Appendix

A Broader related work

Self Supervised Learning - In this section, we detail recent developments in masking-based self-supervised learning approaches:

Masked Image Modeling (MIM) is the strategy of corrupting a data sample by significantly masking a portion of the sample and training a model to recover the missing portion, conditioned on the corrupt sample. It has become a prominent framework in SSL with the success of [23, 68]. An important design consideration here is the output space of the model for supervision, which can be either raw pixels [23, 69] or an alternative representation space [70, 71, 72, 68]. While training Masked auto-encoders is simple, these models are comparatively sample inefficient during training [43].

Self-distillation [73] is the idea of training two (usually identical) networks such that a *student* network learns to predict the output representations of a *teacher* [74] network via a small predictor network when observing augmentations of the same data sample. It has been shown to improve performance significantly even in the case of abundant data [75]. While degenerate constant representations is a concern, a common strategy is to stop gradient backpropagation [25] through the teacher network and employ momentum based weight updates [22]. A concrete instance is DINO [42] utilizing ViTs [76] as the student & teacher encoder networks. More recently DINOv2 [45] improved downstream performance significantly by combining self-distillation and MIM.

Joint-Embedding Predictive Architectures (JEPA) [46] share similarities with MIM, as both rely on masking. However, the JEPA framework conceptually prescribes two key changes: a) information restoration in a latent representation space, rather than in input space (pixels or tokens) b) prediction of latent embedding conditioned on the *masking parameters*. This framework has had success across various modalities, including audio [77, 78], images [43, 79], and pointclouds [80]. Notably, in this paper we consider masking strategies from I-JEPA [43] and V-JEPA [44]. I-JEPA utilizes a spatial block-masking strategy and V-JEPA utilizes tube-masking [81] with varying aspect ratios for learning representations efficiently in latent space circumventing decoding unnecessary pixel-level details.

Representation learning in robotics - Pretraining models for multi-task capability has become popular recently, especially after the success of self-supervised learning (SSL) in computer vision tasks like object classification, segmentation, depth estimation, and image generation. These tasks, while typically tested on computer vision datasets, are also very common in robotics. The idea of using these pre-trained representations for robot learning was initially explored in [82], showing that pre-trained visual representations can sometimes even be better than using ground-truth state representations for training control policies.

Generative SSL via masked image modeling (MIM) [83, 84] has shown successful transfer of pre-trained representations from in-the-wild data to real-robot scenarios, enabling basic motor skills such as reaching, pushing, and picking. Furthermore, many other works investigate contrastive learning approaches to learning general visual representations in robotics [85, 86]. These methods usually employ a pixel reconstruction objective based on a time-contrastive objective or focus on contrasting video clips leveraging natural language for video-language alignment.

The field has been moving towards finding general-purpose representations that work well across a wide range of problems in robot manipulation learning. Voltron [87], is a framework for language-driven visual representation learning for robotics that combines both masked auto-encoding and contrastive learning techniques, focusing on multi-task performance. This model is trained to learn representations that capture both low-level spatial reasoning and high-level semantic understanding by using language supervision from human videos.

Tactile sensor simulation - Multiple simulators have been proposed for vision-based tactile sensors such as [88, 89, 90, 91, 92] with the hope of sim2real generalization of learned policies [93]. However, many of these methods are either limited to marker-based tactile sensors [93], or narrow tasks [94, 95]. Certain other methods [39] also leverage simulated data to train multi-modal representations. However,

	Arch.	EMA decay	LR	Batch size
Sparsh (MAE)	ViT-B/14	N/A	1e-4	100
Sparsh (DINO)	ViT-B/14	0.998	1e-4	150
Sparsh (IJEPA)	ViT-B/14	0.996	6.25e-4	150
Sparsh (VJEPA)	ViT-B/14	0.996	6.25e-4	150

Table 2: Training hyperparameters for Sparsh models. All models run for 150 epochs with optimizer AdamW, a weight decay cosine schedule from 0.04 to 0.4, and a learning rate warmup of 30 epochs.).

in general we find that tactile simulators are still unable to model shadows, as well as real-world per-sensor-instance discrepancies, hampering their potential use for representation learning.

B Touch representation and self-supervision details

To ensure fair evaluation of all models, our SSL algorithms are largely adapted from official MAE, IJEPA, VJEPA, DINO codebases.

B.1 Training details

We train all models on 8 Nvidia A-100 (80G) GPUs. In addition to training losses, to monitor training progress, we rely on online probes. Specifically, we find that for joint embedding predictive architectures, the training losses are not indicative of model convergence during optimization; therefore, proxy metrics such as reconstruction quality are helpful. For all methods, we utilize DPT [96] based decoders to decode the tactile representations back into tactile images. See Figure 5 for some examples of tactile reconstructions from Sparsh embeddings. All encoder models are trained for 150 epochs. 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 without collapse. Additional information of hyperparameters is detailed in Table 2.

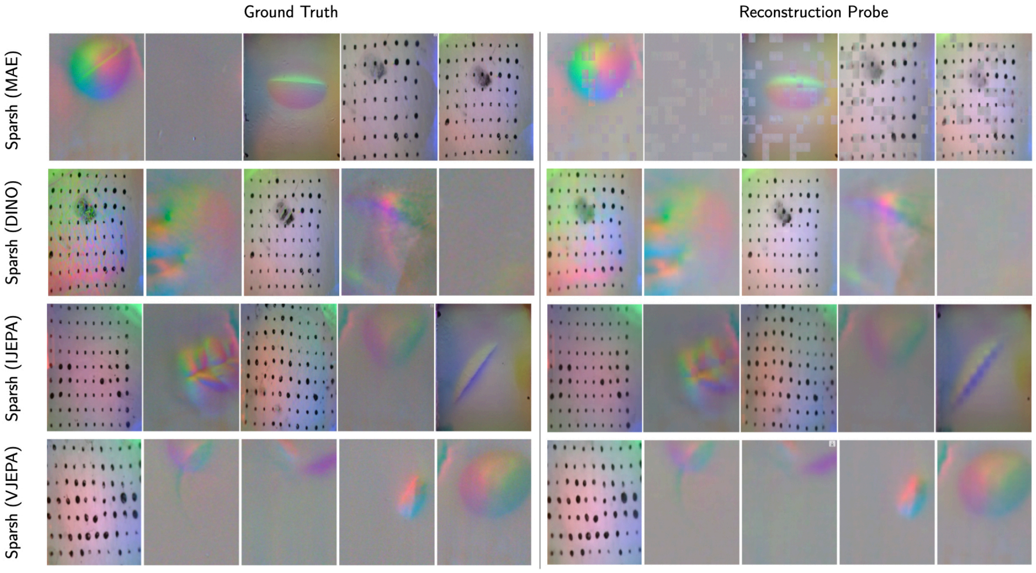

Figure 5: Visualization of reconstructed tactile images using the online probe to monitor SSL training of Sparsh models.

	Sparsh (MAE)	Sparsh (DINO)	Sparsh (IJEPA)	Sparsh (VJEPA)
N. parameters	86254848	86255616	86386944	86537472
FPS	104	112	112	60

Table 3: Number of parameters and inference time for Sparsh backbones

B.2 Architecture details

All encoder models are Vision Transformers (ViT) [76]. Although the main encoder models use ViT-B/14 as the standard architecture, following [43] we use a small ViT as the predictor network. All the models are pretrained without a [cls] token. For DINO, which decodes the [cls] token into classes, we repurpose ViT registers [97] to predict classes. In Table 3 we report the number of parameters for each encoder and their respective inference times.

