TOUCH-JEPA: TACTILE REPRESENTATION LEARNING FROM HAND-OBJECT VIDEO WITH LIMITED TACTILE SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding contact and tactile magnitude in hand-object interaction is a key step toward physically intelligent systems, yet learning such representations is limited by the cost and restricted coverage of tactile measurements. We present Touch-JEPA, a framework for learning tactile representations from hand-object videos with limited tactile supervision. We first introduce a self-supervised pretraining objective that learns from video alone without tactile supervision. Using video-derived hand geometry, Touch-JEPA predicts masked visual features at corresponding hand-surface locations from spatial and temporal context, encouraging the representation to capture latent contact dynamics. We then extend this formulation to joint representation learning, where masked prediction is applied to all videos while tactile supervision is available only for a labeled subset, grounding the shared representation in measured contact and magnitude. With self-supervised pretraining alone, simple supervised readouts on frozen Touch-JEPA features outperform supervised baselines in contact discrimination and tactile magnitude estimation across three human-hand benchmarks, and transfer to robotic hands without robot tactile training data. Joint learning with limited tactile supervision further improves contact and magnitude estimation on the human-hand benchmarks. Together, these results show that hand-object video can provide scalable self-supervision for tactile representation learning, while limited tactile measurements provide complementary physical grounding.

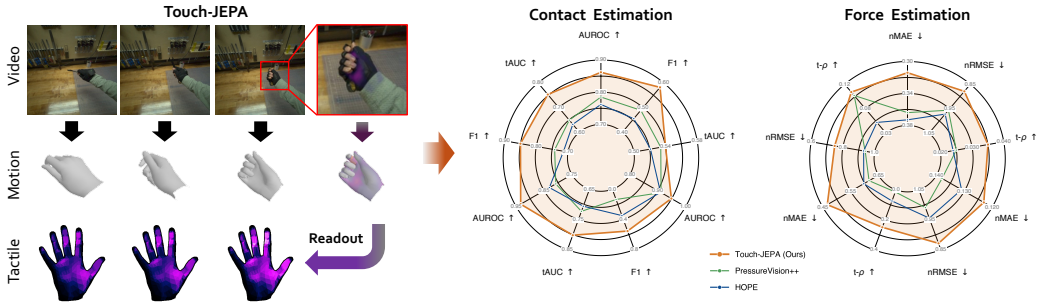


Figure 1: **Touch-JEPA learns tactile representations from video.** Supervised readouts on frozen features outperform the compared supervised baselines in contact and tactile magnitude estimation.

1 INTRODUCTION

Physical understanding of hand-object interaction underpins contact-rich manipulation (Xue et al., 2025), the transfer of dexterous skill from human video to robots (Qin et al., 2022; Shaw et al., 2023), and clinical hand-function assessment (Zhao & Zariffa, 2024). Yet visual observations reveal hand-object geometry and motion while contact state and tactile magnitude remain only indirectly observable (Grady et al., 2022). Learning representations that capture such tactile information from visual and temporal cues could therefore provide a richer understanding of hand-object interaction beyond geometry and motion (Zhao et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Prior work has largely approached tactile understanding from video through supervised prediction. Early approaches exploited load-related visual cues, including soft-tissue deformation, fingertip color changes, and cast shadows, to predict planar pressure maps (Grady et al., 2022; 2024). More recent methods extend this setting to pressure on 3D hand meshes (Zhao et al., 2025; Jeon et al., 2026), forces on hand and object meshes given object geometry (Ilic et al., 2026), and dense hand-surface contact across diverse interactions (Jung & Lee, 2026). Alongside these methods, paired hand-object video and sensor datasets have enabled supervised learning and evaluation for pressure prediction, contact estimation, retrieval, and classification (Song et al., 2025; Zhang et al., 2026; Zhou et al., 2026; Zeng et al., 2026). These advances show the potential of learning tactile information from visual observations, but depend on paired tactile measurements for supervision.

However, paired tactile data are difficult to acquire at scale (Jiang et al., 2024). Collecting such supervision often requires specialized hardware and careful calibration. Moreover, tactile gloves alter bare-hand appearance (Zhang et al., 2026), planar sensors restrict contact geometry, and instrumented objects limit object diversity (Brahmbhatt et al., 2019). Simulation and derived contact labels alleviate some of these sensing constraints but often depend on task-specific pipelines (Wang et al., 2022). Consequently, existing tactile datasets provide strong physical grounding but only fragmented coverage of hand regions, objects, and interactions.

In contrast, large-scale video datasets capture diverse hand-object interactions across broad visual and behavioral distributions, but lack explicit tactile annotation (Damen et al., 2022; Grauman et al., 2022). As such, our key insight is that tactile supervision and large-scale unlabeled video provide complementary learning signals: tactile measurements ground the representation in physically meaningful quantities, while diverse interaction videos provide broader spatial, anatomical, and temporal context. Self-supervised masked prediction Assran et al. (2025) can therefore leverage unlabeled videos to learn interaction regularities beyond the limited tactile-labeled subset, while explicit tactile supervision preserves grounding in contact and force-related quantities. Together, these signals enable tactile representations to benefit from both physically grounded supervision and the scale and diversity of natural interaction videos.

We introduce Touch-JEPA, a framework for tactile representation learning from hand-object interaction videos with limited tactile supervision. In the self-supervised setting, Touch-JEPA learns from video without tactile annotations through masked prediction. Video-derived hand geometry defines corresponding hand-surface locations, allowing visual features to be predicted from spatial and temporal interaction context across heterogeneous videos. A Geometry Encoder integrates evidence across hand regions and frames under interaction and anatomical masking. We then extend the framework to joint representation learning, where the JEPA objective is applied across all videos while tactile supervision is available only for the labeled subset. The two objectives share the encoder, allowing tactile measurements to ground the learned representation in contact and magnitude while self-supervised prediction continues to exploit the scale and diversity of the broader video distribution. We further control interference between the two learning signals during joint training to preserve their complementary information. Our contributions are summarized below:

- We introduce Touch-JEPA, a unified tactile representation learning framework that treats abundant hand-object video and limited tactile measurements as complementary sources of supervision: video provides scalable interaction coverage, while tactile measurements provide direct physical grounding of the shared representation.

- We introduce a geometry-guided self-supervised pretraining objective that learns tactile-relevant hand-surface representations from video alone. Touch-JEPA predicts masked latent features at anatomically corresponding surface locations, using video-derived hand geometry together with interaction and anatomical masking to integrate spatial and temporal evidence without requiring tactile annotations.

- We develop a joint representation learning strategy that incorporates tactile supervision from a limited labeled subset while retaining self-supervised learning across all videos. Tactile supervision shapes the shared encoder and its EMA teacher, allowing physically grounded information to propagate through representation-level self-distillation without tactile pseudo-labels, while gradient control preserves the complementary learning signals.

2 Related Work

Tactile Representation and Estimation. Instrumented gloves enable dense hand-object contact sensing (Sundaram et al., 2019; Song et al., 2025), while prior work learns tactile representations through autoencoding, cross-modal self-supervision, and cross-sensor alignment (Murali et al., 2018; Zambelli et al., 2021; Yang et al., 2022; Kerr et al., 2022; Yang et al., 2024; Feng et al., 2025; Wi et al., 2025). Recent methods further use masked visuo-tactile reconstruction or joint-embedding objectives for tactile observations (Zhu et al., 2026; Higuera et al., 2024). A complementary line of work estimates tactile quantities from vision, including dense contact (Jung & Lee, 2026), planar pressure (Grady et al., 2022; 2024), hand-surface pressure (Zhao et al., 2025; Jeon et al., 2026), and joint hand-object forces (Ilic et al., 2026). Paired video-tactile datasets support supervised prediction (Zhou et al., 2026; Zeng et al., 2026). In contrast, Touch-JEPA learns tactile representations from video without tactile targets during self-supervised pretraining and evaluates whether contact and force can be decoded from the resulting frozen features.

Self-Supervised and Hybrid Representation Learning. Self-supervised visual models learn transferable representations through self-distillation and masked prediction (Caron et al., 2021; Zhou et al., 2021; Oquab et al., 2023; Siméoni et al., 2025). Joint-embedding predictive architectures learn by predicting unobserved representations in images (Assran et al., 2023) and videos (Bardes et al., 2024; Assran et al., 2025; Mur-Labadia et al., 2026). Related work combines representation prediction with geometric, generative, or action objectives. VGGT-Ω (Wang et al., 2026a) learns reconstruction from geometric annotations and subsequently extends training to unlabeled videos through EMA self-distillation. On paired text-to-code tasks, LLM-JEPA (Huang et al., 2026) jointly optimizes JEPA latent prediction and token-level supervision, combining learned representation targets with explicit output sequences. JEPT (Luo & Lu, 2025) combines self-supervised visual transition prediction on labeled and unlabeled videos with supervised inverse dynamics learning on action-labeled demonstrations. Touch-JEPA adapts latent prediction to anatomically corresponding hand-surface locations across time and combines masked prediction on all videos with tactile supervision on the labeled subset.

3 Touch-JEPA

Touch-JEPA learns hand-surface representations by predicting masked visual features at anatomically corresponding locations. The student x-encoder predicts targets from the teacher y-encoder, updated by an exponential moving average (EMA). Visual prediction learns from all videos, while tactile supervision anchors the shared representation to measured tactile-relative intensity.

Problem formulation. Given a T-frame RGB clip x, a frozen HaWoR estimator (Zhang et al., 2025) recovers MANO (Romero et al., 2017) geometry g using dataset-provided camera intrinsics. Each frame contains $V = 778$ vertices and $J = 21$ joints with consistent anatomical indices. For vertex v at time t, $u_{t,v}$ denotes its image projection, $\iota_{t,v}$ its in-image indicator, and $s_{t,v}$ its mesh-based visibility indicator. Mesh visibility requires image inclusion, camera-facing orientation, and no self-occlusion. Occlusion by objects or the other hand is not modeled.

3.1 Geometry-Guided Masked Visual Prediction

The x-encoder produces vertex and joint states $z = f_\theta(\hat{x}, g, a)$ from masked RGB $\hat{x}$, unmasked geometry g, and image-read permissions a. The y-encoder observes unmasked RGB. Its targets are computed directly from the visual field without surface-state feedback. Geometry specifies where to sample, while video determines the target content, requiring prediction of visual features at anatomically corresponding hand-surface locations.

Let $\bar{h}_t$ denote the y-encoder's video-feature field aligned to frame t. For backbone width D and surface width $d = 256$, we define the projected target field and vertex targets as

$$\bar{Y}_t = \texttt{LayerNorm}(\bar{h}_t R), \qquad \bar{y}_{t,v} = \texttt{Sample}(\bar{Y}_t, u_{t,v}), \tag{1}$$

where $R \in \mathbb{R}^{D\times d}$ has fixed orthonormal columns and `Sample` denotes bilinear sampling. Joint targets are sampled analogously at projected joint locations.

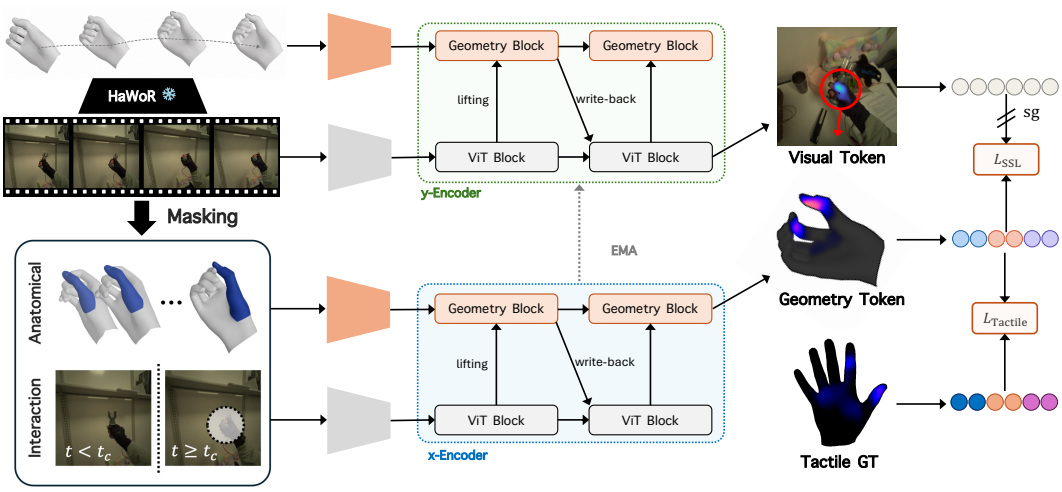


Figure 2: **Touch-JEPA overview.** Given a human-object interaction video, the y-encoder processes unmasked RGB to produce target visual features at geometry-aligned hand-surface locations. The x-encoder combines masked RGB with estimated hand geometry to predict these targets. When tactile annotations are available, joint training additionally supervises contact and force-related predictions decoded from masked x-encoder states before the predictor, providing complementary physical grounding for the shared representation.

