

Tactile-JEPA: Topology-Aware Self-Supervised Representation Learning for Distributed Tactile Sensors

Elizaveta Kovtun^{*,1,2}, Matvey Konovalov^{1,3}, Andrey Sakhovskiy^{1,2}, and Semen Budennyi^{1,4}

Abstract—Tactile sensing is an essential modality for robots performing contact-rich, dexterous manipulation, particularly under visual occlusion. While pre-trained image encoders are standard in robot learning pipelines, tactile encoders are still commonly trained from scratch from raw, noisy signals, which might limit their expressivity. Existing self-supervised learning (SSL) approaches focus predominantly on vision-based tactile sensors, leaving distributed electronic skins largely unaddressed. These sensors, however, have a distinctive property: their sensing elements are sparse and irregularly arranged over the surface they cover, which makes direct reuse of visual SSL methods suboptimal. We present Tactile-JEPA, an efficient self-supervised pre-training method that uses the spatial arrangement of tactile sensors to learn topology-aware representations. Specifically, it is trained to predict the embeddings of masked sensing elements from the unmasked remainder, using the sensor connectivity graph to guide spatial masking. Our analysis shows that effective tactile representations require capturing both local contact details and the global state of the tactile surface, which we achieve through dual-scale masking. Across three diverse datasets spanning magnetic and piezoresistive sensors, different robot embodiments, and single- and paired-sensor configurations, Tactile-JEPA reduces force estimation error by 6.3% and in-hand orientation error by 20.8% over the prior state-of-the-art, with consistent gains in other downstream applications, including policy learning. Overall, our results demonstrate that the benefit of tactile sensing depends critically on the quality of encoder pre-training, a problem which Tactile-JEPA addresses directly. Code is available at <https://github.com/E-Kovtun/tactile>.

I. INTRODUCTION

People increasingly expect robots to assist them across diverse domains, from household tasks to manufacturing and agriculture [1], [2], [3], and this requires generalization across tasks and environments. Vision-Language-Action (VLA) models [4], [5] provide a prominent foundation for this capability, but their only channel for world perception is vision. Relying on vision alone becomes a limiting factor under severe occlusions, dexterous contact-rich manipulation, and contact with fragile objects, in which *tactile sensing* emerges as a critical modality [6], [7], [8]. Recent works explore various ways of integrating touch into VLA learning pipelines [9], [10], demonstrating more precise manipulation in tactile-intensive scenarios.

Equipping robots with a sense of touch requires dedicated sensors, which differ substantially in their operating

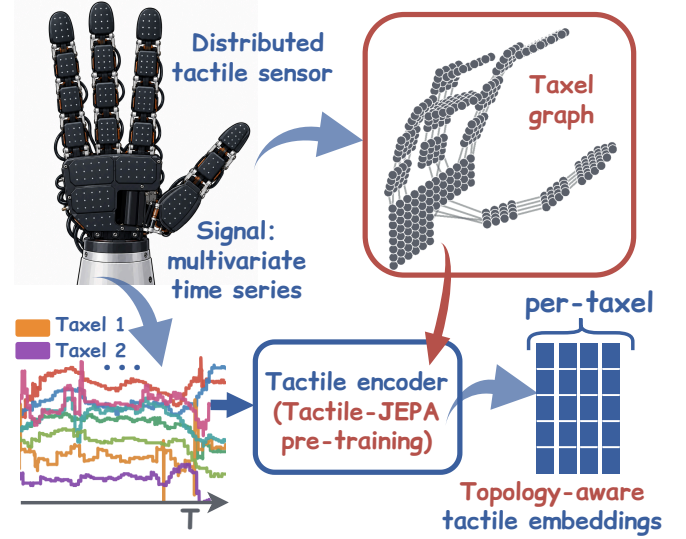


Fig. 1. Tactile-JEPA pre-trains an encoder on signals from distributed tactile sensors in a self-supervised manner, accounting for sensor topology via the taxel graph and producing topology-aware per-taxel embeddings.

principle and design [11]. Among the widely adopted are vision-based sensors, e.g., GelSight [12] and Digit [13], where an internal camera captures the deformation of a soft elastomer, producing an image stream that encodes contact geometry and forces. While this output is high-resolution and visually interpretable, such sensors are bulky and offer limited contact coverage [14]. An alternative is a thin and flexible electronic skin (e-skin) [15], [16], which can be distributed across the entire hand or body rather than confined to fingertips. Regardless of the underlying transduction principle—piezoresistive [17], capacitive (DexSkin [18]), or magnetic (ReSkin [19], Xela uSkin [20], AnySkin [21])—e-skins output a multivariate time series whose channels are the readings of individual sensing elements, or taxels, distributed across the sensing surface [19], [14]. A key question is how to effectively use the signal provided by the tactile sensors. While VLAs inherit vision and language representations from large-scale pre-trained backbones [22], [23], the tactile modality has no such counterpart and is typically learned from scratch on raw, noisy data. This motivates pre-training tactile encoders to obtain effective representations [24], [25].

For vision-based tactile sensors, encoders are pre-trained by adapting self-supervised learning (SSL) approaches from computer vision, such as MAE [26] or DINO [27]. Beyond outperforming end-to-end training, the resulting representations transfer broadly: a single backbone serves different

