Dexterous Tactile World Model

Anonymous authors
Paper under double-blind review

Abstract

World models for manipulation are typically trained from video, yet the events that determine how manipulation unfolds, such as making and releasing contact, are difficult to observe visually and are often easier to sense through touch. We present the Dexterous Tactile World Model (DTWM), a video world model for future-frame prediction of egocentric manipulation from both observed video and tactile signals from a glove worn on each hand. We condition a pretrained video diffusion transformer on each hand's tactile signal through a zero-initialized residual at the corresponding hand location in the video tokens, while a causal mask prevents predicted frames from accessing future information. Compared with a vision-only model matched in architecture, parameters, and training, DTWM reduces the underestimation of hand motion from 23% to 9%, while reducing the perceptual error in the hand region by 7.4% across three training runs per model. The benefit also increases over the prediction horizon, with the improvement in the later predicted chunks being about 4.1x larger than in the first. DTWM also outperforms other visual-tactile world models under the same setting, and training with touch improves future-frame prediction even when no touch is available at inference. Ablations show that the model benefits from both the magnitude and spatial location of force: replacing the tactile signal with binary contact states, either per hand or per location, increases prediction error. The observed course of the force indicates whether the interaction will persist or change.

1 Introduction

A world model predicts how the physical world will evolve from past observations and other conditions, typically as future frames. Such models are useful for physical AI in three ways. They can *synthesize data*: one recording can yield many possible futures, since real-world manipulation data is costly to scale (Yang et al., 2024b; Jang et al., 2025; NVIDIA et al., 2025; Liu et al., 2026). They can serve as *initialization*: a model that predicts the future well implicitly models how objects interact, which makes it a strong initialization for policies and other models of physical interaction (Wu et al., 2024; Hu et al., 2025). They can also serve as an *environment*: a robot policy (Kim et al., 2025; Black et al., 2025; Sun et al., 2026a;b) can be trained or evaluated in the world model instead of the real world (Ha & Schmidhuber, 2018; Hafner et al., 2020; Bruce et al., 2024; Quevedo et al., 2026).

All three uses require accurate predictions where the outcome is decided: at contact. In manipulation, whether a grasp holds, an object slips, or a finger reaches a surface determines what happens next. Yet world models are trained mainly from video, where contact is hard to observe. The hand often occludes the contact region, and touching an object can look nearly identical to pressing firmly against it. The model must therefore infer contact from ambiguous visual evidence, and its uncertainty propagates to the predictions. The failure is visible in the generated video: predicted hands may lose their fingers, merge with the objects they hold, or move less than they should (Figures 1 and 3).

Touch provides a direct signal of these events. A pressure-sensing glove records where and how firmly each hand makes contact, including contacts that are hidden from the camera. Recent datasets pair egocentric videos of everyday manipulation with such tactile measurements from both hands (Zhou et al., 2026a). We therefore condition a world model on dexterous touch: given the observed video frames and the tactile signals from both hands, it predicts how the manipulation unfolds next.

Several recent world models incorporate touch into video prediction (Higuera et al., 2026; Ma et al., 2026; Huang et al., 2026; Zhou et al., 2026b), but differ from ours in data, setting, and model design.

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.

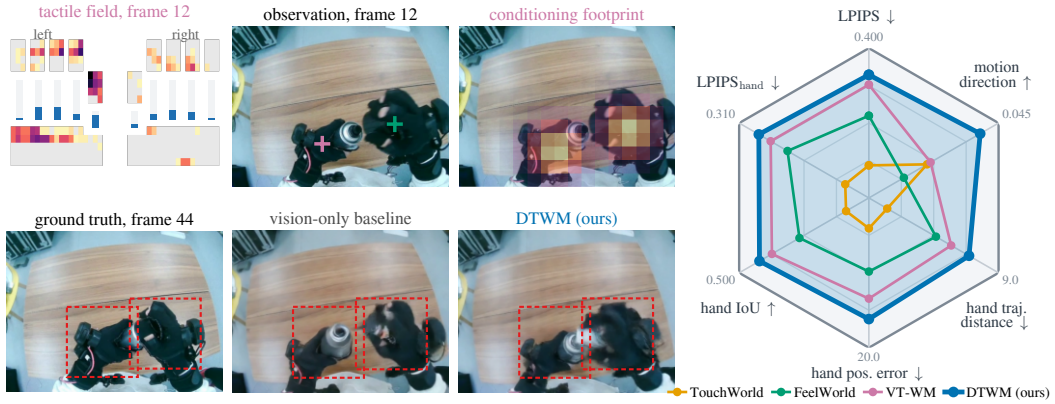


Figure 1: **DTWM shows better future-frame prediction of human manipulation with whole-hand tactile sensing.** Left: a held-out clip in which both hands open a spray can; dashed boxes are the ground-truth hand regions. Right: we achieve better visual quality of the entire frame and the hand crop, and more accurate hand location and motion, than existing visual-tactile world models.

Three are trained on robot trajectories from fingertip or gripper-mounted sensors (Higuera et al., 2026; Ma et al., 2026; Huang et al., 2026), whereas TouchWorld (Zhou et al., 2026b) uses the same human corpus as ours before adapting to a pair of robot hands. All four evaluate touch in robot settings, where the world model is conditioned on the robot's actions or on a subtask and serves planning or a policy. We instead predict egocentric human manipulation from past observations alone, so the model must also infer how the hands themselves will move. Finally, they represent touch as global tactile features, additional tokens or a separate tactile image, none aligned with the image region where contact occurs. We call this missing alignment the *indexing mismatch*: a glove reading is indexed by patches of skin, a video transformer by places in the image, and the touched skin can be anywhere in the frame.

DTWM resolves the mismatch by placing the observed tactile readings at the observed hand location, calculated using rendered hand skeleton (Figure 2). We embed each hand's reading, spread it over the video tokens around that location with a Gaussian footprint, and add it as a residual inside a pretrained video diffusion transformer. The residual is computed by a zero-initialized projection, so the pretrained model is unchanged at initialization and the pathway is learned rather than imposed. The reading is supplied only for the observed frames, and a causal mask keeps every predicted frame from attending to later ones. The model therefore predicts from past observations alone, with no future hand pose, action or reading, as it would at deployment. The backbone, the autoencoder and the text encoder stay frozen, and the tactile pathway adds only a few linear maps.

We evaluate DTWM by observing thirteen frames and predicting the next thirty-six in one pass. The vision-only baseline is matched in architecture, parameters and training, with its tactile input zeroed, so the two models differ only in whether the pathway carries the sensor's signal. The vision-only model underestimates how much the hands move by 23%, and DTWM by 9%. The perceptual error in the hand region falls by 7.4% across three training runs per model, against 5.1% over the whole frame. The improvement also grows along the prediction and is about 4.1x larger in the later chunks than in the first. We also outperform three existing visual-tactile world models under the same setting, with a perceptual error lower by 1.8% to 14.7%.

Ablations of the tactile input show that replacing the full signal with binary contact states, either at each location or for each hand, increases prediction error, with larger reductions in tactile information leading to larger errors. The model therefore uses not only whether a hand is in contact, but also where it is pressed and how firmly. We attribute this to the predictive setting: the observed tactile history captures how force evolves over time, rather than only its value at the end of the observation window. In the corpus, a hand whose force holds steady tends to keep its grasp, whereas a hand whose force decays or rises is 1.8 to 2.5 times as likely to release it within the next half second (Figure 6), and this trend predicts a release beyond what the current force level does. Our contributions are the following:

- **Dexterous tactile world model.** We propose DTWM, a world model for future-frame prediction of egocentric human manipulation. We fine-tune a pretrained video diffusion transformer on egocentric video and tactile signals from both hands, adding each hand's tactile signal to the video tokens at that hand's location through a Gaussian footprint.

- **Improving future-frame prediction of manipulation.** Against a matched vision-only model, it reduces the underestimation of hand motion from 23% to 9% and the hand-region perceptual error by 7.4%. It also outperforms existing visual-tactile world models in the same setting.

- **Dexterous tactile matters.** Conditioning on the location and magnitude of the tactile signal improves prediction more than a binary contact state does. In the training data, how the force changes during the observation indicates whether the interaction will persist or change. Training with touch also improves prediction even when no touch is available at inference.

2 Related Work

Dexterous tactile sensing. Touch is sensed either at high resolution over a small pad or at lower resolution over the whole hand. Optical sensors image the deformation of an elastomer and give dense geometry at a fingertip (Yuan et al., 2017; Lambeta et al., 2020). Taxel arrays and tactile skins trade that resolution for coverage of the palm and all fingers (Sundaram et al., 2019). Robot data comes from teleoperating a gripper or a robot hand carrying such sensors, whereas wearable gloves record people doing everyday tasks, and recent corpora pair egocentric video of bimanual activity with synchronized pressure from both hands (Zhou et al., 2026a; Zhao et al., 2025). These signals serve grasp stability and slip detection (Calandra et al., 2018), visuotactile imitation (Huang et al., 2025), dexterous grasping without sight (Luo et al., 2026), representation learning for control (Lee et al., 2019; Wang et al., 2020; Fu et al., 2024), and unified tactile representations and generation across sensors and modalities (Yang et al., 2024a; Tu et al., 2026). These works use touch to recognize, grasp or represent contact, not to predict how a manipulation unfolds; we instead feed whole-hand touch from both hands to a video world model that predicts future frames.