Interaction masking. We select a split t_c between context and target frames based on valid hand observations. For image-patch center c_p and masking radius r, the interaction mask is

$$m_{t,p} = \mathbf{1}[t \geq t_c] \max_v \{s_{t,v}\, \mathbf{1}[\|c_p - u_{t,v}\| \leq r]\}\,. \tag{2}$$

Pixels in masked patches are zeroed after RGB normalization. The x-encoder must recover withheld evidence in target frames using context frames, remaining scene regions, and target-frame geometry.

Anatomical masking. We additionally select a set of finger or palm regions $\mathcal{H}$ and mask neighborhoods around their projected vertices throughout the clip. Their vertex and joint tokens lose direct image-read access during initial sampling and per-layer lifting. Local image-window attention is also blocked for their vertex tokens. Geometry remains available, together with spatial and temporal communication. Interaction masking preserves visual context before t_c. Anatomical masking additionally removes direct visual evidence for selected regions across all frames, encouraging inference from other hand regions. Sampling rules are provided in the appendix.

3.2 Geometry Encoder

Vertex states $H_t \in \mathbb{R}^{V\times d}$ and joint states $S_t \in \mathbb{R}^{J\times d}$ pass through four layers interleaved with the final four V-JEPA 2.1 ViT-B/16 blocks (Mur-Labadia et al., 2026). Backbone adaptation is restricted to rank-8 LoRA (Hu et al., 2021) in these blocks. The visual path without surface feedback and the interleaved path share backbone weights. Local attention preserves anatomical structure, while global and temporal exchange integrates evidence across regions and frames.

Visual lifting. The initial field F_t sums projected, temporally aligned features of the masked clip $\hat{x}$ from the x-encoder's visual path without surface feedback. For vertex attributes $g_{t,v}$ extracted from g, we initialize

$$H^0_{t,v} = \gamma(g_{t,v}) + \texttt{PE}(v,t) \quad + a_{t,v}\phi_{\text{img}}\big(\texttt{Sample}(F_t, u_{t,v})\big) + (1 - a_{t,v})e_{\emptyset}, \tag{3}$$

where γ embeds geometry and visibility, PE encodes template position, identity, anatomical region, and time, ϕ_{img} is an affine projection, and $e_{\emptyset}$ represents missing image evidence. Permission $a_{t,v}$ equals $\iota_{t,v}$ outside anatomically masked regions and zero otherwise. Joints are initialized analogously. Each layer adds permitted samples from its ViT features. Vertices also attend to local image windows containing initial and current-layer features.

Anatomical attention. Vertices are partitioned into 20 parts $\mathcal{P}$, comprising 15 phalangeal parts and five palm zones assigned by geodesic proximity to the fingers. Within-part attention first updates the vertex states. Pooled part tokens then exchange information and broadcast updates to their constituent vertices through

$$h_{t,P} = \phi_{\downarrow}\left(|P|^{-1}\sum_{w\in P}\texttt{LayerNorm}(H_{t,w})\right),$$
$$H_{t,v} \leftarrow H_{t,v} + \phi_{\uparrow}\big(\texttt{Attn}(h_{t,P(v)}, \{h_{t,P}\}_{P\in\mathcal{P}})\big), \tag{4}$$

where $P(v)$ denotes the part containing vertex v, $\phi_{\downarrow}$ projects to 64 dimensions, $\texttt{Attn}$ denotes multi-head attention, and $\phi_{\uparrow}$ is a zero-initialized projection back to width d. A token-wise feed-forward network completes the block.

Global and temporal exchange. A global attention core combines 195 downsampled surface tokens with 21 joint tokens at width 128. Full self-attention exchanges information across these tokens, with residual updates returned directly to joints and through fixed mesh upsampling to vertices. A shared causal block then integrates each vertex and joint track across frames, while the video encoder remains noncausal. The first three layers feed surface states back to video patches through gated bilinear splatting. Final $\texttt{LayerNorm}$ operations produce z. Architecture and initialization details appear in Appendix C.2.

3.3 Self-Supervised Training

A token-wise residual predictor $q_\omega(z) = z + \texttt{MLP}_\omega(z)$ maps x-encoder states to y-encoder features without mixing tokens or frames. The MLP's final layer is zero-initialized, making q_ω initially the identity. For vertex or joint token i, the dense prediction loss is

$$\mathcal{L}_{\text{dense}} = \Big\langle \|q_\omega(z_{t,i}) - \texttt{sg}(\bar{y}_{t,i})\|_1/d \Big\rangle_w, \tag{5}$$

where $\texttt{sg}$ denotes stop-gradient and $\langle\cdot\rangle_w$ denotes a weighted mean computed within each data lane and then averaged across lanes. Each lane corresponds to a data-source group. Weights prioritize mesh-visible, anatomically masked vertices and visually supported target-frame tokens over context tokens. Visual support requires mesh visibility for vertices and valid image projections for joints. Eligible target vertices receive additional weight based on changes in y-encoder features from the last context frame. Exact weighting rules are provided in the appendix.

A position-conditioned decoder D_ξ predicts patch targets $\bar{F}_{t,p} \in \mathbb{R}^{d_p}$ from vertex and joint states. These targets are the normalized counterpart of F_t computed by the y-encoder without surface feedback. For patch-feature width d_p, the reconstruction loss is

$$\mathcal{L}_{\text{patch}} = \Big\langle \|D_\xi(z)_{t,p} - \texttt{sg}(\bar{F}_{t,p})\|_1/d_p \Big\rangle_{m_{t,p}=1}. \tag{6}$$

The mean is computed over interaction-masked patches within each lane and then averaged across lanes. This loss encourages the representation to retain surrounding interaction evidence beyond individual vertex and joint samples. The target-field construction and decoder are detailed in the appendix. The self-supervised objective is

$$\mathcal{L}_{\text{SSL}} = \mathcal{L}_{\text{dense}} + 0.5\mathcal{L}_{\text{patch}} + 0.05\mathcal{L}_{\text{wSIG}}. \tag{7}$$

Inspired by SIGReg (Balestriero & LeCun, 2025), WeakSIGReg complements this temporal smoothing by encouraging the covariance of randomly projected vertex features toward identity, discouraging dimensional collapse.

3.4 Tactile Supervision

Visual prediction learns from all videos, while tactile labels anchor the shared representation to measured contact and sensor-relative intensity. Fixed dataset-specific correspondences map annotations to sensor-supported vertices, with supervision determined by annotation validity independently of visual visibility. A two-layer head maps masked x-encoder states before the predictor to $(o, \rho) = h_\psi(z)$, with contact logit o and raw magnitude ρ. The contact probability and contact-conditional magnitude are $\hat{c} = \sigma(o)$ and $\hat{m} = \texttt{softplus}(\rho)$, respectively, yielding the gated prediction $\hat{y} = \hat{c}\,\hat{m}$, where σ denotes sigmoid.

For ground-truth magnitude y normalized by the dataset's training-set standard deviation, we define

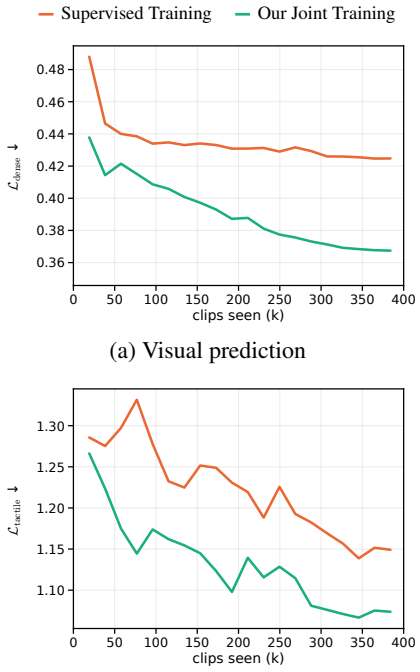
$$\mathcal{L}_{\text{tactile}} = \mathcal{L}_{\text{contact}} + \lambda_m \Big\langle |\log(1+\hat{m}) - \log(1+y)| \Big\rangle_{\mathcal{C}} + \mathcal{L}_{\text{head}}, \tag{8}$$

where $\mathcal{L}_{\text{contact}}$ is class-weighted binary cross-entropy, $\mathcal{C}$ contains valid ground-truth contact entries, and λ_m weights magnitude supervision per dataset. Contact and log-magnitude losses update the encoder and head. The L1 loss $\mathcal{L}_{\text{head}}$ fits gated predictions to y over all valid entries using $\mathtt{sg}(z)$, updating only the head. Losses are normalized within labeled clips and averaged across clips.

The joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{SSL}} + \lambda_{\text{tac}}(k)\mathcal{L}_{\text{tactile}}, \tag{9}$$

where k denotes the training step and λ_{tac} is zero for SSL only or warmed up to one for joint training. At z, we remove the tactile gradient component opposing the SSL gradient (Yu et al., 2020) and cap its remaining norm relative to an SSL-gradient reference. Tactile supervision updates the Geometry Encoder and LoRA-adapted backbone. The EMA y-encoder carries these changes into visual targets on all videos, enabling representation-level self-distillation beyond labeled clips without tactile pseudo-labels. For both variants, tactile estimation removes input masks and fits a new readout on the frozen y-encoder using training-split tactile labels. The SSL predictor and any training-time tactile head are discarded.



(a) Visual prediction

(b) Tactile prediction

Figure 3. Visual and tactile prediction.

We compare joint training with Labels only using the same encoder architecture and matched clip budgets. Labels only omits visual prediction gradients to the encoder, and its dense prediction loss plateaus. Joint training continues to reduce this loss against its own y-encoder targets (Figure 3a) while maintaining lower held-out tactile loss than Labels only throughout training (Figure 3b). These trends support the benefit of visual prediction for tactile learning while retaining the predictive task. Evaluation details are provided in Appendix F.

3.5 TOUCH-JEPA-AC

We assess the utility of the learned representations for simulated robot manipulation through action-conditioned planning (Assran et al., 2025). Touch-JEPA-AC freezes the y-encoder and trains a block-causal predictor to forecast pooled video features, pooled hand-surface representations, and robot state from observed history and issued actions. A MANO proxy fitted to robot fingertips through forward kinematics supplies the hand geometry. Robot contact and force annotations are used only for evaluation.

At decision τ, candidate action $\mathbf{a}$ produces predicted scene and surface features $\hat{B}_{\tau+1}(\mathbf{a})$ and $\hat{U}_{\tau+1}(\mathbf{a})$. We compare them with reference subgoal features using

$$E_\tau(\mathbf{a}) = \mathtt{MAE}\big(\hat{B}_{\tau+1}(\mathbf{a}), B^\star_\tau\big) \quad + w_\tau \mathtt{MAE}\big(\hat{U}_{\tau+1}(\mathbf{a}), U^\star_\tau\big), \tag{10}$$

where $\mathtt{MAE}$ averages absolute differences, starred features encode the current subgoal, and w_τ is a task-dependent surface weight. The cross-entropy method selects palm and grip commands with a one-step horizon and replans after execution. Scripted closing, test lifts, release, and motion constraints complement planning. Details are provided in Appendix H.2.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Implementation details. We use 16-frame clips at 384×384 resolution. The Geometry Encoder has four layers of width 256 with four attention heads. All three training settings process 384,000



Figure 4: **Qualitative comparison of tactile magnitude estimation.** Three interaction sequences unfold from left to right, with brighter colors indicating greater tactile magnitude.

clip presentations over 12,000 updates with batch size 32. We use AdamW with learning rates of 10^{-4} for LoRA and 3×10^{-4} for new modules, with 4% warmup followed by cosine decay. EMA momentum increases from 0.996 to 0.9995. Further training and readout details appear in Appendices A and E.