*Corresponding author: elizaveta.kovtun@skoltech.ru

¹Sber AI, Moscow, Russia

²Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia

³HSE University, Moscow, Russia

⁴Artificial Intelligence Research Institute (AIRI), Moscow, Russia

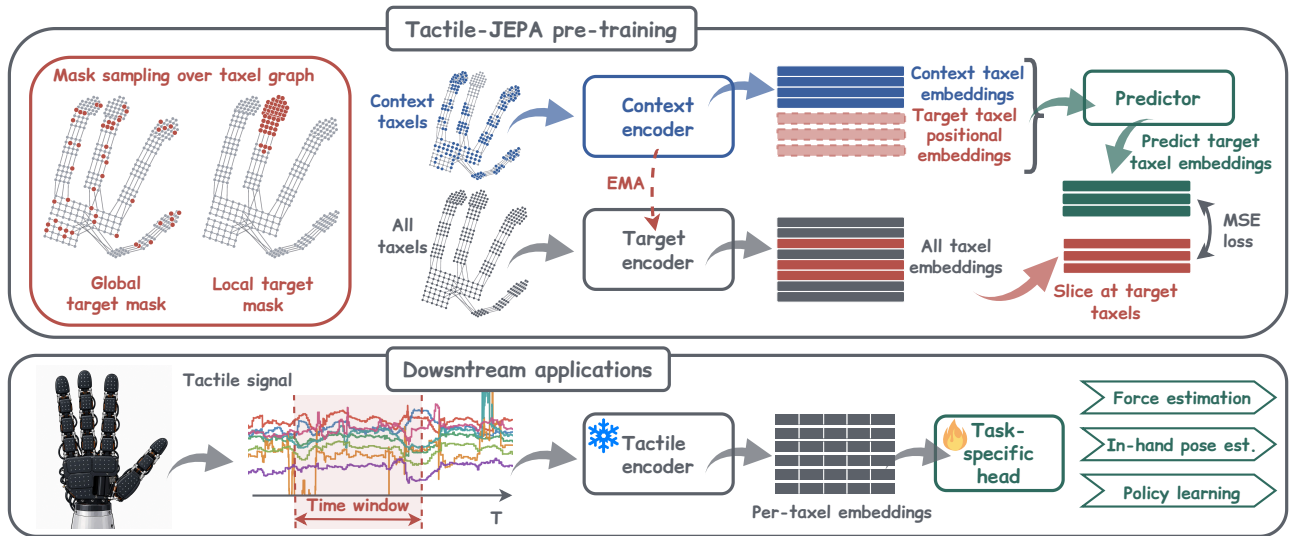


Fig. 2. Overview of Tactile-JEPA. *Top*: self-supervised pre-training. Target masks are sampled over the taxel graph at local and global scales (left), so the encoder learns topology-aware contact patterns at both extents. The predictor infers target embeddings from context embeddings, matching the EMA target encoder outputs via MSE loss (right). *Bottom*: downstream applications. The target encoder from the top is frozen and reused as the tactile encoder, mapping a windowed tactile signal to per-taxel embeddings, on which a task-specific head is trained.

sensors and downstream tasks, remaining effective under limited labeled data [24], [28]. The same strategy can be applied to e-skins, treating their multivariate time-series output as images [29]. However, such approaches do not account for the spatial structure of the sensing surface. We close this gap with **Tactile-JEPA**, a self-supervised method for distributed tactile sensors. Following I-JEPA [30], it predicts the embeddings of masked taxels from the visible ones. The novelty lies in mask sampling: masks are drawn (i) over the sensor connectivity graph, i.e., the taxel graph in Fig. 1, rather than a pixel grid, and (ii) at two scales, with *local* masks covering compact regions and *global* masks spanning taxels distributed across the skin. This produces topology-aware, multi-scale representations. Summing up, this work makes the following **contributions**:

- **Tactile-JEPA**, a self-supervised representation learning method for spatially distributed tactile sensors that operates directly on their multivariate time-series output and produces topology-aware, multi-scale representations.
- Consistent gains over prior tactile representation learning methods across multiple embodiments, sensor types, and downstream tasks, including force and pose estimation, object and action classification, and tactile-conditioned policy learning.
- An experimental study showing that masks sampled over the sensor connectivity graph outperform image-style blocks, and that their scale controls locality: local or global masks favor particular downstream tasks and, when mixed, transfer broadly.

II. RELATED WORK

A. Representation Learning for Vision-based Tactile Sensors

Most representation learning techniques for tactile sensing are developed for vision-based sensors, whose raw output

is an image, making vision SSL objectives directly applicable. A representative example is Sparsh [24], a family of tactile encoders that transfer across several vision-based sensors and are pre-trained with a range of SSL objectives, including masked autoencoding (MAE [26]), self-distillation (DINO [27], DINOv2 [31]), and joint-embedding prediction (I-JEPA [30], V-JEPA [32]). Sparsh-X [25] extends this family to multisensory touch, using SSL to jointly encode tactile image, audio, inertial, and pressure channels into a single embedding. Beyond learning representations from the touch signal alone, another line of work builds a shared latent space over tactile, visual, and linguistic modalities, aligning them through contrastive pre-training that supports cross-modal tasks [33], [34], [35], [36]. All of these methods inherit the pixel grid of vision-based sensors. Distributed sensors provide no such structure: their output is a multivariate time series over a sparse, irregular set of taxels.

B. Representation Learning for Distributed Tactile Sensing

A common workaround reshapes the distributed signal into an image, so that vision SSL remains applicable. T-DEX [29] pre-trains with BYOL [37] on magnetic Xela uSkin [20] pads distributed across a dexterous hand, arranging them into a single three-channel image. Sparsh-skin [14] abandons this image-like representation on the same hardware, encoding each taxel as a separate token trained via self-distillation [31]. Signal structure is exploited spatio-temporally in STAT [38], which combines masked reconstruction with time-order differentiation between signal segments. In both Sparsh-skin and STAT, sensor topology enters only as a per-taxel location feature, leaving the connectivity of the sensing surface unrepresented. This issue is partially addressed in Tactile-GAT [39] and TacGNN [40], which build adjacency graphs over taxels, yet both train end-to-end on labeled data, so no

reusable representation is learned. HyperTixel [41] does pre-train, but its contrastive objective requires contact surface geometry obtainable only in simulation. Consequently, existing approaches rely either on task-specific labels or privileged simulation information, neither of which is readily available in real-world deployment. Therefore, the **research gap** we target is self-supervised pre-training that requires nothing beyond the sensor’s own signal and its known layout, and that explicitly represents the sensor topology.