Visual-tactile world models. Video world models predict how a scene evolves from past frames, first in pixel space (Oh et al., 2015; Finn & Levine, 2017) and now mostly as latent video diffusion transformers (Blattmann et al., 2023b; Peebles & Xie, 2023; Wan et al., 2025), whose high perceptual quality can still hide violations of physical laws (Chen et al., 2026). The fine-tuning of such models can be controlled with an additional modality (Zhang et al., 2023; Mou et al., 2024; Zeng et al., 2025): a global signal such as text or an action (Rombach et al., 2022; Ye et al., 2023; Zeng et al., 2026), a pixel-aligned image such as depth or pose, or extra channels concatenated at the input (Blattmann et al., 2023a), usually added through zero-initialized layers (Zhang et al., 2024; Hu et al., 2022). Several recent world models also condition video prediction on touch. VT-WM (Higuera et al., 2026) concatenates tokens of optical fingertip sensors with visual tokens in an action-conditioned latent world model of a robot hand, and plans with it. FeelWorld (Ma et al., 2026) lets visual tokens attend to one global token per fingertip sensor of a gripper, through a contact-gated attention, and also plans with it. ViTacWorld (Huang et al., 2026) encodes optical tactile images from a gripper with the video autoencoder as an additional view, and uses the model to generate data and to evaluate policies, and Zhang et al. (2026) compare design choices for such models. All of these are trained on robot data, with sensors on the fingertips of a robot hand or on a gripper, whereas we train on egocentric videos of people wearing a tactile glove on each hand. TouchWorld (Zhou et al., 2026b) uses the same corpus as ours: it adapts a pretrained video model to this corpus and then to a pair of robot hands, where it supplies visual-tactile subgoals to a policy. Each of the four is part of a robot system, used to plan actions, generate data and evaluate policies, or supply subgoals to a policy, and all represent the tactile signal as extra tokens, global tactile features or a separate tactile image, none of which is tied to the hand's location in the frame. DTWM is instead a standalone world model of egocentric human manipulation, which predicts future frames from past observations alone and adds each hand's whole-hand tactile reading to the video tokens at its location.

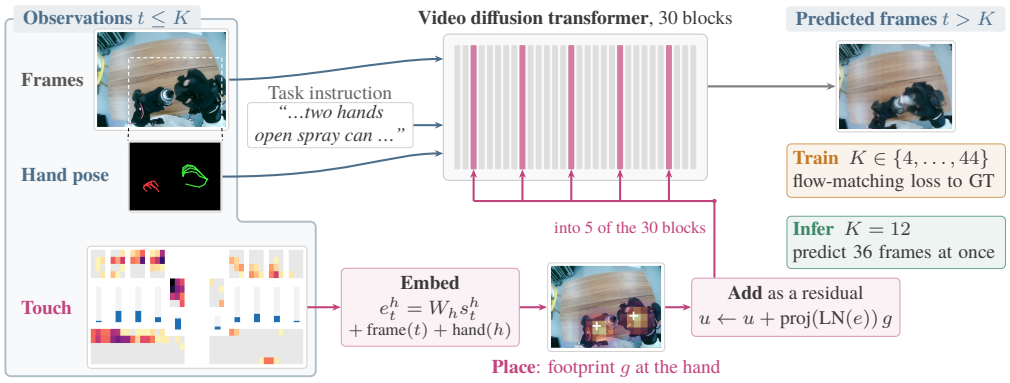


Figure 2: **Pipeline of DTWM.** During both training and inference, the model observes RGB frames, hand poses, and tactile signals from both hands for time steps $t \le K$. We sample one reading from both gloves (255 pressure taxels and 30 finger-flexion channels in total) per video frame. In the touch panel, color shows pressure and blue bars show flexion. The task instruction is encoded by the text encoder, the observed frames are represented as clean latents, and the hand skeleton is encoded using a widened patch embedding. Each hand's tactile signal is embedded as e and added to the video tokens at the hand's projected location using a normalized Gaussian footprint g: $u \leftarrow u + \text{proj}(\text{LN}(e))\, g$. The model then predicts the future RGB frames at time steps $t > K$. During training, K is sampled from $\{4, \dots, 44\}$, and the model is optimized with a flow-matching loss against the ground-truth future frames. At inference, the model observes 13 frames ($K = 12$) and predicts all 36 future frames in a single pass.

3 Method

3.1 Problem setting

We formulate manipulation prediction as block-causal chunked frame prediction with a pretrained flow-matching (Lipman et al., 2023; Liu et al., 2023) video diffusion transformer (Peebles & Xie, 2023; Wan et al., 2025) with 30 blocks and width $d = 3072$. A causal video autoencoder and a $(1, 2, 2)$ patchifier map a clip of $L = 49$ frames at 384×512 to $T = 13$ latent frames on a 12×16 token grid, that is, 2496 video tokens (Appendix C).

The prediction unit is a chunk of $C = 4$ latent frames, or 16 pixel frames. Attention is block-causal (Yin et al., 2025): a token attends bidirectionally inside its own chunk and causally to earlier chunks. Given the observation, pixel frames 0 to K, that is latent frames 0 to $m = K/4$, the model denoises the remaining latents in one pass. Observed latents stay clean with a per-token timestep of zero, and predicted latents share one sampled timestep (Chen et al., 2024).

One pass therefore predicts three chunks jointly. The first predicted chunk attends only to observed latents. Each later chunk also attends to the chunks before it, which are denoised in the same pass, so it depends on the model's own predictions of them. No chunk attends to anything that follows it. During training K is drawn uniformly from $\{4, 8, \dots, 44\}$. At deployment $K = 12$, the model predicts three chunks, or 36 frames, and all of them are evaluated (Section 4.1).

The conditioning contains only information available before the predicted interval (Figure 2). The hand skeleton is encoded by a widened, zero-initialized patch embedding (Appendix C), and no future hand pose or action is provided.

3.2 Tactile conditioning at the hands

The tactile signal is a pressure grid for each hand and frame, indexed somatotopically: each channel is a fixed patch of skin. The transformer is indexed retinotopically, by its token grid. We resolve the mismatch by giving each hand's signal the location of that hand in this grid.

Tactile representation. For each pixel frame t and hand h, the sensor provides a hand-shaped 21×21 raster whose 217 on-hand cells partly share raw glove channels. We remove the duplicates.

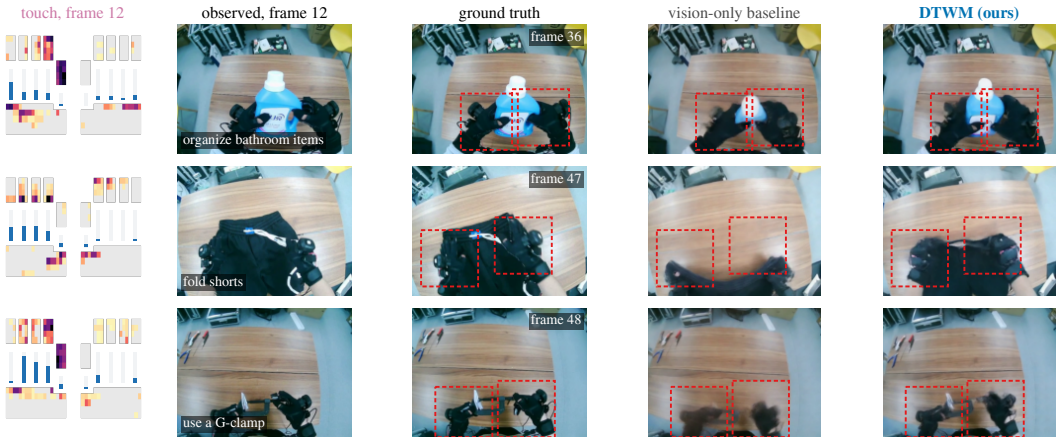


Figure 3: **Comparison with the vision-only baseline.** The hands of the vision-only model lose their definition or stop moving, while ours keep their shape and their motion. Warm colors: pressure, blue bars: finger flexion. Dashed boxes: ground-truth hand regions.

This leaves 138 channels for the left hand and 147 for the right. We standardize each channel with fixed statistics and clip the result to $[-8, 8]$; some channels measure finger flexion (Appendix B). The standardized state s_t^h is embedded as $e_t^h = W_h s_t^h + \text{frame}(t) + \text{hand}(h)$, using a linear map per hand and learned frame and hand embeddings. Embeddings are averaged within each temporal group of the autoencoder, so that t subsequently indexes latent frames.

Section 4.3 replaces this signal with coarser representations, down to a single contact bit per hand and frame, where a hand counts as being in contact when more than 2% of its pressure taxels exceed a normalized pressure of 0.05 (Appendix G).

Projecting the hands into the token grid. The image location of each hand comes from the rendered skeleton the model already receives, produced from per-frame 3D joints (Potamias et al., 2025) under a fixed pinhole projection. The pixel centroid of each hand's color gives normalized coordinates $(y_t^h, x_t^h) \in [0,1]^2$ and a validity flag v_t^h, which is zero when the hand is not drawn. Both are reduced to latent frames by a validity-weighted mean (Appendix D).

Footprint and residual. At cell $p = (i, j)$ of the $H \times W$ token grid, each hand places an isotropic Gaussian that is gated by validity and normalized to unit mass:

$$g_t^h(p) = \exp\Big(-\tfrac{(i-y_t^h H)^2+(j-x_t^h W)^2}{2\sigma^2}\Big), \qquad \hat{g}_t^h(p) = \frac{v_t^h\, g_t^h(p)}{\sum_{p'} v_t^h\, g_t^h(p') + \varepsilon}, \qquad \sigma = 1.8. \tag{1}$$

The normalization makes the injected mass independent of σ, and an undetected hand contributes nothing. Let $u_{t,p}^{(\ell)}$ denote the video token at latent frame t and cell p after block ℓ. Each injection block updates it as

$$u_{t,p}^{(\ell)} \leftarrow u_{t,p}^{(\ell)} + \lambda \sum_{h \in \{\text{L},\text{R}\}} \hat{g}_t^h(p) \Big(W^{(\ell)}\, \text{LN}^{(\ell)}\big(e_t^h\big) + b^{(\ell)} \Big), \tag{2}$$

where $\text{LN}^{(\ell)}$ is a LayerNorm, $W^{(\ell)} \in \mathbb{R}^{d \times d}$ and $b^{(\ell)}$ are a linear map and bias, and λ is a scale fixed at one and stored as a buffer so that the pathway can be disabled exactly at inference (Section 4.3). We inject at blocks $\{0, 6, 12, 18, 24\}$.