Datasets. Encoder training uses three tactile sensing datasets, HandSense (Zhang et al., 2026), PressureVisionDB (Grady et al., 2022), and OpenTouch (Song et al., 2025), alongside standard hand-object interaction datasets, HRDexDB-Human (Lim et al., 2026), DexYCB (Chao et al., 2021), HO3D (Hampali et al., 2021), HO-Cap (Wang et al., 2026b), and SHOWMe (Swamy et al., 2023). All three training settings use the same video sources. Joint training and the Labels only control additionally use tactile measurements from the three tactile datasets. Contact and magnitude estimation are evaluated on held-out test clips from OpenTouch, PressureVisionDB, and HandSense.

Evaluation metrics. Within each comparison, methods are scored on identical valid entries. For contact, we report pooled AUROC and F1. Temporal AUROC (tAUC) is computed separately for each sensor within a clip and averaged across eligible sequences. For tactile magnitude, nMAE and nRMSE measure errors over all valid entries, normalized by each dataset's training-set standard deviation of the target magnitude. Temporal Spearman correlation (t-ρ) is computed separately for each sensor within a clip using only ground-truth contact frames, then averaged across eligible sequences. It measures within-contact temporal rank agreement in sensor-native magnitude, not calibrated force recovery. Additional details for metrics are provided in Appendix E.

4.2 Experimental Results

Contact Estimation. Without tactile annotations during encoder pretraining, Ours (SSL) outperforms all four retrained supervised baselines in pooled AUROC and F1 across all three datasets using supervised readouts on frozen encoders (Table 1). These results support hand-surface masked visual prediction as a source of contact representations. Temporal performance varies across datasets. On PressureVision DB, SSL achieves a tAUC of 0.783 against the strongest baseline's 0.749, while PressureFormer scores higher on OpenTouch and HandSense. Joint training achieves the best AUROC, F1, and tAUC on every dataset, including a tAUC increase over SSL from 0.639 to 0.753 on HandSense. The static prior reaches an OpenTouch AUROC of 0.832 but remains at chance in tAUC, emphasizing the need to evaluate contact timing alongside pooled discrimination.

Table 1: **Contact estimation.** Joint training achieves the best performance across all datasets, while SSL alone outperforms supervised baselines on most metrics. The first, second, and third ranks are assigned among learned methods.

Method	OpenTouch			PVDB			HandSense		
	AUROC ↑	F1 ↑	tAUC ↑	AUROC ↑	F1 ↑	tAUC ↑	AUROC ↑	F1 ↑	tAUC ↑
PressureVision	0.659	0.342	0.509	0.876	0.286	0.749	0.701	0.640	0.578
PressureVision++	0.649	0.342	0.513	0.743	0.179	0.654	0.690	0.645	0.588
PressureFormer	0.785	0.489	0.534	0.912	0.135	0.723	0.808	0.742	0.649
HOPE	0.762	0.455	0.521	0.910	0.348	0.706	0.831	0.714	0.632
Ours (SSL)	0.836	0.549	0.527	0.925	0.475	0.783	0.889	0.791	0.639
Ours (joint)	0.862	0.582	0.540	0.949	0.553	0.802	0.934	0.850	0.753
Static prior	0.832	0.542	0.500	0.819	0.335	0.500	0.559	0.618	0.500

Table 2: **Tactile magnitude estimation.** Joint training achieves the best displayed results, while SSL alone surpasses the retrained supervised baselines on most metrics. Targets are nominal pressure in PVDB and uncalibrated sensor magnitudes in OpenTouch and HandSense. Errors are computed over all valid comparison entries. The t-ρ metric measures temporal rank agreement within ground-truth contact. N/A denotes undefined correlation for the static prior.

Method	OpenTouch			PVDB			HandSense		
	nMAE ↓	nRMSE ↓	t-ρ ↑	nMAE ↓	nRMSE ↓	t-ρ ↑	nMAE ↓	nRMSE ↓	t-ρ ↑
PressureVision	0.363	1.100	0.002	0.129	0.927	0.076	0.734	1.195	0.005
PressureVision++	0.364	1.102	0.000	0.136	0.967	0.038	0.740	1.205	0.007
PressureFormer	0.364	0.956	0.025	0.138	0.991	0.014	0.616	0.941	0.102
HOPE	0.373	0.972	0.024	0.131	0.956	0.074	0.597	0.931	0.047
Ours (SSL)	0.326	0.914	0.030	0.129	0.893	0.239	0.552	0.896	0.079
Ours (joint)	0.314	0.879	0.035	0.123	0.872	0.249	0.466	0.748	0.109
Static prior	0.398	0.898	N/A	0.221	0.955	N/A	0.755	1.047	N/A

Tactile Magnitude Estimation. Supervised readouts on frozen Ours (SSL) features achieve lower nRMSE than all four retrained supervised baselines across the three datasets (Table 2). On PVDB, within-contact temporal correlation reaches 0.239, compared with 0.076 for the strongest supervised baseline, and increases to 0.249 with joint training. Joint training achieves the best results across all metrics in this comparison. The largest error reductions occur on HandSense, where nMAE decreases from 0.552 to 0.466 and nRMSE from 0.896 to 0.748 relative to SSL. However, joint-model temporal correlations remain low on OpenTouch and HandSense, at 0.035 and 0.109. These results support improved sensor-native magnitude estimation through visual prediction and complementary tactile supervision, while fine-grained within-contact temporal tracking remains challenging.

Table 3: **Effectiveness of individual components.**

Variant	Contact			Magnitude		
	AUROC ↑	F1 ↑	tAUC ↑	nMAE ↓	nRMSE ↓	t-ρ ↑
Full model (SSL)	0.883	0.601	0.647	0.331	0.890	0.111
w/o anatomical masking	0.871	0.592	0.633	0.333	0.897	0.098
w/o interaction masking	0.868	0.594	0.638	0.339	0.915	0.081
w/o geometry encoder	0.808	0.518	0.637	0.364	0.973	0.109
w/o video (geometry only)	0.801	0.423	0.605	0.374	0.989	0.055
V-JEPA 2.1 (frozen)	0.824	0.550	0.640	0.350	0.934	0.088

Ablation Studies. Table 3 shows that the full SSL model achieves the best results across all six dataset-averaged metrics. Compared with frozen V-JEPA 2.1, AUROC and F1 improve by 0.059 and 0.051, while nRMSE decreases by 4.7% and t-ρ increases from 0.088 to 0.111. Removing the Geometry Encoder lowers AUROC from 0.883 to 0.808 and raises nRMSE from 0.890 to 0.973. Using geometry alone reduces AUROC to 0.801 and increases nRMSE to 0.989, while t-ρ falls to 0.055, highlighting the contribution of visual evidence. Both masking strategies improve contact and magnitude estimation. Without interaction masking, AUROC falls to 0.868 and nRMSE rises to 0.915, while removing anatomical masking lowers tAUC from 0.647 to 0.633. Together, these results support combining hand geometry and masked visual prediction to learn tactile-relevant representations beyond frozen video features.

Table 4: **Zero-shot transfer to Allegro.**

Method	AUROC ↑	AP ↑	F1 ↑	ρ ↑
PressureVision	0.506	0.443	0.474	−0.032
PressureVision++	0.435	0.395	0.429	−0.039
HOPE	0.582	0.510	0.535	0.159
Ours (SSL)	0.676	0.608	0.608	0.183
Ours (joint)	0.628	0.529	0.569	0.106

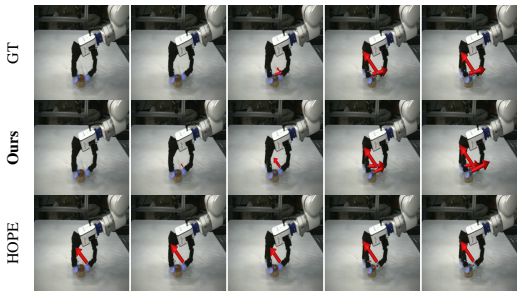


Figure 5: **Comparison on the Allegro hand.**

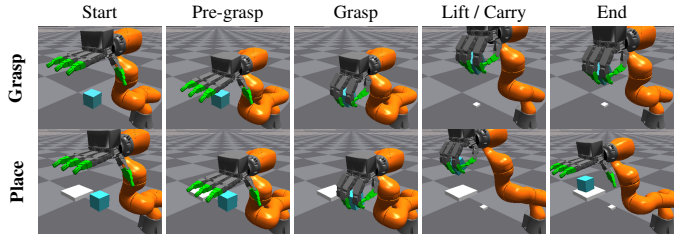


Figure 6: **Robot manipulation rollouts.** Grasp-lift and place with time progressing from left to right.

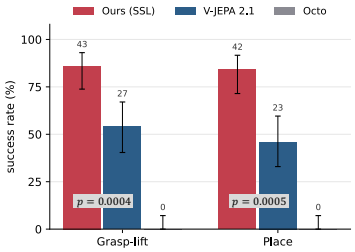


Figure 7: **Task success rates.**

4.3 Robotic Transfer and Manipulation

Transfer to Robotic Hands. We evaluate HandSense-trained predictors without adaptation on 89 real Allegro episodes comprising 10,562 frames. Ours (SSL) leads all four metrics and significantly outperforms the joint model (Table 4). Its AUROC and magnitude correlation are 0.676 and 0.183, compared with 0.628 and 0.106 for joint training. The joint model also outperforms the compared supervised baselines in contact estimation, while SSL achieves the strongest overall transfer performance. We therefore use the SSL encoder for planning.

Action-Conditioned Planning for Robotic Manipulation. Using the frozen SSL encoder, Touch-JEPA-AC plans simulated grasp-lift and place actions toward reference subgoals with an action-conditioned predictor trained on robot trajectories (Appendix H.2). Across 50 matched scenes per task, success rates are 86% for grasp-lift and 84% for place, compared with 54% and 46% for V-JEPA 2.1. Exact two-sided McNemar tests give unadjusted $p = 0.0004$ and $p = 0.0005$, respectively. Octo succeeds in neither task (Figure 7). Representative rollouts appear in Figure 6. These results support the utility of the learned representations for simulated manipulation beyond tactile estimation.

5 Conclusion

We presented Touch-JEPA, a unified framework for tactile representation learning from hand-object video and tactile measurements when available. Geometry-guided masked visual prediction provides a self-supervised foundation, enabling contact and sensor-native magnitude estimation through supervised readouts on frozen features. Joint training incorporates complementary tactile supervision to further improve human-hand estimation. The self-supervised representations also transfer to robotic hands and support simulated manipulation, providing additional evidence of their utility beyond tactile estimation.

Limitations and Future Work. Results use a single training seed, and inference depends on hand geometry. The full video encoder remains noncausal. Magnitude results are sensor-specific and do not establish cross-sensor force calibration. Within-contact temporal correlations remain low on OpenTouch and HandSense, indicating that fine-grained magnitude tracking remains challenging. The current evaluation does not isolate the contributions of visual ambiguity, sensor characteristics, and model limitations to these low correlations. Detailed failure analysis is provided in Appendix G. Future work includes scaling pretraining to more diverse videos, improving robustness to geometry errors, and extending robot evaluation to broader tasks and physical hardware.