Tactile images with a stride of 5 i.e., $\mathbf{I}_t \oplus \mathbf{I}_{t-5} \in \mathbb{R}^{h \times w \times 6}$ are concatenated along the channel dimension before the background is removed and reshaped to 224×224 for ViT processing. We choose a stride of 5 as consecutive images are similar due to high sensor sampling rates, and to match the slip detection window in humans. Ablating the effect of the input image and patch resolution may be important for better performance and is left for future work.

B.3 Dataset splits

We use three available datasets for training Sparsh, namely YCB-Slide [9], Touch-and-Go [20] and Object Folder [37]. The YCB-Slide dataset consist of human sliding interactions with 10 YCB objects. Each object has 5 trajectories, with around 3500 frames each from DIGIT sensors with different optical characteristics (180k frames in total). For each object, we dedicate four trajectories for training and the last one for validation. Touch-and-Go consists of discrete human contact interactions with in-the-wild objects, using a GelSight sensor. It consist of 140 videoclips and plain files with labels for the frames with a clear contact. We use all frames (220k) in the videoclips since we do not rely on labeled data for SSL training, from which 70% is used for training and the remaining for validation. The data used from ObjectFolder consist of 81k frames of robot discrete contact interactions with objects in a controlled setting. We also use a train/val split of 70/30.

To complement the dataset, we collected Touch-Slide with additional human sliding interactions on toy-kitchen objects with the DIGIT sensor. We use 9 objects, shown in Figure 6 and collected 5 trajectories for each, generating 180k frames in total.

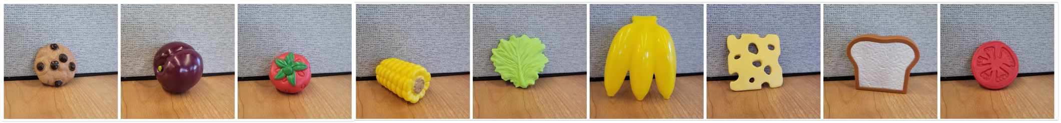

Figure 6: Set of objects for collecting sliding contact trajectories in the Touch-Slide dataset.

C TacBench tasks and evaluation details

C.1 Probe details

The parameters of the model updated via EMA (target encoder for Sparsh (IJEPA) and Sparsh (VJEPA), teacher network for Sparsh (DINO), encoder from Sparsh (MAE)) are fixed and used for evaluation. The features are pooled via attentive pooling for tasks that require global representations, such as slip detection, resultant force estimation, and classification tasks. For tasks that require dense reasoning, we use DPT decoders [96] to decode patch representations into full input resolution quantities such as normal and shear force fields, and reconstructed tactile images. See Figure **??** for a visual illustration of the probe architectures.

We follow attentive probing[44, 52] to assess the capabilities of tactile representations on the benchmark, as this approach allows us to determine what representations capture from self-supervision alone. For most tasks – except force field visualization and policy learning – in the benchmark, we freeze Sparsh and train a cross-attention module (hyperparameters in Table 4) followed by a light 2-layer MLP probe supervised, using the labeled dataset for each task.

Parameter	Setting
Embedding dimension	784
N heads	12
MLP ratio	4.0
Depth	1
Layer normalization	Yes

Table 4: Attentive pooling hyperparameters used for evaluation protocol of representation in downstream tasks.

C.2 [T1] Force estimation

After attentive pooling, the tactile features with 768 dimensions are passed to a 2-layer MLP with 192 and 3 units respectively, to get the 3-axis force estimations. Two independent force decoders are trained using DIGIT and GelSight-mini data respectively, using the sharp and sphere probe data during training and the flat indenter data for testing. The target forces are normalized to be ± 1.0 and scaled back after prediction. We train the force decoder using Adam optimizer with 1e-4 learning rate.

DIGIT. In Table 5 we report the average RMSE over 25k samples of unseen DIGIT data for the force estimation task. We report metrics for each Sparsh model and the E2E approach, under four different budgets of training data. We also provide a 95% confidence interval to ground the error ranges of each model.

In Figure 7 we plot the friction cone from the test data, where the colormap represents the error in mN for each axis. Note that E2E exhibit larger errors (around 500mN) for the tangential component and they are more predominant as the normal force increases. In contrast, the top model Sparsh (DINO) estimates with low error ($<$ 100mN) in general across the whole range of tangential and normal forces.

Model	Full dataset (50k)	$1/3$ dataset	$1/10$ dataset	$1/100$ dataset
E2E	39.34 [39.21, 39.48]	61.42 [61.12, 61.72]	98.22 [97.61, 98.84]	187.51 [185.51, 188.51]
Sparsh (MAE)	36.61 [36.51, 36.71]	45.96 [45.80, 46.12]	58.55 [58.31, 58.79]	115.39 [114.69, 116.09]
Sparsh (DINO)	**36.09** [36.01, 36.17]	**44.03** [43.87, 44.19]	**51.89** [51.69, 52.10]	**97.95** [97.36, 98.52]
Sparsh (IJEPA)	40.27 [40.16, 40.38]	60.04 [59.72, 60.34]	86.57 [86.06, 87.08]	130.37 [129.59, 131.15]
Sparsh (VJEPA)	39.38 [39.30, 39.47]	56.34 [56.07, 56.62]	76.11 [75.67, 76.55]	130.83 [130.29, 131.38]

Table 5: Root Mean Squared Error (mN) and 95% confidence interval for force estimation with DIGIT data. All models were evaluated on flat indenter data over 25k test samples.

GelSight. In Table 6 we report the average RMSE over 25k samples of unseen GelSight data and the corresponding 95% confidence interval. Notice from Figure 8 that the majority of errors are localized around the dynamic shear region. It is worth noting that the errors associated with Sparsh (DINO) remain below 150mN, whereas E2E exhibits higher errors, particularly in the estimation of normal forces.