Initialization and causal masking. $W^{(\ell)}$ and $b^{(\ell)}$ are initialized to zero, so the residual is zero at initialization while the projection receives gradient from the first step (Appendix C). Setting $v_t^h = 0$ for every $t > m$ removes the residual from all predicted latents, in training and evaluation.

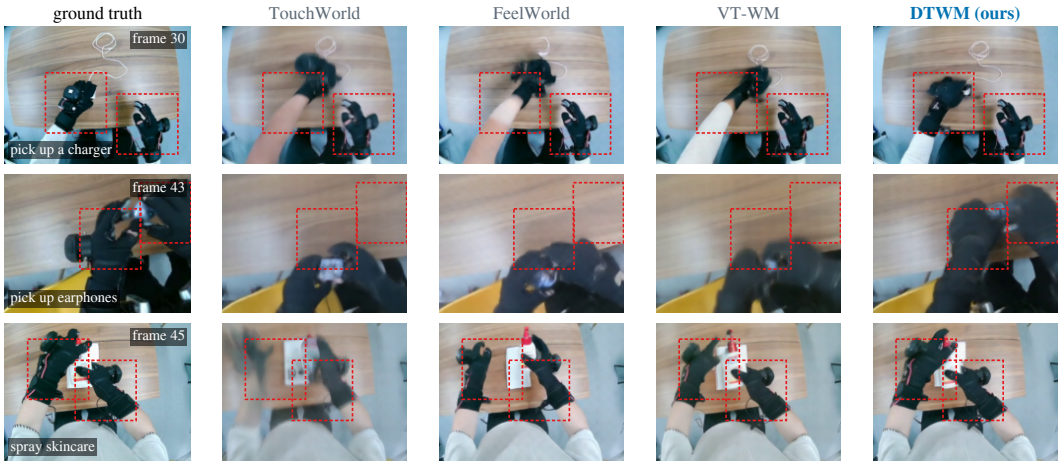


Figure 4: **Comparison with other visual-tactile world models.** In these examples, DTWM preserves hand shape and follows the ground-truth hand positions more closely than the other models.

4 Experiments

We evaluate whether touch improves future-frame prediction and which parts of the tactile signal support that improvement. We first compare DTWM with a matched vision-only baseline and three other visual-tactile world models (Section 4.2), then remove information from the tactile input (Section 4.3). Finally, we examine visual quality and hand motion (Section 4.4), and measure how the gains vary across training runs and prediction horizons (Section 4.5).

4.1 Experimental setup

Data and splits. We use EgoTouch (Zhou et al., 2026a), which pairs egocentric manipulation video with synchronized glove readings from both hands. Its 1,926 episodes yield 43,604 clips of 49 frames, of which 38,600 are used for training. Our main evaluation uses 3,000 clips from recording sessions excluded from training. We divide these by task: 2,728 *in-distribution* (ID) clips have the same, a near-duplicate, or a related task in training; 272 *out-of-distribution* (OOD) clips cover three tasks with no close training counterpart (Table 3). For the three-run and horizon analyses, we use two smaller splits: *held-out episodes*, the first clip of each of 73 held-out episodes, and *training episodes*, 200 reserved clips from the ends of training episodes. The former contains three OOD clips; the latter is entirely ID. Neither split includes clips used for training (Appendix A).

Prediction task. Given RGB frames 0 to 12, the corresponding tactile readings and hand skeletons, and a task instruction, the model predicts frames 13 to 48 (1.2 seconds). No future hand pose, action, or tactile reading is provided. All 36 future frames are generated in one sampling pass with block-causal attention, in chunks covering frames 13 to 28, 29 to 44, and 45 to 48. We score the full predicted interval and analyze shorter windows in Section 4.5.

Metrics and uncertainty. Our primary metric is full-frame LPIPS (Zhang et al., 2018). We also report $\text{LPIPS}_{\text{hand}}$ on an enlarged crop around the ground-truth hands, which usually includes the manipulated object. Both metrics are lower when better. Hand silhouette and optical-flow metrics assess spatial overlap and motion (Appendix D.1). PSNR and SSIM (Wang et al., 2004), together with an analysis of their sensitivity to blur, are reported in Appendix E.1. Percentage reductions are relative to the comparison model, and the statistical procedure is described in Appendix D.

Baselines. The vision-only baseline uses the same architecture, parameter count, and training recipe as DTWM, but its tactile readings are zeroed before the embedding during both training and inference. Its tactile pathway remains trainable and retains the same injection sites and hand locations, so the comparison tests the contribution of the sensor readings. We also compare with three other visual-tactile world models, TouchWorld (Zhou et al., 2026b), VT-WM (Higuera et al., 2026), and FeelWorld (Ma et al., 2026), whose tactile conditioning mechanisms we adapt to our backbone,

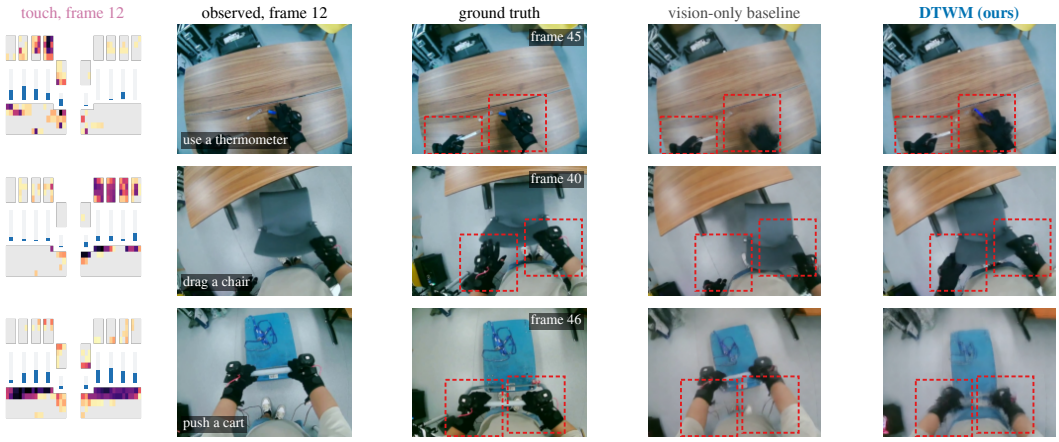


Figure 5: **Examples on OOD tasks.** On tasks with no close counterpart in training, DTWM preserves more hand detail and follows the ground-truth hand positions more closely than the vision-only baseline in the examples shown.

Table 1: **Prediction quality on the held-out set.** DTWM achieves the lowest full-frame and hand-crop LPIPS on both task splits. Hand overlap (IoU) and flow direction agreement are measured over all clips. All rows report one training run.

	ID (2,728 clips)		OOD (272 clips)		Hand, all 3,000 clips	
Model	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	IoU ↑	flow dir. ↑
Vision-only baseline	0.4243	0.3273	0.4882	0.3765	0.475	0.039
TouchWorld (Zhou et al., 2026b)	0.4899	0.3979	0.5341	0.4204	0.376	0.031
VT-WM (Higuera et al., 2026)	0.4237	0.3324	0.4854	0.3797	0.462	0.032
FeelWorld (Ma et al., 2026)	0.4488	0.3465	0.5071	0.3964	0.430	0.027
DTWM (ours)	**0.4157**	**0.3227**	**0.4774**	**0.3668**	**0.476**	**0.041**

glove data, and training recipe (Appendix C.3). ViTacWorld (Huang et al., 2026), which requires optical tactile images, is not included.

Training. We freeze the video autoencoder and text encoder, and adapt the pretrained transformer with rank-64 LoRA (Hu et al., 2022). The widened patch embedding, the tactile embedding, and the LayerNorm and projection that add the tactile residual at five blocks are trained at full rank. The flow-matching loss assigns weights of 0, 1, and 0.2 to observed latent frames, the first predicted chunk, and later frames, respectively. A spatial weight emphasizes the hands; future ground-truth hand locations are used only to weight the loss. All models train for one epoch over 38,600 clips using AdamW at 3×10^{-5} after a 100-step warmup (Appendix C).

4.2 Comparison with the vision-only baseline and other visual-tactile world models

DTWM achieves the lowest LPIPS and $\text{LPIPS}_{\text{hand}}$ on both ID and OOD clips (Table 1). Relative to the vision-only baseline, full-frame LPIPS decreases by 2.0% on ID clips and 2.2% on OOD clips. The same holds on OOD clips, which involve unseen tasks and unseen objects.

The other visual-tactile world models receive the same glove measurements and differ only in how the reading conditions the video model: TouchWorld renders it as image panels, VT-WM appends it as tokens, and FeelWorld adds a gated global feature. DTWM instead adds each hand's reading at that hand's image location, and it has the lowest perceptual error of the four designs. With the same signal and training, how the tactile signal is fed to the model therefore affects prediction quality.

Table 2: **Ablation of tactile input.** The full reading gives the lowest perceptual error. Per-taxel contact retains contact locations and finger flexion but removes pressure magnitude; one bit per hand retains only contact state. The pathway-disabled row evaluates the full-reading checkpoint with its tactile residual switched off. Lower is better for all metrics.

