation that require an instantaneous estimate, we use attentive pooling (see Figure 5a). For tasks such as pose estimation and joystick state estimation, that require sequence reasoning, tactile observations are transformed into tokens at the output frequency through a cascaded application of the backbone network. This is followed by attentive pooling as illustrated in Figure 5b.

Model comparisons. For each of the downstream tasks, we explore multiple variants of the `Sparsh-skin` encoder, along with additional baselines:



(a) Attentive probe: Attentive pooling + small 2-layer MLP for regression tasks

(b) Decoder with a 1-layer transformer block for sequence to sequence prediction tasks

Figure 5: We use two types of decoders for (a) instantaneous, and (b) temporal tasks. Both decoders contain the attentive pooler which uses a learned query token to cross attend to sensor features to output a *single token full-hand* representation.

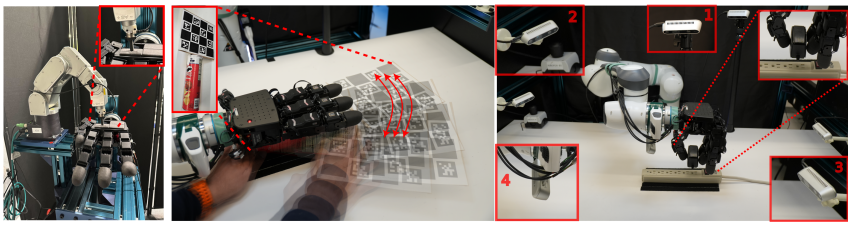


Figure 6: Hardware setup used for downstream tasks: **(Left)** shows the setup for force estimation. We use 3D printed probes attached to a F/T sensor to indent onto the Xela sensors. **(Middle)** shows the setup for pose estimation. We track an object mounted with an ArUco marker to obtain ground truth pose estimates while randomly moving it under the robot hand. **(Right)** shows the setup for plug insertion policy task. We collect tactile measurements and camera observations from three third-person view cameras and a wrist camera view.

1. BYOL*, our reproduction of the BYOL [30] approach to tactile representation learning following [14] using our collected play data and tactile data formatted as images, since the setup used in [14] does not contain palm sensing and uses an older variant of the tactile sensor.
2. End-to-end, training the entire encoder-decoder network with same capacity as `Sparsh-skin` using only labeled task data
3. `Sparsh-skin` (frozen), pretrained representation that uses tactile and hand configuration history.
4. `Sparsh-skin` (finetuned), where the encoder network is finetuned with task-specific data.
5. `Sparsh-skin` (MAE), pretrained representation that uses tactile and hand configuration history trained using MAE supervision instead of self-distillation.

For tasks **(1 - 3)**, we measure performance using the average root mean squared error (RMSE). Additionally, we evaluate each method for sample efficiency by reducing the downstream labeled data accessible during training. Then, for **(4)** plug insertion, we measure success rate (SR) across trials where we select the best model from **(1 - 3)** for comparison against the end-to-end baseline.

4.2 Downstream tasks

(1) Force Estimation. This task involves regressing tactile signals to 3-axis normal and shear forces on a robot hand's palm. We collected force-labeled data using a robot arm with an F/T probe to apply varying normal forces (0.25-5.0N) with hemispherical and flat indenters (see Figure 6 (left)). The probe's position was randomly sampled across the sensor pad, including locations both on and between magnetometers, differing from sensor characterization which only tests atop magnetometers. **Results** (see Figure 7(a)) While the end-to-end model is particularly worse at predicting forces throughout the spectrum, in low data regimes – 3.3% to 10% of the labeled data in this case – it is interesting to note that `Sparsh-skin` (finetuned) and `Sparsh-skin` (frozen) do not see any significant loss in performance. To this end, we test the models with even smaller number of downstream task data samples to find that `Sparsh-skin` (finetuned) is able to predict forces at a reasonable accuracy (350 mN in z) even with only $\sim$ 100 samples. Additionally, we find `Sparsh-skin` (MAE) is worse at predicting forces highlighting that MAE may not be suitable for noisy magnetic flux signals.