AI USE STATEMENT

In this work, we used large language models (LLMs) to refine the manuscript and improve its clarity and readability. We did not use generative AI tools for information retrieval or discovery, or to draft sections of the paper. The generation of synthetic datasets and the formulation or proof of mathematical claims are not applicable to this work. All AI-assisted edits were reviewed and verified by the authors for accuracy and integrity. The authors take full responsibility for the final content of the manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training objectives, data processing and mixture, implementation details, and evaluation protocols in the main paper and appendix. Additional experimental details, including training configurations and evaluation settings, are provided in the appendix to facilitate reproduction of our results.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8701–8711. IEEE, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9040–9049. IEEE, 2021.

Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *International Conference on Learning Representations*, volume 2025, pp. 31265–31285, 2025.

Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. Pressurevision: estimating hand pressure from a single rgb image. In *European Conference on Computer Vision*, pp. 328–345. Springer, 2022.

Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. Pressurevision++: Estimating fingertip pressure from diverse rgb images. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8683–8693. IEEE, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. Ho-3d_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *arXiv preprint arXiv:2107.00887*, 2021.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hai Huang, Yann LeCun, and Randall Balestriero. Llm-jepa: Large language models meet joint embedding predictive architectures. In *International Conference on Learning Representations*, volume 2026, pp. 105717–105737, 2026.

Andela Ilic, Rachel Schuchert, Yijing Jiang, and Christian Holz. Egophi: Estimating 3d hand-object contact and force from egocentric vision. In *European Conference on Computer Vision*, pp. 410–428. Springer, 2026.

Subin Jeon, Byungjun Kim, and Hanbyul Joo. Hope: Hand-object pressure estimation from monocular videos. *arXiv preprint arXiv:2608.06192*, 2026.

Chunpeng Jiang, Wenqiang Xu, Yutong Li, Zhenjun Yu, Longchun Wang, Xiaotong Hu, Zhengyi Xie, Qingkun Liu, Bin Yang, Xiaolin Wang, et al. Capturing forceful interaction with deformable objects using a deep learning-powered stretchable tactile array. *Nature Communications*, 15(1): 9513, 2024.

Daniel Jung and Kyoung Mu Lee. Learning dense hand contact estimation from imbalanced data. *Advances in Neural Information Processing Systems*, 38:120351–120384, 2026.

Justin Kerr, Huang Huang, Albert Wilcox, Ryan Hoque, Jeffrey Ichnowski, Roberto Calandra, and Ken Goldberg. Self-supervised visuo-tactile pretraining to locate and follow garment features. *arXiv preprint arXiv:2209.13042*, 2022.

Jongbin Lim, Taeyun Ha, Mingi Choi, Jisoo Kim, Byungjun Kim, Subin Jeon, and Hanbyul Joo. Hrdexdb: A paired human-robot dataset for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2604.14944*, 2026.

Hao Luo and Zongqing Lu. Learning video-conditioned policy on unlabelled data with joint embedding predictive transformer. In *International Conference on Learning Representations*, volume 2025, pp. 88696–88719, 2025.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Adithyavairavan Murali, Yin Li, Dhiraj Gandhi, and Abhinav Gupta. Learning to grasp without seeing. In *International Symposium on Experimental Robotics*, pp. 375–386. Springer, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pp. 570–587. Springer, 2022.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pp. 654–665. PMLR, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yuxin Ray Song, Jinzhou Li, Rao Fu, Devin Murphy, Kaichen Zhou, Rishi Shiv, Yaqi Li, Haoyu Xiong, Crystal Elaine Owens, Yilun Du, et al. Opentouch: Bringing full-hand touch to real-world interaction. *arXiv preprint arXiv:2512.16842*, 2025.

Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569 (7758):698–702, 2019.

Anilkumar Swamy, Vincent Leroy, Philippe Weinzaepfel, Fabien Baradel, Salma Galaaoui, Romain Brégier, Matthieu Armando, Jean-Sebastien Franco, and Grégory Rogez. Showme: Benchmarking object-agnostic hand-object 3d reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1927–1936. IEEE, 2023.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω, 2026a. URL https://arxiv.org/abs/2605.15195.

Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. Ho-cap: A capture system and dataset for 3d reconstruction and pose tracking of hand-object interaction. *Advances in Neural Information Processing Systems*, 38, 2026b.

Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, 2022.

Youngsun Wi, Jessica Yin, Elvis Xiang, Akash Sharma, Jitendra Malik, Mustafa Mukadam, Nima Fazeli, and Tess Hellebrekers. Tactalign: Human-to-robot policy transfer via tactile alignment. 2025.

Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.

Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26330–26343. IEEE, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Martina Zambelli, Yusuf Aytar, Francesco Visin, Yuxiang Zhou, and Raia Hadsell. Learning rich touch representations through cross-modal self-supervision. In *Conference on Robot Learning*, pp. 1415–1425. PMLR, 2021.

Yuan Zeng, Yujia Shi, Tiao Tan, Xingting Li, Yaqi Qin, Zongqing Lu, Wenming Yang, Jing-Hao Xue, and Qingmin Liao. Egotactile: Learning grasp pressure for everyday objects from egocentric video. *arXiv preprint arXiv:2606.09243*, 2026.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1805–1815. IEEE, 2025.

Xinyu Zhang, Ziyi Kou, Chuan Qin, Mia Huang, Ergys Ristani, Ankit Kumar, Lele Chen, Kun He, Abdeslam Boularias, and Li Guan. Glove2hand: Synthesizing natural hand-object interaction from multi-modal sensing gloves. *arXiv preprint arXiv:2603.20850*, 2026.

Nicholas Zhao and Jose Zariffa. Automated hand prehension assessment from egocentric video after spinal cord injury. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 32:2864–2872, 2024.

Yiming Zhao, Taein Kwon, Paul Streli, Marc Pollefeys, and Christian Holz. Egopressure: A dataset for hand pressure and pose estimation in egocentric vision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27727–27738. IEEE, 2025.

Jianyi Zhou, Ziteng Gao, Feiyang Hong, Zirui Liu, Guannan Zhang, Weisheng Dai, Ruichen Zhen, Chuqiao Lyu, Haotian Wu, Yinian Mao, et al. Touchanything: A dataset and framework for bimanual tactile estimation from egocentric video. *arXiv preprint arXiv:2605.13083*, 2026.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. *Advances in Neural Information Processing Systems*, 38: 153783–153812, 2026.

SUPPLEMENTARY MATERIAL

A DATA AND TRAINING PROTOCOL

Video corpus and tactile coverage. Table 5 summarizes the 110,147 training clips used for encoder learning. Each clip contains $T = 16$ frames. Tactile measurements are available for 30,329 clips, or 27.5% of the corpus, from three sensor datasets. The remaining 79,818 clips provide visual self-supervision. Clip counts refer to temporal windows. OpenTouch windows overlap, and HRDex-human windows are counted separately for each camera view. These counts therefore describe the training roster rather than unique frames.

Table 5: **Encoder-training corpus.** Counts are 16-frame training clips. The four datasets in the HOI group share one sampling lane. Tactile measurements supervise encoder updates only in joint and Labels only training.

Dataset	# Clips	Available tactile measurements
HandSense	3,721	Five fingertip channels in glove counts
PressureVision DB	3,970	Surface pressure in nominal kPa
OpenTouch	22,638	169 taxels in sensor counts
HRDex-human	40,536	None
DexYCB	17,200	None used
HO3D	4,058	None used
HO-Cap	15,346	None used
SHOWMe	2,678	None used
Total	110,147	30,329 clips with tactile measurements

Partitions and sensor support. PressureVision DB uses its participant-based training and test folds, with 24 and 6 participants, respectively. HandSense and OpenTouch use recording-disjoint partitions. HandSense contains 161 training and 18 test recordings, while OpenTouch contains 970 and 123. The corresponding test sets contain 447, 1,224, and 2,519 clips for HandSense, PressureVision DB, and OpenTouch. HandSense uses the released synthesized bare-hand videos and binary contact annotations. PressureVision DB provides pressure on palmar vertices within the sensing pad and facing the pad. OpenTouch retains the 169 supported taxels and uses 3072 minus the released sensor value as magnitude. Sensor validity determines supervised entries independently of image visibility. HRDex contributes human videos, with robot images and measurements excluded from encoder training.

Matched training schedules. SSL, Labels only, and joint training share the encoder initialization, sampling configuration, and seed 912. Each processes 384,000 clip presentations through 12,000 updates with global batch size 32. The first 1,200 updates sample only the three sensor datasets. Each worker receives 3 HandSense, 3 PressureVision DB, and 10 OpenTouch clips per update. Subsequently, these counts become 1, 1, and 3, with 4 HRDex-human and 7 HOI clips. The mixed phase therefore draws 68.75% of its clips from datasets without tactile measurements. Across the complete schedule, this fraction is 61.875%. Within each lane, sampling uses shuffled permutations with disjoint worker shards. An evaluation interval spans 600 updates or 19,200 clip presentations.

Optimization and supervision. Inputs have resolution 384×384. Training selects the observer view or a hand-centered crop with equal probability. Crop expansion is 1.5 times a uniform multiplier in $[0.8, 1.2]$, with independent center shifts of up to 10% of the crop side along each axis. AdamW uses $\beta = (0.9, 0.95)$ and weight decay 0.04, with learning rates 10^{-4} for LoRA and 3×10^{-4} for new modules. A 480-update warmup starts at one sixth of these rates, followed by cosine decay to 10^{-6}. We clip the gradient norm at 1.0 and increase EMA momentum from 0.996 to 0.9995. Training uses bfloat16 and gradient checkpointing. SSL receives no tactile gradient in the encoder. Detached online probes use a separate optimizer and serve only as diagnostics. Joint training adds contact and magnitude supervision from the three sensor datasets, with λ_{tac} increasing from 0.1 to 1 over the first 1,200 updates. Labels only applies the same supervised losses with

unit weight throughout training and no SSL-relative gradient cap. Neither supervised setting uses geometric contact labels from DexYCB or tactile pseudo-labels.

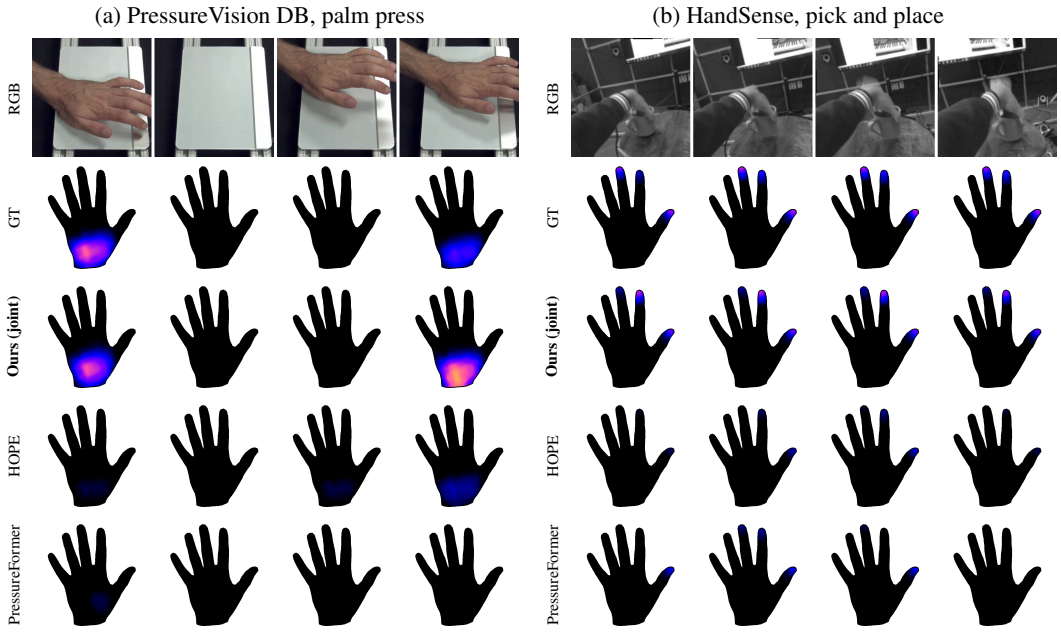


Figure 8: **Additional comparisons on PressureVision DB and HandSense.** Four frames per sequence progress from left to right. Brighter colors indicate greater tactile magnitude.

