ACTS: LEARNING VISUO-TACTILE WORLD MODELS FROM HUMAN PHYSICAL INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models for contact-rich manipulation require data that captures how contact forms, slides, and releases under varying force. Such data are slow to collect through robot teleoperation, which often lacks force feedback and is constrained by gripper geometry. We ask whether these dynamics can instead be learned from human physical interaction and transferred to a robot. We present ACTS, a visuo-tactile world model trained only on human interaction data. Operators manipulate handheld GelSight sensors while we record third-person and wrist video, tactile images, 6-DoF sensor motion, and tactile-derived normal force, yielding React, a 4.9-hour bimanual dataset across three objects. We represent actions at the tactile sensor as motion augmented by force change, providing a shared action space for human collection and robot execution. ACTS jointly predicts future scene, wrist, and tactile observations. On held-out human data, ACTS improves scene-view prediction by 0.7–2.0 dB PSNR over two visuo-tactile world models trained from scratch on React. Without fine-tuning, predicted tactile imprints on robot teleoperation still appear and disappear with recorded contact. Visual transfer is less robust: camera predictions match or exceed frame copying on two of three objects, while wrist views can blur or hallucinate structure. These results suggest that contact dynamics transfer more reliably than visual appearance from human to robot. Project website and supplementary videos: `https://acts-anon.github.io/`.

1 INTRODUCTION

Robot world models predict how candidate actions change future observations, supporting planning and policy learning (Zhu et al., 2025; Hou et al., 2026). Contact-rich manipulation is difficult to model because small changes in contact location and force can determine whether an object sticks, slides, jams, or loses contact. Vision captures scene motion but often cannot resolve the local interactions that cause it. Learning such dynamics therefore requires data spanning contact formation, loading, sliding, and release under varied geometry and force. Collecting this interaction data is the bottleneck we address.

Existing visuo-tactile world models are trained primarily on robot interaction trajectories (Higuera et al., 2026; Zheng et al., 2026; Huang et al., 2026). In parallel, robot-free interfaces such as UMI and TacUMI collect human demonstrations with handheld devices (Chi et al., 2024; Tang et al., 2026). Robot teleoperation is slow for dense contact exploration and typically provides limited force feedback, while handheld grippers remove the robot but retain a fixed contact geometry. In both cases, actions are tied to the collection device. We instead define motion and contact at the tactile sensor, where the same action representation can be used by a human or a robot.

Human operators directly manipulate handheld GelSight sensors while we record 6-DoF sensor poses together with synchronized visual and tactile observations. We represent each action by relative sensor motion augmented with a force-informed displacement along the sensor normal, derived from tactile-estimated normal force (Chen et al., 2025) (Fig. 1). This interface allows operators to vary contact location, orientation, and loading without a gripper constraining how the sensor meets the object. Because the action is defined at the sensor rather than the hand that carries it, human collection and robot execution share the same sensor-level action space.

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.

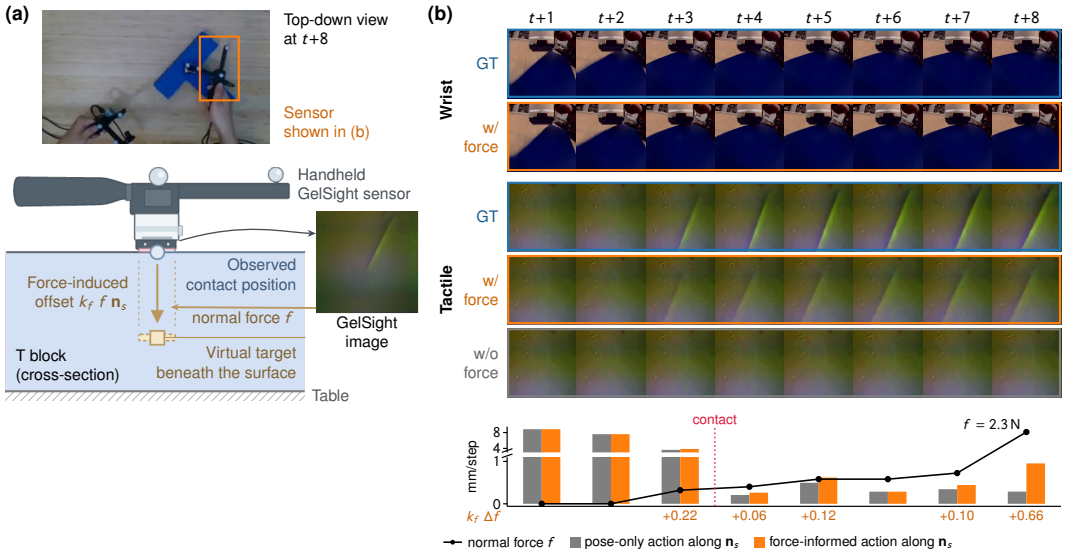


Figure 1: **Force-informed sensor actions.** (a) Top: scene view, with the sensor of (b) boxed. Estimated normal force f shifts the sensor target by $k_f f \mathbf{n}_s$ along the gel normal, so each action includes $k_f \Delta f \mathbf{n}_s$ (Eq. 3). (b) On a held-out PushT sequence, the force-conditioned model forms the tactile imprint at contact, while the pose-only model does not. Tactile crops are shown at 2× contrast.

We use these interactions to train **ACTS**, a visuo-tactile world model adapted from a pretrained video model. Given observation history and sensor actions, ACTS jointly predicts future scene, wrist, and tactile observations. Each action is provided both as force-informed motion in the local sensor frame and as sensor geometry projected into the scene view. We collect **React**, 4.9 hours of curated bimanual interaction with three objects, including multi-view video, two GelSight Mini streams (Johnson & Adelson, 2009; Yuan et al., 2017), tracked sensor poses, and tactile-derived normal force.

We train one model per object using only human data and evaluate zero-shot on robot teleoperation with the same sensing devices. ACTS outperforms visuo-tactile world-model baselines on held-out human scene prediction, while robot rollouts show that predicted tactile contact transitions can transfer without robot-specific fine-tuning. Visual transfer is less robust, revealing a remaining gap between human and robot appearance distributions.

Our contributions are:

- A sensor-level action representation that provides a common interface for human data collection and robot execution.
- **React**, a 4.9-hour human visuo-tactile dataset with tracked sensor poses and tactile-derived normal force.[1]
- **ACTS**, a visuo-tactile world model trained only on human interaction data and evaluated zero-shot on robot teleoperation.

2 Related Work

World models for robot manipulation. World models learn action-conditioned dynamics for planning, simulation, policy learning, and evaluation (Hou et al., 2026; Wang et al., 2026). Visual-foresight methods planned through predicted image trajectories (Finn & Levine, 2017; Dasari et al., 2020), and recent models build on pretrained visual representations and video generators (Zhou et al., 2025; 2024; Zhu et al., 2025; Guo et al., 2025). These models learn from robot trajectories, with actions given as end-effector motion, joint commands, or gripper states (Dasari et al., 2020; Zhu et al., 2025; Guo et al., 2025), tying the learned dynamics to the robot that collected the data.

[1] We will release the React dataset, trained models, and code upon acceptance.

Visuo-tactile world models. Tactile sensing resolves contact that vision leaves ambiguous (Shan et al., 2026), and several world models now predict tactile futures. VT-WM (Higuera et al., 2026) jointly predicts visual and tactile latents for planning, ViTacWorld (Huang et al., 2026) trains on combined real and simulated data, and OmniVTA (Zheng et al., 2026) pairs short-horizon visuo-tactile prediction with a contact-aware policy. ContactWorld (Zhang et al., 2026) compares visual and tactile representations for predictive planning, FeelWorld (Ma et al., 2026) predicts structured contact, force, and slip variables, and Dream-Tac (Lou et al., 2026) and VT-WAM (Tian et al., 2026) couple tactile prediction with action generation. All of these learn from robot trajectories or robot-action datasets. ACTS learns from human interaction instead, with actions defined at the tactile sensor so that the same representation describes a robot moving that sensor.

Visuo-tactile datasets. Visuo-tactile datasets record objects (Suresh et al., 2021; Gao et al., 2022), human touch in natural scenes (Yang et al., 2022), spatially aligned visual–tactile samples (Dou et al., 2024), in-hand manipulation (Suresh et al., 2024), and data pooled across sensors (Zhao et al., 2025). They broaden tactile coverage but generally do not provide the action supervision a world model needs. Handheld collection systems do record actions: UMI (Chi et al., 2024) introduced robot-free demonstration collection, FreeTacMan, exUMI, and TacUMI add tactile, force, and richer motion sensing (Wu et al., 2026; Xu et al., 2025; Cheng et al., 2026), and MimicTouch (Yu et al., 2025) records human tactile demonstrations. These systems target policy learning. React instead pairs visual and tactile observations with sensor-level motion and normal force, the supervision needed to train an action-conditioned world model.

3 DATASET

Human interaction data. React is collected by five operators performing bimanual interactions with a motherboard, a T-shaped block (PushT), and a rope. Each operator holds two independently movable devices, each equipped with a GelSight Mini vision-based tactile sensor (Yuan et al., 2017), an OptiTrack marker assembly for 6-DoF pose tracking, and a local camera. Three fixed cameras record the workspace (Fig. 4, Appendix A). Operators slide, pivot, push and pull the objects while varying contact location, orientation, and loading. The data cover making and breaking contact, sticking, sliding, and object motion. Operators explore these contact conditions without pursuing a prescribed task goal. After curation, the human dataset contains 4.91 hours of interaction across 187 segments from 60 source recordings.

Robot evaluation data. We collect a separate bimanual evaluation set with two YAM arms controlled through GELLO teleoperation interfaces (Wu et al., 2024). Each arm carries the same handheld device used for human collection, with its GelSight Mini, marker assembly, and wrist camera, so robot and human episodes share sensors and action construction. Teleoperators perform non-prehensile manipulation with the same three objects. We record five one-minute episodes per object, yielding 15 episodes and 15 minutes of robot interaction. The world model is trained only on human interaction data; all robot episodes are held out for evaluation.

Post-processing. We align the visual, tactile, and pose streams by timestamp to a common 30 Hz timeline and retain flags that distinguish new tactile captures from repeated frames. We remove intervals with sensor freezes or tracking failures and divide the remaining data into contiguous valid segments. Normal-force estimates derived from the tactile images are combined with the tracked sensor motion to construct force-informed actions. Appendix A gives the alignment procedure, force-estimation method, and dataset statistics.

4 METHOD

We propose ACTS (Fig. 2), an action-conditioned world model jointly predicting future scene video, wrist views, and tactile observations from observation histories and actions for bimanual manipulation. We describe observation encoding and tokenization (Sec. 4.1), then force-aware sensor actions and view-aligned actions encoding contact-related motion and scene-view sensor geometry (Sec. 4.2). We detail unified sequence construction with spatiotemporal alignment and full-window action conditioning (Sec. 4.3), alongside the training objective and inference protocol (Sec. 4.4).