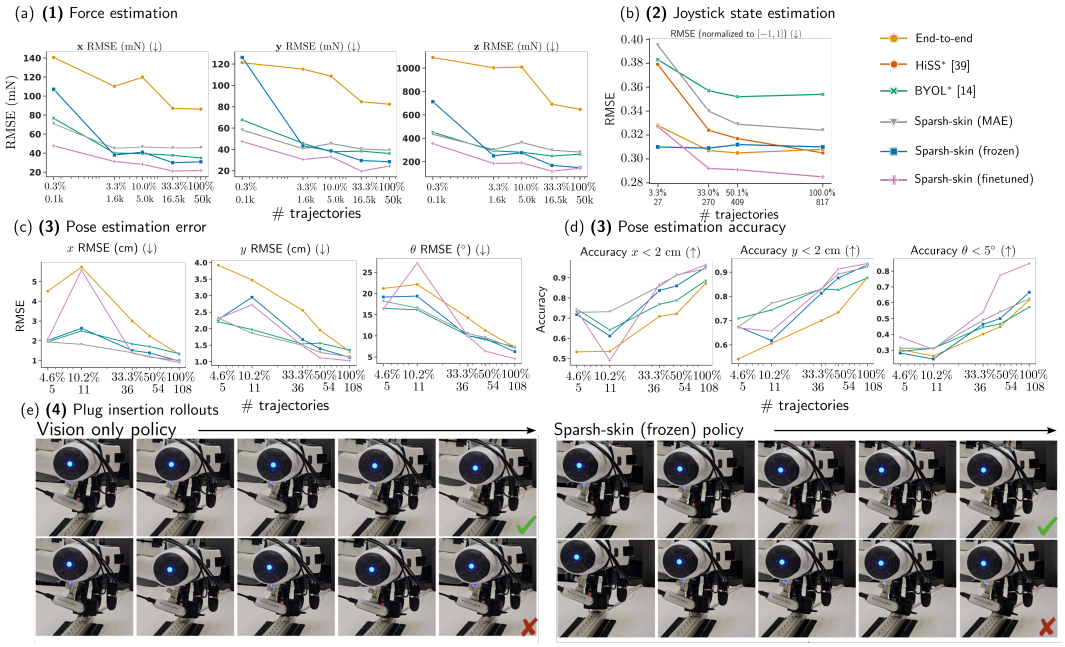


Figure 7: Summary of results comparing `Sparsh-skin` on all tasks. **(a)** Force estimation (RMSE (↓)): BYOL pre-training is less accurate at predicting normal forces. **(b)** Joystick state estimation (↓): `Sparsh-skin` outperforms end-to-end overall and is competitive with HiSS* even when it is given access to only 3.3% of dataset. **(c)** Pose estimation error (↓) and **(d)** Pose estimation accuracy (↑): `Sparsh-skin` (finetuned) has a $\sim 10\%$ improvement over end-to-end for translation and $\sim 20\%$ improvement for rotation. (e) Snapshots of plug insertion policy rollouts (success and failure). Vision-only policy succeeds primarily when the starting position is directly above the socket, while `Sparsh-skin` (frozen) achieves 75% success rate, with failures mainly due to loss of grip when sliding to locate the socket

Furthermore, BYOL* is competitive albeit marginally inferior with respect to `Sparsh-skin` (frozen) as this task tests for instantaneous force decoding. Additional details and results are in the appendix.

(2) Joystick state estimation. We adapt this task from [38] (see Figure 1), as a study of *full-hand* object state estimation. The task is a sequential problem of predicting the joystick states (roll, pitch and yaw) given a short tactile history. In addition to the comparison of `Sparsh-skin` with an end-to-end approach, we also compute the RMSE results from the best reported model in HiSS [38] (denoted as HiSS* in Figure 7(b)). Additional data and pre-processing details are in the appendix.

Results Our model (`Sparsh-skin`) matches baseline (HiSS [38]) performance using full data, despite challenges from jittery teleoperation such as inconsistent touch even with similar joystick maneuvers. Notably, `Sparsh-skin` (frozen) achieves similar performance even with only 3.3% of the data, demonstrating high sample efficiency. `Sparsh-skin` consistently shows lower prediction error across data budgets (Figure 7(b)). Furthermore, `Sparsh-skin` (finetuned) drastically speeds up training, reaching comparable performance to an end-to-end approach in 12k optimization steps versus 220k (a 95% speedup) when using a 33% data budget. An illustration is provided in the Appendix.

(3) Pose estimation. This task tests the ability to track and accumulate slip under the sensors to predict object pose changes $(\mathbf{t}_i^R \triangleq (x, y, \theta)) \in \mathbf{SE}(\mathbf{2})$ using the setup in Figure 6 (middle). We collect 120 trajectories ($\sim$ 30s each), by manually sliding/rotating an object in a range of $\sim (25\text{cm}, 25\text{cm}, 100^\circ)$ under the Allegro hand, tracking ground truth object pose using ArUco tags. These poses in the camera frame are transformed into the robot hand frame and then projected into $\mathbf{SE}(2)$. We use the sequence decoder (Figure 5b) which processes 1-second windows of tactile data (100Hz) and object pose (10Hz). In addition to RMSE, for this task, we also measure performance via pose accuracy (proportion of predictions within 2cm translation and 5° rotation error).

Results (see Figure 7(c)(d)) All representations models pre-trained on play data achieve lower RMSE and higher pose prediction accuracy than the traditional end-to-end approach. In particular for `Sparsh-skin` (finetuned) we find a $\sim 10\%$ improvement over the end-to-end model with the

full dataset for translation, and $\sim$ 20% improvement for rotation. In low data regimes (33%) Sparsh-skin (MAE) outperforms other models since there is a direct correlation between translation and the displacement of the magnetometers on the Xela sensors, while BYOL* shows similar performance to Sparsh-skin (frozen) which maintains $\sim$70% accuracy. Additionally, we observe that allowing in-domain data to fine-tune the Sparsh-skin representations is advantageous, especially for better tracking rotation of the object, which is harder, as it involves torsion.

(4) Policy learning (plug insertion). We train a transformer decoder policy predicting action chunks [39] with Sparsh-skin representations as input for this task. We adapt the insertion task [40, 41, 22] as it is fundamentally tactile requiring touch feedback to observe the alignment state of the plug. The task involves inserting a pre-grasped plug into a fixed socket using a 7-DOF Franka arm and Allegro hand (Fig. Figure 6 right) unlike [22] which used parallel jaw grippers. We collected 100 demonstrations via kinesthetic teleoperation, recording synchronized data: four camera views $(\mathbf{I}_t^{\text{left}}, \dots)$, Allegro tactile readings (z_t), and robot joint states. The arm's initial position was randomized within a $0.05m \times 0.05m \times 0.02m$ volume $\sim$ 10cm above the socket, while the socket position is fixed. The policy predicts sequences of absolute end-effector poses (3D position + axis-angle orientation) $\mathbf{a} \triangleq (\mathbf{T}_t, \mathbf{T}_{t+1}, \dots)$, conditioned on visual and tactile observations but not proprioception (joint states). We evaluate average success rate over 20 trials with randomized start positions, comparing Sparsh-skin variants (V + Sparsh-skin (frozen), V + end-to-end, V + Sparsh-skin (finetuned)) against a vision-only (V) baseline to assess tactile contribution. Further details are in the Appendix.