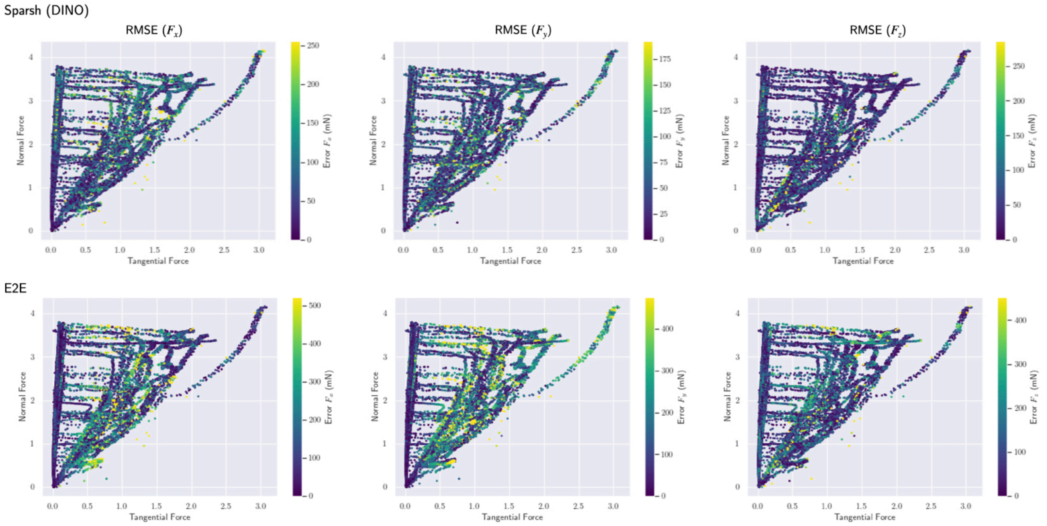

Figure 7: Friction cone of test data and RMSE (mN) for force estimation task with DIGIT sensor.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	57.21 [56.44, 57.98]	59.09 [58.15, 60.04]	57.43 [56.44, 58.42]	82.42 [80.98, 83.86]
Sparsh (MAE)	22.72 [22.27, 23.17]	**23.28** [22.83, 23.72]	33.56 [33.04, 34.08]	78.98 [77.74, 80.21]
Sparsh (DINO)	**20.25** [19.85, 20.65]	23.79 [23.40, 24.18]	**32.17** [31.67, 32.67]	**53.43** [52.69, 54.17]
Sparsh (IJEPA)	27.91 [27.37, 28.44]	35.20 [24.57, 35.82]	44.93 [44.13, 45.73]	91.81 [90.76, 92.86]
Sparsh (VJEPA)	33.26 [32.67, 33.84]	34.07 [33.39, 34.75]	42.35 [41.60, 43.10]	80.36 [79.26, 81.47]

Table 6: Root Mean Squared Error (mN) and 95% confidence interval for force estimation with GelSight-mini data. All models were evaluated on flat indenter data over 25k test samples.

C.3 [T1A] Force field visualization

Since rendering the force field is a dense prediction task, we do not apply the attentive probing protocol. Instead, we follow DPT [53], training a CNN encoder with reassemble-fusion modules at layers 2,5,8,11 of the Sparsh encoder to progressively upsample the representations to obtain a fine-grained prediction of the force field. After the reassemble-fusion modules, we attach two task-specific task head, for normal and shear field prediction.

Since for markerless vision-based sensors is not trivial to get ground truth of the force field, we turn to unsupervised learning. Depth estimation and optical flow are analogous to the estimation of normal and shear force fields, areas where the computer vision community has proposed several unsupervised methodologies [55, 56, 57, 58, 54]. We borrow ideas of unsupervised monocular depth estimation, where from two tactile images I_t and I_{t-n}, we learn a pose estimator for getting the transform between frames. With the sensor intrinsic K, we map image I_t from pixel space to camera plane, translate estimated depth D_t, apply transform from t to $t-n$, and transform back to image plane to get $\hat{I}_{t-n}$. We supervised based on the reprojection error, MSE between I_{t-n} and predicted $\hat{I}_{t-n}$. To reconstruct the shear field, we transfer ideas from unsupervised optical flow, where we warp the features of image I_t to I_{t-n} based on the estimated flow and compute a photometric consistency loss that encourages the estimated flow(shear) to align image patches with a similar appearance. This loss is a linear combination of the Charbonnier loss and the structural similarity (SSIM) between I_{t-n} and $\hat{I}_{t-n}$. We also add a smoothness loss that acts as a regularization term, encouraging the

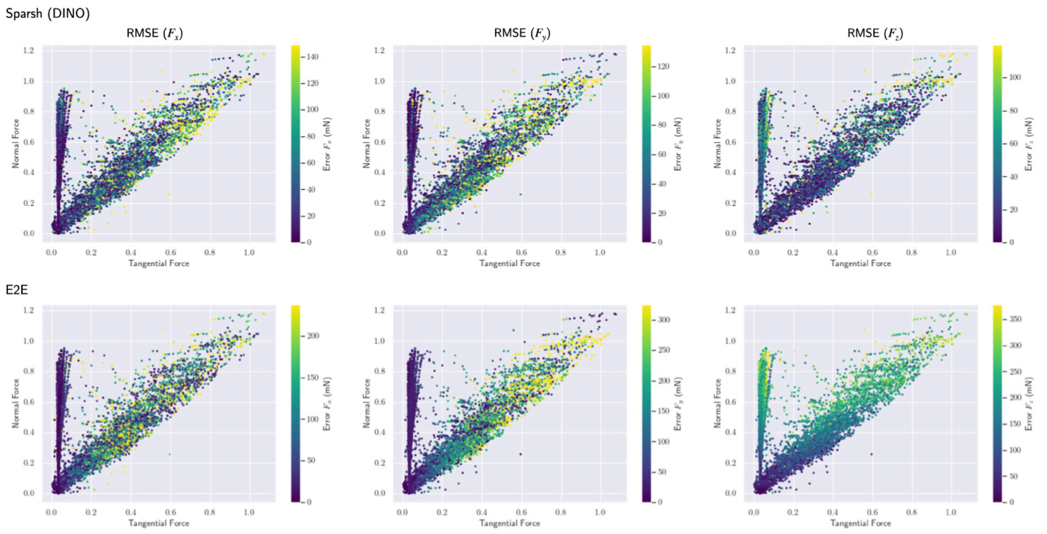

Figure 8: Friction cone of test data and RMSE (mN) for force estimation task with GelSight sensor.

shear field to align the boundaries with the visual edges in the tactile image. In Figure 9 we show snapshots of the normal and shear field predictions during sliding trajectories of the DIGIT sensor on YCB and spherical probe objects.

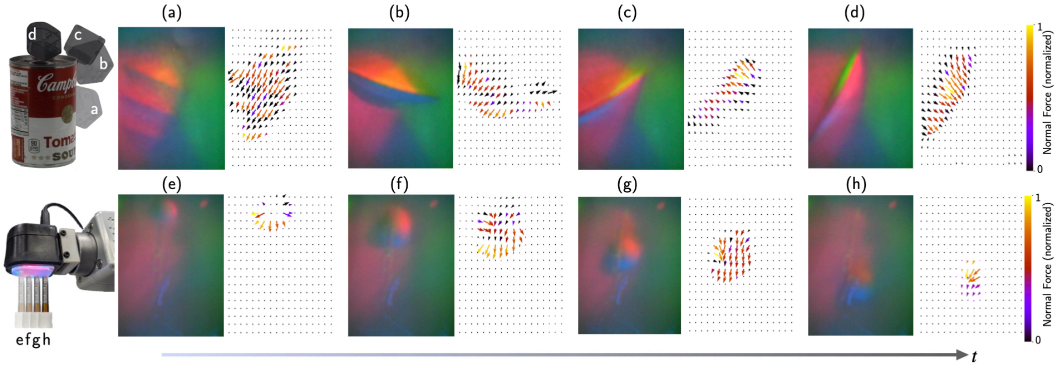

Figure 9: Normalized tactile flow (unitless) visualizations using `Sparsh` (`DINO`). Top row shows predicted force field for four key-frames from a representative YCB-Slide trajectory and bottom row shows interaction with the spherical probe. Arrows represent the tangential forces, while the colors depict the normal forces. These visualizations provide directional information about the relative motion of the contact patch. For instance (a) shows torsional motion resulting from rotating along the edge, (b, c, d) show sliding on the edge, (e) shows a diverging field when making contact with a spherical probe, and (f, g, h) show forces produced by sliding the probe top-down.