III. METHODOLOGY

A. Problem Formulation

Let a distributed tactile sensor covering a robot embodiment comprise N taxels sampled at rate f_s . Over an observation window of duration T_d , it produces $T = \lfloor T_d f_s \rfloor$ frames forming the tactile signal $X = (x_1, \dots, x_N) \in \mathbb{R}^{N \times T \times m}$, where $x_i \in \mathbb{R}^{T \times m}$ the time series produced by the i -th taxel and m is the number of sensing axes per taxel, e.g. $m = 1$ for pressure-based sensors, sensitive to the normal component only, or $m = 3$ for magnetic skins, whose 3-axis magnetic flux readings respond to normal and shear components. A single frame captures the instantaneous state of the skin but carries no information about the contact dynamics. Therefore, we operate on short time windows of duration T_w , comprising $\tau = \lfloor T_w f_s \rfloor$ frames, and consider the slice $X_\tau^{(k)} = X[:, t_k : t_k + \tau, :] \in \mathbb{R}^{N \times \tau \times m}$. The value T_w is chosen so that the window is not a noisy snapshot but a short history of the contact, yet still narrow enough to span a single event. We assume a known sensor layout, i.e., taxel arrangement. Taxel positions $P \in \mathbb{R}^{N \times T \times 3}$ may also be available, but Tactile-JEPA does not require them.

Given an unlabeled dataset $\mathcal{D} = \{X_\tau^{(k)}\}_{k=1}^K$ of tactile windows, **our goal is to pre-train a tactile encoder $\mathbf{E}_\theta$ in a self-supervised manner that accounts for the geometric arrangement of the taxels**. The encoder maps each tactile window to a set of per-taxel embeddings. During downstream evaluation, shown at the bottom of Fig. 2, the pre-trained encoder $\mathbf{E}_\theta$ is kept frozen, and a lightweight task-specific head is trained on top of these embeddings using the corresponding labeled dataset. Importantly, the pre-training objective is defined within a short window and does not model temporal structure across windows, leaving temporal reasoning to the downstream decoder. Such a setup adds flexibility at inference: the encoder can be invoked at the control-loop rate, and longer history is obtained by composing successive embeddings.

B. Tactile-JEPA: Self-Supervised Pre-Training

Tactile-JEPA is a self-supervised representation learning approach tailored to distributed tactile sensing. It learns to predict the representations of hidden taxel signals from visible ones in the embedding space, avoiding direct prediction of noisy sensor signals. The top of Fig. 2 illustrates the Tactile-JEPA pre-training logic. It operates on two subsets of taxels: the visible ones, called the *context*, and the hidden ones, called the *targets*. We call such a subset a *region* and its binary indicator over all taxels a *mask*, using the

two terms interchangeably. These regions are processed by three **components**. The *context encoder* $\mathbf{E}_\theta$ embeds only the context taxels. The *target encoder* $\mathbf{E}_{\bar{\theta}}$ embeds all taxels, and its outputs at the target taxels serve as the prediction goals. The *predictor* $\mathbf{P}_\phi$ receives the context embeddings together with the positions of the target taxels and predicts their embeddings. The self-supervised loss measures how closely these predictions match the target encoder outputs.

Below, we first describe how the tactile input signals are represented. We then introduce the **two key design elements** of Tactile-JEPA: the construction of the taxel connectivity graph and the mask sampling strategy based on this graph. Finally, we describe the Tactile-JEPA workflow, including the encoding, prediction, and SSL training objective.

Input representation. Let $x_i^{(k)} = X[i, t_k : t_k + \tau, :] \in \mathbb{R}^{\tau \times m}$ be the response of taxel i over the τ frames. Following [14], we map this response to an embedding of dimension d by an affine projection shared across all taxels and all windows, followed by layer normalization (LN):

$$\tilde{x}_i^{(k)} = \text{LN}\left(W \text{vec}\left(x_i^{(k)}\right) + b\right) \in \mathbb{R}^d, \quad (1)$$

where $\text{vec}(\cdot)$ flattens the windowed signal of each taxel into a vector of length $m\tau$. The parameters $W \in \mathbb{R}^{d \times m\tau}$ and $b \in \mathbb{R}^d$ define the projection matrix and bias. We set the window duration T_w to capture a single contact event. The full windowed response of a taxel therefore forms one semantic unit. The encoder maps this entire sequence to a single embedding. Stacking these embeddings across all taxels yields the representation $\tilde{X}_\tau^{(k)} \in \mathbb{R}^{N \times d}$. The dimension of this tensor depends strictly on taxel count and remains independent of the frame count within the window.

Taxel connectivity graph. We capture the spatial structure of the distributed tactile sensor with a taxel connectivity graph $G = (V, E)$, whose nodes $V = \{1, \dots, N\}$ correspond to the N taxels and whose edges $(i, j) \in E$ connect taxels that are adjacent in the known sensor layout. The graph therefore requires no taxel coordinates and remains fixed across time windows. For sensors with two separate parts (e.g., two hands), G consists of two connected components.

Graph-based mask sampling. The SSL task divides the taxels into visible context regions and hidden target regions. The encoder predicts target representations from context representations. We sample n_c context regions $\{\mathcal{C}_j\}_{j=1}^{n_c}$ and n_t target regions $\{\mathcal{T}_l\}_{l=1}^{n_t}$, with $\mathcal{C}_j, \mathcal{T}_l \subset V$. Notably, Tactile-JEPA samples these regions on the taxel connectivity graph rather than on a pixel grid. We define two types of target masks. A *local* mask is a connected subgraph that covers a compact sensor region and captures localized contact patterns. Instead, a *global* mask consists of taxels scattered across the graph, reflecting the overall contact state of the embodiment (e.g., the hand) rather than any single region. Both types are used jointly: each set of target masks contains an equal mix of local and global masks. Masks are assigned a budget, the number of taxels they contain. To sample a local mask, we choose a seed taxel at random and grow the mask by Dijkstra expansion, repeatedly adding the connected

TABLE I

CHARACTERISTICS OF OPEN-SOURCE TACTILE DATASETS. NOTATION FOLLOWS SEC. III-A: N TAXELS, m SENSING AXES, f_s SENSOR SAMPLING RATE, AND $\tau = \lfloor T_w f_s \rfloor$ FRAMES PER TIME WINDOW. COORD. INDICATES WHETHER TAXEL COORDINATES ARE PROVIDED. HOURS AND #FRAMES GIVE EACH DATASET’S TOTAL DURATION AND FRAME COUNT. τ MATCHES HUMAN SLIP-RESPONSE LATENCY ($T_w \approx 0.1$ s) [42].