Model	SR (↑)
VisuoSkin [22]	0.66
Vision only (V)	0.20
End-to-end V+T	0.40
Sparsh-skin V+T (frozen)	0.75
Sparsh-skin V+T (finetuned)	0.70

Table 1: Policy learning (plug insertion): Success rate percentage reported over 20 trials, while ensuring identical initial conditions during each trial for the tested policy variants. VisuoSkin results are obtained from [22]

Results: (see Table 1) We find that policies conditioned on pretrained Sparsh-skin features outperform the end-to-end model. In Figure 7(e) (see supplementary for video), we present snapshots of real-world policy deployments for both vision-only and visuo-tactile Sparsh-skin (frozen) policies.Without tactile modality (vision only), we find that the policy is able to get close to the socket but indefinitely continues to search for it and does not push the plug in, even when it is directly above the socket. Further, we find that this policy tends to keep pushing the plug to the left of the socket, which we hypothesize is due to perceptual aliasing, where the plug incorrectly appears to be right above the socket from the wrist camera. On the other hand, all model variants with access to the tactile modality observe respectable success rates. In qualitative inspection, we find that the policies using Sparsh-skin (V+T frozen) representations slides after making contact with the extension board, while Sparsh-skin (V+T end-to-end) and Sparsh-skin (V+T finetuned) tends to retry by lifting the plug, when mistakes occur. As noted earlier, in comparison with [22, 12] which trains end-to-end visuo-tactile policies, our setup uses a multifinger Allegro hand as the manipulator, where the plug is grasped using three fingers; nevertheless, we find that policies trained with tactile features from Sparsh-skin are competitive.

5 Conclusion

We present Sparsh-skin, a high-performance tactile representation model trained via self supervision for magnetic skins on dexterous hands. Through evaluations across tactile-centric estimation and policy learning tasks, we demonstrate the efficacy of our supervision objective, tokenization, masking strategies and pretraining of the model over a large unlabeled dataset containing $\sim$4 hours of atomic contact interactions with household objects. In experiments, when considering sample efficiency (training on 33% downstream data), we find that Sparsh-skin (frozen) outperforms the end-to-end model baseline by $\sim$56%, our adaptation of the BYOL approach to tactile representation learning by $\sim$28% and the Sparsh-skin (MAE) baseline by $\sim$53%. We believe that Sparsh-skin represents a step toward foundation models for full-hand tactile representations that enables high-dexterity robotics tasks.

Limitations.

We identify the following limitations for the model design and evaluation framework:

1. **Model design:** Although `Sparsh-skin` handles contact dynamics implicitly by learning representations over windows of tactile signal, the data corruption strategy is inherently spatial. Future work may consider explicitly handling temporal correlation learning via temporal prediction tasks.
2. **Pose estimation:** Our pose estimation task is not designed for applications where the hand is not static. While pose estimation in its current iteration is designed to track a 2D pose with a flat fixed hand configuration, real object interactions involve both 3D pose changes, and simultaneously changing hand configurations. There are also exciting avenues to study tasks such as grasp stability prediction [42] and slip prediction.
3. **Manipulation policy:** Although our experiments with the real physical system support the hypothesis that tactile information from magnetic skin sensors improves policy performance, we still need to assess their ability to generalize. Visuo-tactile policies can overfit to the specific tactile signatures of objects and environments used in data collection, which raises an open question: how can we achieve generalization across diverse tactile feedback signals while maintaining data efficiency?

Acknowledgments

The authors thank Unnat Jain, Tarasha Khurana, Hung-Jui Huang, Jessica Yin, Changhao Wang, Luis Pineda, Mrinal Kalakrishnan and Youngsun Wi for helpful discussion and reviews of the paper. This work is supported by Meta FAIR labs.

References

[1] 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*, pages 2549–2564. PMLR, 2023. URL https://proceedings.mlr.press/v229/qi23a/qi23a.pdf.

[2] B. Romero, H.-S. Fang, P. Agrawal, and E. Adelson. EyeSight Hand: Design of a Fully-Actuated Dexterous Robot Hand with Integrated Vision-Based Tactile Sensors and Compliant Actuation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860, 2024. doi:10.1109/IROS58592.2024.10802778. URL https://arxiv.org/abs/2408.06265.

[3] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, and M. Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. 2024. URL https://openreview.net/forum?id=xYJn2e1uu8.

[4] M. Yang, C. Lu, A. Church, Y. Lin, C. Ford, H. Li, E. Psomopoulou, D. A. Barton, and N. F. Lepora. AnyRotate: Gravity-Invariant In-Hand Object Rotation with Sim-to-Real Touch. *arXiv preprint arXiv:2405.07391*, 2024. URL https://arxiv.org/abs/2405.07391.

[5] 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*, 5(3):3838–3845, 2020. doi:10.1109/LRA.2020.2977257. URL https://ieeexplore.ieee.org/document/9018215.

[6] W. Yuan, S. Dong, and E. Adelson. GelSight: High-resolution Robot Tactile Sensors for Estimating Geometry and Force. *Sensors: Special Issue on Tactile Sensors and Sensing*, 17(12): 2762 – 2782, November 2017. URL https://www.mdpi.com/1424-8220/17/12/2762.

[7] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934, 2018. doi:10.1109/IROS.2018.8593661. URL https://dl.acm.org/doi/10.1109/IROS.2018.8593661.

[8] S. Wang, Y. She, B. Romero, and E. H. Adelson. GelSight Wedge: Measuring High-Resolution 3D Contact Geometry with a Compact Robot Finger. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. URL https://dl.acm.org/doi/10.1109/ICRA48506.2021.9560783.

[9] T. P. Tomo, A. Schmitz, W. K. Wong, H. Kristanto, S. Somlor, J. Hwang, L. Jamone, and S. Sugano. Covering a Robot Fingertip With uSkin: A Soft Electronic Skin With Distributed 3-Axis Force Sensitive Elements for Robot Hands. *IEEE Robotics and Automation Letters*, 3 (1):124–131, 2018. doi:10.1109/LRA.2017.2734965. URL https://ieeexplore.ieee.org/document/8000399.