C.4 `[T2]` Slip detection

To collect labeled slip data we perform a normal/shear load test. Using a firmly affixed hemispherical probe on a flat surface, a robot presses the DIGIT sensor toward the probe, applying random normal forces of up to 5N. Upon reaching the target normal force, the robot slides the probe 2mm to a randomly selected position on the sensor surface, allowing us to capture the shear profile with a F/T sensor. To label slip, we rely on the friction cone to identify samples on the sticking and slippage regions. A description of the procedure is illustrated in Figure 10.

As eluded to in Section 3, `Sparsh`'s inference window is approximately 80 milliseconds. This is appropriate since this duration matches the reaction time needed by humans to adjust the grip

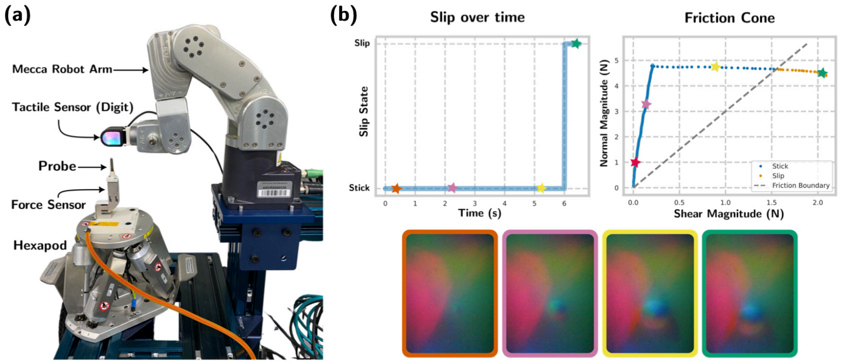

Figure 10: (a) Data collection setup for **[T1] Force Estimation** and **[T2] Slip Detection**. The Mecca Robot Arm with DIGIT / Gelsight is pressed against a static probe with random normal force. The arm then slides the sensor over the probe which induces shear forces. (b) Slip states over one representative stroke. When the sensor is pressed against the probe the normal force increases. The gel sensor initially resists sliding due to friction, but gives in, which results in a slight drop in normal force while the magnitude of shear force increases.

force when detecting partial slip [47]. We train two heads: one for slip detection and the other for the estimation of normalized force change (Δ). We find empirically that training both heads simultaneously improves slip detection, given their high correlation. The MLP probes are trained with cross-entropy for slip detection and mean absolute error (MAE) for Δ force regression as loss functions. Our dataset comprises 125k samples, with only 13% corresponding to slip instances. We reserve 25k samples for evaluating model performance.

Table 7 provides F1-score metrics for all models under different amounts of training data. `Sparsh` (`VJEPA`) outperforms all models, even when trained under low data regimes. In Figure 11 we contrast the predictions over time for a sample trajectory between `Sparsh` (`VJEPA`) and `E2E` models trained with 33% of the data. Note that for `Sparsh` (`VJEPA`) the errors are around the friction boundary, where the probe is starting to slide. Also, it is worth noticing that a poor estimation of changes in shear and normal forces is reflected in the accuracy of distinguishing between slip and no-slip. In Figure 12, we illustrate a failure case for `Sparsh` (`VJEPA`), as its results do not align with the ground truth. However, it is important to note that slip labeling is prone to errors due to its reliance on an experimental coefficient of friction. Despite the inaccuracies in the friction boundary for this trajectory, `Sparsh` (`VJEPA`) successfully detects the slip samples.

Model	Full dataset	$1/3$ dataset	$1/10$ dataset	$1/100$ dataset
`E2E`	0.767	0.238	0.299	0.214
`Sparsh (MAE)`	0.783	0.818	0.691	0.269
`Sparsh (DINO)`	0.685	0.561	0.548	0.489
`Sparsh (IJEPA)`	0.776	0.791	0.775	0.726
`Sparsh (VJEPA)`	**0.820**	**0.828**	**0.800**	**0.760**

Table 7: Performance of models on slip detection task under different budgets of training data. We use F1 score as metric, given that it ensures the model accurately identifies slip events without favoring the majority class. A high F1 score indicates effective and reliable slip detection.

C.5 `[T3]` Pose estimation

We collect a dataset of trajectories with time-synchronized pairs of object pose measurements and sensor observations using an Allegro hand equipped with DIGIT sensors on each finger, mounted on a robot arm. The object was placed on a table and with the palm facing downward, we pressed against it with the fingertips (see Figure 3). We manually perturbed the object's pose by sliding and rotating it under the Allegro fingertips. The pose of the object was tracked using ArUco tags. Given ground truth object pose measurements in the world frame, we preprocess them into relative pose change $(\Delta x, \Delta y, \Delta\theta) \in \mathrm{SE}(2)$ in the sensor frame.

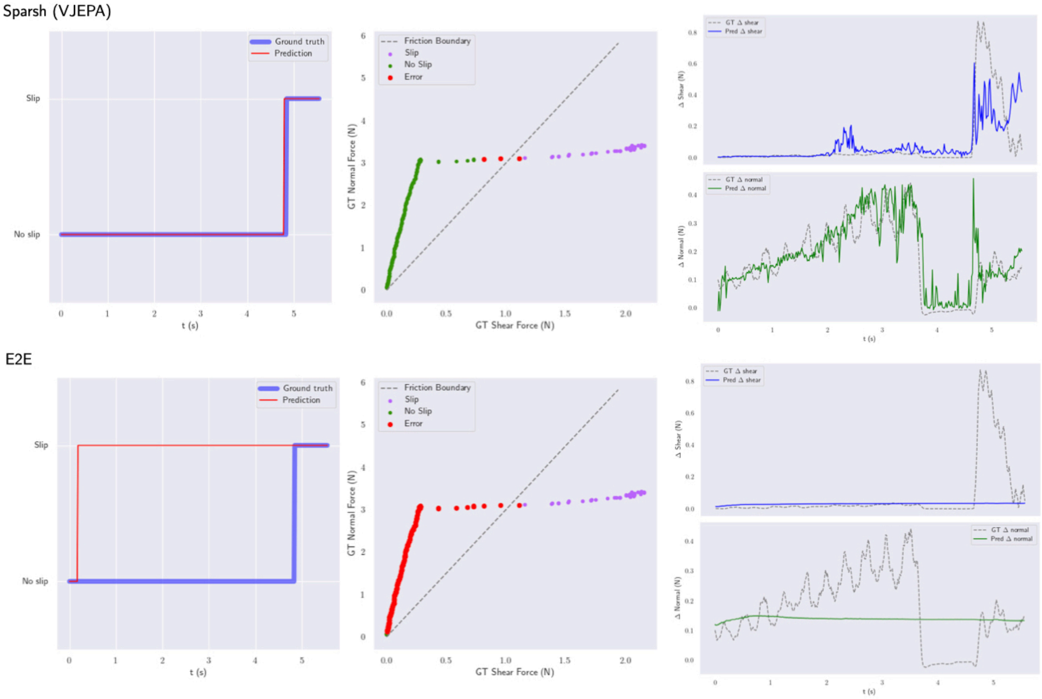

Figure 11: Contrast between `Sparsh` (`VJEPA`) and `E2E` for a test trajectory with a spherical probe sliding on the DIGIT sensor. `Sparsh` (`VJEPA`), even though trained only on 33% of the data, can detect slip accurately, which is correlated with its ability to estimate changes in normal and shear forces.