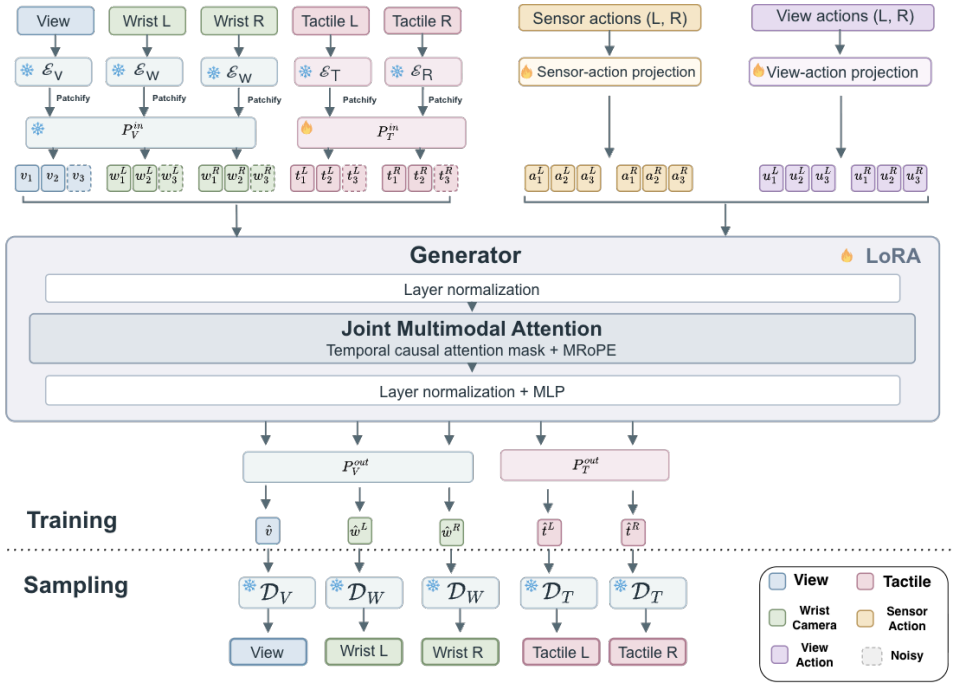


Figure 2: **Overview of ACTS.** Frozen encoders produce observation latents, whose spatial patches are mapped to tokens by input projectors P^{in}. A generator initialized from pretrained Cosmos3 weights jointly processes noisy future tokens conditioned on clean history, force-aware sensor actions, and view-aligned actions. Output projectors P^{out} map target-token features to latent velocity predictions. The scene and two wrist streams share the RGB projector pair $(P_V^{\text{in}}, P_V^{\text{out}})$, while both tactile streams share $(P_T^{\text{in}}, P_T^{\text{out}})$. Sampling updates only future latents, which frozen decoders reconstruct into synchronized multimodal predictions.

4.1 Multimodal Observation Representation

To capture manipulation from complementary perspectives, we jointly model scene video, wrist views, and tactile observations in a shared latent prediction framework. We define the observation streams as $\mathcal{M} = \{V, W^L, W^R, T^L, T^R\}$, where V denotes the top-down scene RGB view, W^L and W^R denote the left and right wrist-camera RGB views, and T^L and T^R denote the corresponding GelSight observations. For each stream $m \in \mathcal{M}$, let X_H^m and X_F^m denote its observed history and future sequence, respectively. Given synchronized histories and action conditions $\mathcal{A}$, our goal is to model

$$p_\theta(\{X_F^m\}_{m\in\mathcal{M}} \mid \{X_H^m\}_{m\in\mathcal{M}}, \mathcal{A}), \tag{1}$$

where θ denotes the model's trainable parameters.

To obtain compact representations while preserving temporal causality, we encode each observation stream as

$$Z^m = \mathcal{E}_m(X^m), \qquad m \in \mathcal{M}, \tag{2}$$

where X^m concatenates history X_H^m and future X_F^m in temporal order. The scene and two wrist-camera streams share the pretrained Wan2.2 causal VAE encoder (Wan et al., 2025), while both GelSight streams share a variant adapted to our visuo-tactile dataset.[2] All encoders are frozen, and observations are encoded offline. We partition Z^m into history latents Z_H^m and future latent blocks $Z_F^m = (z_1^m, \ldots, z_Q^m)$. We spatially patchify each latent block and linearly project the patches into a common d-dimensional token space, adding learned stream embeddings to distinguish their sources. Let $\mathbf{e}_{q,p}^m \in \mathbb{R}^d$ denote the token for patch p of future block q in stream m; history blocks are tokenized identically. The scene and two wrist-camera streams share one input projection and one output projection, while the two GelSight streams share a separate pair. The output projections map hidden states back to the corresponding latent patch spaces.

[2]Details of tactile VAE adaptation are provided in Appendix D.

4.2 Action Representation

We connect motion, contact dynamics, and scene geometry through two complementary action representations. Following IRASim (Zhu et al., 2025) and Cosmos3 (NVIDIA et al., 2026), we provide both representations for both hands over the complete window, collectively denoted by $\mathcal{A}$, as clean conditions to every predicted observation block.

Force-aware sensor actions. Inspired by DexForce (Chen et al., 2025), we augment sensor-frame translation with a force-dependent normal displacement to capture contact-loading changes beyond observed motion. For hand h at frame t, let $(R_t^h, \mathbf{p}_t^h)$ denote the sensor pose and f_t^h its estimated normal force. We define

$$\begin{aligned} \Delta R_t^h &= (R_t^h)^\top R_{t+1}^h, \\ \Delta \widetilde{\mathbf{p}}_t^h &= (R_t^h)^\top (\mathbf{p}_{t+1}^h - \mathbf{p}_t^h) + k_f \Delta f_t^h \mathbf{n}_s, \\ \mathbf{a}_{t+1}^h &= [\Delta \widetilde{\mathbf{p}}_t^h; \mathrm{Rot6D}(\Delta R_t^h)] \in \mathbb{R}^9, \end{aligned} \tag{3}$$

where $\Delta f_t^h = f_{t+1}^h - f_t^h$, $\mathbf{n}_s$ is the unit gel normal in sensor coordinates, $k_f = 0.5\,\mathrm{mm/N}$, and $\mathrm{Rot6D}$ is the 6-D rotation encoding of Zhou et al. (2019).

View-aligned actions. Inspired by Action Images (Zhen et al., 2026), we encode sensor position and orientation in the scene view using three sensor-fixed points: the gel center and offsets along the gel normal and an in-plane axis. For each sensor h and frame t, camera calibration maps these points to normalized pixel coordinates and camera-space depths, which we concatenate into $\mathbf{u}_t^h \in \mathbb{R}^9$. Depth retains geometric information lost during projection. Appendix E details the force augmentation and reference-point projection.

4.3 Multimodal Sequence Construction

Sequence construction. To jointly predict future observations from history and actions, we pack clean history tokens, action tokens, and noisy future target tokens from all streams into a single sequence. We initialize the generator from pretrained Cosmos3 (NVIDIA et al., 2026) weights to adapt its spatiotemporal representations to visuo-tactile prediction. To preserve conditioning information, we corrupt only future target latents and apply noise-level embeddings only to their tokens.

Temporal alignment. To align observations and actions by physical time, we adopt Cosmos3's multimodal rotary position embedding (MRoPE) (NVIDIA et al., 2026), which encodes time, height, and width in separate rotary dimensions. Each token is assigned a timestamp τ_i, where observation tokens use latent-block endpoints, sensor-action tokens use transition endpoints, and view-aligned action tokens use their corresponding frames. We map these timestamps to a shared temporal axis using Cosmos3's pretrained time scale, aligning streams in time while preserving spatial structure within observations.

Attention mask. To predict observation dynamics under a known action trajectory, we allow all tokens to access the complete action window while restricting attention to observation keys at the same or earlier timestamps. We implement this through an additive attention mask:

$$M_{ij} = \begin{cases} 0, & j \in \mathcal{I}_A \text{ or } \tau_j \le \tau_i, \\ -\infty, & \text{otherwise}, \end{cases} \tag{4}$$

where i, j index query and key tokens within a sample, τ_i denotes token time, and $\mathcal{I}_A$ contains sensor-action and view-action token indices. Applied in every layer, this mask preserves temporal ordering in direct observation attention while allowing same-time tokens to exchange information across streams for joint prediction. To prevent ground-truth leakage, we additionally construct each prediction prefix without later observation blocks or clean copies of the current target in Sec. 4.4.

4.4 Training and Inference

Training. To learn sequential dynamics across observation streams, we train ACTS with block-wise teacher forcing. Each future block q is predicted from clean observation history, preceding ground-truth future blocks, and the full action window, collectively denoted by $\mathcal{C}_q$. Following Cosmos3 (NVIDIA et al., 2026), we use rectified flow matching (Lipman et al., 2023; Liu et al., 2023)

with noisy targets

$$\widetilde{z}_q^m = (1-\sigma) z_q^m + \sigma \epsilon_q^m, \qquad \epsilon_q^m \sim \mathcal{N}(0, I), \tag{5}$$

where m indexes observation streams and σ is the noise level. The model jointly predicts velocities $v_q^{m,*} = \epsilon_q^m - z_q^m$ by minimizing

$$\mathcal{L} = \mathbb{E}\left[\frac{1}{Q} \sum_{q=1}^{Q} \sum_{m \in \mathcal{M}} \frac{\left\| v_\theta^m(\widetilde{Z}_q, \mathcal{C}_q, \sigma) - v_q^{m,*} \right\|_2^2}{D_z} \right], \tag{6}$$

where $\widetilde{Z}_q$ collects the noisy target streams, Q is the number of future blocks, and D_z is the number of scalar elements per stream's latent block. Only current target tokens are corrupted and supervised.

Inference. To follow the sequential prediction process used during training, we generate future latent blocks autoregressively within each prediction window. Each block is conditioned on the observed history, previously generated blocks, and the complete action sequence. For the first block, the observation context contains only the recorded history. We initialize the target latents of all streams with independent Gaussian noise and jointly transform them into clean predictions by integrating the learned velocity field, while keeping the conditioning tokens fixed. The completed block is then appended as clean latent context for predicting the next block, replacing the ground-truth context used during teacher forcing. This process proceeds entirely in latent space without intermediate decoding. After all future blocks are generated, the corresponding decoders $\{\mathcal{D}_m\}_{m \in \mathcal{M}}$ reconstruct the predicted latents into synchronized scene, wrist, and tactile observations. Appendix F provides further training and inference details.

5 Experiments

5.1 Baseline Models

We compare ACTS with two action-conditioned visuo-tactile world models, training both from scratch on React. Among recent visuo-tactile world models, only ContactWorld (Zhang et al., 2026) and Dream-Tac (Lou et al., 2026) release training code; VT-WM (Higuera et al., 2026), OmniVTA (Zheng et al., 2026), ViTacWorld (Huang et al., 2026), FeelWorld (Ma et al., 2026), and VT-WAM (Tian et al., 2026) had not released model code at the time of writing. **ContactWorld** is a decoder-free JEPA-style (Assran et al., 2023) model, which we train with its official code, adapted to our observation and action spaces. **VT-WM-style** is our reimplementation of VT-WM, preserving its autoregressive latent predictor and action conditioning while adapting its interfaces to our data. We intended to compare with Dream-Tac but could not adapt it to React: it generates actions jointly with future observations instead of conditioning on given actions, and its released code expects a robot camera layout and proprioception that React lacks. Implementation details and deviations appear in Appendices I and G.

5.2 Evaluation Metrics

We compare ACTS with baseline models on future-frame prediction using held-out human recordings and zero-shot transfer to robot episodes.