[10] T. P. Tomo, W. K. Wong, A. Schmitz, H. Kristanto, A. Sarazin, L. Jamone, S. Somlor, and S. Sugano. A modular, distributed, soft, 3-axis sensor system for robot hands. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 454–460. IEEE, 2016. URL https://ieeexplore.ieee.org/document/7803315.

[11] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *5th Annual Conference on Robot Learning*, 2021. URL https://proceedings.mlr.press/v164/bhirangi22a/bhirangi22a.pdf.

[12] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch. *arXiv preprint arXiv:2409.08276*, 2024. URL https://arxiv.org/abs/2409.08276.

[13] PaXini. ITPU multi-dimensional tactile sensing unit, 2025. URL https://www.paxini.com/ax.

[14] I. Guzey, B. Evans, S. Chintala, and L. Pinto. Dexterity from Touch: Self-Supervised Pre-Training of Tactile Representations with Robotic Play, 2023. URL https://arxiv.org/abs/2303.12076.

[15] T. Wu, J. Li, J. Zhang, M. Wu, and H. Dong. Canonical Representation and Force-Based Pretraining of 3D Tactile for Dexterous Visuo-Tactile Policy Learning, 2024. URL https://arxiv.org/abs/2409.17549.

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[17] J. A. Fishel and G. E. Loeb. Sensing tactile microvibrations with the BioTac — Comparison with human sensitivity. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1122–1127, 2012. doi:10.1109/BioRob.2012.6290741. URL https://ieeexplore.ieee.org/document/6290741.

[18] B. Liang, W. Liang, and Y. Wu. Tactile-Guided Dynamic Object Planar Manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3203–3209, 2022. doi:10.1109/IROS47612.2022.9981270. URL https://ieeexplore.ieee.org/document/9981270.

[19] J. Wei, S. Cui, J. Hu, P. Hao, S. Wang, and Z. Lou. Multimodal Unknown Surface Material Classification and Its Application to Physical Reasoning. *IEEE Transactions on Industrial Informatics*, 18(7):4406–4416, 2022. doi:10.1109/TII.2021.3126601. URL https://ieeexplore.ieee.org/document/9609678.

[20] J. Gao, Z. Huang, Z. Tang, H. Song, and W. Liang. Visuo-Tactile-Based Slip Detection Using A Multi-Scale Temporal Convolution Network, 2023. URL https://arxiv.org/abs/2302.13564.

[21] H. Li, S. Dikhale, J. Cui, S. Iba, and N. Jamali. HyperTaxel: Hyper-Resolution for Taxel-Based Tactile Signals Through Contrastive Learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7499–7506, 2024. doi:10.1109/IROS58592.2024.10802001. URL https://ieeexplore.ieee.org/document/10802001.

[22] V. Pattabiraman, Y. Cao, S. Haldar, L. Pinto, and R. Bhirangi. Learning Precise, Contact-Rich Manipulation through Uncalibrated Tactile Skins. *arXiv preprint arXiv:2410.17246*, 2024. URL https://arxiv.org/abs/2410.17246.

[23] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[24] J. Zhao, Y. Ma, L. Wang, and E. Adelson. Transferable Tactile Transformers for Representation Learning Across Diverse Sensors and Tasks. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=KXsropnmNI.

[25] G. Cao, J. Jiang, D. Bollegala, and S. Luo. Learn from Incomplete Tactile Data: Tactile Representation Learning with Masked Autoencoders. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10800–10805. IEEE, 2023. URL https://ieeexplore.ieee.org/document/10341788.

[26] 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 *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=7yMZAUkXa4.

[27] F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024. URL https://ieeexplore.ieee.org/document/10658351.

[28] V. Dave, F. Lygerakis, and E. Rueckert. Multimodal visual-tactile representation learning through self-supervised contrastive pre-training. *arXiv preprint arXiv:2401.12024*, 2024.

[29] A. George, S. Gano, P. Katragadda, and A. B. Farimani. VITaL Pretraining: Visuo-Tactile Pretraining for Tactile and Non-Tactile Manipulation Policies, 2024. URL https://arxiv.org/abs/2403.11898.

[30] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. URL https://papers.nips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, 2023. URL https://arxiv.org/abs/2304.07193.

[32] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022. URL https://proceedings.mlr.press/v162/baevski22a/baevski22a.pdf.

[33] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020. URL https://arxiv.org/abs/1910.03135.

[34] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[35] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.pdf.

[36] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/papers/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.pdf.

[37] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

[38] R. Bhirangi, C. Wang, V. Pattabiraman, C. Majidi, A. Gupta, T. Hellebrekers, and L. Pinto. Hierarchical state space models for continuous sequence-to-sequence modeling. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2025. URL https://dl.acm.org/doi/10.5555/3692070.3692223.

[39] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.016.

[40] 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 *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6437–6443. IEEE, 2021.

[41] Y. Wu, Z. Chen, F. Wu, L. Chen, L. Zhang, Z. Bing, A. Swikir, S. Haddadin, and A. Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. *arXiv preprint arXiv:2409.11047*, 2024.

[42] 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*, 3(4):3300–3307, 2018. doi:10.1109/LRA.2018.2852779. URL https://arxiv.org/abs/1805.11085.

[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. URL https://arxiv.org/abs/2010.11929.