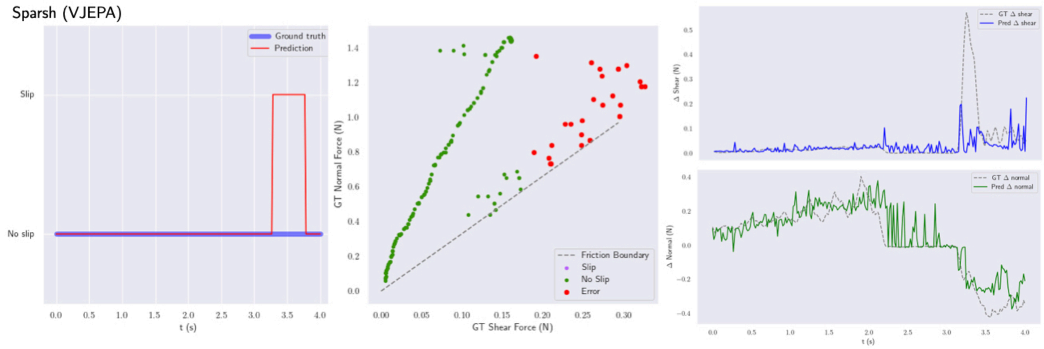

Figure 12: Failure case where the ground truth does not reflect slip since it relies on an experimental coefficient of friction. Despite the inaccuracies in the friction boundary for this trajectory, `Sparsh` (`VJEPA`) successfully detects slip samples.

Since we follow a regression-by-classification approach, we discretize the range of motion for each degree of freedom into multiple intervals in Log-uniform space. This allows us to achieve a better data distribution across all classes, as most pose changes are concentrated around zero. The strategy of classification-regression is also commonly explored for monocular depth estimation [98].

After attentive pooling, the features are passed to three heads, one for each degree of freedom. Each head is an MLP with two layers, which outputs the probability distribution over 11 classes (pose change bins). In Figure 13 we present the binning as well as the confusion matrices on test data for each degree of freedom, comparing `E2E`, `Sparsh` (`DINO`) and `Sparsh` (`IJEPA`) for pose estimation when trained on 33% of the available labeled data. Note that `Sparsh` can accurate distinguish pose changes in a low data regime, while a conventional task-specific approach struggles discerning the differences between adjacent bins, and finally tends to default to zero or maximum relative pose change, losing resolution in estimation.

Figure 14 shows a test trajectory over time with its ground truth labels. The colors on the plot represent the class agreement between the pose decoders trained with Sparsh (DINO) (using 33% of the data) and the ground truth. Darker colors indicate no error, while brighter colors indicate greater misclassification. In Table 8 we report for each model accuracy in pose estimation over 630 test samples and 95% confidence interval.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.812 [0.811, 0.813]	0.245 [0.244, 0.247]	0.162 [0.160, 0.164]	0.162 [0.160, 0.164]
Sparsh (MAE)	0.896 [0.896, 0.897]	0.719 [0.718, 0.721]	0.417 [0.414, 0.420]	0.223 [0.221, 0.225]
Sparsh (DINO)	**0.913** **[0.912, 0.914]**	**0.834** **[0.832, 0.836]**	**0.460** **[0.457, 0.461]**	**0.242** **[0.240, 0.245]**
Sparsh (IJEPA)	0.851 [0.850, 0.852]	0.601 [0.599, 0.603]	0.323 [0.321, 0.325]	0.212 [0.210, 0.215]
Sparsh (VJEPA)	0.856 [0.854, 0.857]	0.648 [0.646, 0.651]	0.368 [0.367, 0.370]	0.228 [0.225, 0.231]

Table 8: Accuracy and 95% confidence interval for pose estimation task following the regression-by-classification paradigm. Relative pose between object and ring finger. Metrics computed over 630 test samples.

C.6 [T4] Grasp stability

We use the Feeling of Success dataset [8], which contains data from a pair of GelSight sensors (with markers) attached to a jaw gripper (left and right fingers). The goal is to determine the success or the failure of the grasp attempt.

We pass to the SSL model the ‘before’ and ‘during’ as tactile history. We create our randomized split with all objects, using approximately 8k grasps for training and the remaining 1.3k grasps for evaluation. Using attentive probing, we freeze Sparsh and train a 2-layer MLP with two output units for grasp success classification.

In Table 9 report the accuracy for binary classification to compare the performance of the models across different training budgets, including a 95% confidence interval. Figure 15 shows the confusion matrices on test samples for E2E, Sparsh (DINO) and Sparsh (IJEPA) trained on a 33% of labeled data.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.784 [0.783, 0.785]	0.725 [0.722, 0.728]	0.682 [0.680, 0.684]	0.478 [0.472, 0.482]
Sparsh (MAE)	0.815 [0.813, 0.817]	0.696 [0.691, 0.702]	0.764 [0.761, 0.768]	0.466 [0.461, 0.471]
Sparsh (DINO)	0.780 [0.777, 0.782]	0.706 [0.702, 0.710]	0.773 [0.772, 0.775]	0.473 [0.467, 0.479]
Sparsh (IJEPA)	0.802 [0.800, 0.804]	**0.782** **[0.779, 0.784]**	**0.768** **[0.766, 0.770]**	**0.598** **[0.597, 0.601]**
Sparsh (VJEPA)	**0.809** **[0.805, 0.813]**	0.702 [0.700, 0.704]	0.743 [0.740, 0.746]	0.523 [0.519, 0.527]

Table 9: Accuracy and 95% confidence interval for grasp stability classification over different budget sizes of training data, using Feeling of Success dataset. Results over 1.3k grasps.