Video Generation Metrics. Given eight observed frames and recorded actions, each model predicts the next eight frames (0.53 s at 15 fps). We evaluate only predicted frames, separately for the scene view, two wrist cameras, and two GelSight streams. We report PSNR for pixel-level fidelity, SSIM (Wang et al., 2004) for structural similarity, and LPIPS (Zhang et al., 2018) for perceptual distance. Higher PSNR and SSIM and lower LPIPS indicate better results. For both human and robot data, reference frames are decoded from ground-truth latents using the frozen RGB and tactile VAEs. All comparisons use identical test windows and seeds. Appendix B provides further details.

Robot Experiments. We apply models trained on human data to held-out robot episodes (Sec. 3) without fine-tuning. Robot recordings are normalized to match the human data's tactile color and illumination and wrist-camera tone. Actions are constructed from OptiTrack sensor poses and GelSight-estimated force following Sec. 4.2, as for the human data, rather than from robot joint readings. Per-step translations from forward kinematics differ from the OptiTrack ones by a median

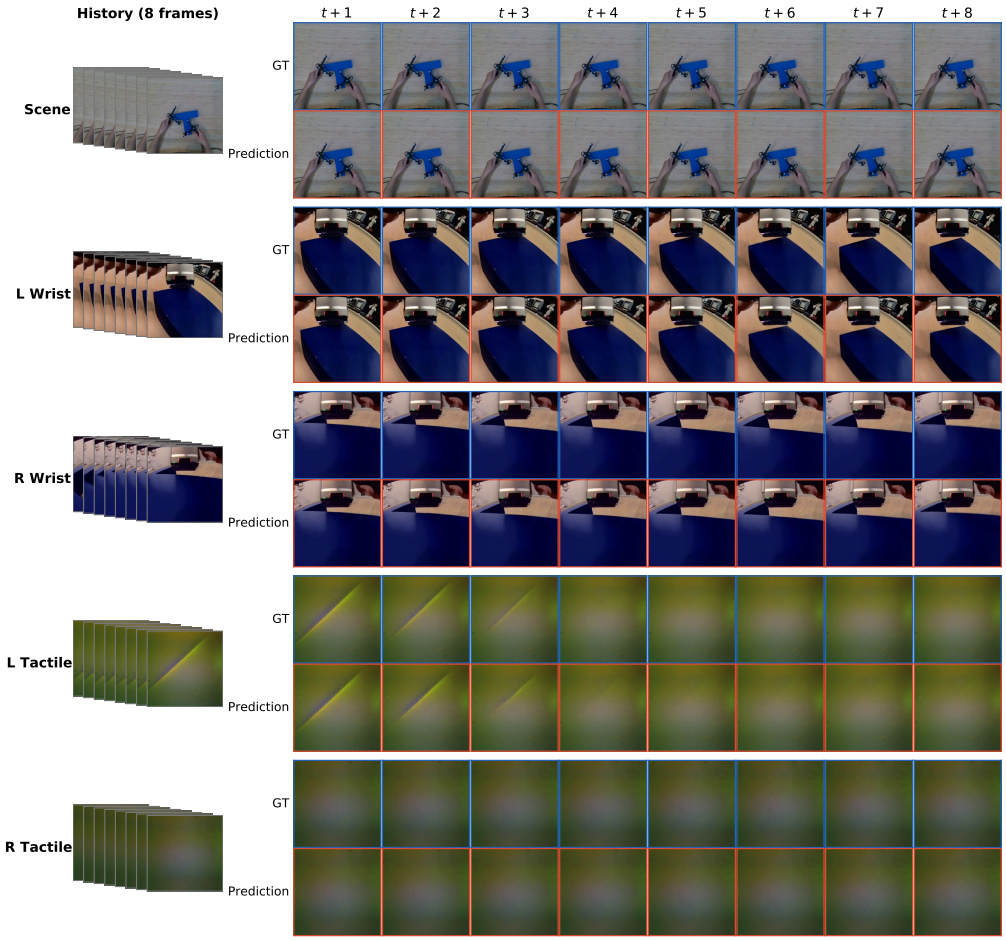


Figure 3: **Qualitative results** on a held-out PushT window. Given eight observed frames (left) and the recorded sensor actions, ACTS predicts the next eight frames of the scene view, both wrist views, and both GelSight streams. Ground truth in blue boxes, predictions in orange boxes. Videos and results on all tasks: `https://acts-anon.github.io/`.

of 0.33 mm, about half the median per-step motion, so OptiTrack keeps robot actions consistent with the human training data (Appendix B). We use the same eight-frame observation and prediction protocol on windows in which contact is made or broken, and report the same per-stream metrics.

6 Results

6.1 Implementation Details

We train ACTS separately per task with the same architecture and training recipe, initializing from Cosmos3-Nano (NVIDIA et al., 2026). Observation VAEs and backbone weights remain frozen, while 45.3M parameters are trained through LoRA (Hu et al., 2021) and modality-specific interfaces. All models are trained for 20 epochs on four NVIDIA H100 GPUs, with training, validation, and test splits separated by source episode. Appendix F provides implementation and dataset details.

6.2 Qualitative Examples

Fig. 3 shows a held-out PushT window in which the operator lifts the left sensor off the block; more rollouts are on the project page. The recorded left imprint fades over $t{+}1$–$t{+}3$, and the predicted imprint fades on the same frames. The block stays in place, and the scene and wrist predictions remain consistent with the recording. Predicted imprints are slightly smoother than recorded ones, which we attribute to the tactile VAE.

Table 1: Short-horizon prediction quality over 8 frames (0.53 s). *Copy* repeats the last observed frame across the prediction window. ACTS consistently improves the camera streams, while high temporal persistence makes full-frame tactile metrics less discriminative.

		Video		Wrist (L)		Wrist (R)		Tactile (L)		Tactile (R)	
Task	Metric	Copy	Ours	Copy	Ours	Copy	Ours	Copy	Ours	Copy	Ours
Rope	PSNR↑	25.10	**28.92**	17.66	**22.30**	19.66	**23.70**	44.74	**46.08**	46.46	**46.51**
	SSIM↑	0.848	**0.907**	0.657	**0.769**	0.704	**0.796**	0.986	**0.988**	0.990	**0.990**
	LPIPS↓	0.115	**0.045**	0.261	**0.131**	0.234	**0.129**	0.033	**0.021**	0.027	**0.020**
PushT	PSNR↑	25.21	**29.69**	19.32	**22.86**	19.20	**22.76**	46.24	**46.42**	**48.73**	48.37
	SSIM↑	0.861	**0.925**	0.762	**0.834**	0.766	**0.844**	0.986	**0.988**	0.991	**0.991**
	LPIPS↓	0.103	**0.038**	0.176	**0.097**	0.194	**0.111**	0.027	**0.022**	0.016	**0.015**
Mother-board	PSNR↑	26.62	**27.74**	16.13	**17.68**	19.73	**20.77**	**41.94**	41.65	**46.67**	45.70
	SSIM↑	0.865	**0.880**	0.494	**0.525**	0.651	**0.686**	**0.979**	0.977	**0.988**	0.988
	LPIPS↓	0.065	**0.046**	0.307	**0.258**	0.188	**0.162**	0.043	**0.041**	0.031	**0.029**

Table 2: Scene-view prediction on held-out human data over 8 frames (0.53 s). Both baselines are trained from scratch on React. Best results per object are shown in bold.

Task	Model	PSNR↑	SSIM↑	LPIPS↓
Rope	ContactWorld (2026)	26.03	0.846	0.164
	VT-WM (2026)	28.05	0.8878	0.1118
	Ours	**28.92**	**0.907**	**0.045**
PushT	ContactWorld (2026)	26.35	0.858	0.139
	VT-WM (2026)	27.69	0.8916	0.1047
	Ours	**29.69**	**0.925**	**0.038**
Motherboard	ContactWorld (2026)	25.85	0.835	0.120
	VT-WM (2026)	27.01	0.8692	0.0837
	Ours	**27.74**	**0.880**	**0.046**

6.3 Quantitative Results

Tab. 1 compares ACTS with a persistence baseline that repeats the last observed frame. ACTS improves all camera streams by 1.0–4.6 dB PSNR, indicating that it captures scene motion and wrist-camera ego-motion beyond temporal persistence. On the scene view, ACTS also outperforms ContactWorld and VT-WM-style on all three objects in PSNR, SSIM, and LPIPS (Tab. 2).

Tactile scores require more careful interpretation. GelSight observations change only locally during contact, so a copied frame already achieves high PSNR and SSIM over the short prediction horizon. ACTS shows little consistent gain in full-frame tactile PSNR, although LPIPS improves on all tactile streams. These metrics are also computed against VAE-reconstructed ground truth, and therefore measure prediction error in the decoder image space rather than fidelity to the raw tactile image. They capture appearance similarity but not the correctness of contact dynamics, so we treat them as a limited diagnostic and assess contact evolution in the rollout videos. Appendix H ablates the force term in the sensor action (Eq. 3): it leaves the camera streams unchanged and moves tactile PSNR by at most 0.6 dB, yet it decides whether an imprint forms when the sensor reaches the block (Fig. 1b).

6.4 Zero-shot Transfer to Robot Teleoperation

Tab. 3 applies the same models, without fine-tuning, to held-out robot teleoperation, scored as in Tab. 1 against VAE-reconstructed ground truth. Robot motion is slower than human manipulation, so repeating the last frame is a stronger baseline than on human data. Transfer varies by object. On PushT, ACTS improves all three camera streams in PSNR and SSIM, by up to 2.1 dB PSNR on the scene view. On Rope, the results are mixed: scene PSNR rises by 0.7 dB while right-wrist PSNR drops by 0.4 dB. On Motherboard, ACTS falls below frame copying on every camera stream. The motherboard has the most complex geometry of the three objects, with connectors, slots, and

Table 3: Zero-shot short-horizon prediction on robot teleoperation (8 frames, 0.53 s), scored against VAE-reconstructed ground truth. *Copy* repeats the last observed frame; slow robot motion makes it a strong baseline.

		Video		Wrist (L)		Wrist (R)		Tactile (L)		Tactile (R)	
Task	Metric	Copy	Ours	Copy	Ours	Copy	Ours	Copy	Ours	Copy	Ours
Rope	PSNR↑	31.57	**32.31**	27.29	**27.88**	**27.70**	27.28	**42.74**	42.56	**44.73**	44.65
	SSIM↑	**0.944**	0.942	0.827	**0.845**	0.838	**0.856**	0.980	**0.980**	0.986	**0.986**
	LPIPS↓	**0.021**	0.022	0.055	**0.055**	**0.054**	0.057	0.043	**0.029**	0.040	**0.025**
PushT	PSNR↑	27.75	**29.85**	25.84	**26.71**	23.96	**25.62**	**45.66**	44.15	**45.49**	44.25
	SSIM↑	0.915	**0.924**	0.850	**0.863**	0.827	**0.853**	0.980	**0.982**	0.984	**0.984**
	LPIPS↓	0.035	**0.030**	**0.044**	0.048	0.064	**0.053**	0.031	**0.024**	0.040	**0.032**
Mother-board	PSNR↑	**32.46**	30.63	**26.72**	25.71	**25.88**	24.92	**46.79**	44.65	44.39	**44.60**
	SSIM↑	**0.941**	0.922	**0.819**	0.753	**0.817**	0.811	**0.990**	0.987	0.986	**0.987**
	LPIPS↓	**0.015**	0.024	**0.050**	0.073	**0.052**	0.059	**0.016**	0.023	0.035	**0.025**

heat sinks, so a small misalignment in the predicted view displaces many fine edges at once. The robot wrist cameras also record these episodes darker than the human recordings, which hides the fine detail the model relies on in the human data. Across objects, the wrist views degrade most: predictions are blurrier than on human data and sometimes contain object structure absent from the recording. On the tactile streams, frame copying scores higher PSNR on five of the six streams, by up to 2.1 dB, while ACTS scores lower LPIPS on five of the six.