	ID (2,728 clips)		OOD (272 clips)	
Tactile input	LPIPS	$\text{LPIPS}_{\text{hand}}$	LPIPS	$\text{LPIPS}_{\text{hand}}$
zeroed reading (vision-only baseline)	0.4243	0.3273	0.4882	0.3765
pathway disabled at inference	0.4208	0.3297	0.4838	0.3754
one contact bit per hand	0.4307	0.3318	0.5014	0.3864
per-taxel contact, no force	0.4200	0.3246	0.4823	0.3745
DTWM (ours, full reading)	**0.4157**	**0.3227**	**0.4774**	**0.3668**

4.3 What information in touch matters?

Table 2 compares two coarser tactile inputs with the full reading, using one training run per input and keeping the architecture and training recipe fixed. It also evaluates the full-reading model with its tactile pathway disabled at inference.

Pressure magnitude improves prediction. Thresholding each pressure taxel keeps the spatial pattern of contact and the finger-flexion channels but removes pressure magnitude. This increases full-frame and hand-crop LPIPS on both splits. Replacing the reading with one contact bit per hand, which also discards where the hand is pressed and how its fingers are flexed, further increases error and makes the model worse than the vision-only baseline. The model therefore uses more than whether a hand is in contact.

Touch history correlates with future manipulation. The observation records how pressure changes over time. In the training corpus, a hand whose force holds steady rarely releases its grasp within the next chunk, whereas a hand whose force rises or decays is 1.8 to 2.5 times as likely to release it (Figure 6 and Appendix G). The trend remains predictive after accounting for the current force level. Binarizing the pressure removes both the force level and its trend, and with them this cue to how the interaction will continue.

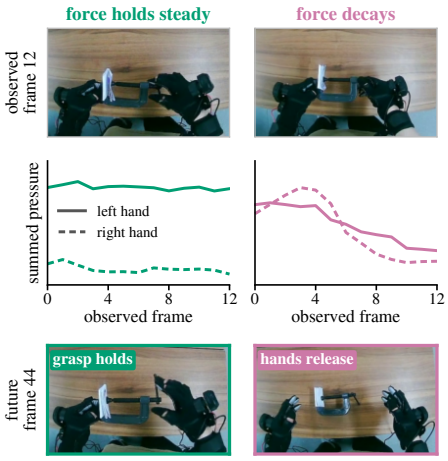


Figure 6: **Touch history correlates with future manipulation.** The observed frames look alike, but the observed touch evolves differently, and so do the manipulations that follow.

Some benefit persists without touch at inference. Setting $\lambda = 0$ in Equation (2) disables the tactile residual of the trained full-reading model. This worsens all four perceptual scores relative to using the full reading, but full-frame LPIPS remains below the vision-only baseline on both ID and OOD clips. We attribute this to training on paired touch and video. Every training clip shows which visual appearance corresponds to which contact, where the hand touches and how firmly, and the model learns this association, which remains useful when the tactile pathway is off.

4.4 Visual quality and hand motion

Visual detail. The examples in Figures 3 to 5 show the improvement most clearly around the hands. In the predictions of the other models, fingers can merge, hand outlines blur, and hands can separate from the objects they hold. DTWM better preserves these details in the examples shown. Across clips, its predictions also retain more high-frequency content than those of the vision-only baseline, although both remain less sharp than the ground truth (Appendix D.2). PSNR and SSIM favor the vision-only baseline; a controlled blur analysis shows that these metrics can improve as predictions lose detail, while LPIPS worsens (Appendix E.1). We therefore assess perceptual quality alongside the hand-behavior metrics below.

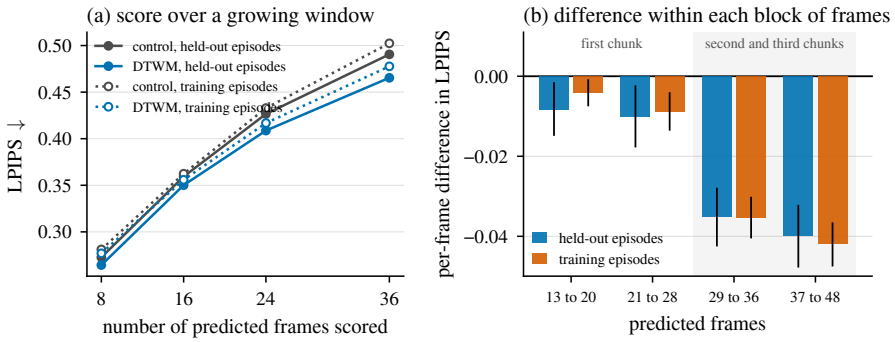


Figure 7: **The LPIPS improvement grows over the prediction horizon.** (a) Cumulative scoring windows: DTWM has lower LPIPS at every window, with a larger gap beyond the first chunk. (b) Disjoint temporal segments: on held-out episodes, the average per-frame gap after the first chunk is about 4.1 times that within it. Scores are averaged over three runs per model.

Hand motion. DTWM improves all reported hand-behavior metrics over the other visual-tactile world models (Table 9). Against the matched vision-only baseline, touch corrects how far the hands move and how large they appear (Table 10). The underestimation of optical-flow magnitude falls from 23% to 9%, and the silhouette area ratio moves closer to one.

4.5 Variation across runs and prediction horizons

Training runs. Across three independently trained runs per model, DTWM has lower mean LPIPS on both smaller evaluation splits (Table 6). It wins 7 of 9 cross-model run comparisons on held-out episodes and 7 of 9 on training episodes (Table 7). The advantage therefore holds on average and in most pairings of individual runs.

Prediction horizon. The LPIPS gap is small in the first predicted chunk and grows in the later chunks on both splits (Figure 7 and Table 8). On held-out episodes, the mean per-frame improvement after the first chunk is about 4.1 times that within the first chunk. Because the same generated videos are scored over each window, this pattern reflects where the benefit appears within a prediction. It is consistent with touch resolving uncertainty about the observed interaction: an error in that estimate barely affects the first predicted frames, but later chunks attend to earlier ones and inherit their errors.

5 Conclusion

We presented DTWM, a world model that predicts egocentric human manipulation from the observed video and the tactile signals of both hands. We finetune a pretrained video diffusion transformer and add each hand's tactile signal to the video tokens at that hand's location through a zero-initialized, causally masked residual. Compared with the vision-only baseline and existing visual-tactile world models, DTWM better predicts how the hands move and appear under the same setting, and its advantage over the vision-only baseline grows over the prediction horizon. Training with touch improves future-frame prediction even when no touch is available at inference. Ablations show that the model benefits from the magnitude and location of the force rather than from contact alone. Touch thus provides a direct signal of the contact events that video struggles to capture, and it makes the predicted hand-object interaction more accurate.

Limitations. To our knowledge, the corpus we use is the only open dataset that pairs egocentric video of everyday human manipulation with tactile signals from both hands, and at about twenty hours it is small for video models. Hand locations come from an off-the-shelf reconstruction that misses some frames. Finally, the model predicts human manipulation, so it cannot yet be placed directly inside a robot planner or policy. Extending it to a robot embodiment is a natural next step.

Ethics statement

This work uses recordings of people who wore instrumented gloves while carrying out everyday tasks in homes, offices, shops, workshops and outdoor settings (Zhou et al., 2026a). The frames may show private interiors, public spaces, personal belongings and, incidentally, other people, and the gloves record the wearer's hands throughout. We collected no new recordings, used the corpus for research purposes only, and do not redistribute raw video. The corpus authors state that the data, code and benchmark will be released publicly (Zhou et al., 2026a); we will follow the license terms they publish, and our own release will cover code, configurations, split lists and per-clip scores rather than imagery. As with manipulation research in general, the method has potential dual uses. Improved video prediction of manipulation can serve robot learning, teleoperation and assistive devices, and we are not aware of a misuse specific to conditioning a video model on a body-worn sensor beyond those already associated with video generation and robot learning.

Reproducibility statement

The appendices describe the corpus and splits (Appendix A), the tactile channels (Appendix B), the model and training recipe (Appendix C), the evaluation protocol and statistics (Appendix D), and the complete results (Appendix E). We will release code, per-model configurations, split lists and per-clip scores.

AI use statement

We used an LLM-based coding assistant during this project and describe its use in the categories requested by the conference policy.

Tasks for which it was used. The assistant helped draft and revise the text of this paper, including this statement. It assisted with literature search, in particular for the recent visuotactile world models discussed in Section 2. It wrote and executed training and evaluation scripts, the statistical analysis and the figure scripts, and it implemented and launched several of the ablation models.

Tasks for which it was not used. No part of the data is synthetic or generated by a model; all data are the released EgoTouch recordings. The paper contains no mathematical result whose proof was produced by an LLM.

Review of the assisted work. Every score, difference and interval in the text and tables is emitted by a script that reads the per-clip score files, so no reported figure can diverge from the data underlying it. Descriptive statistics of the corpus and configuration constants are transcribed from the pipeline's logs. Every bibliography entry was checked against its Crossref or arXiv record. Figures are generated from held-out clips by released scripts, and the frames shown were selected by documented rules. Statements about prior work were checked against the original papers rather than against summaries produced by the assistant. We reviewed all assisted work and take full responsibility for the content of this paper, including text, claims and artifacts produced with the help of generative AI.

References

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems XXI*, 2025. doi: 10.15607/RSS.2025.XXI.010.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable Video Diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a. doi: 10.48550/arXiv.2311.15127.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575. IEEE, 2023b. doi: 10.1109/CVPR52729.2023.02161.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. doi: 10.1109/LRA.2018.2852779.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, pp. 24081–24125, 2024. doi: 10.52202/079017-0759. URL http://papers.nips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html.