Dataset	Collection	Sensor	Type	Platform	Setup	Coord.	N	m	f_s [Hz]	τ	Hours	#Frames
Sparsh-skin [14]	Teleop. play	Xela uSkin	Magnetic	Allegro hand	Unimanual	✓	368	3	100	10	4.6	1.7 M
Tactile socks [17]	Human activity	Knitted fabric	Piezores.	Human feet	Bipedal	✓	237 + 216	1	14	5	5.9	298 k
DECO-50 [8]	Teleop. demos	Inspire FTP	Piezores.	Inspire hands	Bimanual	✗	1062×2	1	30	3	16.6	1.8 M

taxels, until the budget is filled; a global mask is formed by selecting taxels uniformly at random over the whole graph until the same condition is met. The context mask follows the logic of the global mask construction. All target taxels are removed from the context to prevent overlap. For ratios and numbers of context and target masks, see Section IV-D.

Encoding. The tactile encoder $\mathbf{E}_\theta$ is a transformer operating over the prepared tactile representation $\tilde{X}_\tau^{(k)} \in \mathbb{R}^{N \times d}$. Before entering the transformer blocks, each taxel embedding $\tilde{x}_i^{(k)}$ is summed with a taxel positional embedding $e_i \in \mathbb{R}^d$ drawn from a learnable table $\{e_i\}_{i=1}^N$. Since each taxel contributes a single element to the sequence, bidirectional attention is computed across the taxel dimension, and the encoder outputs updated per-taxel embeddings $Z \in \mathbb{R}^{N \times d}$.

During pre-training, two instances of the same tactile encoder are used. The context encoder $\mathbf{E}_\theta$ processes only the taxel embeddings contained in a context region:

$$Z_{context}^{\mathcal{C}_j} = \mathbf{E}_\theta(\{\tilde{x}_i^{(k)} + e_i\}_{i \in \mathcal{C}_j}) \in \mathbb{R}^{|\mathcal{C}_j| \times d} \quad (2)$$

The target encoder $\mathbf{E}_{\bar{\theta}}$ has the same architecture but receives all taxels:

$$Z_{target} = \mathbf{E}_{\bar{\theta}}(\{\tilde{x}_i^{(k)} + e_i\}_{i \in V}) \in \mathbb{R}^{N \times d} \quad (3)$$

Prediction. The goal of the transformer-based predictor $\mathbf{P}_\phi$ is to infer the representations of a hidden target region from the representation of the observed context region. For a context-target pair $(\mathcal{C}_j, \mathcal{T}_l)$, the predictor receives the context representation $Z_{context}^{\mathcal{C}_j}$ together with the positional embeddings of the taxels to be predicted, each summed with a learnable mask vector $u \in \mathbb{R}^d$:

$$\hat{Z}^{(\mathcal{C}_j, \mathcal{T}_l)} = \mathbf{P}_\phi(Z_{context}^{\mathcal{C}_j} \parallel \{e_i + u\}_{i \in \mathcal{T}_l}) \in \mathbb{R}^{|\mathcal{T}_l| \times d}, \quad (4)$$

where $\parallel$ denotes concatenation along the sequence dimension. Attention within $\mathbf{P}_\phi$ is computed over the concatenated sequence, and only the outputs at the target positions are retained, thereby determining the output dimension of $\hat{Z}^{(\mathcal{C}_j, \mathcal{T}_l)}$.

SSL objective. For each context region, all target regions are predicted. Let $\hat{z}_i^{(j,l)}$ denote the prediction for taxel i from $\hat{Z}^{(\mathcal{C}_j, \mathcal{T}_l)}$, and $z_i^{(l)}$ the representation of the same taxel i obtained by slicing the target encoder output Z_{target} at the taxels of $\mathcal{T}_l$. The SSL objective of Tactile-JEPA is the mean squared error (MSE) between them, averaged over all pairs:

$$\mathcal{L}_{SSL} = \frac{1}{n_c n_t} \sum_{j=1}^{n_c} \sum_{l=1}^{n_t} \frac{1}{|\mathcal{T}_l|} \sum_{i \in \mathcal{T}_l} \|\hat{z}_i^{(j,l)} - \text{sg}(z_i^{(l)})\|_2^2, \quad (5)$$

Where $\text{sg}(\cdot)$ denotes the stop-gradient operator. Backpropagation updates only the context encoder. The target encoder parameters are maintained as an exponential moving average (EMA) of the context encoder weights.

IV. EXPERIMENTS AND RESULTS

A. Datasets

We evaluate Tactile-JEPA on three publicly available tactile datasets, chosen to span distinct transduction principles, embodiments, and downstream tasks. We consider **Sparsh-skin** [14], **Tactile socks** [17], and **DECO-50** [8], which correspond to teleoperated play data collected with a sensorized robotic hand, human locomotion recorded from wearable tactile socks, and teleoperated demonstrations of contact-rich manipulation tasks on a bimanual robot, respectively. For DECO-50, we use only the data subset related to *Assembly* task as the most tactile-intensive one [8]. The characteristics of the datasets are provided in Table I.

B. Downstream evaluation

Tasks and metrics. The Sparsh-skin dataset provides three downstream tasks [14]. In *force estimation*, the model regresses tactile signals to 3-axis normal and shear forces. Labels are collected by indenting the palm sensor pad with a force/torque probe. We report the per-axis and total RMSE in centinewtons (cN). In *in-hand pose estimation*, the model tracks the planar pose $(x, y, \theta) \in SE(2)$ of an object sliding under a static robotic hand. The pose is expressed in the hand frame and labeled using ArUco tags. We report the RMSE for x and y in centimeters (cm) and for θ in degrees. We also report pose accuracy, defined as the fraction of predictions within 2 cm and 5° of the ground truth. *Object classification* identifies the manipulated object from a set of 14 items in the play data. We evaluate this task using top-1 accuracy.

In the Tactile socks dataset, there are two downstream tasks [17]. *Action classification* predicts which of 9 activities the person wearing a pair of sensor socks is performing (e.g., walking, climbing up or down stairs, jumping) from a window of pressure frames; we evaluate it with top-1 accuracy. *Full-body pose estimation* regresses the wearer’s pose, represented as 19 relative joint angles spanning the

legs, torso, and arms. We report mean joint RMSE in radians (rad) over the predicted joint angles.