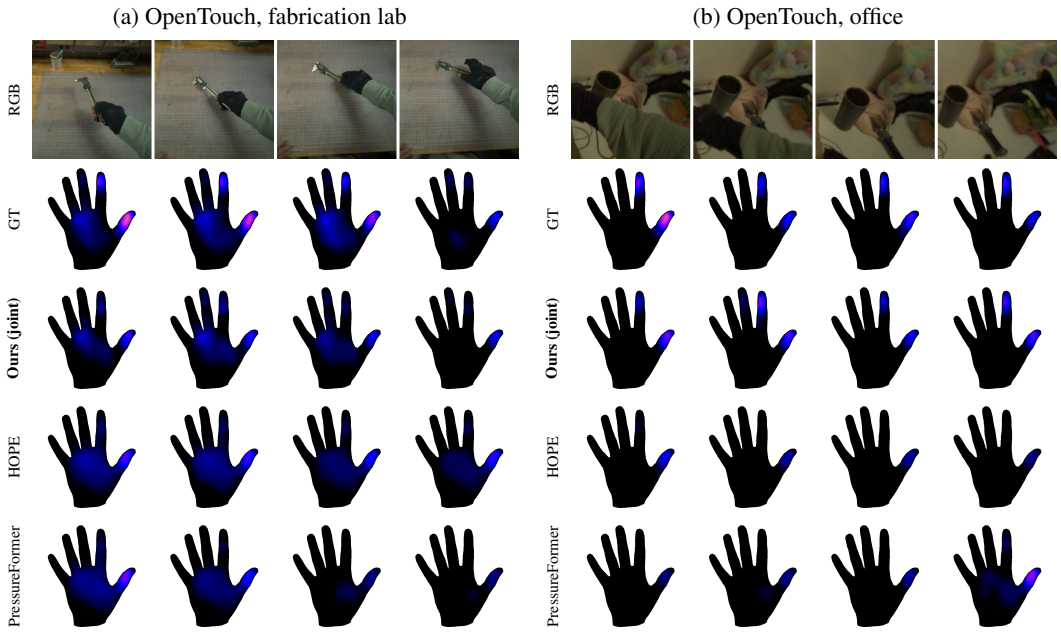


Figure 9: **Additional comparisons on OpenTouch.** Display conventions follow Figure 8.

B Additional Visualizations

Figures 8 and 9 compare tactile magnitude predictions with ground truth across three datasets.

C Architecture and Masking

C.1 Geometry and Visual Feature Paths

Hand geometry provides correspondence between image observations and a common anatomical surface. Vertex and joint coordinates are centered at the wrist and divided by the root mean squared vertex distance to the wrist in each frame. Vertex attributes additionally include surface normals, mesh visibility, and image inclusion. The anatomical positional encoding sums a vertex embedding, a region embedding, six-frequency Fourier features of the template coordinates, and a linear embedding of normalized frame index. Joints use joint identity and frame index. An additional palm frame, defined by the wrist, index metacarpophalangeal joint, and ring metacarpophalangeal joint, supplies canonical coordinates through zero-initialized affine projections. The original camera-oriented coordinates remain available. Geometry estimation is frozen throughout training.

The backbone has width $D = 768$, patch size 16, and temporal tubelets of two frames. We number its twelve blocks from 0 to 11. The initial visual field F_t sums features from blocks 5, 8, and 11, each projected to $d_p = 128$ dimensions and aligned to the input frames by a learned temporal decoder. Dense teacher targets instead use the final block's features, projected to $d = 256$ by the fixed matrix R in Equation 1. We construct R by QR decomposition of a Gaussian matrix with seed 912. The target-field $\mathrm{LayerNorm}$ has no learnable affine parameters. Each tubelet feature supplies both constituent frames before bilinear sampling.

Table 6 distinguishes the visual target path from the surface representation path. The interleaved path branches after block 8 and reruns blocks 9 through 11 with the same backbone and LoRA parameters. Thus, surface feedback changes the features read by subsequent Geometry Encoder layers without entering the visual features used as prediction targets.

Table 6: **Information flow in Touch-JEPA.** The visual and interleaved paths share backbone parameters within each encoder. The EMA copy supplies targets and the representation used for evaluation.

Path	RGB input	Surface feedback	Output and use
Student visual	Masked	None	Initial field F_t
Student surface	Masked	First three layers	States z for prediction and tactile supervision
EMA visual	Unmasked	None	Dense and patch targets
EMA surface	Unmasked	First three layers	Frozen states for downstream readouts

C.2 Global and Temporal Exchange

Each of four Geometry Encoder layers uses width 256 and four attention heads. Within-part attention operates on the 20 template parts described in the main text. Cross-part attention has width 64 and four heads, followed by a projection to 256 dimensions. The local feed-forward network has hidden width 512 and GELU activation. Its output projection and the cross-part output projection are zero-initialized.

The fixed downsampling matrix $D_0 \in \{0,1\}^{195\times 778}$ selects 195 distinct template vertices with one nonzero per row. The fixed sparse matrix $U_0 \in \mathbb{R}^{778\times 195}$ maps coarse updates back to the full surface. For layer ℓ, the persistent frame core $C_t^\ell \in \mathbb{R}^{216\times 128}$ is updated by

$$\begin{aligned} C_t^{\ell+1} &= \mathrm{Core}_\ell\big(C_t^\ell + \phi_{\mathrm{in}}^\ell\big(\mathrm{Concat}(S_t, D_0 H_t)\big)\big)\,, \\ \Delta_t^\ell &= \phi_{\mathrm{out}}^\ell(C_t^{\ell+1}), \\ S_t &\leftarrow S_t + [\Delta_t^\ell]_{\mathrm{joint}}, \qquad H_t \leftarrow H_t + U_0[\Delta_t^\ell]_{\mathrm{surf}}. \end{aligned} \tag{11}$$

Concatenation joins token rows, with joints preceding surface tokens. The core uses full four-head self-attention and a feed-forward network of hidden width 512. The affine output map ϕ_{out}^ℓ has zero-initialized weights and bias.

Initial core embeddings and the first three core attention modules use a FreiHAND-trained FastMETRO checkpoint (Cho et al., 2022). Its learned 512-to-128 projection maps the vertex and joint

embeddings to core width. We transfer the attention query, key, value, and output projections and their associated attention normalization. The fourth attention module copies the third at initialization and is subsequently trained independently. The core feed-forward networks and remaining Geometry Encoder modules are initialized separately. All training variants use the same initialization.

After each core update, a temporal block processes each anatomical track independently. Vertices and joints share its weights within a layer, while layers have separate parameters. Attention is causal along the track, and its feed-forward hidden width is 1024. This restriction does not make the video backbone causal. After the first three layers, normalized in-image vertex states are bilinearly splatted to the four neighboring patches. Contributions are divided by their accumulated weights and averaged over each two-frame tubelet. A bias-free projection and zero-initialized channel gate add the result before the next interleaved ViT block. Separate final $\mathrm{LayerNorm}$ operations yield vertex and joint states.

C.3 Mask Construction and Image Access

The interaction radius is $48 \times 384/224 \simeq 82.3$ pixels. To select t_c, we find the first and last frames containing supported vertices. Candidate bounds reserve three frames on either side and are clamped to $[6, 10]$. An integer is drawn uniformly from the resulting half-open interval when it is nonempty. Otherwise, we use the midpoint split of the observed interval, clamped to $[6, 10]$.

For anatomical masking, a clip selects a thumb and another finger with probability 0.30, the palm with probability 0.175, one finger with probability 0.35, or no region with probability 0.175. The single-finger case adds a different second finger with probability 0.5. A selection is redrawn up to four attempts if a chosen region is visible in less than 5% of its vertex-frame entries. For hand bounding-box diagonal b_t in pixels, the anatomical pixel mask uses discs of radius $r_t^{\mathrm{anat}} = \mathrm{clip}(0.05b_t + 3, 5, 17)$ around selected in-image vertices in every frame. Both masks zero RGB values after input normalization.

Anatomically hidden vertices lose image access during initialization, per-layer lifting, and local image-window attention. Associated joints also lose direct samples. The local window radius is approximately 109.7 pixels. For hidden vertices, the attention residual is explicitly zeroed even when numerical handling of an empty attention row admits image keys. Geometry and communication through other parts, the core, and temporal tracks remain available. This closes direct image reads while preserving the inference paths used to recover withheld evidence.

D Training Objectives and Joint Optimization

D.1 Dense Prediction and Patch Reconstruction

Let h_v indicate membership in an anatomically hidden region, $I_t = \mathbf{1}[t \geq t_c]$, and $j_{t,j}$ indicate a valid joint projection. Before change-based reweighting, the dense weights are

$$\begin{aligned} w^0_{t,v} &= s_{t,v}\big(h_v + (1 - h_v)[I_t + \lambda_c(k)(1 - I_t)]\big), \\ w^0_{t,j} &= j_{t,j}[I_t + \lambda_c(k)(1 - I_t)], \qquad \lambda_c(k) = 0.2 \min\left(1, \frac{k}{0.1K}\right), \end{aligned} \tag{12}$$

where k is the training step and $K = 12{,}000$ is the planned number of updates. For unhidden target vertices supported both currently and at $t_c - 1$, we multiply $w^0_{t,v}$ by their mean absolute teacher-feature change divided by the mean change over eligible entries, clipped to $[0.25, 4]$. Other weights are unchanged. Within each data lane, weighted vertex and joint errors share one normalization denominator. The dense objective averages these normalized losses over lanes with nonzero support.

The residual predictor uses $\mathrm{LayerNorm}$ followed by a 256-to-512-to-256 MLP with GELU and zero-initialized final affine layer. The patch decoder uses a learned query plus an affine embedding of normalized patch position and frame index. One four-head cross-attention block reads the same frame's 778 vertex and 21 joint states. Output normalization and affine maps produce $d_p = 128$ features. The patch target is the normalized EMA counterpart of F_t. Only interaction-masked patches in target frames contribute to Equation 6, with the same within-lane normalization followed by averaging across lanes.

D.2 Covariance Regularization

WeakSIGReg samples at most 1024 supported vertex states per device and uses an independently redrawn orthonormal sketch $Q \in \mathbb{R}^{256\times 64}$ at each step. For sampled states $Z \in \mathbb{R}^{n\times 256}$, define centered sketches $\widetilde{Z} = ZQ - \mathbf{1}\,\mathrm{mean}_{\mathrm{rows}}(ZQ)$. The regularizer is

$$\mathcal{L}_{\mathrm{wSIG}} = \left\| \frac{\widetilde{Z}^\top \widetilde{Z}}{n-1} - I_{64} \right\|_F . \tag{13}$$

The Frobenius norm is not squared, and fewer than eight supported rows contribute zero. This covariance constraint complements the temporal smoothing supplied by EMA.

D.3 Tactile Supervision and Gradient Coupling

Sensor annotations are lifted to vertices through fixed correspondences. Each covered vertex receives the value and validity of its dominant sensor channel. PressureVision DB already supplies vertex-level targets. The tactile head uses a 256-unit hidden layer, GELU, dropout 0.1, and two zero-initialized outputs. For training contact prevalence r, positive BCE entries receive weight $\mathrm{clip}((1-r)/r, 1, 10)$. Magnitude weights λ_m are 1.0, 1.0, and 0.3 for HandSense, PressureVision DB, and OpenTouch. The magnitude normalization uses the encoder-training statistics in Table 7.

The BCE and contact-conditioned log-magnitude terms update the encoder and head. For the all-valid gated L1 term, evaluating the head on $\mathrm{sg}(z)$ preserves its parameter gradients while blocking encoder gradients. Each component is normalized over its eligible entries within a clip and the sum is averaged over labeled clips. The final configuration uses no tactile pseudo-label, transition, increment, or geometry-derived contact term.