Nuo Chen, Lulin Liu, Zihao Li, Ziyao Zeng, Zihao Zhu, Wenyan Cong, Junyuan Hong, Yunhao Yang, Zhengzhong Tu, Yan Wang, Boris Ivanovic, Marco Pavone, Zhangyang Wang, Yang Zhou, and Zhiwen Fan. A physics-grounded benchmark for multi-agent dynamics in world models. In *Computer Vision – ECCV 2026: 19th European Conference, Malmö, Sweden, September 8–12, 2026, Proceedings, Part LXXIV*, Lecture Notes in Computer Science, pp. 547–565. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37029-7_32.

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.

Letian Fu, Gaurav Datta, Huang Huang, William Chung-Ho Panitch, Jaimyn Drake, Joseph Ortiz, Mustafa Mukadam, Mike Lambeta, Roberto Calandra, and Ken Goldberg. A touch, vision, and language dataset for multimodal alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14080–14101. PMLR, 2024. URL https://proceedings.mlr.press/v235/fu24b.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 2455–2467, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models. arXiv preprint arXiv:2602.06001, 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. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2557–2578. PMLR, 2025. URL https://proceedings.mlr.press/v270/huang25e.html.

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, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. doi: 10.1109/LRA.2020.2977257.

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8943–8950. IEEE, 2019. doi: 10.1109/icra.2019.8793485.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Lulin Liu, Nuo Chen, Yan Wang, Bangya Liu, Wenyan Cong, Hezhen Hu, Boris Ivanovic, Hao Wang, Ziyao Zeng, Xinyu Gong, Yang Zhou, Zixiang Xiong, Dilin Wang, Zhangyang Wang, Weisong Shi, Ruohan Zhang, Marco Pavone, and Zhiwen Fan. OpenLongTail: Generative scaling of long-tail driving data. arXiv preprint arXiv:2607.09655, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Shengcheng Luo, Xiyan Huang, Zhe Xu, Wanlin Li, Ziyuan Jiao, and Chenxi Xiao. Blind dexterous grasping via Real2Sim2Real tactile policy learning. arXiv preprint arXiv:2606.11767, 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. arXiv preprint arXiv:2607.24267, 2026.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4296–4304. Association for the Advancement of Artificial Intelligence (AAAI), 2024. doi: 10.1609/aaai.v38i5.28226.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12254. IEEE, 2025. doi: 10.1109/CVPR52734.2025.01143.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hidBHy1CAw.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022. doi: 10.1109/CVPR52688.2022.01042.

Xiatao Sun, Chen Liang, Ziyao Zeng, Qian Wang, Haoyang Zhang, Yue Sun, Qiucheng Li, and Daniel Rakita. Decoupling vision, language, and action for efficient multi-task robot policies. arXiv preprint arXiv:2609.18374, 2026a.

Xiatao Sun, Yuan Zhuang, Mateo Sanchez Lopez Negrete, Matei-Victor Coldea, Chen Liang, Haoyang Zhang, Che Liu, Ziyao Zeng, Shawn Li, Qian Wang, Fei Miao, and Daniel Rakita. Artificial foveated perception for mitigating shortcut learning in robotic foundation models. arXiv preprint arXiv:2607.10655, 2026b.

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. doi: 10.1038/s41586-019-1234-z.

Jiahang Tu, Fengyu Yang, Chenyang Ma, Xihang Yu, Ziyao Zeng, Shaokai Wu, Hanbin Zhao, Zhi Tao, Chao Zhang, Hui Qian, and Alex Wong. UniTac: A unified multimodal model for cross-sensor tactile understanding and generation. In *European Conference on Computer Vision (ECCV)*, 2026.

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. URL https://arxiv.org/abs/2503.20314.

Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. SwingBot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5633–5640. IEEE, 2020. doi: 10.1109/IROS45743.2020.9341006.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero 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.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. Binding touch to everything: Learning unified multimodal tactile representations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26330–26343, 2024a. doi: 10.1109/CVPR52733.2024.02488.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024b.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721, 2023.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, Nashville, TN, USA, 2025. IEEE. doi: 10.1109/cvpr52734.2025.02138.

Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. ISSN 1424-8220. doi: 10.3390/s17122762.

Ziyao Zeng, Jingcheng Ni, Ruyi Liu, and Alex Wong. Coffee: Controllable diffusion fine-tuning. arXiv preprint arXiv:2511.14113, 2025.

Ziyao Zeng, Jingcheng Ni, Daniel Wang, Patrick Rim, Younjoon Chung, Fengyu Yang, Byung-Woo Hong, and Alex Wong. Iris: Integrating language into diffusion-based monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34193–34205, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00355.

Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. LLaMA-Adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=d4UiXAHN2W.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595. IEEE, 2018. doi: 10.1109/cvpr.2018.00068.

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

Yiming Zhao, Taein Kwon, Paul Streli, Marc Pollefeys, and Christian Holz. EgoPressure: A dataset for hand pressure and pose estimation in egocentric vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 27727–27738. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02582.

Jianyi Zhou, Ziteng Gao, Feiyang Hong, Zirui Liu, Guannan Zhang, Weisheng Dai, Ruichen Zhen, Chuqiao Lyu, Haotian Wu, Yinian Mao, Xushi Wang, Yuxiang Jiang, Wenbo Ding, and Shuo Yang. TouchAnything: A dataset and framework for bimanual tactile estimation from egocentric video. arXiv preprint arXiv:2605.13083, 2026a. URL https://arxiv.org/abs/2605.13083.

Jianyi Zhou, Feiyang Hong, Yunhao Li, Yicheng Zhao, Yongjue Cen, Zirui Liu, Jiakang Huang, Zirui Chen, Ruiyang Zhang, Weizhuo Zhu, Xuhua Song, and Shuo Yang. TouchWorld: A predictive and reactive tactile foundation model for dexterous manipulation. arXiv preprint arXiv:2607.07287, 2026b.

A Dataset and preprocessing

Corpus. All experiments use the EgoTouch corpus (Zhou et al., 2026a), released as the HuggingFace dataset `zhouzhoujy/EgoTouch`. It records everyday bimanual manipulation with a head-mounted wide-angle RGB camera and a pressure-sensing glove on each hand. Our copy contains 1933 episode directories from five scenes (Home, Office, Outdoor, Retail and Workbench). Each episode provides frame-aligned egocentric RGB video at 480×640 and 30 fps, two wrist-mounted RGB streams, three sets of 3D hand-joint annotations (WiLoR, HaMeR and Rokoko), per-hand 21×21 pressure grids, raw 256-dimensional glove vectors, hand segmentation masks, wrist tracker poses and an episode-level contact label. We use three of these streams: the egocentric video as context and prediction target, the pressure grids as tactile input, and the WiLoR joints (Potamias et al., 2025), which are projected into the image to render the hand-conditioning stream and to place the tactile residual.

Clips. The clip planner retains 1926 of the 1933 episodes. Three episodes are shorter than one clip and four contain video files that cannot be decoded. The retained episodes contain 2,183,052 frames, or 20.21 hours at 30 fps. Each egocentric stream is divided into contiguous, non-overlapping windows of 49 frames, yielding 43,604 clips of 1.63 seconds. Clips are stored at 432×320 together with a per-clip pressure tensor of shape $(49, 2, 21, 21)$ and the on-hand mask. The loader resizes all frames to a single 384×512 resolution, and inference applies the same resizing, so the evaluation geometry matches training. The stored container rate of 15 fps is metadata only; the frames are consecutive 30 fps samples, so the 36 predicted frames span 1.2 seconds.

Splits. We define our own splits rather than using the split file released with the corpus. The held-out-episode split is disjoint from training at the episode level, and the training-episode split is disjoint at the clip level. Because the corpus often contains several consecutive episodes of the same task, we additionally apply a session holdout: every episode that shares a recording session with an evaluation episode, meaning the same scene and task with consecutive timestamps at most 300 seconds apart, is removed from training. This removes 588 episodes and 4,704 clips. The held-out-episode split consists of the first clip of each of 73 held-out episodes from the Home scene, covering 45 task types. The training-episode split consists of at most the last two clips of each of 100 long training episodes, 200 clips in total, with one 49-frame clip discarded between the training portion of each episode and its evaluation clips so that no evaluation clip is adjacent to a training clip. The large held-out set contains 3,000 of the 4,704 clips removed by the session holdout, none of which is used in training; its division by task is described below. The training split consists of the remaining 38,600 clips from 1338 episodes and 174 task types, amounting to 17.51 hours of video, with a scene composition of 53.3% Home, 16.1% Retail, 13.4% Workbench, 12.8% Outdoor and 4.4% Office.

In-distribution and out-of-distribution tasks. The session holdout makes the held-out set disjoint from training in episodes and sessions but not in content: 39 of its 45 task names never occur in training, yet most of these tasks have a close counterpart among the 174 training tasks. Because task instructions are generated from the task name, we compared every held-out task name with the training task names and assigned it to one of four groups (Table 3). A task is *in-distribution* if the same task, a near-duplicate or a task with a related object or action occurs in training, and *out-of-distribution* if nothing similar does. The out-of-distribution group consists of three tasks, pushing a cart, dragging a chair and using a thermometer, with 272 of the 3,000 clips. The held-out-episode split contains three out-of-distribution clips and the training-episode split none, so both smaller splits are in-distribution.

Task instruction. Every clip is paired with a task instruction generated from a fixed template, "egocentric first-person view of two hands {task}, hand-object manipulation", in which underscores in the task name are replaced by spaces. Instructions contain between 9 and 15 words, with a mean of 11.2. The instruction is always provided, and no model drops it during training.