On DECO-50, the goal is to *learn a visuo-tactile manipulation policy* $\pi(A_t|O_t)$ that predicts a chunk of future actions $A_t = [a_t, a_{t+1}, \dots, a_{t+H}]$, where H is the prediction horizon and each action $a_t \in \mathbb{R}^{12}$ specifies the target joint positions of the two dexterous hands (6 joints per hand), as commanded during bimanual teleoperation of the contact-rich Assembly task (plugging a socket held in one hand with a plug held in the other). The observation $O_t = [I_{1,t}, \dots, I_{n,t}, X_\tau^{(t)}]$ consists of the $n = 2$ images from the binocular head camera at time t together with the tactile history window $X_\tau^{(t)} = X[:, t - \tau : t, :] \in \mathbb{R}^{N \times \tau \times m}$ covering the τ frames preceding t . We report RMSE in normalized-action units between predicted and teleoperated actions, averaged over the $H = 16$ horizon steps and 12 joints.

Task heads. We adopt the downstream head architectures of [14], and train a separate head per task on top of the outputs of the frozen pre-trained tactile encoder.

Window-level head. For force estimation and action and object classification, each target corresponds to one tactile window; for force, the window precedes the label, keeping the estimate causal. The head is an attentive probe, i.e., a one-layer, 3-head transformer whose learned query token cross-attends to the taxel embeddings and pools them into one embedding, followed by a two-layer MLP.

Sequence-to-sequence head. Pose estimation on Sparsh-skin and Tactile socks requires longer temporal context, since pose is inferred from how contact evolves. We therefore encode consecutive windows with the frozen encoder, pool each with the attentive probe, and pass the resulting sequence to a one-layer transformer decoder that predicts the pose after each window.

Visuo-tactile policy. For DECO-50, the embeddings pooled by the attentive probe, ResNet-18 visual embeddings, and a learnable action token form one sequence processed by a two-layer bidirectional transformer. A two-layer MLP maps the output action token to the action chunk.

C. Baselines

For a temporally matched comparison, we evaluate Tactile-JEPA against baselines pre-trained in the same regime as ours, on short time windows or instantaneous frames rather than on explicit long-horizon context.

BYOL [29]. T-DEX treats an instantaneous tactile frame as a three-channel image, with the sensing pads laid out spatially, and pre-trains an ImageNet-initialized AlexNet encoder on it with BYOL. In our setup, we feed it the middle frame of each time window.

MAE [14]. Sparsh-skin (MAE) is trained by masked reconstruction of the taxel readings, with taxel coordinates concatenated to the input signal; otherwise, the tactile data view and the transformer encoder coincide with those of Tactile-JEPA.

DINO [14]. Sparsh-skin builds on the DINOv2 paradigm, where a student predicts the prototype logits of an EMA

teacher from a more heavily masked view. Masking is taxel-wise; apart from concatenated taxel coordinates, the input data view and the encoder are the same as in Tactile-JEPA.

End-to-end. Without any pre-training, the tactile encoder is randomly initialized and trained jointly with the head on the labeled data of each task. On the tactile socks only, we replace the transformer with the CNN&GRU encoder of [17].

D. Experimental Setup

Data preprocessing. For Sparsh-skin, magnetic readings are baseline-corrected by subtracting each taxel’s mean response in a no-contact configuration, which removes the static offset [14]; the residuals are standardized per sensing axis. Piezoresistive skins need no such correction, as they rest at zero without contact. Tactile socks are therefore standardized with a single scalar mean and standard deviation, computed separately for each labeled subset, and DECO-50 signals are divided by a fixed per-hand maximum. All statistics are computed on training data only.

Temporal windowing. In Tactile-JEPA, we operate over short windows of tactile history rather than instantaneous frames. We set the number of frames per window τ so that $T_w \approx 0.1$ s for Sparsh-skin and DECO-50 (Table I), motivated by the latency at which humans adjust grip force following slip [42]. The tactile socks sample at 14 Hz, where this would reduce the window to a single frame; we therefore use $\tau = 5$, giving $T_w \approx 0.36$ s. For the pose estimation tasks, the sequence decoder consumes 10 consecutive windows on Sparsh-skin and 12 on the Tactile socks.

Taxel grouping. For the DECO-50 dataset, the two hands together contain $N = 1062 \times 2 = 2124$ taxels. Since our tactile encoder attends over taxels, such a sequence length is computationally expensive, so we group $n_g = 4$ –5 adjacent taxels and project their joint response $\mathbb{R}^{n_g \times \tau \times m}$ to a single embedding. This reduces the sequence length to $\approx N/n_g$ tokens.

Mask sampling. For Tactile-JEPA, we sample 1 context and 4 targets (2 local, 2 global), with 1 local and 1 global target per sensing part in two-part embodiments. The context initially covers 50–90% of the taxels and each target 10–18% of its sampling domain. Targets are drawn independently and may overlap; their union is removed from the context.

Data splits. For Sparsh-skin, object classification and pre-training share the same trajectories, split 80/10/10% (train/val/test) within each object; test is held out. Force (64/16/20%) and pose (71.7/18.3/10.0%, trajectory-level) use separately collected data, disjoint from pre-training. For Tactile socks, action classification follows a chronological split (72.7/9.1/18.2%) and pose follows the predefined splits (83.5/3.6/12.9%). Encoders are pre-trained per task on the training portion. DECO-50 is split 70/15/15% by complete demonstrations, with the training demonstrations shared between tactile encoder pre-training and policy learning.

Architecture details. In Tactile-JEPA, the encoder $\mathbf{E}_\theta$ follows the ViT-Tiny configuration [43], [44]: 12 transformer blocks with embedding dimension $d = 192$, 3 attention heads, and an MLP expansion ratio of 4. The predictor $\mathbf{P}_\phi$

TABLE II

COMPARISON OF THE GENERAL-PURPOSE QUALITY OF LEARNED TACTILE REPRESENTATIONS ACROSS TWO DISTINCT ROBOTIC EMBODIMENTS AND DIVERSE DOWNSTREAM TASKS. BEST **BOLD**, SECOND UNDERLINED.