Because full-frame tactile metrics do not show whether contact is predicted correctly, we also inspect rollouts. Fig. 5 (Appendix C) shows a held-out robot PushT window in which the sensor breaks contact with the block. The predicted left GelSight imprint fades within one frame of the recorded one, and the scene prediction keeps the block in place. The left wrist view shows the failure described above: the recorded view barely changes, while the predicted block face grows and shifts from $t{+}4$ onward. Across the robot rollouts we inspected, including those on the project page, predicted imprints appear and disappear as contact is made and broken in the recording, even where the wrist predictions degrade.

7 Limitations

Force estimation. The estimator predicts only normal force, which governs making and breaking contact but not the shear that governs sliding. Its error of 0.318 N (6.4%) is measured on CNC calibration data (Appendix A.3); the human recordings have no force ground truth, so its error during manipulation is unmeasured.

Evaluation metrics. All streams are scored against VAE-reconstructed ground truth, so errors shared by the model and the VAE go undetected. On GelSight images, full-frame metrics are dominated by static gel and do not show whether contact is made or broken at the right time, which we assess only qualitatively.

Motion capture. Sensor poses come from OptiTrack for both human collection and robot evaluation, which confines React to one instrumented lab. Forward kinematics is not a substitute: its per-step translations differ from OptiTrack by a median of 0.33 mm, about half the median per-step motion and equal to the force term for a 0.66 N change (Appendix B). Both robot collection and deployment therefore need external tracking, so contact-rich tactile data with accurate sensor-level actions remains difficult to scale.

8 Conclusion

We presented ACTS, a visuo-tactile world model trained entirely on human physical interaction, with actions defined at the tactile sensor as motion augmented with force change. It outperforms two visuo-tactile baselines on held-out human data, and on zero-shot robot teleoperation its tactile imprints track contact transitions while wrist views degrade. Improving visual transfer and using ACTS for robot planning are future work.

Ethics statement

React records five human operators who are members of the research team and gave informed consent to the recording and release of the data. The third-person cameras capture the operators' hands and forearms, and no frame captures an operator's face. The dataset contains no personal information beyond these recordings. The objects are inanimate, and the robot experiments involve no human subjects. We see no application of this work that raises concerns beyond those common to robot manipulation research.

Reproducibility statement

Appendix A describes data collection, stream alignment, curation, and normal-force estimation, with dataset statistics in Tab. 4. Appendix F gives the model architecture, training hyperparameters, data splits, and compute, and Appendix D the tactile VAE adaptation. Baseline configurations and every deviation from the published models are listed in Appendices I and G. Evaluation protocol and metrics are defined in Sec. 5.2, and all comparisons use identical test windows and seeds. We will release the React dataset, the trained models, and the training and evaluation code upon acceptance.

Use of generative AI

In this work, we used generative AI tools (Claude, Anthropic) to provide feedback on experimental design, to help interpret results, and to help build the automatic pipeline that cleans and reformats raw recordings into the dataset. We did not use generative AI tools to generate synthetic data, develop theoretical models or conceptual frameworks, propose or refine hypotheses, implement methods, or translate; formulating or proving mathematical claims and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, draft and edit parts of the paper, and identify and format references. We have reviewed all AI-assisted work. Experimental changes suggested by the AI tool, such as additional baselines and contact-timing metrics, were treated as proposals, and the authors decided which to pursue. The curation pipeline applies fixed rules, removing intervals with camera freezes or invalid pose tracking, and keeps the source-frame index of every aligned sample so that its output can be audited against the raw recordings (Appendix A). Every description of a qualitative result was checked against the corresponding figure by the authors, and every explanation of a quantitative result reflects the authors' own analysis. Bibliography entries added with AI assistance were retrieved from their DOIs and checked against CrossRef, and no reference was written from the model's memory. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Claire Chen, Zhongchun Yu, Hojung Choi, Mark Cutkosky, and Jeannette Bohg. Dexforce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation, 2025. URL https://arxiv.org/abs/2501.10356.

Tailai Cheng, Kejia Chen, Lingyun Chen, Liding Zhang, Yue Zhang, Yao Ling, Mahdi Hamad, Zhenshan Bing, Fan Wu, Karan Sharma, and Alois Knoll. TacUMI: A multi-modal universal manipulation interface for contact-rich tasks. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-scale multi-robot learning.

In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 885–897. PMLR, 2020.

Yiming Dou, Fengyu Yang, Yi Liu, Antonio Loquercio, and Andrew Owens. Tactile-augmented radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26529–26539, 2024.

Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XToAemis1h.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017. doi: 10.1109/ICRA.2017.7989324.

Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. ObjectFolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10598–10608, 2022.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas J. Guibas. Alltracker: Efficient dense point tracking at high resolution, 2025. URL https://arxiv.org/abs/2506.07310.

Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, and Mustafa Mukadam. Tactile beyond pixels: Multisensory touch representations for robot manipulation. *arXiv preprint arXiv:2506.14754*, 2025.

Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models, 2026. URL https://arxiv.org/abs/2602.06001.

Bohan Hou, Gen Li, Jindou Jia, Tuo An, Xinying Guo, Sicong Leng, Haoran Geng, Yanjie Ze, Tatsuya Harada, Philip Torr, et al. World model for robot learning: A comprehensive survey. *arXiv preprint arXiv:2605.00080*, 2026.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Hung-Jui Huang, Michael Kaess, and Wenzhen Yuan. NormalFlow: Fast, robust, and accurate contact-based object 6DoF pose tracking with vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 10(1):452–459, 2025a. doi: 10.1109/LRA.2024.3505815.

Hung-Jui Huang, Mohammad Amin Mirzaee, Michael Kaess, and Wenzhen Yuan. GelSLAM: A real-time, high-fidelity, and robust 3d tactile slam system, 2025b. URL https://arxiv.org/abs/2508.15990.

Yunao Huang, Shiyu Sang, Haotao Lu, Suting Ni, Shijie Wu, Ziyang Guo, Ye Shi, and Jingya Wang. ViTacWorld: Scaling visuo-tactile world models for contact-rich robot manipulation. *arXiv preprint arXiv:2607.22530*, July 2026. doi: 10.48550/arXiv.2607.22530. URL https://arxiv.org/abs/2607.22530.

Micah K. Johnson and Edward H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1070–1077. IEEE, 2009. doi: 10.1109/CVPR.2009.5206534.

Rohan Kota, Kaival Shah, J. Edward Colgate, and Gregory Reardon. 3D Cal: An open-source software library for depth reconstruction on vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 2026. doi: 10.1109/LRA.2026.3673994.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105. Curran Associates, Inc., 2012.

Mike Lambeta, Tingfan Wu, Ali Sengul, Victoria Rose Most, Nolan Black, Kevin Sawyer, Romeo Mercado, Haozhi Qi, Alexander Sohn, Byron Taylor, Norb Tydingco, Gregg Kammerer, Dave Stroud, Jake Khatha, Kurt Jenkins, Kyle Most, Neal Stein, Ricardo Chavira, Thomas Craven-Bartle, Eric Sanchez, Yitian Ding, Jitendra Malik, and Roberto Calandra. Digitizing touch with an artificial multimodal fingertip. *arXiv preprint arXiv:2411.02479*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

Yunfan Lou, Yifan Ye, Yankai Fu, Jun Cen, Xiaowei Chi, Yaoxu Lyu, Peidong Jia, Sirui Han, Zhihe Lu, and Shanghang Zhang. Dream-Tac: A unified tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2606.08737*, 2026.

Wenxuan Ma, Chaofan Zhang, Chao Xue, Yinghao Cai, Guocai Yao, Shaowei Cui, and Shuo Wang. FeelWorld: Visuo-tactile world model for hierarchical contact prediction and planning, 2026. URL `https://arxiv.org/abs/2607.24267`.

NVIDIA, :, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, Aarti Basant, Mukesh Beladiya, Mohammad Qazim Bhat, Zaid Pervaiz Bhat, Dan Blick, Vanni Brighella, Han Cai, Tiffany Cai, Eric Cameracci, Jiaxin Cao, Yulong Cao, Mark Carlson, Carlos Casanova, Ting-Yun Chang, Yan Chang, Yu-Wei Chao, Prithvijit Chattopadhyay, Roshan Chaudhari, Chieh-Yun Chen, Junyu Chen, Ke Chen, Qizhi Chen, Wenkai Chen, Xiaotong Chen, Yu Chen, An-Chieh Cheng, Click Cheng, Xiu Chia, Jeana Choi, Chaeyeon Chung, Wenyan Cong, Yin Cui, Magdalena Dadela, Nalin Dadhich, Wenliang Dai, Joyjit Daw, Alperen Degirmenci, Rodrigo Vieira Del Monte, Robert Denomme, Sameer Dharur, Marco Di Lucca, Ke Ding, Wenhao Ding, Yifan Ding, Yuzhu Dong, Nicole Drumheller, Yilun Du, Aigul Dzhumamuratova, Aleksandr Efitorov, Hamid Eghbalzadeh, Naomi Eigbe, Imad El Hanafi, Hassan Eslami, Benedikt Falk, Jiaojiao Fan, Jim Fan, Amol Fasale, Sergiy Fefilatyev, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Vikram Fugro, Prashant Gaikwad, TJ Galda, Katelyn Gao, Yihuai Gao, Wenhang Ge, Sreyan Ghosh, Arushi Goel, Vivek Goel, Akash Gokul, Rama Govindaraju, Jinwei Gu, Miguel Guerrero, Elfie Guo, Aryaman Gupta, Siddharth Gururani, Hugo Hadfield, Song Han, Ankur Handa, Zekun Hao, Mohammad Harrim, Ali Hassani, Nathan Hayes-Roth, Yufan He, Chris Helvig, Cyrus Hogg, Madison Huang, Michael Huang, Sophia Huang, Yufan Huang, Jacob Huffman, DeLesley Hutchins, Suneel Indupuru, Boris Ivanovic, Arihant Jain, Joel Jang, Ryan Ji, Yanan Jian, Dongfu Jiang, Jingyi Jin, Atharva Joshi, Nikhilesh Joshi, Pranjali Joshi, Andy Ju, Jaehun Jung, Weiwei Kang, Scott Kassekert, Jan Kautz, Ashna Khetan, Julia Kiczka, Slawek Kierat, Gwanghyun Kim, Kuno Kim, Sunny Kim, Kezhi Kong, Xin Kong, Zhifeng Kong, Tomasz Kornuta, Egor Krivov, Hui Kuang, Saurav Kumar, Chia-Wen Kuo, George Kurian, Wojciech Kutak, JF Lafleche, Himangshu Lahkar, Omar Laymoun, Jayjun Lee, Sanggil Lee, Gabriele Leone, Boyi Li, Freya Li, Jiajun Li, Jinfeng Li, Ling Li, Pengcheng Li, Shangru Li, Tingle Li, Xiaolong Li, Xuan Li, Zhaoshuo Li, Zhiqi Li, Hao Liang, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Ming-Yu Liu, Sifei Liu, Zihan Liu, Hai Loc Lu, Xiangyu Lu, Alice Luo, Ruipu Luo, Wenjie Luo, Jiangran Lyu, Martin Ding Ma, Nic Ma, Qianli Ma, Dawid Majchrowski, Louis Marcoux, Miguel Martin, Qing Miao, Ashkan Mirzaei, Shreyas Misra, Kaichun Mo, Durra Mohsin, Hyejin Moon, Pawel