Handling of data irregularities. The 21×21 pressure grid is a hand-shaped raster rather than the sensor topology. Of its 441 cells, 217 lie on the hand and carry finite values, and the remaining 224 are undefined by design. The loader sets the undefined cells to zero, retains their pattern as a fixed per-hand mask, and restricts all tactile computations to the on-hand cells. The released grids are

Table 3: **Only three held-out tasks have no counterpart in training.** Held-out tasks grouped by their closest counterpart in training, by task name.

Group	Examples (held-out task: training task)	Tasks	Clips
Same task	pick up toothpaste; organize a suitcase; use a microwave	6	479
Near-duplicate	pick up earphones: over-ear headphones; grasp sunscreen: grasp body lotion; fold shorts: fold clothes; organize medicine: sort medicine	18	1,251
Related object or action	squeeze a duck toy: pick up a toy racket; use a G-clamp: rotate a table clamp; pick up a thermos cup: wash a cup	18	998
Nothing similar	push a cart; drag a chair; use a thermometer	3	272

baseline-corrected and normalized to $[0, 1]$ per episode, with separate maxima for the pressure and flexion channels. Hand detections are missing on 10.65% of frames for the left hand and on 12.14% for the right. A joint record that is null, irregular, non-finite or entirely zero is marked invalid. Gaps of at most 30 frames are filled by linear interpolation in 3D, whereas longer gaps leave the hand undrawn. A hand's image location is recovered from the rendered skeleton by color thresholding and is considered present only when at least eight pixels of its color are drawn. When a hand is absent, its tactile residual is zero and the hand-weighted loss has no peak at that hand.

B Tactile sensor layout

The 21×21 grid. Tactile input is provided per frame as a pair of 21×21 grids, one for each hand, and stored per clip as a float16 array of shape $(49, 2, 21, 21)$. The grid is a hand-shaped visualization raster and not the physical sensor layout. Exactly 217 of its 441 cells are finite and lie on the hand; the other 224 are undefined, and we set them to zero and keep their pattern as a fixed mask. The released data are baseline-corrected and rescaled to $[0, 1]$ per episode (Appendix A).

Duplicate cells, pressure channels and flexion channels. The on-hand cells are not independent measurements. Each cell maps to one index of the glove's raw 256-dimensional reading, and up to five cells can share an index. Removing duplicates leaves 138 channels for the left hand and 147 for the right, ordered by raw index and represented by the first cell that uses each index in row-major order. Twelve channels per hand are shared by five cells in rows 7 to 11 and are finger-flexion sensors; on a random sample of 800 training clips such a channel exceeds 0.1 on 27.7% of channel-frames, compared with 9.9% for a pressure taxel. Three further channels per hand also behave as flexion sensors although they are shared by only two cells. They are located on the thumb block in rows 11 and 12, and they are channels 93 to 95 on the left hand and 42 to 44 on the right. Over 60 episodes they exceed 0.15 while every other cell of the thumb block remains below 0.02 on 0.3% of frames, compared with 0.4% for the five-cell flexion channels and 0.0% for any pressure taxel, and they exceed 0.1 on 26.5% of channel-frames, compared with 12.7% for a fingertip taxel. Each hand therefore has 15 flexion channels, three per digit, together with 123 pressure taxels on the left and 132 on the right. All channels are provided to the model; the distinction only affects how the readings are displayed in the figures and how contact is measured in Appendix G.

Standardization. Each channel is standardized with a mean and standard deviation computed over training frames in the $[0, 1]$ grid space. The standard deviation is bounded below by 10^{-3}, and the standardized value is clipped to $[-8, 8]$. One left-hand channel and eighteen right-hand channels are constant in the training data and remain at the lower bound. These statistics were computed on 1,282 clips of the Home scene. The per-frame embeddings, rather than the channel values, are averaged within each temporal group of the video autoencoder, which reduces 49 pixel frames to 13 latent frames.

Validation of region positions. Region positions are not read from the raster. Fingertips are joints 4, 8, 12, 16 and 20, and the palm is the mean of joints 0, 5, 9, 13 and 17, projected with the same function that renders the skeleton, so the two are aligned by construction. A region is invalid when

the hand is undetected or its projected point lies outside the image. The resulting array of shape $(49, 2, 6, 3)$ was computed for all 1926 episodes without failure. At the start of training we verified that the region path was active, that the validity of each region was determined independently, and that the injected residual was exactly zero.

C Model and training details

Table 4 lists the complete training recipe. DTWM, the capacity-matched control, every ablation and every other visual-tactile world model use these settings and differ only in the tactile pathway under study.

Backbone and geometry. DTWM finetunes Wan2.2-Fun-5B-InP (Wan et al., 2025), a flow-matching video transformer with 5,001,016,512 parameters, hidden dimension 3072, 30 blocks, 24 attention heads of dimension 128 and a feed-forward width of 14,336. A training example is a clip of 49 frames at 384×512, which the causal video autoencoder compresses by a factor of four in time and sixteen in space into 48-channel latents; a non-overlapping $(1, 2, 2)$ patchifier turns these into 2496 video tokens. Apart from text cross-attention and the tactile residual, all conditioning is by channel concatenation. The pretrained inpainting input layer reads 100 channels, namely 48 noisy latents, a 4-channel observation mask and 48 reference latents. We extend the patch embedding to 148 channels by appending the encoded hand-skeleton stream, copy the pretrained weights into the first 100 input channels and initialize the 48 new channels to zero, so the extended model is functionally identical to the pretrained model at initialization. Text conditioning uses a frozen umt5-xxl encoder with a 512-token context. The noise schedule is a flow-matching Euler discrete scheduler with 1000 training timesteps and shift 12.0, and the same scheduler is used at inference (Appendix D).

Trained parameters. The autoencoder, the text encoder and the transformer weights are frozen. Adaptation uses two disjoint sets of parameters. The first is a LoRA of rank 64 and scale 64 (Hu et al., 2022) on all 306 linear modules of the transformer. The second consists of densely trained modules: the extended patch embedding, the tactile embedding and the five-block spatial injector. Modules of the tactile pathway are excluded from the LoRA, so the pathway is trained at full rank and does not share LoRA capacity. The capacity-matched control instantiates and trains exactly the same modules with the same initialization. Its pressure tensor is set to zero before the tactile embedding at every training and evaluation step. The embedding still adds its learned frame and hand terms, so the control's residual is a trainable signal placed at the hands that carries no sensor information. The two models therefore differ only in whether the reading contains the sensor's signal.

Optimizer, precision and schedule. A single AdamW optimizer (Loshchilov & Hutter, 2019) updates both parameter sets with a learning rate of 3×10^{-5}, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$ and weight decay 0.01. The learning rate is constant after a linear warmup of 100 steps, and gradients are clipped to a global norm of 1.0. Training uses bf16 precision with gradient checkpointing.

Distributed setup. Each run uses two GPUs with `accelerate` and DeepSpeed ZeRO-2 (Rajbhandari et al., 2020), with optimizer states offloaded to the CPU. Each GPU processes one clip per step and the parameters are updated after every step, so the effective batch size is two clips and one epoch comprises 19,300 updates. Peer-to-peer and InfiniBand transports are disabled, and all-reduce operations pass through host memory. All models use identical settings.

Budget and hardware. Each model is finetuned for exactly one epoch over the 38,600 training clips, and the checkpoint at the end of the epoch is evaluated. Validation during training is disabled, so all reported evaluations come from a separate inference pass. Each run uses two NVIDIA RTX 6000 Ada GPUs with 49 GB of memory, of which about 38 GB per GPU are used, and takes between 5 h 20 and 6 h 30.

C.1 Further details

Alignment of latent and pixel frames. The causal autoencoder maps the 49 pixel frames of a clip to 13 latent frames: latent frame 0 contains pixel frame 0, and latent frame $k > 0$ contains pixel

Table 4: **One recipe for every model in the paper.** Complete training recipe.

Backbone	
Pretrained model	Wan2.2-Fun-5B-InP (flow-matching video transformer)
Transformer parameters	5,001,016,512 (after the channel extension)
Hidden dimension, blocks, heads	3072, 30, 24 (head dimension 128)
Feed-forward dimension	14,336
Patchifier	3D convolution, kernel and stride $(1, 2, 2)$
Video autoencoder	48 latent channels, $4\times$ temporal and $16\times$ spatial compression
Clip length and resolution	49 frames, 384×512
Latent and token grid	$13 \times 24 \times 32$ latents, $13 \times 12 \times 16 = 2496$ tokens
Patch-embedding channels	148: 48 noise, 4 mask, 48 reference, 48 hand skeleton
Text encoder	umt5-xxl, frozen, 512 tokens
Noise schedule	flow-matching Euler discrete, 1000 training timesteps, shift 12.0
Trained parameters	
LoRA rank, scale, targets	64, 64, all 306 linear modules of the transformer
LoRA parameters	164,261,888
Dense parameters	patch embedding 1,821,696; tactile embedding 4,349,725; injector 47,232,000
Total trainable parameters	217,665,309 (4.35%), identical for DTWM, the control and every tactile input
Optimization	
Optimizer	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01
Learning rate	3×10^{-5}, constant after a 100-step warmup
Gradient clipping	global norm 1.0
Precision	bf16, gradient checkpointing
Clips per GPU and number of GPUs	1 and 2
Effective batch size	2 clips per update
Updates per epoch	19,300
Budget	1 epoch, 38,600 clips
Systems	
Software	`accelerate` 1.2.1, DeepSpeed 0.19.2 (ZeRO-2, CPU optimizer offload), PyTorch 2.6.0
Hardware per run	2 NVIDIA RTX 6000 Ada (49 GB), about 38 GB used per GPU
Duration per run	5 h 20 to 6 h 30

frames $4k - 3$ to $4k$. This grouping determines the boundary of the observation, since pixel frames 0 to K correspond to latent frames 0 to $m = K/4$, and it determines the temporal reduction of the tactile embeddings (Appendix B).