Dataset	Downstream	Metric	End-to-end	BYOL [29]	MAE [14]	DINO [14]	Tactile-JEPA (ours)
Sparsh-skin [14]	Object classification	Accuracy $\uparrow$	70.61 $\pm$ 2.85	64.30 $\pm$ 1.38	63.44 $\pm$ 0.64	83.26 $\pm$ 0.79	83.27$\pm$3.36
	Force estimation	RMSE [cN] $\downarrow$	<u>15.00$\pm$0.92</u>	23.79 $\pm$ 0.96	21.55 $\pm$ 2.06	16.66 $\pm$ 1.17	14.05$\pm$0.73
		x -RMSE [cN] $\downarrow$	3.84$\pm$0.20	5.70 $\pm$ 0.38	5.68 $\pm$ 0.57	4.23 $\pm$ 0.18	<u>3.91$\pm$0.24</u>
		y -RMSE [cN] $\downarrow$	<u>4.34$\pm$0.48</u>	5.55 $\pm$ 0.32	5.69 $\pm$ 0.78	4.46 $\pm$ 0.67	4.23$\pm$0.40
		z -RMSE [cN] $\downarrow$	<u>25.32$\pm$1.57</u>	40.42 $\pm$ 1.66	36.44 $\pm$ 3.49	28.19 $\pm$ 2.00	23.63$\pm$1.26
	In-hand pose est.	x -RMSE [cm] $\downarrow$	<u>1.13$\pm$0.09</u>	1.31 $\pm$ 0.13	1.60 $\pm$ 0.30	1.20 $\pm$ 0.08	0.96$\pm$0.05*
		y -RMSE [cm] $\downarrow$	<u>1.19$\pm$0.16</u>	1.36 $\pm$ 0.05	1.42 $\pm$ 0.19	1.31 $\pm$ 0.02	1.06$\pm$0.04
		θ -RMSE [$^\circ$] $\downarrow$	<u>7.20$\pm$0.87</u>	6.92 $\pm$ 0.36	8.38 $\pm$ 0.70	8.27 $\pm$ 0.44	5.48$\pm$0.39*
		x -Accuracy $\uparrow$	<u>91.93$\pm$2.05</u>	89.12 $\pm$ 1.94	85.13 $\pm$ 4.15	90.27 $\pm$ 2.21	95.74$\pm$1.05*
		y -Accuracy $\uparrow$	<u>90.30$\pm$3.21</u>	86.97 $\pm$ 0.87	87.12 $\pm$ 2.54	88.95 $\pm$ 1.03	93.14$\pm$0.99
		θ -Accuracy $\uparrow$	<u>62.85$\pm$6.24</u>	<u>63.03$\pm$2.05</u>	53.54 $\pm$ 3.10	54.02 $\pm$ 1.84	71.75$\pm$2.56*
Tactile socks [17]	Action classification	Accuracy $\uparrow$	92.97 $\pm$ 1.26	collapsed	collapsed	<u>95.63$\pm$1.53</u>	96.91$\pm$1.92
	Full-body pose est.	RMSE [rad] $\downarrow$	14.37 $\pm$ 0.30	collapsed	collapsed	<u>13.78$\pm$0.19</u>	13.42$\pm$0.33*

*Bold is statistically better than underlined ($p \leq 0.05$, one-sided Welch t -test).

TABLE III

COMPARISON OF TACTILE REPRESENTATIONS FOR VISUO-TACTILE POLICY LEARNING ON DECO-50. BEST **BOLD**, SECOND UNDERLINED.

Tactile representation	RMSE $\downarrow$
Vision-only	0.5132 $\pm$ 0.0103
Vision + Random embeddings	0.4969 $\pm$ 0.0118
Vision + End-to-end tactile encoder	0.4922 $\pm$ 0.0058
Vision + BYOL-pretrained encoder	collapsed
Vision + MAE-pretrained encoder	0.4400$\pm$0.0021
Vision + DINO-pretrained encoder	collapsed
Vision + Tactile-JEPA encoder (ours)	<u>0.4634$\pm$0.0029</u>

uses the same configuration with only 4 blocks, keeping the information needed for prediction in the encoder representations rather than in the predictor.

Training details. Tactile-JEPA is trained with AdamW, a learning rate warmed up linearly from 10^{-5} to 10^{-4} and then cosine-decayed to 10^{-6} , and weight decay increased from 0.04 to 0.4. Sparsh-skin and the Tactile-socks encoders are trained for 500 epochs at a global batch size of 256 (30 warm-up epochs); the DECO-50 encoder for 150 epochs at batch size 2048 (5 warm-up epochs). Downstream task heads are trained with AdamW at a learning rate of 10^{-4} , with early stopping on validation and batch sizes of 4-64 depending on the task. The DECO-50 policy is trained for 150 epochs at batch size 1024. Full hyperparameters are available at <https://github.com/E-Kovtun/tactile>. We adapt the DINO and MAE configurations from Sparsh-skin,¹ the CNN&GRU end-to-end baselines from the original Tactile socks implementation,² and BYOL from T-DEX.³

Evaluation protocol. For each SSL method, we pre-

train three encoders with different seeds and train three downstream heads per encoder on Sparsh-skin and four on Tactile Socks and DECO-50, yielding 9 or 12 runs per task. End-to-end baseline follows three or four independent runs. We report means and sample standard deviations. The two top-ranked methods are compared with a one-sided Welch t -test on head-averaged results per pre-training seed ($n = 3$; independent runs for non-pretrained baselines), at nominal $p \leq 0.05$ without multiple-comparison correction.

Compute and runtime. Tactile-JEPA pre-training uses four NVIDIA A100 80 GB GPUs: 6.1 h for Sparsh-skin, 0.7 h and 2.0 h for the Tactile socks action and pose encoders, and 5.5 h for DECO-50. Downstream heads are trained on a single GPU, taking 0.6–2.1 h, 0.2–0.4 h, and 4.0 h respectively.