Morkisz, Saeid Motiian, Kirill Motkov, Seungjun Nah, Yashraj Narang, Deepak Narayanan, Thabang Ngazimbi, Julian Ouyang, Shubham Pachori, David Page, Yatian Pang, Sehwi Park, Mahesh Patekar, Mostofa Patwary, Marco Pavone, Trung Pham, Wei Ping, Soha Pouya, Shrimai Prabhumoye, Varun Praveen, Delin Qu, Hesam Rabeti, Morteza Ramezanali, Marilyn Reeb, Xuanchi Ren, Kristen Rumley, Wojciech Rymer, Jun Saito, Yeongho Seol, John Shao, Piyush Shekdar, Tianwei Shen, Humphrey Shi, Min Shi, Stella Shi, Kevin Shih, Mohammad Shoeybi, Mateusz Sieniawski, Shuran Song, Alexander Sotelo, Amir Sotoodeh, Sunil Srinivasa, Vignesh Srinivasakumar, Bartosz Stefaniak, Rahul Heinrich Steiger, Shangkun Sun, Jiaxiang Tang, Shitao Tang, Yangyang Tang, Yue Tang, Tolou Tavakkoli, Kayley Ting, Krzysztof Tomala, Wei-Cheng Tseng, Jibin Varghese, Sergei Vasilev, Thomas Volk, Raju Wagwani, Roger Waleffe, Andrew Z. Wang, Boxiang Wang, Haoxiang Wang, Qiao Wang, Shihao Wang, Shijie Wang, Ting-Chun Wang, Yan Wang, Yu Wang, Rohit Watve, David Wehr, Fangyin Wei, Xinshuo Weng, Jay Zhangjie Wu, Kedi Wu, Hongchi Xia, Summer Xiao, Tianjun Xiao, Kevin Xie, Daguang Xu, Jiashu Xu, Mengyao Xu, Ruqing Xu, Xingqian Xu, Yao Xu, Dinghao Yang, Dong Yang, Hans Yang, Xiaodong Yang, Xuning Yang, Yichu Yang, Yurong You, Zhiding Yu, Hao Yuan, Simon Yuen, Xiaohui Zeng, Pengcuo Zeren, Cindy Zha, Haotian Zhang, Jenny Zhang, Jing Zhang, Liangkai Zhang, Paris Zhang, Shun Zhang, Xuanmeng Zhang, Zhizheng Zhang, Ann Zhao, Yilin Zhao, Yuliya Zhautouskaya, Charles Zhou, Fengzhe Zhou, Shilin Zhu, Yuke Zhu, Dima Zhylko, and Artur Zolkowski. Cosmos 3: Omnimodal world models for physical ai, 2026. URL https://arxiv.org/abs/2606.02800.

Tim Schneider, Guillaume Duret, Cristiana de Farias, Roberto Calandra, Liming Chen, and Jan Peters. Tactile MNIST: Benchmarking active tactile perception, 2025. URL https://arxiv.org/abs/2506.06361.

Amir-Hossein Shahidzadeh, Gabriele Caddeo, Koushik Alapati, Lorenzo Natale, Cornelia Fermüller, and Yiannis Aloimonos. FeelAnyForce: Estimating contact force feedback from tactile sensation for vision-based tactile sensors, 2024. URL https://arxiv.org/abs/2410.02048.

Shilin Shan, Chuhao Zhou, Ruize Wang, Xinyan Chen, Xiangyu Chen, Xinyu Zhou, Boyu Ma, Iris Yuxuan Hu, Jingliang Li, Celeste Yuxuan Hu, Geng Li, Guohao Chen, Tianrui Zhu, Zhe Li, Yanjie Ze, Haoran Geng, Zhiyang Dou, Jianxin Bi, Yuejiang Liu, Jianshu Zhou, Jiachen Li, Paul Liang, Tatsuya Harada, Robert Katzschmann, Harold Soh, Na Li, Edward Johns, Danica Kragic, Jan Peters, Wojciech Matusik, Masayoshi Tomizuka, Jitendra Malik, and Jianfei Yang. Learning physical interaction: A survey of tactile- and force-aware robot learning, 2026.

Sudharshan Suresh, Zilin Si, Joshua G. Mangelson, Wenzhen Yuan, and Michael Kaess. Efficient shape mapping through dense touch and vision. *arXiv preprint arXiv:2109.09884*, 2021.

Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joseph Ortiz, and Mustafa Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. *Science Robotics*, 9(96):adl0628, 2024.

Tutian Tang, Xingyu Ji, Wanli Xing, Ce Hao, Wenqiang Xu, Lin Shao, Cewu Lu, Qiaojun Yu, Jiangmiao Pang, and Kaifeng Zhang. Towards human-like manipulation through rl-augmented teleoperation and mixture-of-dexterous-experts vla. *arXiv preprint arXiv:2603.08122*, 2026.

Shuai Tian, Yupeng Zheng, Yuhang Zheng, Songen Gu, Yujie Zang, Yuxing Qin, Weize Li, Haoran Li, Wenchao Ding, and Dongbin Zhao. VT-WAM: Visual-tactile world action model for contact-rich manipulation, 2026. URL https://arxiv.org/abs/2607.02503.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen

Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Fangyuan Wang, Ziyuan Wang, Guorui Pei, Mengshi Zhang, Canxi Liang, Jun Hu, Zhongxuan Li, Jinsong Wu, Ning Han, Zeqing Zhang, Jiaming Qi, Hongmin Wu, Shiyao Zhang, Pai Zheng, Jia Pan, David Navarro-Alarcon, Sichao Liu, and Peng Zhou. World models for robotic manipulation: A survey. *SmartBot*, pp. e70053, 2026. doi: 10.1002/smb2.70053.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Longyan Wu, Checheng Yu, Jieji Ren, Li Chen, Ran Huang, Guoying Gu, and Hongyang Li. FreeTacMan: Robot-free visuo-tactile data collection system for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12156–12163. IEEE, 2024.

Yue Xu, Litao Wei, Pengyu An, Qingyu Zhang, and Yong-Lu Li. exUMI: Extensible robot teaching system with action-aware task-agnostic tactile representation. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2536–2554. PMLR, 2025.

Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. In *Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. MimicTouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4844–4865. PMLR, 2025.

Wenzhen Yuan, Siyuan Dong, and Edward Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. doi: 10.3390/s17122762.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhiyuan Zhang, Pokuang Zhou, Kaidi Zhang, Adeesh Desai, Temitope Amosa, Davood Soleymanzadeh, Jiuzhou Lei, Minghui Zheng, and Yu She. Contactworld: What matters in vision-tactile world models for contact-rich manipulation. *arXiv preprint arXiv:2606.13877*, 2026.

Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3766–3779. PMLR, 2025.

Haoyu Zhen, Zixian Gao, Qiao Sun, Yilin Zhao, Yuncong Yang, Yilun Du, Pengsheng Guo, Tsun-Hsuan Wang, Yi-Ling Qiao, and Chuang Gan. Action images: End-to-end policy learning via multiview video generation, 2026. URL `https://arxiv.org/abs/2604.06168`.

Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, et al. Omnivta: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

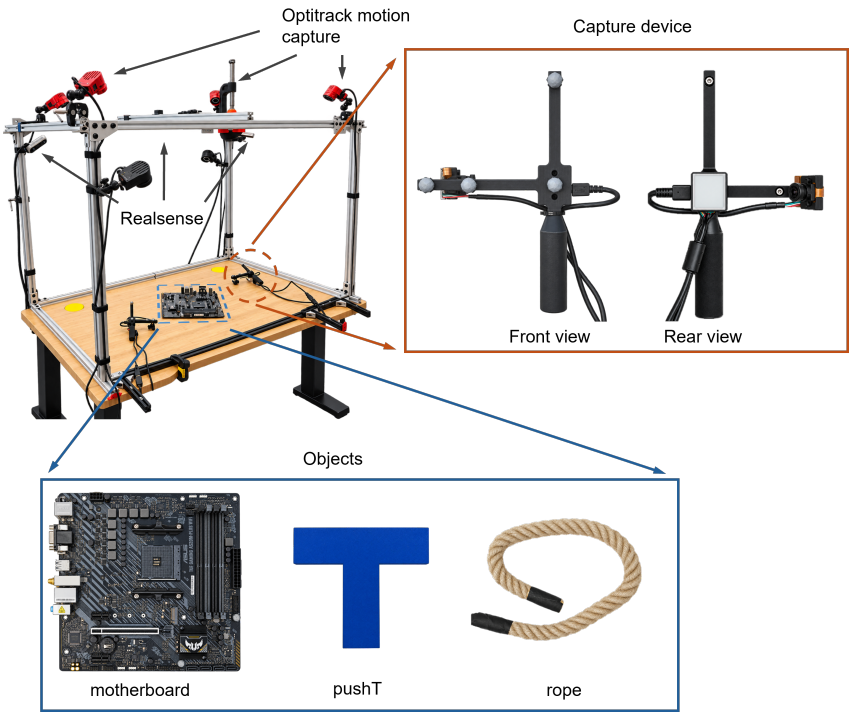


Figure 4: **Human data collection setup.** Left: the workspace and three fixed cameras. Top right: the two handheld devices and their sensing hardware. Bottom: the motherboard, PushT block, and rope used for collection.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 2024.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2406.14540.

A Dataset and Force-Estimation Details

A.1 Dataset Composition

The curated dataset contains 529,839 aligned time steps from 60 source recordings. Removing intervals with invalid sensor or tracking data leaves 187 variable-length segments, totaling 4.91 hours at the nominal 30 Hz alignment rate. Multiple segments may come from the same source recording and therefore do not represent independent collection sessions. Table 4 gives the distribution across objects.

A.2 Temporal Alignment and Curation

Visual streams are recorded at a nominal 30 Hz and OptiTrack poses at 100 Hz. We align all modalities by timestamp to a common 30 Hz timeline. Because the tactile cameras operate at a

Table 4: Dataset statistics after curation. Duration is computed from the number of aligned time steps at 30 Hz and counts each bimanual interaction once.

Object	Recordings	Segments	Duration (min)
Motherboard	12	25	93.9
PushT	30	92	96.4
Rope	18	70	104.1
Total	60	187	294.4

lower frame rate, adjacent time steps may reuse the same tactile image. For each sensor, a freshness flag and source-frame index distinguish new captures from repeated observations.

We remove intervals containing camera freezes or invalid pose tracking and divide each recording into contiguous valid segments. Force is estimated for each new tactile capture and held constant on aligned rows that reuse the same image. We retain the source-frame indices to audit this alignment.