Why a gated projection does not train. An earlier version of the injector used a randomly initialized projection behind a per-block scalar gate, setting the scale in Equation (2) to $\lambda^{(\ell)} = \tanh(\gamma^{(\ell)})$ with $\gamma^{(\ell)}$ initialized to zero. The gradient with respect to $W^{(\ell)}$ then contains the factor $\tanh(\gamma^{(\ell)})$, which is zero at initialization, so the projection receives no gradient and retains its random values. The gate can move, but opening it only injects a random projection of the reading, so nothing drives it open. We confirmed numerically that the projection's gradient was zero and that the gates remained closed. Initializing the projection itself to zero, as in zero convolutions (Zhang et al., 2023) and zero-initialized low-rank adapters (Hu et al., 2022), keeps the residual exactly zero at initialization while providing gradient to the projection from the first step.

Loss weighting. Over latent frames, the loss weight is 0 on observed latents $0, \ldots, m$, 1 on the following chunk $m+1, \ldots, m+C$ and 0.2 on all later latents. Spatially, the weight is multiplied by $1 + 5.0\,\mathcal{H}$, where $\mathcal{H} \in [0, 1]$ is the maximum over both hands of a unit-peak Gaussian of width 2.4 latent cells centered at the hand's latent location. This weighting uses ground-truth hand locations of the predicted frames only as supervision, and it is never provided as input.

C.2 Training schedules and variants

Three-run comparison. The full-reading model and the control were each trained three times with the recipe of Table 4, and Table 7 lists the result of each run.

Tactile inputs. The inputs of Table 2 were trained in a later round with the recipe of Table 4, one run is reported for each input, and all were evaluated on the session-held-out set. They differ from DTWM only in the values written into the 21×21 grids before the embedding. *Per-taxel contact* thresholds each pressure taxel at 0.05 and writes a constant, leaving the flexion channels unchanged, so only the force magnitude is removed. The *contact bit* sets every cell of a hand to that constant when more than 2% of the hand's pressure taxels exceed 0.05 and to zero otherwise.

C.3 Other visual-tactile world models

We compare the ways in which existing visual-tactile world models feed a tactile signal to a video predictor, not complete systems. The three original systems differ from ours in predictor, embodiment and task (Table 5), so their reported numbers are not comparable with ours. TouchWorld pretrains on the same corpus as ours; VT-WM and FeelWorld use other sensors on a robot hand or gripper. For each we keep the way the tactile signal is fed to the predictor and take everything else from DTWM: the pretrained backbone and its LoRA adapter, the hand-skeleton conditioning, the training clips, the recipe of Table 4, the sampling of the observation length, the restriction of the tactile input to frames 0 to K, and the evaluation. All three receive the glove reading of DTWM, standardized as in Section 3.2, which is the signal TouchWorld was pretrained on and replaces the original sensor of VT-WM and FeelWorld, so the rows of Table 1 differ only in how the tactile signal is fed to the model. Each was trained once.

Table 5: **Each existing visual-tactile world model keeps its original mechanism and receives the same glove reading as DTWM.** The original system and our implementation.

	TouchWorld (Zhou et al., 2026b)	VT-WM (Higuera et al., 2026)	FeelWorld (Ma et al., 2026)
Original sensor	the same pressure gloves (our corpus), and tactile gloves on two robot hands	four Digit 360 optical fingertip sensors on an Allegro hand	DM tactile sensors, 3D contact point cloud
Original predictor	video model finetuned from Wan2.2-TI2V-5B	action-conditioned latent transformer over Cosmos tokens	latent dynamics predictor over V-JEPA 2 features
Original mechanism	readings rendered as tactile images beside the RGB views	Sparsh-X tactile tokens concatenated with the visual tokens	one global tactile token per fingertip sensor, attended by the visual tokens through a contact-gated attention
Our implementation	grids drawn as image panels in the hand-skeleton conditioning stream	one token per latent frame and hand appended to the video tokens	one tactile feature per hand added to all video tokens, scaled by a learned gate
Tactile parameters	51,581,725 (zeroed residual pathway kept)	23,236,381	51,597,090

TouchWorld: rendered panels. Each hand's 21×21 grid, including the flexion cells, is clipped to $[0, 1]$ and drawn in grayscale at 4 pixels per cell, an 84×84 panel with off-hand cells black and a thin gray frame, 6 pixels from the top-left corner (left hand) and the top-right corner (right hand) of the hand-skeleton video. Grayscale keeps the panels out of the red and green that identify the two hands in the skeleton render. The panels therefore reach the transformer through the same widened patch embedding as the skeleton, at a fixed image position rather than at the hand, and are blank after frame K together with the skeleton. The residual pathway of Section 3.2 is kept with a zeroed reading, exactly as in the control, so this model is the control with the panels added.

VT-WM: appended tokens. The standardized readings are averaged over the four pixel frames of each latent frame and embedded by the per-hand linear map of Section 3.2 together with the frame, hand and observed-frame embeddings. A residual two-layer network (LayerNorm, linear, SiLU, linear, with a zero-initialized output layer) stands in for the original tactile encoder. This gives one token per latent frame and hand, 26 per clip, which are appended after the $2{,}496$ video tokens

and processed by all thirty blocks together with them. Readings after the observation are set to the standardized mean, and the block-causal mask is extended with the chunk index of each tactile token, so that a video token attends to the tactile tokens of its own and earlier chunks only. The tactile tokens are discarded at the output. There is no per-block projection, which is why this design has fewer tactile parameters.

FeelWorld: gated global feature. The tactile embedding is projected at the five injection blocks as in Equation (2), but the footprint $\hat{g}$ is replaced by a uniform distribution over the 12×16 token grid, so each hand contributes one global feature per latent frame. At each injection block the projected feature is multiplied by a gate $\text{sigmoid}(w^{(\ell)\top}\text{LN}^{(\ell)}(e) + c^{(\ell)})$ predicted from the tactile embedding, which adds $5 \times 3{,}073 = 15{,}365$ parameters. The original model gates the attention from the visual tokens to its tactile tokens by a separately predicted contact probability; we keep the residual form shared by the other models and learn the gate end to end, so that this model differs from them only in the gate and in the absence of placement.

D Evaluation protocol

Deployment setting and sampling. All generated-video results in this paper use one evaluation setting. A 49-frame clip is divided at $K = 12$: pixel frames 0 to 12, that is, thirteen frames spanning 0.43 seconds and four latent frames, form the observation, and the model denoises the remaining nine latent frames, corresponding to 36 pixel frames, in a single pass. Observed latents are held at their clean encodings with timestep zero throughout sampling. Attention is block-causal over chunks of four latent frames, so the first predicted chunk (pixel frames 13 to 28) attends only to observed latents, the second (frames 29 to 44) attends in addition to the first, and the third (frames 45 to 48) attends to everything before it. The hand skeleton is black and the tactile residual is zero after frame K, so no future information is supplied. Sampling uses 30 flow-matching steps with scheduler shift 12.0 and classifier-free guidance 6.0 against the negative prompt "bad detailed". The footprint width at evaluation is the inference default $\sigma = 3.0$, whereas training uses $\sigma = 1.8$; evaluating one checkpoint at both widths changes LPIPS by $+0.0005$ on the held-out-episode split, and all models share the same evaluation width. Generated frames at 384×512 are resized bilinearly to the stored 320×432 resolution before scoring. All 36 predicted frames are scored. The analysis in Section 4.5 scores shorter windows of the same generated videos without regenerating them.

Metrics. LPIPS uses the AlexNet backbone with the calibrated linear head, inputs mapped to $[-1, 1]$, computed per frame and averaged over the scored frames. $\text{LPIPS}_{\text{hand}}$ applies the same metric to one crop per clip, defined as the union of rendered skeleton pixels over the scored window, enlarged by 25% of its height and width on each side and resized to 128×128. Because the enlarged crop usually contains the grasped object, it measures the manipulation region rather than the hands alone, and it is undefined for the few clips without a usable skeleton. PSNR is computed as $10 \log_{10}(255^2/\text{MSE})$ over full RGB frames. SSIM is the scikit-image implementation on the mean of the color channels with a data range of 255 and default settings.

Statistical procedure. The unit of pairing is the clip. For a model with several runs, the per-clip scores are first averaged over its runs and then compared with the other model, so that run-to-run variation is included in the estimate rather than being treated as signal. The vector of per-clip differences, with fewer entries where $\text{LPIPS}_{\text{hand}}$ is undefined, is resampled with replacement $20{,}000$ times with a fixed generator. We report the mean difference and the 2.5 and 97.5 percentiles of the resampled means. For sets of more than 400 clips the normal approximation to this interval is used, which agrees with the bootstrap to the reported precision. The interval accounts for clip sampling and evaluation noise but not for the variation between independently trained runs, which is therefore reported separately through the results of individual runs (Table 7).

D.1 Hand-behavior metrics

Frame-level perceptual metrics are dominated by background texture and are largely insensitive to whether the hands behave correctly. We therefore measure hand behavior directly, computing every quantity with the same estimator on ground-truth and generated video so that estimator bias cancels in the difference. The glove silhouette is extracted as dark pixels inside a crop centered on the

ground-truth hand location. The crop is necessary because, over the full frame, the wearer's body and shadows dominate the dark pixels. From the silhouette we compute its overlap with the ground truth (IoU), the distance between centroids in pixels, the ratio of generated to ground-truth silhouette area (ideal value one), and the dynamic time warping distance between the two centroid trajectories (DTW). Optical flow within the same crop gives the agreement of flow directions and the ratio of total flow magnitude to the ground truth (ideal value one). Frames are averaged within each clip before pooling, since pooling individual frames would treat correlated observations as independent. The window is the full predicted interval, frames 13 to 48.