E. Results

Tactile representation evaluation. We first evaluate whether Tactile-JEPA learns general-purpose tactile representations that transfer across downstream tasks. For each task, we train the task-specific heads described in Section IV-B on the frozen representations from the Sparsh-skin and Tactile Socks datasets. Results are reported in Table II. Compared with the strongest baseline on each metric, Tactile-JEPA reduces θ -RMSE by 20.8% and force RMSE by 6.3%. On Tactile socks, it improves action classification accuracy by 1.28% and reduces full-body pose RMSE by 2.6%. Given the substantial and consistent margin over existing methods, we conclude that *Tactile-JEPA learns representations that generalize across downstream tasks and embodiments*.

Tactile-aware policy learning. We next investigate whether the learned tactile representations improve contact-rich policy learning on DECO-50 when integrated with visual information. We compare vision-only, vision augmented with random embeddings, a tactile encoder trained jointly with the policy from random initialization, and encoders pre-trained with SSL methods. Results are given in Table III. MAE

¹<https://github.com/facebookresearch/sparsh-multisensory-touch>

²<https://github.com/YunzhuLi/senstextile>

³<https://github.com/irmakguzey/tactile-dexterity>

achieves the lowest error on DECO-50 but does not maintain consistent superiority across datasets. Tactile-JEPA achieves the second-best results with stable training and no adaptation, while DINO collapses in its standard configuration and becomes viable only after disabling iBOT loss.

Across three datasets spanning different distributed tactile sensors, Tactile-JEPA shows *robust training dynamics on heterogeneous tactile signals*: it pre-trains stably on all of them under the same masking configuration, with no per-dataset tuning. In contrast, several baselines exhibit training instability: BYOL collapses on both Tactile Socks and DECO-50, while MAE collapses on Tactile Socks.

Computational efficiency. Tactile-JEPA pre-trains 65–75% faster than DINO across all datasets while matching or exceeding its downstream performance.

V. ABLATION STUDY

Masking strategy ablation. Since Tactile-JEPA derives its topology awareness and multi-scale representations from sampling target masks over the connectivity graph, we primarily ablate the target-mask sampling strategy. In particular, we compare block-based masking from I-JEPA [30] (4I-JEPA), four local masks (4L), four global masks (4G), and our default mix of two local and two global (2L+2G). Results are provided in Table IV. As local targets focus on fine contact detail and global ones on the whole-hand contact state, single-scale targets can favor specific tasks. On DECO-50, 4G yields the lowest policy error, consistent with action prediction depending on the contact state of the whole hand rather than on any single contact region. In contrast, our default 2L+2G mix achieves the best force RMSE and in-hand x -accuracy, ranks second on DECO-50 close to 4G, and outperforms graph-agnostic I-JEPA masking on all three tasks. *Multi-scale targets thus provide a robust choice for general-purpose pre-training.*

We additionally analyze the default globally sampled context mask with a locally sampled one ([†] in Table IV). The multi-scale target setup 2L+2G with local context improves force estimation but degrades pose and policy performance substantially, confirming global context as the better general-purpose choice. Notably, 4L target masks with local context outperform I-JEPA masking on both force estimation and policy learning. Since both use compact, connected masks with the same budget and differ only in whether masks are sampled over the taxel graph, this comparison isolates and confirms the benefit of graph-based sampling.

VI. CONCLUSION

We introduce Tactile-JEPA, a self-supervised pre-training method that exploits the spatial arrangement of taxels in distributed tactile sensors. The key trait of Tactile-JEPA is multi-scale mask sampling over the taxel graph, which makes the representations topology-aware, encoding both localized tactile cues and the overall contact state of the hand. Since the taxel graph is used only for mask sampling during pre-training, the encoder requires no layout information in

TABLE IV
EVALUATION OF DIFFERENT TARGET-MASK SAMPLING STRATEGIES FOR TACTILE-JEPA ON SPARSH-SKIN (FORCE AND IN-HAND POSE EST.) AND DECO-50. BEST **BOLD**, SECOND UNDERLINED.

	Force est.	In-hand pose est.	Policy learning
Targets	RMSE ↓	x -Acc. (%) ↑	RMSE ↓
4I-JEPA	14.99±1.03	93.50±1.46	0.4959±0.0083
4L	15.52±1.03	94.32±1.86	0.4705±0.0145
4G	15.17±0.96	95.32±1.65	0.4600±0.0019
2L+2G	14.05±0.73	95.74±1.05	<u>0.4635±0.0044</u>
4L [†]	13.93±1.03	91.29±1.49	0.4893±0.0130
4G [†]	15.14±1.00	95.16±0.73	0.4807±0.0078
2L+2G [†]	13.28±0.83	92.83±2.17	0.4828±0.0102

[†]Context mask sampled locally instead of globally (default); excluded from the best and second-best marking.

downstream tasks. The resulting embeddings are general-purpose, transferring across task types and embodiments, and reduce error over the strongest baselines by 20.8% in in-hand orientation estimation and 6.3% in force estimation. Moreover, Tactile-JEPA pre-trains stably across all sensor types, whereas competing SSL methods collapse on some datasets. Overall, Tactile-JEPA brings effective wide-coverage tactile sensing within practical reach of robot learning.

ACKNOWLEDGMENT

The authors used an LLM for language editing and grammar checking, and take full responsibility for all content.

REFERENCES

- [1] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, *et al.*, “Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 80–93.
- [2] A. Keshvarparast, D. Battini, O. Battaia, and A. Pirayesh, “Collaborative robots in manufacturing and assembly systems: literature review and future research agenda,” *Journal of Intelligent Manufacturing*, vol. 35, no. 5, pp. 2065–2118, 2024.
- [3] M. Spagnuolo, G. Todde, M. Caria, N. Furnitto, G. Schillaci, and S. Failla, “Agricultural robotics: A technical review addressing challenges in sustainable crop production,” *Robotics*, vol. 14, no. 2, p. 9, 2025.
- [4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [5] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [6] K. Zhang, H. Zhang, Z. Xu, Z. Zhang, M. R. I. Prince, X. Li, X. Han, Y. Zhou, A. Ajoudani, and Y. She, “Tacvla: Contact-aware tactile fusion for robust vision-language-action manipulation,” *arXiv preprint arXiv:2603.12665*, 2026.
- [7] Z. Cheng, Y. Zhang, A. Tang, K. Wang, W. Zhang, H. Li, H. Zhang, and L. Song, “OmniTvl: Vision-tactile-language-action models with semantic-aligned tactile sensing,” *IEEE Robotics and Automation Letters*, 2026.
- [8] X. Li, Y. Sun, L. Zhang, B.-S. Huang, Y. Peng, Y. Meng, H. Jiang, S. Xie, G. Yao, A. Knoll, Z. Bing, X. Wang, and Z. Sun, “DECO: Decoupled multimodal diffusion transformer for bimanual dexterous manipulation with a plugin tactile adapter,” in *Forty-third International Conference on Machine Learning*, 2026.