A.3 Normal-Force Estimation

We calibrate a sensor-specific normal-force estimator for the GelSight Mini (Yuan et al., 2017) sensors used in our collection setup. For each recording, we construct a reference image from low-contact observations and compute features describing the contact area, surface deformation, and RGB change relative to this reference.

We standardize these features and map them to normal force using support vector regression with a radial basis function kernel (RBF-SVR). The kernel provides a nonlinear mapping in which tactile observations with similar feature vectors receive similar force predictions. We train the regressor on separate CNC calibration data collected with known normal loads over 0.07–15.00 N. A reference-dependent noise gate suppresses responses near the no-contact state.

Position-grouped cross-validation on 4,709 CNC calibration samples gives a mean absolute error of 0.318 N (95% CI: 0.300–0.338 N) and a mean absolute percentage error of 6.40%. The percentage error is computed per sample relative to its measured force.

This calibration is specific to our sensors. Optical tactile images vary with the sensor unit, gel, illumination, camera response, and gel condition, so we do not assume that the mapping transfers unchanged to another sensor. The human interaction recordings also lack synchronized force ground truth. We use the estimates as approximate loading signals for action construction, not as measured force annotations.

B Evaluation Details

Prediction Protocol. Each test window contains eight observed frames followed by eight target frames, corresponding to a prediction horizon of approximately 0.53 s at 15 fps. Models receive the observed frames and recorded action sequence. We score only the predicted frames, separately for the scene camera, left and right wrist cameras, and left and right GelSight sensors (Yuan et al., 2017). All checkpoints and ablations are evaluated on identical test windows with identical random seeds.

Reference Frames. For both human recordings and robot episodes, reference frames are obtained by decoding ground-truth latents with the frozen RGB and tactile VAEs. These scores therefore measure prediction fidelity relative to VAE reconstructions rather than raw recordings.

Visuotactile Prediction Metrics. We compute PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018)independently for each stream. For a predicted window $\widehat{\mathbf{Y}}$ and its reference $\mathbf{Y}$, PSNR is

$$\mathrm{PSNR} = 10 \log_{10} \frac{R^2}{\mathrm{MSE}(\widehat{\mathbf{Y}}, \mathbf{Y})}, \tag{7}$$

where R is the peak value of the image intensity range and the mean squared error is computed over all pixels, channels, and predicted frames in the window. PSNR measures pixel-level fidelity and is

sensitive to spatial alignment and contact imprints, making it our primary metric for the low-texture tactile streams.

SSIM (Wang et al., 2004) compares local luminance, contrast, and structure. We compute it per predicted frame and average over the window. LPIPS (Zhang et al., 2018) measures perceptual distance using pretrained AlexNet (Krizhevsky et al., 2012) features and is likewise averaged over predicted frames. Higher PSNR and SSIM and lower LPIPS indicate better fidelity.

Zero-Shot Robot Evaluation. Models trained on human recordings are applied to held-out robot episodes (Sec. 3) without fine-tuning. Before inference, robot observations are mapped to the human recording conventions for tactile color and illumination and wrist-camera tone. Action representations are constructed from OptiTrack sensor poses and GelSight-estimated force using the procedure in Sec. 4.2. We use OptiTrack rather than the robot's joint readings so that human and robot actions are computed by the same pipeline from the same tracking system. The two sources disagree at the scale of a single step. Per-step translations computed from forward kinematics differ from those computed from OptiTrack by a median of 0.33 mm (90th percentile 0.99 mm), while the sensor moves a median of 0.60 mm per step (90th percentile 3.06 mm). Grouped by speed, the difference is 20–40% of the per-step motion, and it exceeds the motion when the robot is nearly still. Per-step rotations differ by a median of 0.1–0.2°. Likely sources are the calibration between the robot base and the OptiTrack frame, joint backlash, and arm compliance under contact load, none of which forward kinematics observes. Actions built from forward kinematics would therefore depart from the human training data by a large fraction of each step, whereas OptiTrack measures the sensor directly in both settings. We select evaluation windows in which contact is made or broken and apply the same eight-frame observation and eight-frame prediction protocol. All metrics are computed per stream against the VAE-reconstructed reference frames described above.

C Qualitative Robot Rollouts

Fig. 5 shows a zero-shot rollout on a held-out robot episode from the evaluation set of Sec. 3, produced by the model trained only on human data and discussed in Sec. 6.4. Rollouts on all three objects are on the project page.

D Tactile Encoder

We adapt the pretrained causal 3D VAE of Wan2.2-TI2V-5B (Wan et al., 2025) to tactile video with LoRA, keeping all base weights frozen. Training uses 6,533 left and right gel-sensor videos (64 held out for validation) from visual-tactile dataset including GelSLAM (Huang et al., 2025b), FeelAnyForce (Shahidzadeh et al., 2024), Tactile MNIST (Schneider et al., 2025), FoTa (Zhao et al., 2025), TacQuad (Feng et al., 2025), 3D Cal (Kota et al., 2026), and TactileTracking (Huang et al., 2025a), covering 26 manipulation tasks. Each sample is a 17-frame clip resampled from 30 to 6 fps and center-cropped to 2564×256. Rank-32 LoRA (Hu et al., 2021) adapters (α = 32) go on all 72 convolutional layers of the encoder and decoder, 30.6M trainable parameters (4.2% of the VAE). The objective is an L1 reconstruction loss weighted by optical-flow magnitude, so regions of the gel that deform get more weight. The flow is AllTracker (Harley et al., 2025) reference flow from frame 0 to frame t. Two more terms are added: a spatial-gradient (edge) L1 loss (weight 1.0) and LPIPS (weight 0.1). We train for 4,000 optimizer steps with AdamW (Loshchilov & Hutter, 2019), learning rate 1e-4 with 100 warmup steps, an effective batch size of 4 and bf16, on a single H100. Validation PSNR is 40.2 dB.

E Action Representation Details

Sensor-frame motion and force. For each hand $h \in \{L, R\}$, the sensor pose comprises an orientation $R_t^h \in \mathrm{SO}(3)$ and position $\mathbf{p}_t^h \in \mathbb{R}^3$ in a common reference frame. The estimated normal force f_t^h is obtained using the procedure in Appendix A.3.

Sensor pose captures motion but does not fully describe contact loading. Following the force-informed formulation of Chen et al. (2025), we interpret normal force as a virtual displacement

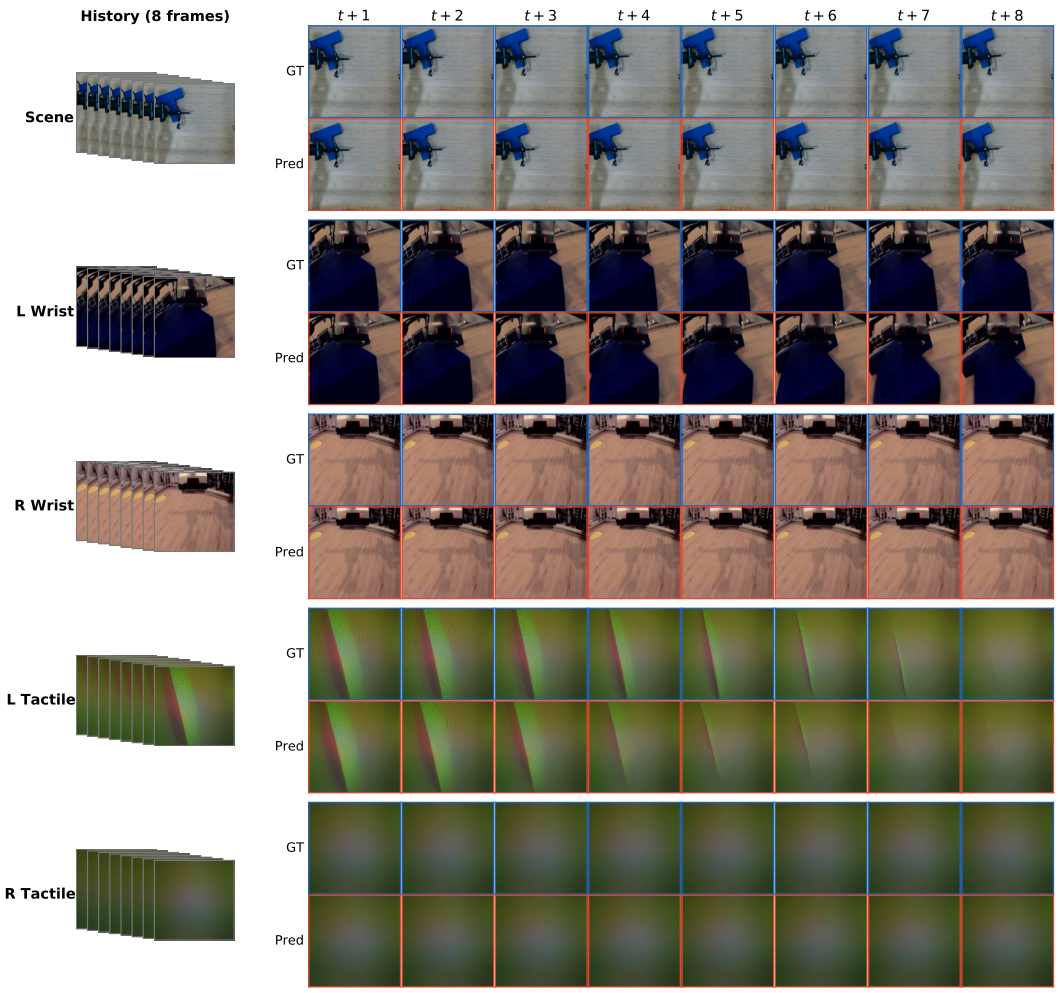


Figure 5: **Zero-shot rollout on robot teleoperation.** A held-out robot PushT window in which the sensor breaks contact with the block, predicted without fine-tuning by the model trained on human data. Given eight observed frames (left) and the robot's sensor actions, ACTS predicts the next eight frames of the scene view, both wrist views, and both GelSight streams. Ground truth in blue boxes, predictions in orange boxes. The scene and left tactile predictions follow the recording as the sensor lifts off. In the left wrist view, the recorded frames barely change, while the predicted block face grows and shifts from $t{+}4$ onward. More robot rollouts: `https://acts-anon.github.io/`.

along the sensor's local pressing direction:

$$\widetilde{\mathbf{p}}_t^h = \mathbf{p}_t^h + k_f f_t^h R_t^h \mathbf{n}_s, \tag{8}$$

where $\mathbf{n}_s$ is the unit gel normal in sensor coordinates and $k_f = 0.5\,\mathrm{mm/N}$ is an assumed controller compliance. The orientation remains unchanged. As illustrated in Fig. 1a, the offset represents a virtual controller displacement, not a measurement of gel compression.

To construct actions, we express relative motion in the sensor coordinate system at frame t and augment its translation with the change in force-induced displacement:

$$\begin{aligned} \Delta R_t^h &= (R_t^h)^\top R_{t+1}^h, \\ \Delta \mathbf{p}_t^h &= (R_t^h)^\top (\mathbf{p}_{t+1}^h - \mathbf{p}_t^h), \\ \Delta \widetilde{\mathbf{p}}_t^h &= \Delta \mathbf{p}_t^h + k_f \Delta f_t^h \mathbf{n}_s, \end{aligned} \tag{9}$$

where $\Delta f_t^h = f_{t+1}^h - f_t^h$. The force increment is applied directly along the local normal, rather than obtained by differencing the virtual positions in Eq. 8. Concatenating the augmented translation

Table 5: Dataset splits and training sizes. Splits are defined by complete source episodes. Window counts correspond to examples used for training and evaluation.