Appendix

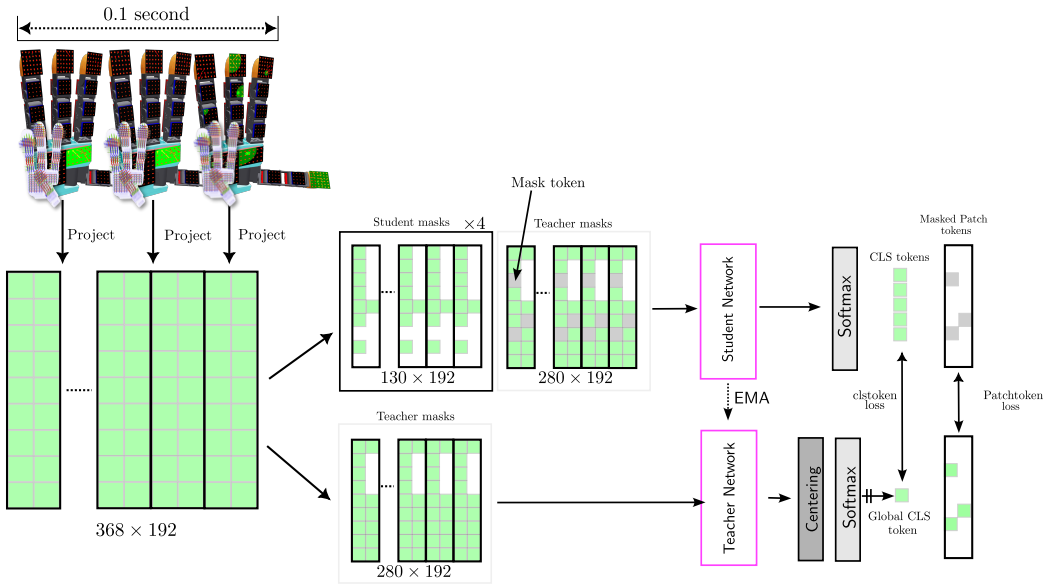


Figure 8: `Sparsh-skin` block diagram for self-supervised learning of skin representations. Our approach follows the student-teacher framework and loss functions used in self-distillation. However, we adapt the transformer input tokenization to accommodate time-series Xela data.

A `Sparsh-skin` self-supervision details

A.1 Training details

We train `Sparsh-skin` on 8 Nvidia A-100 (80G) GPUs. To monitor learning, we use reconstruction online probe and classification via linear probing. We use AdamW optimizer and use a linear rampup followed by a cosine schedule as the learning scheduler. Further, we find that tuning momentum value as well as the weight decay factor was important in observing training convergence. Additional information of hyperparameters is detailed in Table 2.

Architecture	ViT-Tiny (adapted)
Embedding dim	192
EMA decay	[0.994, 1.0]
LR	1e-4
Batch size	64

Table 2: Training hyperparameters for `Sparsh-skin`. All models run for 500 epochs with optimizer AdamW, a weight decay cosine schedule from 0.04 to 0.4, and a learning rate warmup of 30 epochs.).

A.2 Architecture details

Our encoder model is a modified version of Vision Transformers [43]. Specifically, we adapt the tokenization of the time-series Xela with sensor pose data. After flattening the 3D-axis magnetic reading per magnetometer (368) and concatenating their corresponding pose in chunks of 0.1 second, the inputs $x \in \mathbb{R}^{10\times368\times6}$ are tokenized through a linear projection to the dimension d of the representation $f_{linear}(x) \in \mathbb{R}^{368\times d}$. We use a tiny model with $d = 192$. We add a learnable embedding to identify different types of xela pads (palm, phalanges and fingertips). Then, we

construct different cropped view of the data, two global views and eight local views. We mask sensor data from contiguous blocks by removing those sensors from the input. For the local view we retain between 10% and 40% of the tactile signal, whereas for the global views we retain 40% to 100%. An illustration of the masking and diagram block of the pipeline for self-supervised learning of Xela representations is shown in Figure 8.

The student and teacher share the same encoder and projector head architecture, both initialized with the same weights. The projector head corresponds to a 3-layer MLP with an output dimension of $k = 65536$. We use the projection head for the proxy prediction task to distill knowledge to match output distributions over k dimensions between student and teacher networks. The student network is updated via back-propagation, while the teacher network is updated at a lower frequency via exponential moving average (EMA) on the student weights. We pass the global and local views to the student encoder, while the teacher only has access to the global views. The register tokens from global/local views are passed through the projection head. For the teacher only, the output is also centered and sharpened via softmax normalization.

B Additional task details

We provide additional information about the decoder architectures for each task, as well as additional results to highlight the performance on downstream tasks when using frozen or fine-tuned `Sparsh-skin` representations. Also, please refer to Table 3 for details on labeled data curation for evaluation tasks.

Task	Dataset	Size	Collector	Label
Force estimation	Normal load (indenter: sphere, flat)	50k datapoints	Robot	3-axis force
Pose estimation	Object sliding	108 trajectories	Human	Object pose $\mathbf{SE}(2)$
Joystick state estimation	Joystick motion	817 trajectories	Human	Normalized roll, pitch, yaw
Plug insertion	Demonstrations	100 trajectories	Human	Absolute EE pose

Table 3: Datasets for evaluating `Sparsh-skin` representations on downstream tasks.

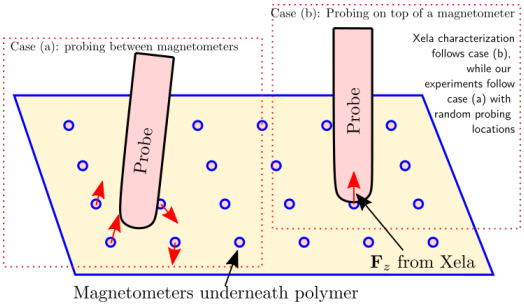
B.1 Force estimation

`Sparsh-skin` features are pooled via attentive pooling to obtain a full-hand representation $z_{hand} \in \mathbb{R}^d$. The force decoder consist of shallow 2-layer MLP with 3 outputs regressing to normalized force for each axis.

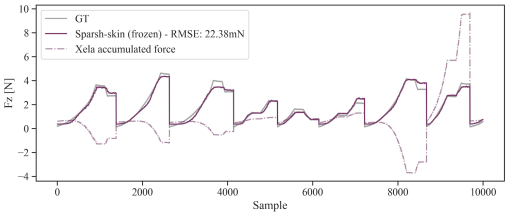
In Figure 9 we illustrate the data protocol followed for force estimation, which we note is different from the protocol that is usually followed for force characterization of tactile sensors. We note that we indent the tactile sensor pads at both, positions on top of the sensor as well as positions in between magnetometer locations, while choosing these positions randomly. This results in cases where the probe may slide and present slightly uneven force outputs. Specifically, in figure 9(b) we note that `Sparsh-skin` predicts the correct normal forces, while accumulation (mean) of normal forces from the magnetometers over the sensor pad results in inconsistent force outputs compared to ground truth.