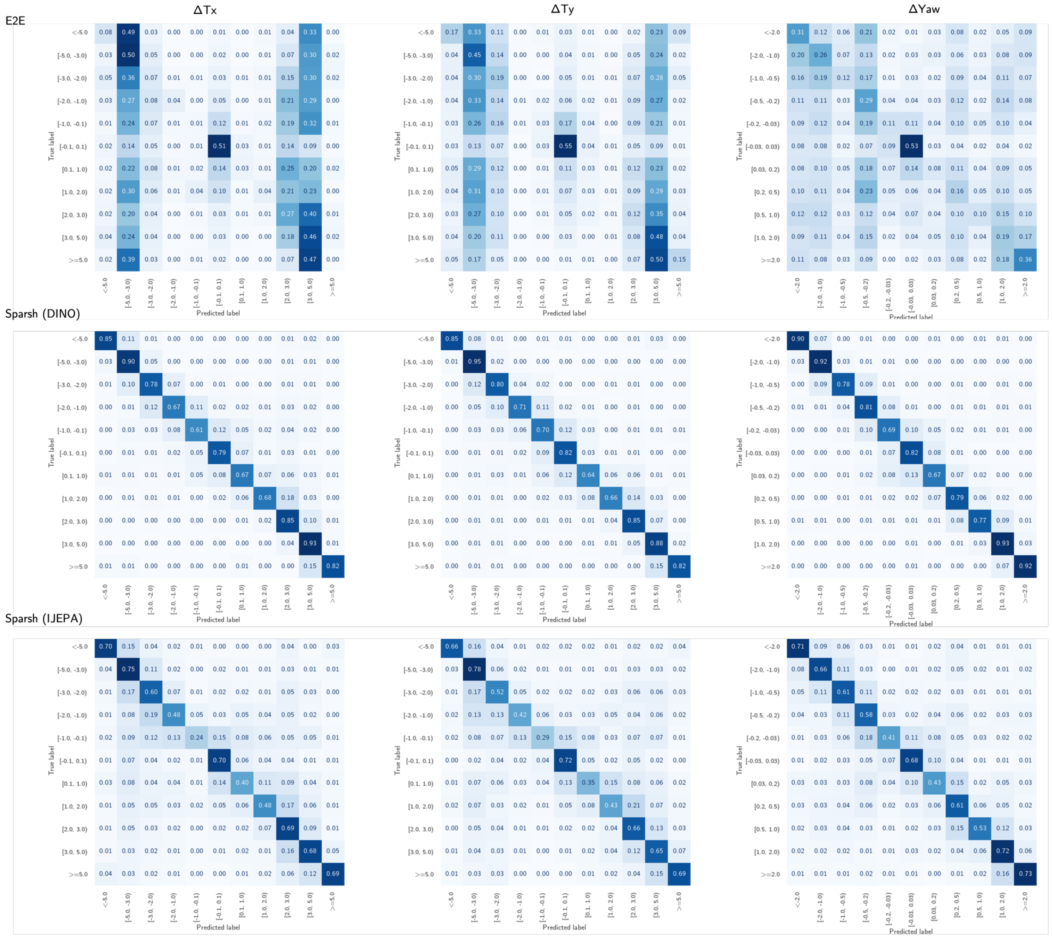

Figure 13: Confusion matrix on test data for ΔT_x, ΔT_y, ΔYaw for `E2E`, `Sparsh` (`DINO`) and `Sparsh` (`IJEPA`) trained on 33% of the available labeled data. The test dataset consist of 630 samples.

C.7 `[T5]` Bead maze

The goal in bead maze is to guide the bead along the wire, as shown in Figure 3. We don't rely on vision for hand-eye coordination, making the task fundamentally tactile since forces in the fingers indicate whether the bead is moving smoothly or encountering resistance. In our setup, we use a Franka arm with a robotic hand mounted on the wrist and DIGIT sensors on the fingers. To collect demonstrations for training the policy, we start the task with the bead grasped between the thumb and index fingers and move the arm to guide the bead along the wire. We collect 30 demonstrations on different maze patterns with mix of VR-based and manual kinesthetic-based teleoperation, corresponding to a total of ∼34k training pairs of tactile images and robot joint angles.

For training the policy, we adapt Diffusion Policy [64] to our problem setting. Given a small history of tactile images $(\ldots, \mathbf{z}_{t-1}, \mathbf{z}_t)$, and robot proprioception $(\ldots, q_{t-1}, q_t)$, we train the policy to predict changes in joint angles as actions $\mathbf{a} \triangleq (\Delta q_t, \Delta q_{t+1}, ...)$; $\Delta q \in \mathbb{R}^7$, instead of position control. Following the guidelines in Diffusion Policy, we use an observation horizon of 2 and an action prediction horizon of 8. We adhere to the official implementation for policy architecture and training hyper-parameters. For conditioning on tactile input, we modify the CNN encoder from Diffusion Policy and replace it with `Sparsh` backbones with fixed parameters. For training an end-to-end policy, the encoder corresponds to a ViT-Base encoder with randomly initialized weights.

In Table 10 we report to position error of `E2E`, `Sparsh` (`DINO`) and `Sparsh` (`IJEPA`) with respect to test demonstrations on an unseen maze, highlighting the fidelity of `Sparsh` (`DINO`) and `Sparsh` (`IJEPA`) to

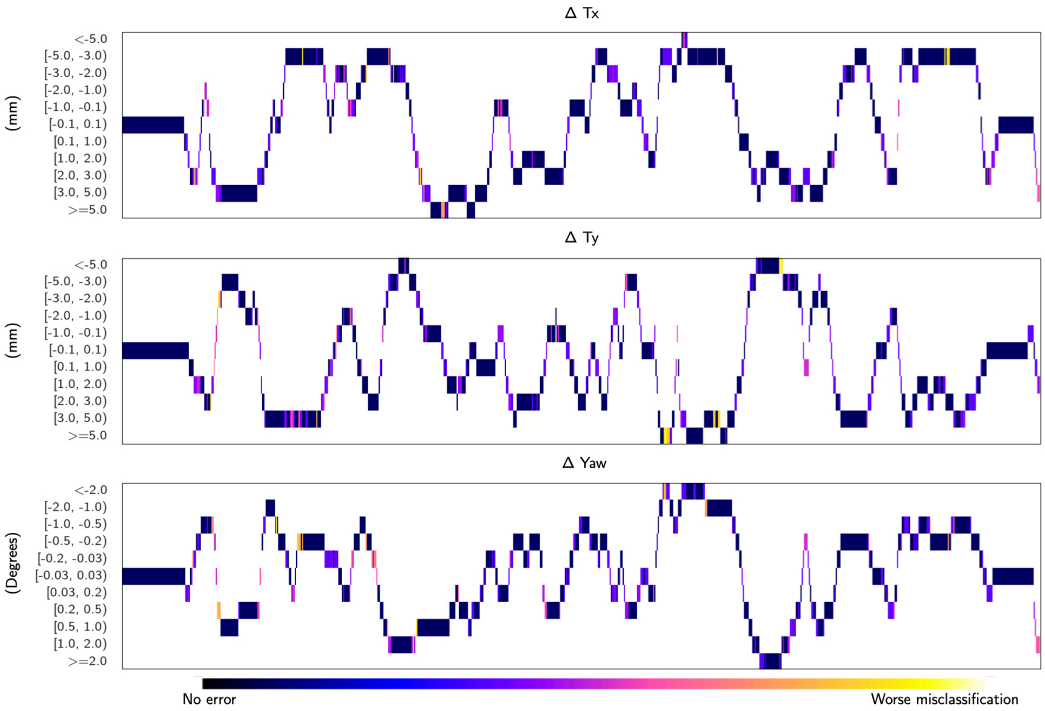

Figure 14: Ground truth relative pose classes for T_x, T_y, and Yaw for a test trajectory. The colormap represents the class agreements between the ground truth and the pose decoder, with darker colors indicating no error and brighter colors indicating greater misclassification.