Task	Episodes			Windows			Steps/epoch
	Train	Val	Test	Train	Val	Test	
PushT	56	4	8	55,199	4,439	5,756	6,900
Rope	53	8	9	77,188	6,363	8,468	9,648
Motherboard	21	2	5	71,386	5,411	16,699	8,923

with the six-dimensional rotation representation of Zhou et al. (2019) gives

$$\mathbf{a}_{t+1}^h = [\Delta\widetilde{\mathbf{p}}_t^h; \mathrm{Rot6D}(\Delta R_t^h)] \in \mathbb{R}^9. \tag{10}$$

The index $t+1$ identifies the transition endpoint.

Reference-point projection. We define three fixed points in sensor coordinates:

$$\mathbf{o}_1 = \mathbf{o}_c, \qquad \mathbf{o}_2 = \mathbf{o}_c + \ell_n \mathbf{n}_s, \qquad \mathbf{o}_3 = \mathbf{o}_c + \ell_p \mathbf{e}_s, \tag{11}$$

where $\mathbf{o}_c$ is the gel center, $\mathbf{e}_s$ is a unit in-plane axis, and ℓ_n and ℓ_p are fixed offset distances. These points jointly encode sensor position and orientation.

Their coordinates in the common reference frame are $\mathbf{q}_{t,j}^h = R_t^h \mathbf{o}_j + \mathbf{p}_t^h$. Using the calibrated transformation $(R_C, \mathbf{t}_C)$ from this frame to the scene camera, we obtain

$$\mathbf{q}_{t,j}^{C,h} = R_C \mathbf{q}_{t,j}^h + \mathbf{t}_C, \qquad z_{t,j}^h = [\mathbf{q}_{t,j}^{C,h}]_3. \tag{12}$$

We project each point using the scene-camera calibration and normalize its pixel coordinates to obtain $(\bar{x}_{t,j}^h, \bar{y}_{t,j}^h)$. Concatenating these coordinates and camera-space depths for all three points produces $\mathbf{u}_t^h \in \mathbb{R}^9$. Unlike $\mathbf{a}_{t+1}^h$, which describes a transition, $\mathbf{u}_t^h$ describes sensor geometry at frame t.

Action construction and normalization. Sensor and view actions each have nine dimensions per hand. We use $n_s = (0, -1, 0)^\top$ and $k_f = 5 \times 10^{-4}\,\mathrm{m/N}$. Rot6D concatenates the first two columns of the relative rotation matrix. View conditions contain $(x/W, y/H, d)$ for each of the three reference points, with depth d in meters. Sensor and view representations are normalized separately, pooling both hands to estimate training-set percentiles:

$$\widetilde{b}_j = 2\frac{b_j - q_{0.01,j}}{q_{0.99,j} - q_{0.01,j}} - 1. \tag{13}$$

F Implementation and Training Details

F.1 Data Preparation

Dataset splits. We train separate models for PushT, Rope, and Motherboard using the same architecture and training recipe. Each dataset is partitioned by complete source episodes, preventing windows from the same episode from appearing in different splits. Table 5 reports episode counts, window counts, and optimization steps per epoch. The PushT training set excludes windows with invalid action transitions. For Motherboard, two test episodes are in-distribution, while three were recorded in a separate session with different hardware and force calibration and are treated as out-of-distribution. Wrist observations are available in every episode, and wrist dropout is disabled.

Observation preprocessing. We sample all five observation streams at 15 fps and resize and center-crop frames to 256×256. Each window contains eight history frames and eight future frames. The scene and two wrist streams use the pretrained Wan2.2-TI2V-5B VAE (Wan et al., 2025), while both tactile streams use its GelSight-specific (Yuan et al., 2017) LoRA-adapted variant. All VAEs remain frozen, and observation latents are encoded offline and cached. The VAEs produce 48-channel latents with temporal and spatial compression factors of four and sixteen. We prepend a copy of the initial frame before encoding and remove the resulting anchor latent, obtaining $48 \times 4 \times 16 \times 16$ latents per stream. The four latent blocks comprise two history blocks and two future blocks. Spatial patches of size 2×2 yield an 8×8 token grid per block, with 192 input features per token.

F.2 Model Initialization and Adaptation

Backbone. We initialize the generator branch of Cosmos3-Nano (NVIDIA et al., 2026) from the official checkpoint. It contains 36 transformer layers with hidden dimension 4,096. The understanding branch is not used. We retain the pretrained generator, RGB input and output projections, final normalization layers, and noise-level embedder. The native action projection is replaced by our action interfaces, initialized from its pretrained weights. Each task starts independently from the same checkpoint with fresh adapters and interfaces, without warm-starting from another task's trained model.

LoRA adapters. The base generator remains frozen. We apply LoRA (Hu et al., 2021) with rank 16 and scaling parameter $\alpha = 32$ to the query, key, value, output, gate, up, and down projections in every transformer layer. These $36 \times 7 = 252$ linear layers introduce 43.6M trainable parameters. The LoRA matrices A and B are initialized with Kaiming-uniform values and zeros, respectively, yielding zero initial adapter updates.

Modality interfaces. The scene and wrist streams share the frozen RGB input and output projections. Both tactile streams share a trainable projection pair initialized from the corresponding RGB projections. Sensor and view actions use separate nine-dimensional input projections, each shared by both hands. These projections are initialized from the first nine input channels of the UMI-domain slice of the pretrained action-conditioning projection. All input projections map to the same hidden dimension.

Stream embeddings and parameter count. A learned 9×4096 embedding distinguishes five observation streams and four action streams. The embeddings are initialized from a truncated normal distribution, with action-stream rows additionally offset by the pretrained action-modality embedding. Together, the LoRA (Hu et al., 2021) adapters, tactile and action projections, and stream embeddings contain 45.3M trainable parameters, approximately 0.30% of the 15.2B backbone. The RGB projections and noise-level embedder remain frozen.

F.3 Sequence Packing and Temporal Alignment

Token layout. We concatenate streams into contiguous slices, ordering observation tokens by latent block, patch row, and patch column. A complete window contains $5 \times 4 \times 64 = 1{,}280$ observation tokens, $2 \times 15 = 30$ sensor-action tokens, and $2 \times 16 = 32$ view-action tokens, totaling 1,342 tokens. Each token retains its stream identity, frame index, spatial coordinates, and target indicator. Positional coordinates and attention masks are derived from this metadata rather than storage offsets.

Temporal and spatial coordinates. All streams use a shared 15 fps clock. Observation blocks receive the endpoint indices $3, 7, 11, 15$ of their four-frame intervals. Sensor-action tokens receive transition endpoints $1, \ldots, 15$, and view-action tokens receive frame indices $0, \ldots, 15$. We convert frame index k to the pretrained temporal scale:

$$\tau(k) = r_{\mathrm{C}} \frac{k}{r}, \qquad r = 15, \quad r_{\mathrm{C}} = 6. \tag{14}$$

Observation tokens use patch row and column indices as spatial coordinates. Both spatial coordinates are zero for action tokens. Frame indices retain their original values across prediction prefixes within a window, so extending the prefix does not change existing token coordinates.

F.4 Training Procedure

Teacher-forcing prefixes. For each future block q, we construct a separate prediction prefix with clean context

$$\mathcal{C}_q = (Z_H, Z_{F,<q}, \mathcal{A}), \tag{15}$$

where $Z_H = \{Z_H^m\}_{m \in \mathcal{M}}$ contains observation histories, $Z_{F,<q} = \{(z_1^m, \ldots, z_{q-1}^m)\}_{m \in \mathcal{M}}$ contains preceding ground-truth future blocks, and $\mathcal{A}$ contains both action representations for both hands over the complete window. The current target appears only in noisy form. Later observation blocks and clean copies of the current target are excluded to prevent ground-truth leakage. We apply the attention mask in Sec. 4.3, retaining access to the full action trajectory.

Each window produces two teacher-forced passes. The first contains two history blocks and the first future block from each stream, while the second additionally includes the preceding future block as clean context. With all 62 action tokens retained, these prefixes contain 1,022 and 1,342 tokens, respectively. Each pass contains 320 current target tokens. Their losses contribute equally to one optimizer update.

Noise sampling. We sample one noise level σ per window and share it across both target blocks and all observation streams. Following the pretrained generator's Waver rule with shift parameter 3.0, we sample

$$\xi \sim \mathcal{U}(0,1), \tag{16}$$

$$\eta = 1 - \xi - 1.29\left[\cos^2\left(\frac{\pi\xi}{2}\right) - 1 + \xi\right], \tag{17}$$

$$\sigma = \frac{3\eta}{1+2\eta}. \tag{18}$$

Gaussian noise ϵ_q^m is sampled independently for each block and stream, with identity covariance. History, preceding ground-truth blocks, and actions remain clean. Noise-level embeddings are applied only to current target observation tokens.

Flow-matching supervision. The interpolation in Eq. 5 defines the target velocity $v_q^{m,*} = \epsilon_q^m - z_q^m$. In Eq. 6, each stream's squared error is normalized by its latent-block size D_z. Stream losses are summed with equal weight and averaged over future blocks. We apply no additional noise-dependent loss weighting, and clean context tokens receive no prediction loss. The expectation is over training windows and sampled noise.

Optimization. We use AdamW (Loshchilov & Hutter, 2019) with learning rate 5×10^{-5}, $(\beta_1,\beta_2) = (0.9, 0.95)$, and weight decay 0.01. The learning rate warms up for 100 steps and remains constant thereafter. Gradients are clipped to a global norm of 1.0. Training uses four NVIDIA H100 GPUs with two windows per GPU, giving a global batch size of eight. Trainable parameters are maintained in FP32 with BF16 autocast, and activation checkpointing is enabled. Each model is trained for 20 epochs, with a checkpoint saved after every epoch.

F.5 Inference and Rollout

Within-window generation. Inference follows the training prefix construction, replacing preceding ground-truth future blocks with generated latents. For each target block, all streams are initialized with independent Gaussian noise and jointly denoised using UniPC with 35 steps, without classifier-free guidance. The sampler integrates the learned velocity field from noise to clean latents, corresponding to $\sigma = 1$ to $\sigma = 0$. Only current target latents are updated. Observed history, previously generated blocks, and the complete action window remain fixed. The first generated block serves directly as clean latent context for the second, without intermediate decoding or re-encoding. After restoring the VAE anchor, the corresponding decoders $\{\mathcal{D}_m\}_{m\in\mathcal{M}}$ reconstruct the eight predicted frames per stream.

Across-window rollout. To extend generation beyond one window, we use the eight predicted frames as the next observation history. For each stream, we prepend a copy of its first predicted frame and re-encode the sequence, obtaining a new anchor and two history latent blocks. We advance the action window by eight frames and reset the temporal origin for the next prediction window. Thus, blocks within a window share latent context directly, while successive windows pass context through decoded and re-encoded observations.