In Figure 10, we present the correlation metrics between ground truth and predicted forces on test data for decoders trained with a 33% data budget. The results show that end-to-end training leads to overfitting, resulting in poor generalization to unseen strokes and essentially random normal force predictions. In contrast, using `Sparsh-skin` (frozen) representations yields better fitting, which can be further improved by adapting these representations to in-domain data.

In Figure 11, we present a comparison between ground-truth testing strokes (normal loading sequences) and their reconstructed counterparts, obtained by passing Xela data through the frozen force decoder to recreate the sequences. The forces estimated via `Sparsh-skin` (frozen) are able to capture increasing/ decreasing changes in the normal loading, as opposed to the end-to-end model. Shear



(a) Illustration of data collection procedure followed in our setup vs procedure followed during Xela force calibration



(b) Illustration of how the Normal Xela force output is inconsistent with GT force obtained from F/T probe, and the corresponding prediction from `Sparsh-skin`

Figure 9: Illustration of data collection protocol follwed for Force estimation with Xela sensors

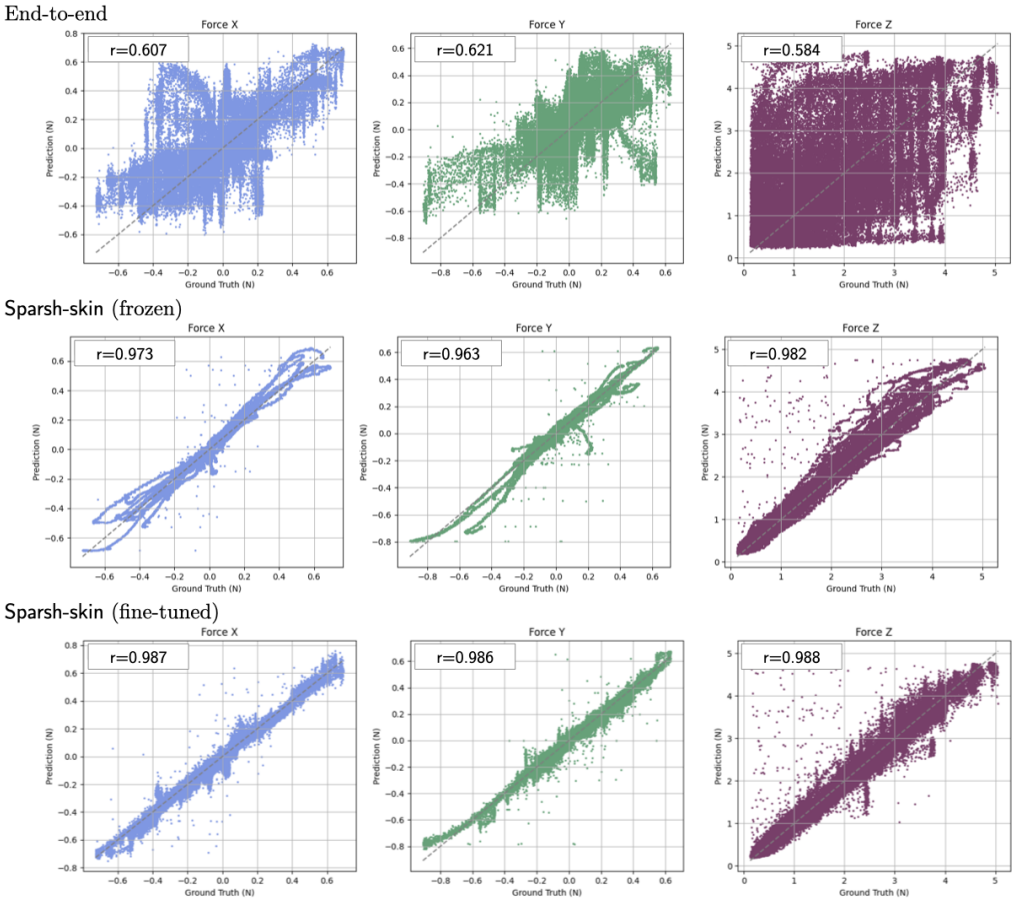


Figure 10: Correlation between ground truth and predicted forces on unseen normal loading with an indenter on Xela sensors.

from skin representations is not as accurate as normal force prediction, but the trend of the tangential forces matches the ground truth.

B.2 Joystick state estimation

For this task, we highlight that when we train decoders using pretrained representations as the input, the convergence rate of the validation RMSE is significantly higher (see Figure 12) than training the decoder using raw observations through uninitialized models. Specifically observe that `Sparsh-skin`