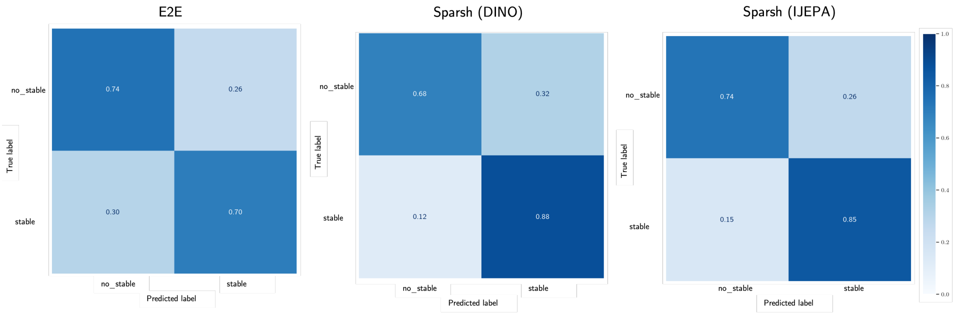

Figure 15: Confusion matrix on test data for grasp stability, comparing `E2E`, `Sparsh (DINO)` and `Sparsh (IJEPA)` trained on 33% of the available labeled data. The test dataset consist of 1.3k grasps.

follow a similar trajectory. Nevertheless, this doesn't necessarily transfer to real-world performance, since the locality of the observations and predictions make the errors in the adjusted joint angles to compound fast, which results in unforeseen collisions and the subsequent lose of the grasp. In an overfitting setting, training a policy for a single maze, policies using `Sparsh (DINO)` and `Sparsh (IJEPA)` are able to complete almost 30% of the maze on the real robot. However, it is expected an specialist policy trained end-to-end to perform better in the overfitting setting. Experimentally, we found than an `E2E` policy trained for a single maze is able to complete almost 80% of the maze running on the real robot.

In Table 11 we summarize the performance of `Sparsh` across the benchmark. We find that with respect to an `E2E` approach, with `Sparsh` we can achieve an improvement of 98.75% on average. `Sparsh (DINO)` and `Sparsh (IJEPA)` are in general the best models across the board, showing the benefits of learning touch representations in latent space. An MAE approach, which relies on pixel space supervision, is still competitive, although it was not evaluated on the policy task.

Model	Full dataset	$1/2$ dataset	$1/10$ dataset
Sparsh-(E2E)	8.46 [7.61, 9.32]	7.14 [6.26, 8.05]	9.80 [8.78, 10.82]
Sparsh-(DINO)	5.54 [4.90, 6.17]	5.98 [5.29, 6.67]	5.71 [5.13,6.29]
Sparsh-(IJEPA)	5.47 [4.82, 6.13]	5.72 [5.05, 6.40]	5.46 [4.82, 6.10]

Table 10: Position error (mm) and 95% confidence interval for the Bead Maze task. We compare the ground truth trajectory from a test demonstration in an unseen maze against the compounded trajectory from the predicted delta joint angles from each policy.

Task	Best SSL vs E2E	DINO vs IJEPA	MAE vs Best	VJEPA vs Best
Force estimation (DIGIT)	28.31%	26.67%	-4.38%	-27.96%
Force estimation (GelSight)	59.74%	32.41%	1.72%	-64.23%
Slip detection	242.70%	29.08%	-1.21%	0.00%
Pose estimation	235.89%	-37.91%	-13.81%	-22.33%
Grasp stability	5.14%	8.45%	$-10/17\%$	-7.83%
Bead maze	19.72%	-5.26%	-	-
Average	$\mathbf{98.75}\%$	8.91%	-5.57%	-24.47%

Table 11: Performance of Sparsh across TacBench and comparison between SSL approaches.

References

[1] 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.

[2] 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.

[3] 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.

[4] N. F. Lepora, Y. Lin, B. Money-Coomes, and J. Lloyd. Digitac: A digit-tactip hybrid tactile sensor for comparing low-cost high-resolution robot touch. *IEEE Robotics and Automation Letters*, 7(4):9382–9388, 2022.

[5] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6437–6443, 2021. URL `https://api.semanticscholar.org/CorpusID:233004667`.

[6] C. Higuera, J. Ortiz, H. Qi, L. Pineda, B. Boots, and M. Mukadam. Perceiving extrinsic contacts from touch improves learning insertion policies. *arXiv preprint arXiv:2309.16652*, 2023.

[7] J. Lloyd and N. F. Lepora. Goal-driven robotic pushing using tactile and proprioceptive feedback. *IEEE Transactions on Robotics*, 38(2):1201–1212, 2022. doi:10.1109/TRO.2021.3104471.

[8] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017.

[9] S. Suresh, Z. Si, S. Anderson, M. Kaess, and M. Mukadam. MidasTouch: Monte-Carlo inference over distributions across sliding touch. In *Proc. Conf. on Robot Learning, CoRL*, Auckland, NZ, Dec. 2022.

[10] M. Bauza, A. Bronars, and A. Rodriguez. Tac2Pose: Tactile object pose estimation from the first touch. *The International Journal of Robotics Research*, 42(13):1185–1209, 2023. doi:10.1177/02783649231196925. URL `https://doi.org/10.1177/02783649231196925`.

[11] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, J. Ortiz, and M. Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. In *arXiv preprint arXiv:2312.1346*, Dec. 2023.

[12] A. Church, J. Lloyd, R. Hadsell, and N. F. Lepora. Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard. *IEEE Robotics and Automation Letters*, 5(4): 6145–6152, 2020.

[13] H. Xu, Y. Luo, S. Wang, T. Darrell, and R. Calandra. Towards learning to play piano with dexterous hands and touch. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10410–10416. IEEE, 2022.

[14] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora. Tactile gym 2.0: Sim-to-real deep reinforcement learning for comparing low-cost high-resolution robot touch. *IEEE Robotics and Automation Letters*, 7(4):10754–10761, 2022.

[15] B. Zandonati, R. Wang, R. Gao, and Y. Wu. Investigating vision foundational models for tactile representation learning. *arXiv preprint arXiv:2305.00596*, 2023.

[16] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311, 2015. doi:10.1109/ICRA.2015.7139016.

[17] Z. Lu, Z. Liu, X. Zhang, Y. Liang, Y. Dong, and T. Yang. 3d force identification and prediction using deep learning based on a gelsight-structured sensor. *Sensors and Actuators A: Physical*, 367:115036, 2024.

[18] D. C. Bulens, N. F. Lepora, S. J. Redmond, and B. Ward-Cherrier. Incipient slip detection with a biomimetic skin morphology. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8972–8978, 2023. doi:10.1109/IROS55552.2023.10341807.

[19] Y. Du, G. Zhang, Y. Zhang, and M. Y. Wang. High-resolution 3-dimensional contact deformation tracking for fingervision sensor with dense random color pattern. *IEEE Robotics and Automation Letters*, 6(2):2147–2154, 2021. doi:10.1109/LRA.2021.3061306.

[20] F. Yang, C. Ma, J. Zhang, J. Zhu, W. Yuan, and A. Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.

[21] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu. ObjectFolder 2.0: A multisensory object dataset for sim2real transfer. In *CVPR*, 2022.

[22] 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.

[23] 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.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20j.html`.

[25] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[26] U. Ozbulak, H. J. Lee, B. Boga, E. T. Anzaku, H. min Park, A. V. Messem, W. D. Neve, and J. Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=Ma25S4ludQ`. Survey Certification.

[27] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum. A cookbook of self-supervised learning, 2023.

[28] I. H. Taylor, S. Dong, and A. Rodriguez. GelSlim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787, 2022. doi:10.1109/ICRA46639.2022.9811832.