The tactile weight follows $\lambda_{\mathrm{tac}}(k) = \min(1, 0.1 + 0.9k/(0.1K))$. Let g_s and g_t denote the corresponding unweighted SSL and tactile gradients at the shared vertex states, and write $\lambda = \lambda_{\mathrm{tac}}(k)$. We apply

$$\begin{aligned} g_t^+ &= g_t - \frac{\min(0, \langle g_t, g_s\rangle)}{\max(\|g_s\|_2^2, \epsilon)} g_s, \\ \alpha &= \min\left(1, \frac{0.4\max(\|g_s\|_2, r_k)}{\max(\lambda\|g_t^+\|_2, \epsilon)}\right), \qquad g_z = g_s + \lambda\alpha g_t^+ . \end{aligned} \tag{14}$$

Here $\epsilon = 10^{-20}$ and r_k is the running mean SSL gradient norm collected at steps 100 through 599, held fixed afterwards and zero before collection. The projection removes the opposing component at z (Yu et al., 2020). The cap uses the larger of the current and reference norms, while leaving g_s unchanged. The head receives ordinary gradients of the weighted tactile loss.

EMA updates the trainable visual and Geometry Encoder parameters after optimization. The SSL predictor, patch decoder, and tactile head are outside this EMA copy. Consequently, tactile gradients can affect later visual targets through the shared adapted backbone without generating tactile pseudo-labels. For Labels only, the dense predictor is trained on detached states, so the encoder receives only tactile supervision. Its tactile weight is one from the start and no gradient projection or cap is applied. All downstream readouts use unmasked frozen EMA states and discard the training predictor and tactile head.

E Readouts and Evaluation Protocol

Frozen representations and sensor correspondence. Evaluation fixes the EMA encoder and removes input masks. The SSL predictor and the tactile head used in joint training are discarded. A deterministic hand crop with scale 1.5 produces vertex features of shape $16 \times 778 \times 256$. Fixed row-normalized operators pool these features to the sensor grid before the readout. PressureVision DB uses the identity over 778 vertices, OpenTouch maps to 169 taxels, and HandSense maps to five fingertip channels. The same operators and training clips are used for each Touch-JEPA training setting.

Readout fitting. For each dataset, we sample $1{,}500$ training clips with seed 912 and at most 400 valid entries per clip. A clip-level 20% holdout is reserved for model selection. Each feature dimension is standardized using the fitting subset mean and standard deviation, with 10^{-6} added to the denominator. The readout contains a 256-unit linear layer, GELU, dropout 0.1, and a two-output linear layer initialized to zero. The contact probability $\hat{c}$ is obtained by applying sigmoid to the contact logit. The final normalized magnitude prediction is $\hat{y} = \hat{c}\hat{m}$, where $\hat{m}$ is the nonnegative magnitude output. We minimize class-balanced binary cross-entropy plus unit-weight L1 error of the gated magnitude. The positive-class weight is the negative-to-positive ratio in the fitting entries. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, and batch size $16{,}384$ for 40 epochs, with seed 0 for head initialization. We restore the epoch with the lowest total holdout loss, without refitting on the holdout.

Magnitude normalization. Encoder supervision and readout evaluation use separate fixed training subsets of 120 and 60 clips per dataset. Each scalar is the population standard deviation of all valid measurements in its subset, pooled across clips, frames, and channels. Table 7 distinguishes these scales. The normalized target y in Equation 8 is the raw measurement divided by ν_{enc}. Readout targets are divided by ν_{eval}. Before evaluation, the final gated prediction is converted back to sensor units as $\hat{y}^{\mathrm{raw}} = \nu_{\mathrm{eval}}\hat{y}$. The reported errors of every method are normalized by the same ν_{eval}. OpenTouch contact uses a positive magnitude exceeding the training-subset tenth percentile plus 0.25 times its standard deviation. This gives thresholds of 108.54 counts during encoder supervision and 107.58 counts for readout fitting and evaluation.

Table 7: **Fixed magnitude scales and evaluation support.** Normalization uses training data only. All-valid entries are used for objective comparisons and ablations. Main baseline comparisons use common entries.

Dataset	ν_{enc}	ν_{eval}	All valid	Common
OpenTouch	434.146	430.309	6,811,376	3,328,284
PressureVision DB	5.354	5.160	3,230,307	3,142,493
HandSense	191.711	181.521	18,816	18,816

Comparison populations. Within each comparison, the evaluation mask intersects annotation validity with the availability of every method's predictions. Image-based pressure outputs restrict the main comparison to their projection support. The Touch-JEPA objective comparisons and ablations use all annotation-valid entries, including entries outside that support. This distinction explains why the same checkpoint can have different scores across these tables.

Contact metrics. For Touch-JEPA, contact metrics use $\hat{c}$ from the same readout head. AUROC and average precision pool valid test entries and use $4{,}000$-bin score histograms. F1 pools true-positive, predicted-positive, and ground-truth-positive counts. Temporal AUROC averages over clip-channel sequences with at least four valid frames and both contact classes. Within-frame AUROC averages over frames containing both classes.

Magnitude metrics. For Touch-JEPA, all magnitude metrics use the final native-unit prediction $\hat{y}^{\mathrm{raw}} = \nu_{\mathrm{eval}}\hat{c}\hat{m}$, where $\hat{c} = \sigma(\ell)$ is the contact probability and $\hat{m}$ is the nonnegative normalized magnitude from the same readout head. Compared methods are evaluated using their own final pressure predictions. For annotation-valid comparison entries $\mathcal{E}$, let $e_i = \hat{y}_i^{\mathrm{raw}} - y_i^{\mathrm{raw}}$ denote the native-unit prediction error. We compute

$$\mathrm{nMAE} = \frac{\sum_{i\in\mathcal{E}} |e_i|}{\nu_{\mathrm{eval}}|\mathcal{E}|}, \qquad \mathrm{nRMSE} = \frac{1}{\nu_{\mathrm{eval}}}\sqrt{\frac{\sum_{i\in\mathcal{E}} e_i^2}{|\mathcal{E}|}}. \tag{15}$$

Temporal Spearman correlation is computed separately for each sensor within a clip using only valid ground-truth contact frames. Eligible sequences contain at least four such frames with nonconstant target magnitude. For a fixed comparison support, eligibility does not depend on predicted contact. We correlate each method's final magnitude predictions with the targets and average across eligible sequences. Constant predictions from learned models contribute zero under our aggregation

convention. The static prior is reported as N/A because its temporal Spearman correlation is mathematically undefined. For Touch-JEPA, conditioning on ground-truth contact does not fix predicted contact confidence, so this metric evaluates the combined output rather than isolating the magnitude branch. Pooled contact-region correlation ρ_c is Spearman correlation over valid ground-truth contact entries. For PressureVision DB, volumetric IoU is the ratio of summed pointwise minima to summed pointwise maxima of nonnegative predicted and target magnitudes. Native errors retain each sensor's units.

Baselines. The main comparison matches a budget of $6{,}144{,}000$ frame presentations. PressureFormer is reimplemented with frozen HaMeR (Pavlakos et al., 2024) image features and a UV pressure decoder, using supervised PVDB vertices for its projection loss. All outputs are mapped to the same sensor grids. The static prior uses fixed training-set contact rates and mean magnitudes per sensor entry, without video.

Uncertainty. Paired bootstrap intervals resample test clips 200 times with seed 912, applying the same resample to all compared methods. Difference intervals use the 2.5th and 97.5th percentiles. They characterize test-sample variation for the single training seed and fixed fitted readouts.

F Complementarity and Component Analysis

Controlled objectives. We compare SSL, Labels only, and joint training with the encoder architecture, initialization, and sampling schedule described in Appendix A. In Labels only, tactile supervision updates the encoder and head. Dense visual prediction trains only the predictor from detached encoder states, allowing us to monitor the prediction task without an SSL gradient to the encoder. The remaining SSL terms are disabled. Joint training uses both objectives and the gradient control described in Section 3.4.

Held-out learning curves. Figure 3a evaluates dense prediction on 96 fixed test clips from each of HandSense, PressureVision DB, OpenTouch, and HRDex-human. Mask draws are fixed across checkpoints and training settings. We average the four dataset losses with equal weight and use context weight 0.2 throughout evaluation. Each checkpoint supplies its own EMA visual targets. The curves therefore describe continued optimization of each model's prediction task rather than errors against a shared fixed teacher. Figure 3b uses the corresponding sensor-dataset sample, comprising 287 clips with valid tactile supervision. It evaluates the training tactile loss in evaluation mode and averages across these clips, without the warmup multiplier λ_{tac}. Joint training has lower held-out tactile loss at all 20 checkpoints, ending at 1.074 against 1.149 for Labels only, while continuing to reduce dense loss.

Table 8: **Complementarity with frozen readouts.** All settings use $384{,}000$ clip presentations and the same supervised readout protocol. Evaluation uses all valid test entries, as defined in Appendix E. Contact and magnitude panels report every main metric. Colors indicate first and second ranks. Ties at the displayed precision share a rank.

	OpenTouch			PressureVision DB			HandSense		
Training	AUROC ↑	F1 ↑	tAUC ↑	AUROC ↑	F1 ↑	tAUC ↑	AUROC ↑	F1 ↑	tAUC ↑
SSL	0.840	0.555	0.527	0.923	0.469	0.781	0.889	0.791	0.639
Labels only	0.861	0.585	0.545	0.942	0.480	0.788	0.921	0.823	0.755
Joint	0.867	0.590	0.548	0.946	0.543	0.798	0.934	0.850	0.753
	nMAE ↓	nRMSE ↓	t-ρ ↑	nMAE ↓	nRMSE ↓	t-ρ ↑	nMAE ↓	nRMSE ↓	t-ρ ↑
SSL	0.316	0.895	0.026	0.127	0.883	0.237	0.552	0.896	0.079
Labels only	0.308	0.879	0.042	0.122	0.877	0.224	0.549	0.854	0.111
Joint	0.304	0.864	0.047	0.122	0.864	0.248	0.466	0.748	0.109

Representation-level comparison. Table 8 evaluates frozen EMA encoders after discarding the training heads. Joint training achieves higher contact AUROC and lower magnitude nRMSE than

either single objective on all three datasets. Table 9 reports the paired differences for these metrics. Their intervals exclude zero for both reference settings. The same ordering in AUROC and nRMSE holds at 192,000 and 288,000 clip presentations. These results support complementary contributions to the learned representation beyond the training heads. The benefit is metric dependent. On HandSense, joint training and Labels only have similar temporal point estimates, with tAUC values of 0.753 and 0.755 and t-ρ values of 0.109 and 0.111, respectively.

Table 9: **Paired improvements at** $384{,}000$ **clip presentations.** Evaluation uses the same annotation-valid entries as Table 8. Entries give joint minus the reference with a 95% paired clip-bootstrap interval. Positive AUROC and negative nRMSE differences favor joint. Differences are calculated before rounding.

Dataset	Reference	ΔAUROC	ΔnRMSE
OpenTouch	SSL	$+0.027$ $[0.025, 0.029]$	-0.031 $[-0.034, -0.026]$
	Labels only	$+0.005$ $[0.004, 0.007]$	-0.016 $[-0.019, -0.012]$
PressureVision DB	SSL	$+0.024$ $[0.021, 0.026]$	-0.019 $[-0.027, -0.013]$
	Labels only	$+0.005$ $[0.003, 0.006]$	-0.013 $[-0.018, -0.009]$
HandSense	SSL	$+0.045$ $[0.032, 0.063]$	-0.148 $[-0.194, -0.115]$
	Labels only	$+0.014$ $[0.004, 0.023]$	-0.106 $[-0.129, -0.075]$

Component protocol. Table 3 compares trained SSL variants at 192,000 clip presentations, corresponding to $6{,}000$ updates. It additionally includes frozen V-JEPA 2.1 as a pretrained representation baseline without further encoder adaptation. Each metric is averaged equally across the three datasets on all annotation-valid entries. The main baseline tables instead evaluate the final models after $384{,}000$ clip presentations on common entries.