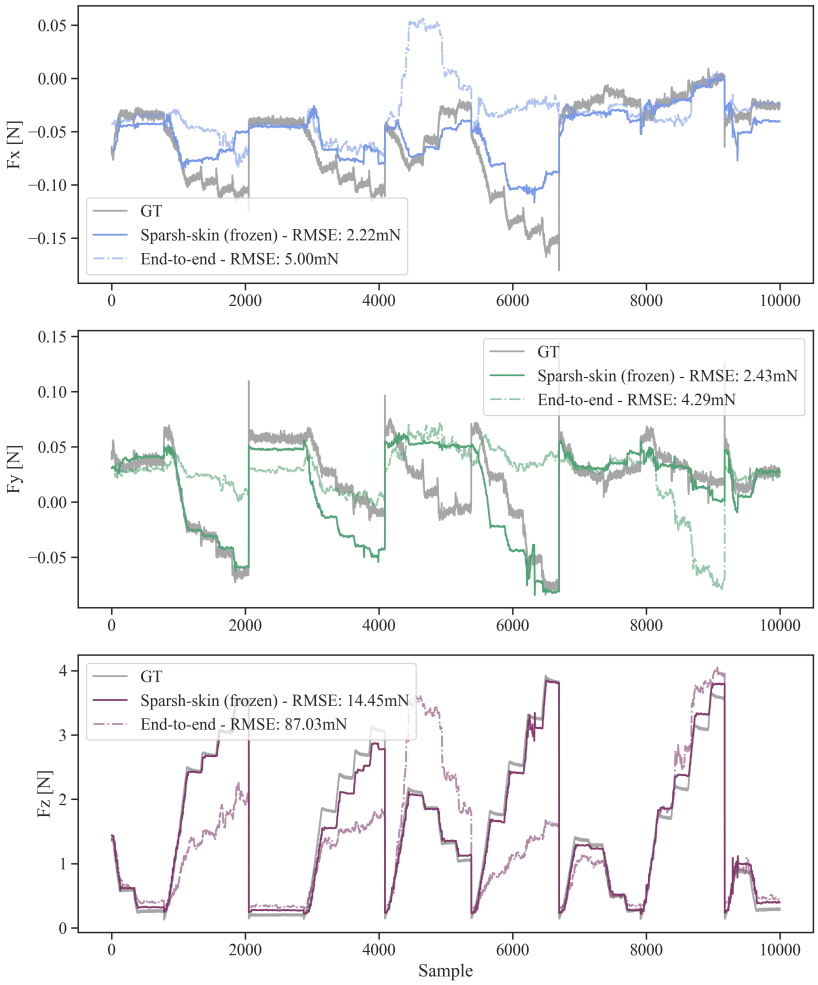


Figure 11: Ground truth tangential and normal force from test strokes with flat indenter (gray) and force sequence reconstruction from `Sparsh-skin` (frozen) and end-to-end model.

(fine-tuned) is able to reach performance on par with end-to-end pretrained model within 12.9k optimization steps.

B.3 Pose estimation

In this task, we aim to predict the object pose over 1-second trajectories. Xela observations at 100Hz are converted into tactile representation tokens at the output frequency using `Sparsh-skin` in a cascaded manner. Following attentive pooling, a single-layer transformer block is applied to reason about the 1-second context window of full-hand tactile features.

Figure 13 compares ground-truth test pose sequences with their reconstructed counterparts, obtained from task models trained on 100% and 33% of the available data. The results show that fine-tuning `Sparsh-skin` on the full dataset yields higher accuracy in estimating object pose changes over time compared to traditional end-to-end approaches. Moreover, even with a drastic reduction in labeled samples (to 33%), the model still achieves relatively good performance, particularly in tracking translation changes. Furthermore, for this task, we also visualize that this tasks requires *full-hand* sensing. For instance, in Figure 14, we observe that when we use `Sparsh-skin` by removing palm sensing on the Xela hand results in $\geq 10\%$ drop in pose tracking performance.

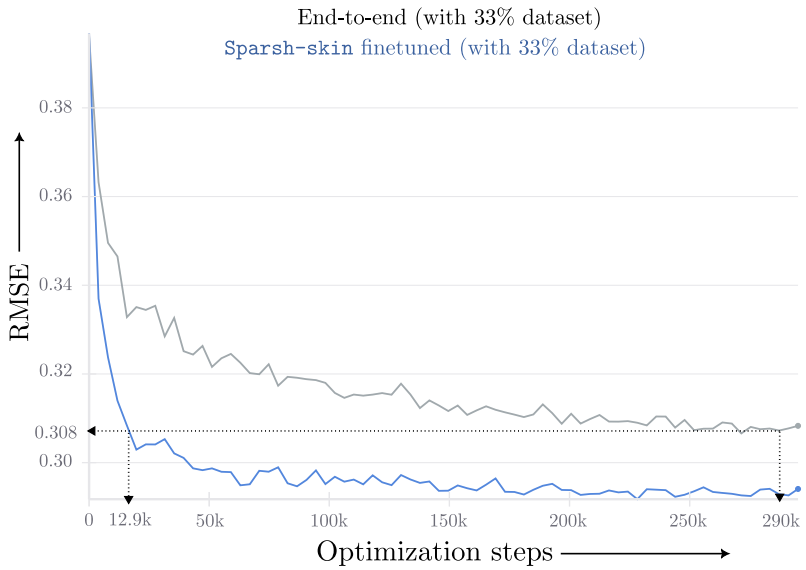


Figure 12: Validation RMSE convergence rates between `Sparsh-skin` fine-tuned and `Sparsh-skin` end-to-end: We find that `Sparsh-skin` fine-tuned allows the model to generalize and learn the patterns required to infer joystick states significantly faster during training.