G VT-WM-style Baseline

Predictor. A 12-layer transformer, hidden dimension 384, 6 heads, MLP ratio 4; 116.6 M parameters. Inputs are the cached Wan2.2 latents our model uses: $48\times4\times16\times16$ per stream, five streams, 2×2 patches giving 64 tokens per latent block. Attention is factorized into spatial and temporal passes with axial RoPE. The three camera streams share an input projection and output head; the two tactile streams share a second pair. Training minimises L_1 on the next latent block, teacher-forced and sampled losses weighted equally, with the horizon-4 rollout generated without gradients.

Action conditioning. After every self-attention block, camera and tactile tokens cross-attend to action tokens, which carry sinusoidal positional embeddings. Actions are the force-informed sensor deltas our model receives, 36 dimensions per frame. One latent transition spans five frames; each receives its five raw transitions concatenated (180 dimensions) rather than averaged, since averaging shares raw transitions 3, 7 and 11 between adjacent latent transitions and leaks a later action into an earlier prediction. Latent block t is conditioned on the action producing block $t{+}1$, the block it is supervised against.

Training. AdamW, peak learning rate 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, gradient clipping at 1.0, 500 warmup steps into cosine decay. Batch size 8 with accumulation 4, bfloat16, one A100-80GB, 10 epochs. Early stopping is off so every task gets the same budget.

Deviations. (i) The cross-attention residual is gated by a zero-initialised scalar; the backbone relies on adaLN-zero identity at initialisation, which an ungated sub-layer breaks. (ii) Context is eight frames at 15 fps (0.53 s) rather than nine at 6 fps (1.5 s), inherited from our encoder. (iii) Tactile tokens come from our GelSight tokenizer rather than Sparsh-X (Higuera et al., 2025) on Digit 360 (Lambeta et al., 2024), which needs audio, IMU and pressure channels this dataset lacks. (iv) Three camera views rather than one, matching our model's inputs. (v) 116.6 M parameters for 20 epochs on one GPU, against 173 M for 80k updates on 32 A100s.

Validation. Before training we check identity at initialisation, action reachability, temporal causality and gate-gradient flow. During training we track *action gain*: rollout L_1 under a held control action divided by the same under the recorded action. A value near 1 means the model ignores its conditioning, which validation loss cannot detect. The control is $\mathrm{Rot6D}(I)$ with zero translation, not a zero vector: an all-zero rotation is degenerate and never occurs in training. Our runs reach 1.47–1.87.

Checkpoint selection. Both models are evaluated at their final epoch. VT-WM's validation L_1 bottoms at epoch 4–6, but we do not select on it: L_1 is minimised by the conditional mean, and across the nine camera task–stream pairs the best-validation checkpoint wins on PSNR and loses on LPIPS in eight. Action gain also rises monotonically to the final epoch on all three tasks while rollout L_1 bottoms mid-training. Selecting on validation instead moves scene-view PSNR by under 0.1 dB on all three tasks and 0.4–0.7 dB on the motherboard wrist streams, and changes no conclusion.

H FORCE-INFORMED ACTION ABLATION

Tab. 6 compares models trained with and without the force term, on identical test windows. Both are trained for 10 epochs rather than the 20 used for Tab. 1, which halves the cost of the ablation; in our main runs the second 10 epochs change the test metrics only marginally. Scene and wrist streams are unchanged, as expected: the force term only alters the normal component of the sensor action. Tactile PSNR differs by at most 0.6 dB, with significant gains on PushT and on the left Motherboard sensor, and a small loss on the right Rope sensor.

These averages hide what the term does. Contact changes occupy few windows and an imprint occupies few pixels, so frame metrics are dominated by static gel. Fig. 6 shows a PushT window in which the right sensor touches the block. The pose-only action advances the sensor 21.7 mm along the gel normal over the eight predicted steps, 20.1 mm of it before contact; the force term adds 1.2 mm in total, up to 0.66 mm per step. Yet only the force-conditioned model forms an imprint. The sensor-frame pose says that the sensor advances, not where the block surface is; the force change marks the contact. The pose-only prediction stays blank and still scores slightly higher PSNR, since a blank gel costs less than a faint, slightly displaced imprint. The converse case, a release in which the pose-only model holds the imprint two frames past lift-off (43.8 vs. 37.0 dB), is on the project page: there the pose moves under 1 mm along the normal while the force-informed action retreats 1.9 mm.

I CONTACTWORLD

ContactWorld (Zhang et al., 2026) is an action-conditioned vision–tactile world model based on JEPA, with modality-specific CNN encoders and a GRU latent predictor but no image decoder. Each model observes one visual and one tactile stream with single-frame context. After verifying

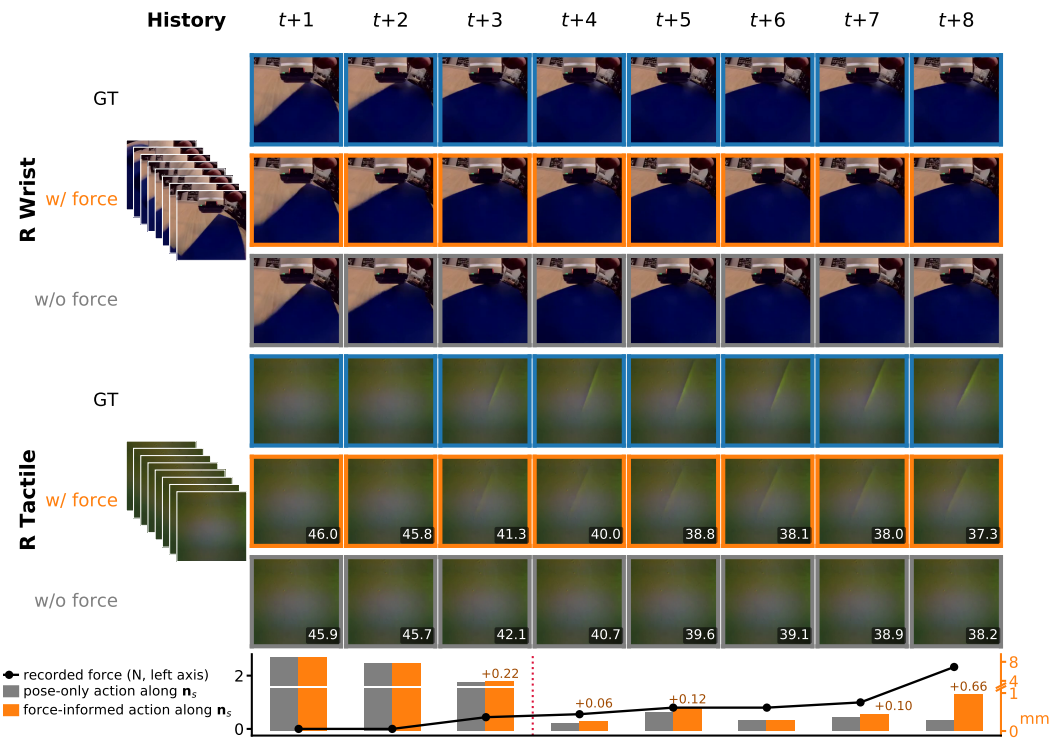


Figure 6: **Effect of the force term on tactile prediction.** A held-out PushT window in which the right sensor touches the block. Rows: ground truth (blue), force-informed actions (orange), pose-only actions (gray) for the right wrist and GelSight streams; numbers are per-frame PSNR. Bottom: recorded normal force (black, left axis) and per-step action along the gel normal $\mathbf{n}_s$ (Eq. 3; right axis, broken), pose-only (gray) and force-informed (orange), with the force increment $k_f\,\Delta f$ above each orange bar; the dotted line marks contact. The force term adds 1.2 mm over the window, and only the model that receives it forms the imprint.

our setup using its released simulation checkpoints, we train ContactWorld from scratch using the official code and hyperparameters for approximately 100k steps. We use the same training episodes, source frames, 36-dimensional normalized actions, and held-out evaluation windows as ACTS.

Five-stream evaluation. To cover all five observation streams without modifying the architecture, we train three models per task on (scene, tactile-left), (wrist-left, tactile-left), and (wrist-right, tactile-right). Scene predictions come from the first model, left wrist and tactile predictions from the second, and right wrist and tactile predictions from the third. Each model retains its original two-stream, single-frame context, whereas ACTS conditions on all five streams over eight frames.

Pixel-space readout. We freeze the trained ContactWorld models and fit a separate readout for each output stream to enable pixel-space evaluation. Each readout maps four consecutive predicted ContactWorld latents, each comprising 512-dimensional visual and tactile embeddings, to one temporally compressed Wan2.2 (Wan et al., 2025) latent block. A two-layer MLP embeds these latents and broadcasts its output over the spatial latent grid. Conditioned additionally on the last ground-truth history VAE latent, four convolutional layers predict a residual relative to that history latent. Each readout contains approximately 1.5M parameters and uses a zero-initialized output layer. Readouts are trained on the GRU's own rollouts from training windows using latent-space MSE and validation early stopping, while ContactWorld remains frozen. Predicted VAE latents are decoded and evaluated using the same modality-specific VAEs and metrics as ACTS.

We additionally evaluate an oracle-input condition using encoded ground-truth future frames and a context-only control replacing ContactWorld latents with their training-set mean. These conditions

Table 6: Ablation of force information in action conditioning. All observation streams are retained. A checkmark indicates that force information is included in the action representation. Within each dataset, bold values indicate a significant difference between the force and no-force configurations for the same observation stream (paired t-test, $p < 0.05$). All configurations are evaluated after 10 training epochs for a fair comparison.

Task	Force	Stream	PSNR↑	SSIM↑	LPIPS↓
Rope	–	Video	28.50	0.9021	0.0495
		Wrist (L)	21.68	0.7583	0.1444
		Wrist (R)	23.62	0.7966	0.1353
		Tactile (L)	45.90	0.9874	0.0230
		Tactile (R)	**46.30**	0.9898	0.0188
	✓	Video	28.56	0.9028	0.0490
		Wrist (L)	21.84	0.7639	0.1401
		Wrist (R)	23.45	0.7932	0.1354
		Tactile (L)	45.73	0.9876	0.0221
		Tactile (R)	**46.00**	0.9893	0.0210
PushT	–	Video	29.25	0.9202	0.0408
		Wrist (L)	22.61	0.8286	0.1067
		Wrist (R)	22.62	0.8432	0.1102
		Tactile (L)	**45.90**	0.9868	0.0256
		Tactile (R)	**47.71**	0.9900	0.0148
	✓	Video	29.14	0.9199	0.0409
		Wrist (L)	22.29	0.8222	0.1110
		Wrist (R)	22.52	0.8412	0.1130
		Tactile (L)	**46.52**	0.9875	0.0219
		Tactile (R)	**47.91**	0.9900	0.0149
Motherboard	–	Video	27.79	0.8798	0.0468
		Wrist (L)	17.71	0.5280	0.2597
		Wrist (R)	20.73	0.6863	0.1626
		Tactile (L)	41.09	**0.9778**	**0.0410**
		Tactile (R)	45.53	0.9882	0.0273
	✓	Video	27.70	0.8783	0.0482
		Wrist (L)	17.57	0.5275	0.2598
		Wrist (R)	20.70	0.6842	0.1612
		Tactile (L)	41.36	**0.9800**	**0.0360**
		Tactile (R)	45.73	0.9886	0.0277

assess readout performance without latent rollout error and without predictive latent information, respectively.