- [9] X. Li, M. Cai, J. Xu, J. Zhu, H. Fan, Y. Shen, G. Ren, and H. Dong, "At-vla: Adaptive tactile injection for enhanced feedback reaction in vision-language-action models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2026, pp. 28 764–28 774.
- [10] J. Bi, K. Y. Ma, C. Hao, M. S. Zheng, and H. Soh, "Vla-touch: Enhancing vision-language-action model with dual-level tactile feedback," *IEEE Robotics and Automation Letters*, 2026.
- [11] N. F. Lepora, "Tactile robotics: Past and future," *The International Journal of Robotics Research*, p. 02783649261421615, 2026.
- [12] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.
- [13] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.
- [14] A. Sharma, C. Higuera, C. K. Bodduluri, Z. Liu, T. Fan, T. Hellebrekers, M. Lambeta, B. Boots, M. Kaess, T. Wu, F. R. Hogan, and M. Mukadam, "Self-supervised perception for tactile skin covered dexterous hands," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 305. PMLR, 27–30 Sep 2025, pp. 2311–2328.
- [15] G. Cheng, E. Dean-Leon, F. Bergner, J. R. G. Olvera, Q. Leboutet, and P. Mittendorf, "A comprehensive realization of robot skin: Sensors, sensing, control, and applications," *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2034–2051, 2019.
- [16] F. Liu, S. Deswal, A. Christou, Y. Sandamirskaya, M. Kaboli, and R. Dahiya, "Neuro-inspired electronic skin for robots," *Science robotics*, vol. 7, no. 67, p. eabl7344, 2022.
- [17] Y. Luo, Y. Li, P. Sharma, W. Shou, K. Wu, M. Foshey, B. Li, T. Palacios, A. Torralba, and W. Matusik, "Learning human–environment interactions using conformal tactile textiles," *Nature Electronics*, vol. 4, no. 3, pp. 193–201, 2021.
- [18] S. Wistreich, B. Shi, S. Tian, S. Clarke, M. Nath, C. Xu, Z. Bao, and J. Wu, "Dexskin: High-coverage conformable robotic skin for learning contact-rich manipulation," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 305. PMLR, 27–30 Sep 2025, pp. 769–793.
- [19] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta, "Reskin: versatile, replaceable, lasting tactile skins," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 08–11 Nov 2022, pp. 587–597.
- [20] 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)*. IEEE, 2016, pp. 454–460.
- [21] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto, "Anyskin: Plug-and-play skin sensing for robotic touch," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16 563–16 570.
- [22] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, *et al.*, "Paligemma: A versatile 3b vlm for transfer," *arXiv preprint arXiv:2407.07726*, 2024.
- [23] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, *et al.*, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.
- [24] 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," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 06–09 Nov 2025, pp. 885–915.
- [25] C. Higuera, A. Sharma, T. Fan, C. K. Bodduluri, B. Boots, M. Kaess, M. Lambeta, T. Wu, Z. Liu, F. R. Hogan, and M. Mukadam, "Tactile beyond pixels: Multisensory touch representations for robot manipulation," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 305. PMLR, 27–30 Sep 2025, pp. 105–123.
- [26] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE, 2022, pp. 15 979–15 988.
- [27] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *2021 IEEE/CVF international conference on computer vision (ICCV)*. IEEE, 2021, pp. 9630–9640.
- [28] J. Zhao, Y. Ma, L. Wang, and E. Adelson, "Transferable tactile transformers for representation learning across diverse sensors and tasks," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 06–09 Nov 2025, pp. 3766–3779.
- [29] I. Guzey, B. Evans, S. Chintala, and L. Pinto, "Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 229. PMLR, 06–09 Nov 2023, pp. 3142–3166.
- [30] 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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023, pp. 15 619–15 629.
- [31] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khaidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," *Transactions on Machine Learning Research Journal*, 2024.
- [32] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas, "Revisiting feature prediction for learning visual representations from video," *arXiv preprint arXiv:2404.08471*, 2024.
- [33] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 26 330–26 343.
- [34] Y. Zhou, M. Xu, J. Shi, Q. Li, and J. Chen, "Collaborative representation learning for alignment of tactile, language, and vision modalities," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 22, 2026, pp. 18 864–18 872.
- [35] R. Feng, J. Hu, W. Xia, T. Gao, A. Shen, Y. Sun, B. Fang, and D. Hu, "Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 31 265–31 285.
- [36] R. Feng, Y. Zhou, S. Mei, D. Zhou, P. Wang, S. Cui, B. Fang, G. Yao, and D. Hu, "Anytouch 2: General optical tactile representation learning for dynamic tactile perception," in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 3543–3577.
- [37] 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*, vol. 33, pp. 21 271–21 284, 2020.
- [38] J. Lin, J. Li, J. Gao, W. Ma, and Y. Liu, "Jointly modeling spatio-temporal features of tactile signals for action classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 12, 2024, pp. 13 817–13 825.
- [39] L. Chen, Y. Zhu, and M. Li, "Tactile-gat: tactile graph attention networks for robot tactile perception classification," *Scientific Reports*, vol. 14, no. 1, p. 27543, 2024.
- [40] L. Yang, B. Huang, Q. Li, Y.-Y. Tsai, W. W. Lee, C. Song, and J. Pan, "Tacgnn: Learning tactile-based in-hand manipulation with a blind robot using hierarchical graph neural network," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3605–3612, 2023.
- [41] 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)*. IEEE, 2024, pp. 7499–7506.
- [42] R. S. Johansson and G. Westling, "Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip," *Experimental brain research*, vol. 66, no. 1, pp. 141–154, 1987.
- [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.

- [44] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.