G Failure Analysis

Analysis protocol. We examine challenging conditions using the test predictions and common valid entries from the main comparisons. Qualitative examples are selected to illustrate failures and do not estimate their prevalence.

Motion and limited surface visibility. The OpenTouch examples in Figure 10 show incomplete recovery of tactile responses under image motion and limited surface visibility. In the kitchen sequence, Ours underestimates the broad palm response present in the ground truth, with predictions concentrated near the fingertips. In the restaurant sequence, the predicted palm response is similarly weak, while the thumb-tip response is overestimated relative to the ground truth. These examples illustrate errors in both the spatial distribution and magnitude of predicted tactile responses.

Sparse contact events. Figure 11 illustrates weak or missing predictions for localized fingertip contacts. In the piano sequence, the brief index-fingertip response in the ground truth is substantially weaker in the prediction. In the bottle-opening sequence, Ours fails to reproduce the index-fingertip response, while PressureFormer captures part of it. These examples show that brief or localized contact can remain underrepresented in the predicted tactile maps despite joint training.

H Robotic Transfer and Manipulation

The robot studies examine two uses of the learned representation. Zero-shot transfer applies a human-trained tactile predictor to real Allegro-hand video. Action-conditioned planning instead trains a dynamics predictor on simulated robot trajectories while keeping the visual encoder frozen. The latter uses observed actions and robot state but no tactile targets for learning or planning.

H.1 Zero-Shot Tactile Estimation on Robotic Hands

Data and correspondence. We evaluate the Allegro portion of the paired human and robot recordings in HRDexDB (Lim et al., 2026). The collection contains 89 robot episodes. Available synchro-

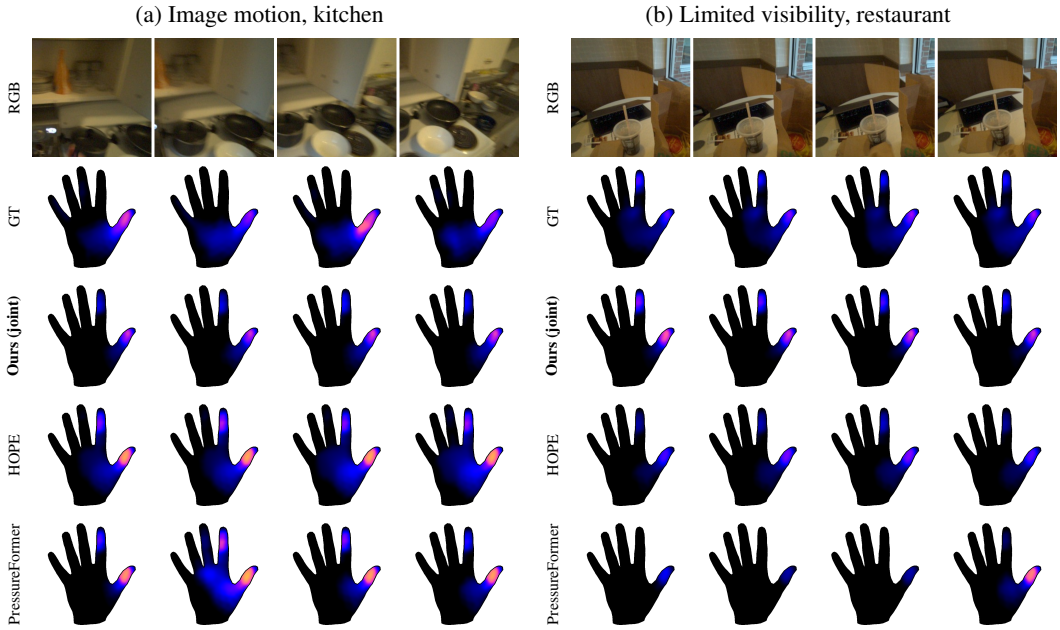


Figure 10: **Failure cases under image motion and limited visibility.** OpenTouch examples show missed or underestimated tactile responses. GT and predictions share a color scale within each clip, with brighter colors indicating greater sensor-native magnitude.

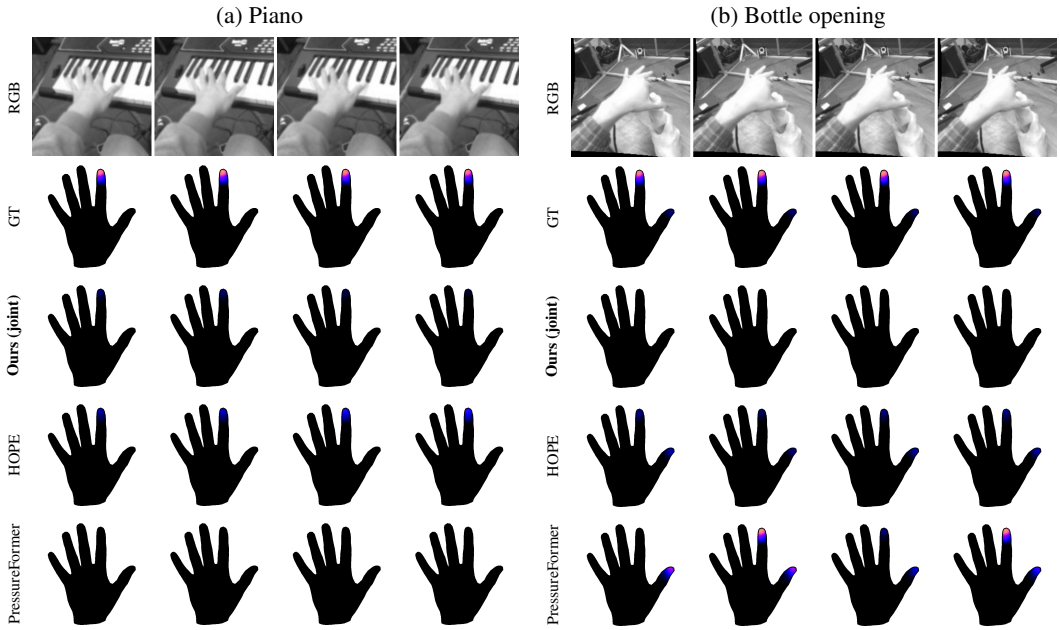


Figure 11: **Failure cases for sparse fingertip contacts.** Selected HandSense examples show weak responses during brief contact in (a) and missed fingertip contact in (b). PressureFormer captures part of the contact missed by Ours (joint) in (b). Display conventions follow Figure 10.

nized camera views provide 2,754 valid contexts covering 10,562 distinct robot frames. There are four views in 78 episodes, three in five episodes, and two in six episodes.

Measured robot joint angles drive forward kinematics. A fixed retargeting maps the index, middle, ring, and thumb fingertips to the corresponding MANO fingertips. We fit the MANO pose with 500 Adam steps at learning rate 0.05 and a pose-prior weight of 0.01. This geometric proxy provides image sampling locations without fitting to tactile data. The virtual little finger has no robot target and is excluded from scoring. In this real-video pipeline it remains in the encoder input under the pose prior. Hand crops use an expansion factor of 1.5 and are resized to 384×384 for Touch-JEPA and 224×224 for the compared supervised models.

Human-trained predictions. For Touch-JEPA, the fixed HandSense correspondence operator pools vertex features into fingertip features before standardization and readout. We retain its four rows with robot counterparts. The saved heads are fitted on HandSense TRAIN with the same fitting recipe used for human evaluation in Appendix E. No robot examples fit or adapt these heads. HOPE pools its predicted vertex field, while PressureVision and PressureVision++ pool projected class logits before decoding magnitude. For each robot frame, finite predictions from all covering contexts and camera views are averaged. All methods are scored on the same 42,207 valid frame and fingertip entries.

Targets and statistics. A robot tactile sample is valid when its nearest timestamp is within 50 ms of the video frame. Contact labels use a channel-specific threshold equal to the tenth percentile plus 0.25 standard deviations of tactile readings from the designated robot TRAIN recordings. These measurements define evaluation labels and do not update the predictors. Pooled AUROC and AP use contact scores over the common entries.

Magnitude correlation is computed within each episode and fingertip using at least eight valid samples with nonconstant targets and predictions. We average finite Spearman correlations over fingertips and then over episodes. This differs from the contact-conditioned, within-clip temporal correlation in the human benchmark. Paired bootstrap comparisons resample episodes 1,000 times with seed 140914 and report the 2.5 and 97.5 percentiles of each difference. For joint minus SSL, the intervals are $[-0.087, -0.014]$ for AUROC, $[-0.119, -0.040]$ for AP, $[-0.075, -0.008]$ for F1, and $[-0.136, -0.017]$ for magnitude correlation. These intervals support the SSL encoder choice in the planning experiment.

Additional evaluation on the Inspire hand. We additionally evaluate zero-shot tactile estimation on sequences of the five-finger Inspire hand from HRDexDB (Lim et al., 2026). Table 10 reports results on its five fingertip channels. Ours (SSL) achieves the highest scores across all four metrics among the compared methods.

Table 10: **Zero-shot transfer to the Inspire hand.** Evaluation covers five fingertip channels.

Method	AUROC ↑	AP ↑	F1 ↑	ρ ↑
PressureVision	0.494	0.145	0.218	−0.023
PressureVision++	0.479	0.137	0.213	0.002
HOPE	0.568	0.178	0.257	0.014
Ours (SSL)	0.608	0.195	0.284	0.059
Ours (joint)	0.509	0.150	0.222	−0.015

H.2 Observations and Action-Conditioned Learning

Simulated observations. We use a fixed-base KUKA iiwa arm and an Allegro hand in Isaac Gym. Physics runs at 120 Hz with two substeps. A calibrated RGB camera records 384×384 images at 4 Hz with a 55-degree field of view. Each encoder input contains the trailing 16 actual frames, spanning 3.75 seconds and ending at the current observation. Although the backbone is noncausal within this window, it receives no frames after the decision time. The input is the full camera image with ImageNet normalization.

The MANO proxy is fitted to the four fingertip pads using forward kinematics and the pose fitting procedure above. In simulation, the unsupported little finger's 114 vertices and four joints are marked invalid, leaving 664 supported vertices. At decision τ, the observation is $\mathbf{o}_\tau = (B_\tau, U_\tau, \mathbf{p}_\tau)$. Scene tokens $B_\tau \in \mathbb{R}^{36\times768}$ are obtained by 4×4 average pooling of the final backbone layer's 24×24 patch grid at its last tubelet. Surface tokens $U_\tau \in \mathbb{R}^{32\times256}$ average current-frame vertex states within 32 fixed groups of supported vertex indices. The state $\mathbf{p}_\tau \in \mathbb{R}^{35}$ concatenates 23 joint angles and the flattened 3×4 palm pose. Each feature group uses channel-wise training means and standard deviations, with a standard-deviation floor of 0.05.

Predictor and training data. The action-conditioned predictor has 24 Transformer blocks of width 1024, 16 attention heads, and feed-forward width 4096. Separate projections embed 36 scene tokens, 32 surface tokens, one state token, and one action token. Learned slot and step embeddings identify these 70 tokens per decision. Block-causal attention permits all tokens from the current and preceding decisions within an eight-step context.

Training uses 696 trajectories from 408 initial scenes, with 232 trajectories each for grasp-lift, place, and push-align. Push-align is used only for training. Trajectories and reference demonstrations use scene seeds distinct from evaluation. An eight-dimensional issued action contains three palm-translation commands, four finger-flexion commands, and thumb opposition. Action inputs record the issued commands after controller projection.