[29] D. Ma, E. Donlon, S. Dong, and A. Rodriguez. Dense tactile force estimation using gelSlim and inverse FEM. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424. IEEE, 2019.

[30] M. Lambeta, H. Xu, J. Xu, P.-W. Chou, S. Wang, T. Darrell, and R. Calandra. PyTouch: A machine learning library for touch processing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13208–13214. IEEE, 2021.

[31] J. Hansen, F. Hogan, D. Rivkin, D. Meger, M. Jenkin, and G. Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE, 2022.

[32] Y. Chen, M. V. d. Merwe, A. Sipos, and N. Fazeli. Visuo-tactile transformers for manipulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 2026–2040. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/chen23d.html.

[33] 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.

[34] T. P. Tomo, M. Regoli, A. Schmitz, L. Natale, H. Kristanto, S. Somlor, L. Jamone, G. Metta, and S. Sugano. A new silicone structure for uskin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot icub. *IEEE Robotics and Automation Letters*, 3(3): 2584–2591, 2018.

[35] I. Guzey, B. Evans, S. Chintala, and L. Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=EXQ0eXtX3OW.

[36] R. Gao, Y. Dou, H. Li, T. Agarwal, J. Bohg, Y. Li, L. Fei-Fei, and J. Wu. The objectfolder benchmark: Multisensory learning with neural and real objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17276–17286, 2023.

[37] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021.

[38] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu. ObjectFolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10598–10608, 2022.

[39] 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. *arXiv preprint arXiv:2401.18084*, 2024.

[40] S. Yu, K. Lin, A. Xiao, J. Duan, and H. Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv preprint arXiv:2405.02794*, 2024.

[41] Y. Dou, F. Yang, Y. Liu, A. Loquercio, and A. Owens. Tactile-augmented radiance fields. *arXiv preprint arXiv:2405.04534*, 2024.

[42] 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.

[43] 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 (CVPR)*, pages 15619–15629, June 2023.

[44] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. V-JEPA: Latent video prediction for visual representation learning. 2023.

[45] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[46] Y. LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022.

[47] A. Zangrandi, M. D'Alonzo, C. Cipriani, and G. Di Pino. Neurophysiology of slip sensation and grip reaction: insights for hand prosthesis control of slippage. *Journal of neurophysiology*, 126(2):477–492, 2021.

[48] S. Dong, W. Yuan, and E. H. Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–144. IEEE, 2017.

[49] F. Veiga, H. van Hoof, J. Peters, and T. Hermans. Stabilizing novel objects by learning to predict tactile slip. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5065–5072, 2015. doi:10.1109/IROS.2015.7354090.

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

[51] 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.

[52] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, 2024.

[53] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[54] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

[55] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[56] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[57] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.

[58] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova. What matters in unsupervised optical flow. *arXiv preprint arXiv:2006.04902*, 2020.

[59] S. Kim, A. Bronars, P. Patre, and A. Rodriguez. Texterity–tactile extrinsic dexterity: Simultaneous tactile estimation and control for extrinsic dexterity. *arXiv preprint arXiv:2403.00049*, 2024.

[60] S. Kanitkar, H. Jiang, and W. Yuan. PoseIt: a visual-tactile dataset of holding poses for grasp stability analysis. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 71–78. IEEE, 2022.

[61] Z. Si, Z. Zhu, A. Agarwal, S. Anderson, and W. Yuan. Grasp stability prediction with sim-to-real transfer from tactile sensing. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7809–7816, 2022. doi:10.1109/IROS47612.2022.9981863.

[62] R. Kolamuri, Z. Si, Y. Zhang, A. Agarwal, and W. Yuan. Improving grasp stability with rotation measurement from tactile sensing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6809–6816. IEEE, 2021.

[63] F. R. Hogan, J.-F. Tremblay, B. H. Baghi, M. Jenkin, K. Siddiqi, and G. Dudek. Finger-STS: Combined proximity and tactile sensing for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(4):10865–10872, 2022.

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

[65] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[66] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[67] N. Hansen, Z. Yuan, Y. Ze, T. Mu, A. Rajeswaran, H. Su, H. Xu, and X. Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

[68] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[69] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[70] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[71] P. Gao, T. Ma, H. Li, Z. Lin, J. Dai, and Y. Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022.

[72] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[73] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[74] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[75] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[76] 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.

[77] 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 K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1298–1312. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/baevski22a.html.

[78] Z. Fei, M. Fan, and J. Huang. A-JEPA: Joint-embedding predictive architecture can listen. *ArXiv*, abs/2311.15830, 2023. URL https://api.semanticscholar.org/CorpusID:265456289.

[79] A. Bardes, J. Ponce, and Y. LeCun. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features, 2023.

[80] A. Saito and J. Poovvancheri. Point-JEPA: A joint embedding predictive architecture for self-supervised learning on point cloud. 2024. URL https://api.semanticscholar.org/CorpusID:269362564.

[81] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[82] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The unsurprising effectiveness of pre-trained vision models for control. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17359–17371. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/parisi22a.html.

[83] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[84] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[85] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[86] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[87] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023.

[88] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra. TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):3930–3937, 2022. ISSN 2377-3766. doi:10.1109/LRA.2022.3146945. URL https://arxiv.org/abs/2012.08456.

[89] Z. Si and W. Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *arXiv preprint arXiv:2109.04027*, 2021.

[90] D. F. Gomes, P. Paoletti, and S. Luo. Beyond flat gelsight sensors: Simulation of optical tactile sensors of complex morphologies for sim2real learning. 2023. URL https://www.roboticsproceedings.org/rss19/p035.pdf.

[91] Z. Si, G. Zhang, Q. Ben, B. Romero, Z. Xian, C. Liu, and C. Gan. DIFFTACTILE: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eJHnSg783t.

[92] Z. Chen, S. Zhang, S. Luo, F. Sun, and B. Fang. Tacchi: A pluggable and low computational cost elastomer deformation simulator for optical tactile sensors. *IEEE Robotics and Automation Letters*, 8(3):1239–1246, 2023. doi:10.1109/LRA.2023.3237042.

[93] W. Chen, J. Xu, F. Xiang, X. Yuan, H. Su, and R. Chen. General-purpose sim2real protocol for learning contact-rich manipulation with marker-based visuotactile sensors. *IEEE Transactions on Robotics*, 40:1509–1526, 2024. doi:10.1109/TRO.2024.3352969.

[94] Y. Zhao, X. Jing, K. Qian, D. F. Gomes, and S. Luo. Skill generalization of tubular object manipulation with tactile sensing and sim2real learning. *Robotics and Autonomous Systems*, 160:104321, 2023. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2022.104321. URL https://www.sciencedirect.com/science/article/pii/S092188902200210X.

[95] Z. Si, Z. Zhu, A. Agarwal, S. Anderson, and W. Yuan. Grasp stability prediction with sim-to-real transfer from tactile sensing. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7809–7816. IEEE, 2022.

[96] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[97] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

[98] S. Shao, Z. Pei, X. Wu, Z. Liu, W. Chen, and Z. Li. Iebins: Iterative elastic bins for monocular depth estimation. *Advances in Neural Information Processing Systems*, 36, 2024.