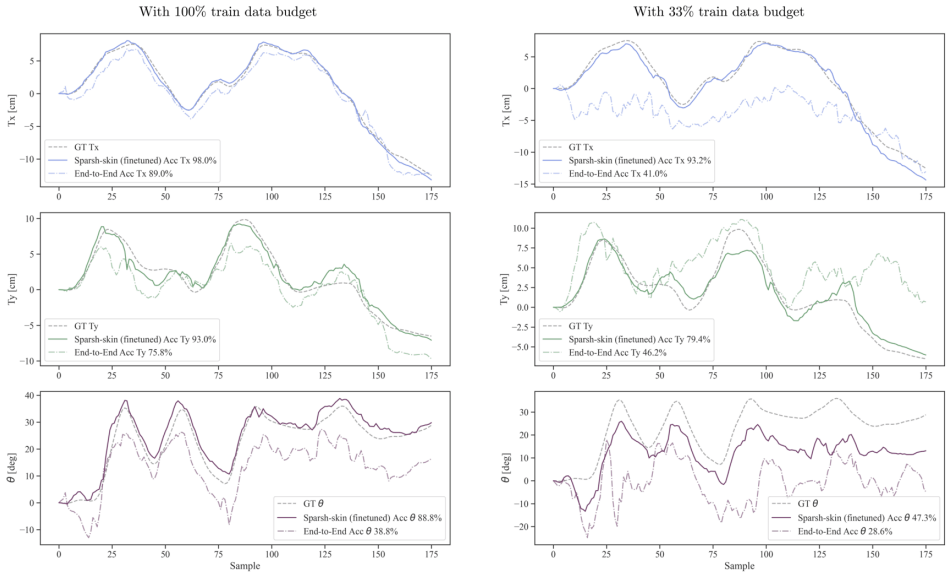


Figure 13: Ground truth pose sequence for object in test set and reconstructed trajectory via end-to-end and `Sparsh-skin` (finetuned) representations. (left) Task decoders trained with 100% of train data budget, corresponding to 108 sequences. (right) Task decoders trained with 33% of train sequences.

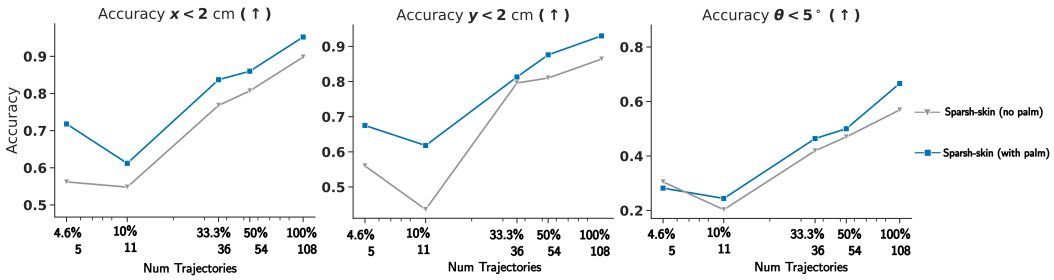


Figure 14: Comparison of pose estimation accuracy of `Sparsh-skin` with and without palm sensing.

B.4 Policy learning (plug insertion)

For this task, we use a transformer decoder to predict action sequences given camera and tactile observations. Figure 15 illustrates the architecture of the transformer decoder used in this work. Images are encoded using a Resnet18 CNN, which are trained from scratch to produce image features, while the tactile observations are processed through `Sparsh-skin`. Further, a learnable token (CLS / action) token is also concatenated with the observation tokens. After processing through the transformer, we extract the action token, which is then passed into a small 2-layer MLP to predict a sequence of actions. For this task, follow an *receding-horizon* control approach, where we choose a prediction action sequence length of 16, of which 8 actions are executed, given only the observations from the current timestep.

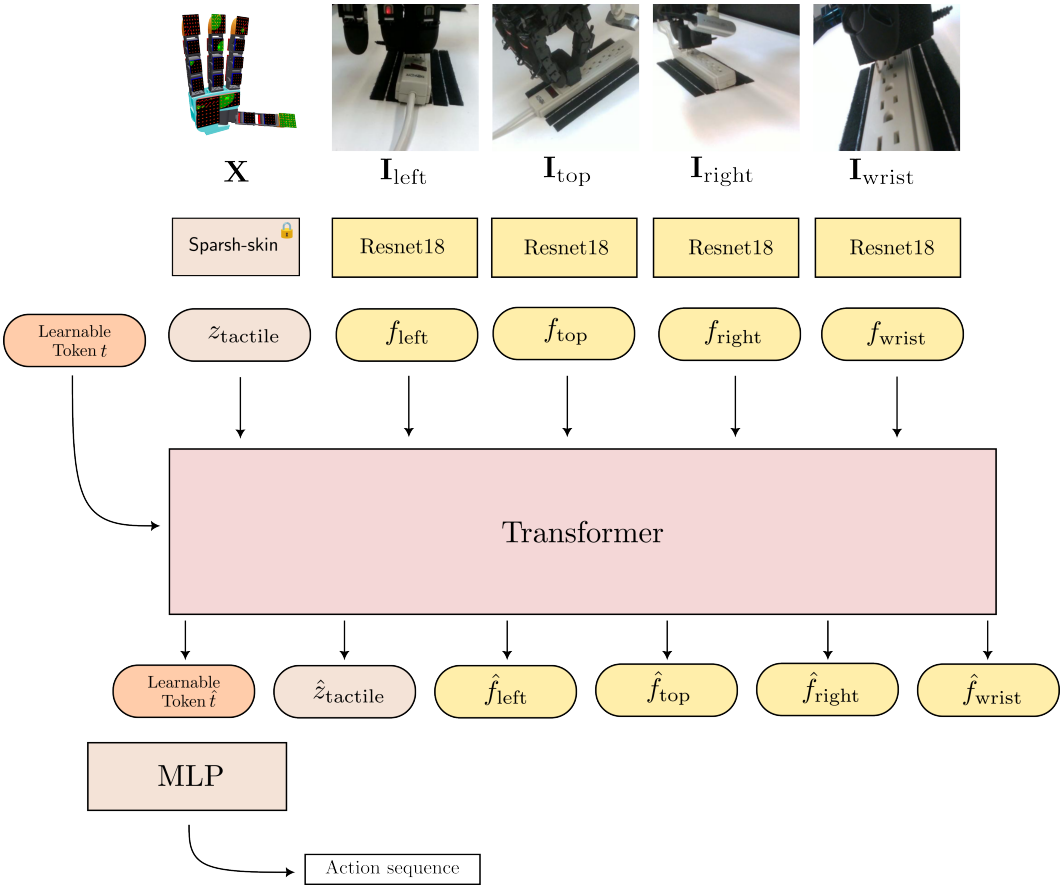


Figure 15: Illustration of the policy architecture: We use a transformer to fuse information from visual and tactile modalities, through the use of a learnable action token, which is then used to subsequently predict action sequences.