For group $r \in \{B, U, \mathbf{p}\}$, let $\mathcal{L}_r^{\mathrm{tf}}$ be mean absolute prediction error over eight transitions conditioned on observed history. Let $\mathcal{L}_r^{\mathrm{ar}}$ be the corresponding mean over a two-step autoregressive rollout. The predictor objective is

$$\mathcal{L}_{\mathrm{AC}} = \sum_{r\in\{B,U,\mathbf{p}\}} \lambda_r\big(\mathcal{L}_r^{\mathrm{tf}} + \mathcal{L}_r^{\mathrm{ar}}\big), \qquad (\lambda_B, \lambda_U, \lambda_{\mathbf{p}}) = (1, 1, 0.1). \tag{16}$$

The autoregressive rollout feeds back predicted scene features, surface features, and robot state, with gradients through both steps. Future measured state is not supplied. Encoder features are frozen targets, and object state, contact, and force are excluded from this loss. AdamW uses 1,024 updates of 256 windows, weight decay 0.04, gradient clipping at norm 1, and seed 917. The learning rate warms up for 49 updates to 4.25×10^{-4}, remains constant through update 975, and decays linearly to zero. Evaluation uses the final predictor checkpoint.

H.3 Planning and Control

Planning minimizes the scene and surface distance in Equation 10 with a one-step horizon. Reference observations at demonstration endpoints 2, 4, 14, and 18 provide successive subgoals for 2, 2, 10, and 4 decisions. Each encoder computes its own reference features using its training normalization. Predicted robot state is rolled forward by the predictor but does not enter the goal distance.

At each planning decision, the cross-entropy method samples 400 candidates for 10 refinement iterations and retains 10 elites. The search variables are three palm translations and one grip value, which is copied to the five hand commands. Initial search standard deviation is 0.7, with a floor of 0.03. The controller executes the projected final mean. Translation is limited to an L1 norm of 7.5 cm per decision and a fixed workspace. Inverse kinematics and a smooth joint ramp execute each command.

H.4 Planning and Control

Planning minimizes the scene and surface distance in Equation 10 with a one-step horizon. Reference observations at demonstration endpoints 2, 4, 14, and 18 provide successive subgoals for 2, 2, 10, and 4 decisions. Each encoder computes its own reference features using its training normalization. Predicted robot state is rolled forward by the predictor but does not enter the goal distance.

At each planning decision, the cross-entropy method samples 400 candidates for 10 refinement iterations and retains 10 elites. The search variables are three palm translations and one grip value, which is copied to the five hand commands. Initial search standard deviation is 0.7, with a floor of 0.03. The controller executes the projected final mean. Translation is limited to an L1 norm of 7.5 cm per decision and a fixed workspace. Inverse kinematics and a smooth joint ramp execute each command.

One-step energy landscape. Following the qualitative energy analysis in V-JEPA 2.1 (Mur-Labadia et al., 2026), Figure 12 visualizes the scene-feature prediction energy for a reference carry transition in the place task. The goal is the next observation in the demonstration. In this example, the sampled minimum lies 1.0 cm from the demonstrated command for Touch-JEPA-AC, compared with 2.5 cm for V-JEPA 2.1 in the horizontal command plane.

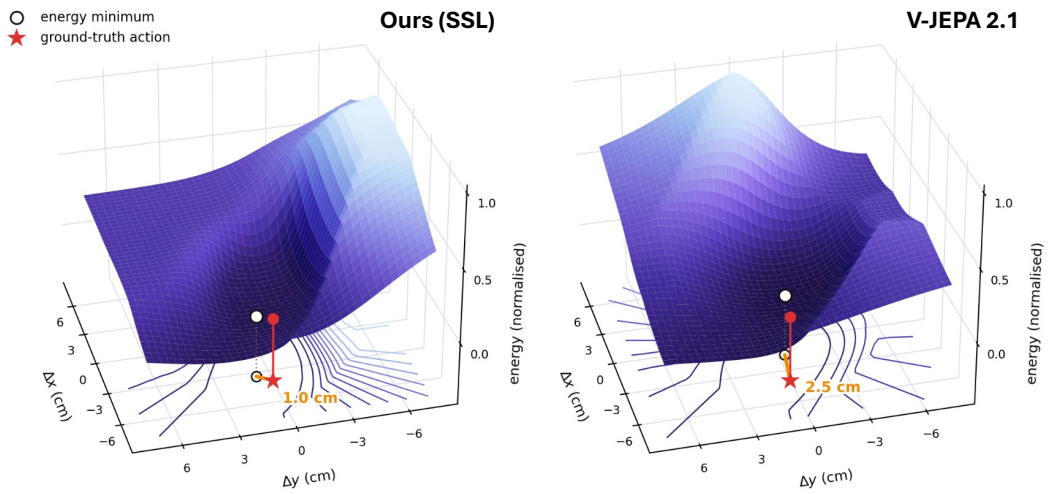


Figure 12: **One-step scene prediction energy during object transport.** Horizontal translation commands are varied on a 41×41 grid, with remaining command dimensions fixed before controller projection. Energy is the scene-feature L1 distance to the next reference observation, normalized independently to $[0, 1]$ within each panel. White circles mark sampled minima and red stars denote the demonstrated command. Annotated distances are measured in the displayed command plane.

H.5 Baselines and Evaluation Criteria

Comparison settings. The V-JEPA 2.1 baseline uses the same released visual backbone initialization, camera observations, fitted geometry, trajectories, predictor training budget, subgoal schedule, CEM settings, and controller as Touch-JEPA-AC. Its surface input samples concatenated features from backbone blocks 5, 8, and 11, followed by a fixed orthonormal projection to 768 channels and the same vertex grouping. Thus, its surface tokens have width 768, compared with Touch-JEPA's learned width 256. Predictor input and output projections accommodate this difference.

Octo is adapted as a direct policy with a frozen backbone and trainable action head and proprioceptive tokenizer. It receives two RGB frames, two 35-dimensional state vectors, and a goal image. A diffusion head predicts a four-action chunk, of which only the first action is executed. The five hand commands are averaged into one grip value. Training uses the same 696 trajectories and 1,024 updates of 256 samples. AdamW uses learning rate 3×10^{-5} after a 100-update warmup and weight decay 0.1.

Success criteria. Each task uses 50 matched scenes across methods. Grasp-lift succeeds when the minimum object-height increase during the final second is at least 10 cm and the maximum summed fingertip-pad force over the episode is at most 8 N. Place succeeds when, throughout the final second, horizontal target error is at most 3 cm, vertical error is at most 1 cm, and the object has been released. Object pose and contact measurements are used to score these conditions.

For grasp-lift, both planners succeed in 25 scenes, only Touch-JEPA-AC succeeds in 18, only V-JEPA 2.1 succeeds in two, and both fail in five. The corresponding place counts are 18, 24, five, and three. Let b count successes unique to Touch-JEPA-AC and c count successes unique to V-JEPA 2.1. The exact two-sided McNemar probability is

$$p = \min\left(1,\ 2\sum_{j=0}^{\min(b,c)} \binom{b+c}{j} 2^{-(b+c)}\right). \tag{17}$$

Table 11: **Planning success with uncertainty.** Counts use the same 50 scenes per task. Parentheses give Wilson 95% confidence intervals for success percentages.

Method	Grasp-lift	Place
Touch-JEPA-AC	43/50 (73.8, 93.0)	42/50 (71.5, 91.7)
V-JEPA 2.1	27/50 (40.4, 67.0)	23/50 (33.0, 59.6)
Octo	0/50 (0.0, 7.1)	0/50 (0.0, 7.1)

This yields unadjusted $p = 4.02 \times 10^{-4}$ and $p = 5.46 \times 10^{-4}$, respectively, consistent with the rounded values in the main text. Confidence intervals describe variation across evaluation scenes for one predictor training seed.

H.6 Qualitative Rollout Comparisons

Figure 6 illustrates Touch-JEPA-AC rollouts from the planner camera. Figures 13 and 14 additionally compare Touch-JEPA-AC and V-JEPA 2.1 on matched scenes. For each task, we show the lowest-index evaluation scene in which Touch-JEPA-AC succeeds and V-JEPA 2.1 fails. Rollout and initial-state panels are rendered from recorded robot and object states, while goal panels show the final reference camera images supplied to the planner. The accompanying curves report simulator measurements used for evaluation.

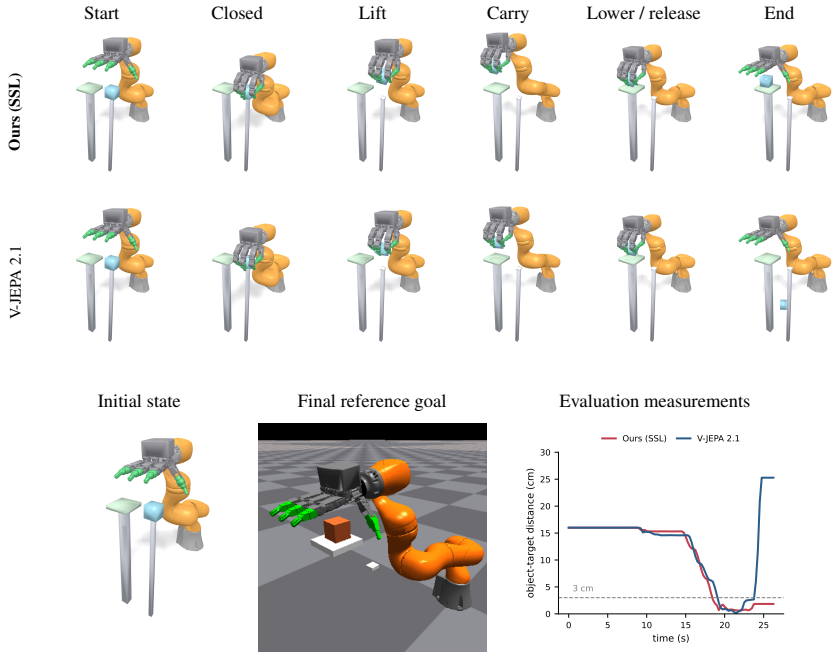


Figure 13: **Place rollout comparison.** Time progresses from left to right. Curves show planar object-target distance. Touch-JEPA-AC releases the object with a maximum planar target error of 1.8 cm over the final second, while V-JEPA 2.1 drops it beside the target.

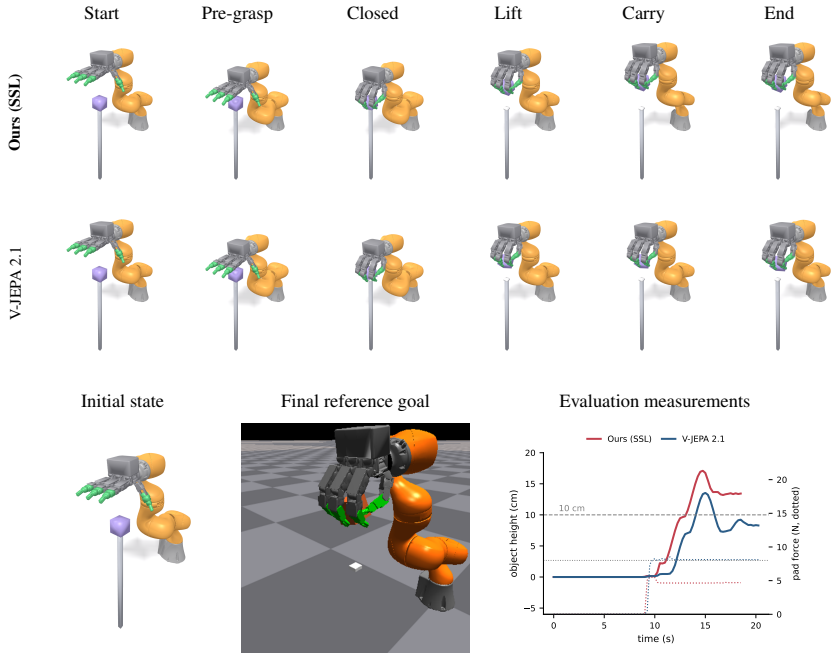


Figure 14: **Grasp-lift rollout comparison.** Time progresses from left to right. Curves show object height above its initial position and summed fingertip-pad force. Both methods grasp and lift the object. Touch-JEPA-AC maintains a minimum lift of 13 cm over the final second, whereas V-JEPA 2.1 maintains 8.2 cm and therefore fails the 10 cm height criterion.