D.2 Sharpness

For Section 4.4 we compute, for each frame, the variance of the Laplacian of the grayscale image, a standard measure of focus, and for each clip the ratio of the mean over generated frames to the mean over ground-truth frames within the scored window. On the 3,000-clip set, this ratio is 0.146 for the capacity-matched control, 0.171 for the full-reading model, and 0.167 for that checkpoint with its tactile pathway disabled ($+0.025$ [$+0.024$, $+0.026$] for DTWM relative to the control). No model exceeds the sharpness of the ground truth, so the ordering reflects how much of the ground-truth detail each model reproduces rather than the addition of spurious texture.

E Complete results

This appendix contains the complete tables for Section 4. Unless stated otherwise, all scores are computed over the 36 predicted frames of the 49-frame clips. LPIPS and $\text{LPIPS}_{\text{hand}}$ are better when lower, and PSNR and SSIM when higher. Every difference is clip-paired (Appendix D). Models are comparable within a table; absolute scores from different training rounds or clip lengths should not be compared.

Table 6: **Three independent training runs per model.** Mean $\pm$ standard deviation over runs. The difference is clip-paired.

	Held-out episodes		Training episodes	
Model	LPIPS	$\text{LPIPS}_{\text{hand}}$	LPIPS	$\text{LPIPS}_{\text{hand}}$
Vision-only baseline	0.4905 ± 0.0328	0.3817 ± 0.0439	0.5024 ± 0.0417	0.3992 ± 0.0488
DTWM (ours)	$\mathbf{0.4654} \pm 0.0109$	$\mathbf{0.3536} \pm 0.0119$	$\mathbf{0.4777} \pm 0.0170$	$\mathbf{0.3733} \pm 0.0149$
Difference	-0.0252	-0.0281	-0.0247	-0.0259

Individual runs. Table 7 lists the mean LPIPS of each run of the full-reading model and of the control in the three-run comparison. Because the runs of the two models are trained independently, we compare every run of DTWM with every run of the control. The pooled clip-paired interval of Table 6 accounts for clip sampling and evaluation noise; this table shows the variation between training runs.

Table 7: **DTWM wins 7 of the 9 run pairings on each split, and its worst run is below the baseline's mean.** Mean LPIPS of each independently trained run in the main comparison, and the number of run comparisons in which DTWM obtains the lower score.

	Held-out episodes		Training episodes	
	control	DTWM	control	DTWM
run 1	0.5281	0.4777	0.5505	0.4963
run 2	0.4677	0.4571	0.4771	0.4630
run 3	0.4758	0.4612	0.4796	0.4737
mean over runs	0.4905	0.4654	0.5024	0.4777
run comparisons won by DTWM	7 of 9		7 of 9	

Scoring window. Table 8 contains the values shown in Figure 7. The upper block scores the same predictions over the first 8, 16, 24 and 36 predicted frames. The lower block gives the mean per frame within each block of frames, obtained by differencing cumulative means: $\ell_{a+1..b} = (b\,\bar{\ell}_{1..b} - a\,\bar{\ell}_{1..a})/(b-a)$, with indices counted over predicted frames.

Table 8: **The per-frame improvement is smallest in the first block and several times larger in the later blocks.** LPIPS as a function of the scored window, and per frame within each block of predicted frames, averaged over the runs of Table 6.

	Held-out episodes			Training episodes		
Predicted frames scored	control	DTWM	difference	control	DTWM	difference
Cumulative window, mean over the frames scored						
frames 13 to 20	0.2727	**0.2644**	−0.0083	0.2812	**0.2770**	−0.0042
frames 13 to 28	0.3593	**0.3501**	−0.0092	0.3624	**0.3559**	−0.0065
frames 13 to 36	0.4265	**0.4087**	−0.0178	0.4329	**0.4168**	−0.0161
frames 13 to 48	0.4905	**0.4654**	−0.0252	0.5024	**0.4777**	−0.0247
Mean per frame within each block						
frames 13 to 20 (first chunk)	0.2727	**0.2644**	−0.0083	0.2812	**0.2770**	−0.0042
frames 21 to 28 (first chunk)	0.4459	**0.4358**	−0.0101	0.4436	**0.4348**	−0.0088
frames 29 to 36 (second chunk)	0.5610	**0.5260**	−0.0350	0.5739	**0.5386**	−0.0353
frames 37 to 48 (second and third chunks)	0.6185	**0.5786**	−0.0399	0.6414	**0.5994**	−0.0420

Hand behavior. Table 10 reports the hand-behavior metrics of Appendix D.1 over the full predicted window, for DTWM and the vision-only baseline on the same 3,000 clips.

Metrics of Figure 1. Table 9 lists the six metrics shown in the radar chart of Figure 1 for every model on the held-out set. Each value is averaged over clips.

Table 9: **DTWM is ahead of every existing visual-tactile world model on all six metrics.** The metrics of Figure 1 on the 3,000-clip held-out set, predicted frames 13 to 48.

Model	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	IoU ↑	centroid (px) ↓	DTW (px) ↓	flow dir. ↑
TouchWorld	0.494	0.400	0.376	28.0	14.2	0.031
FeelWorld	0.454	0.351	0.430	25.1	11.9	0.027
VT-WM	0.429	0.337	0.462	23.3	11.2	0.032
DTWM (ours)	**0.421**	**0.327**	**0.476**	**21.9**	**10.4**	**0.041**

Table 10: **Touch changes how much of the hands is drawn and how much they move: the silhouette area and flow magnitude ratios move toward one.** Hand-behavior metrics over frames 13 to 48 on the held-out set, clip-paired.

	DTWM against the vision-only baseline		
Metric	vision-only	DTWM	difference
silhouette IoU ↑	0.475	**0.476**	+0.001
silhouette area ratio (ideal 1)	0.920	**1.032**	+0.111
trajectory DTW (px) ↓	10.41	**10.37**	−0.04
flow direction agreement ↑	0.039	**0.041**	+0.003
flow magnitude ratio (ideal 1)	0.771	**0.912**	+0.141

E.1 PSNR AND SSIM

Tables 12 and 13 repeat the main-text tables with PSNR and SSIM added. On every split the vision-only baseline has the higher PSNR and SSIM while DTWM has the lower LPIPS, and the existing visual-tactile world models are ordered differently by the two kinds of metric as well. The pixel

metrics are reported here rather than in the main text because on this task they reward a prediction for committing to less, which Table 11 shows directly. Blurring the output of DTWM with a Gaussian of increasing width raises its PSNR from 14.94 to 15.30 and its SSIM from 0.535 to 0.571, while its LPIPS rises from 0.412 to 0.625; the blurred prediction scores higher on PSNR on 98% of the clips and on SSIM on 100%.

The direction of the gap between the two models follows from the same property. Of the -0.30 dB by which DTWM trails the baseline on 150 clips, 61% disappears when each predicted frame's mean and standard deviation are matched to the ground truth's before scoring, and only 9% when prediction and ground truth are both blurred: the gap is carried by global brightness and contrast, not by the high-frequency detail measured in Appendix D.2. On 150 clips the baseline's frames are lower in contrast and in saturation than the ground truth by -7.8 and -16.4 grey levels, those of DTWM by -3.9 and -11.0. Under a squared error, shrinking contrast toward the mean is the optimal hedge for an uncertain prediction, so the model whose frames are closer to the ground truth in contrast pays for it in PSNR and SSIM.

Table 11: **PSNR and SSIM improve when the prediction is blurred; LPIPS penalizes blur.** LPIPS, PSNR and SSIM of the two models and of degraded versions of the DTWM prediction on 150 held-out clips, frames 13 to 48; the last row is the models' reconstruction of the observed frames, which no prediction can pass. Best value of each column in bold.

Prediction	LPIPS ↓	PSNR ↑	SSIM ↑
Vision-only baseline	0.414	15.24	0.547
DTWM	**0.412**	14.94	0.535
DTWM, Gaussian blur $\sigma = 1$	0.476	15.08	0.551
DTWM, Gaussian blur $\sigma = 2$	0.565	15.21	0.564
DTWM, Gaussian blur $\sigma = 3$	0.625	**15.30**	**0.571**
reconstruction of the observed frames	0.031	35.50	0.960

Table 12: **PSNR and SSIM order the models differently from LPIPS.** Table 1 with PSNR and SSIM added, on the same clips.

		ID (2,728 clips)				OOD (272 clips)			
Model	Params	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	PSNR ↑	SSIM ↑	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	PSNR ↑	SSIM ↑
Vision-only baseline	51.6M	0.4243	0.3273	**15.28**	0.5095	0.4882	0.3765	**13.99**	0.5144
TouchWorld (Zhou et al., 2026b)	51.6M	0.4899	0.3979	14.80	0.5088	0.5341	0.4204	13.88	**0.5218**
VT-WM (Higuera et al., 2026)	23.2M	0.4237	0.3324	15.11	**0.5102**	0.4854	0.3797	13.91	0.5203
FeelWorld (Ma et al., 2026)	51.6M	0.4488	0.3465	14.40	0.4896	0.5071	0.3964	13.36	0.5012
DTWM (ours)	51.6M	**0.4157**	**0.3227**	15.06	0.4994	**0.4774**	**0.3668**	13.87	0.5068

Table 13: **Over three runs, PSNR and SSIM favor the baseline where LPIPS favors DTWM.** Table 6 with PSNR and SSIM added, three runs per model.

	Held-out episodes				Training episodes			
Model	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	PSNR ↑	SSIM ↑	LPIPS ↓	$\text{LPIPS}_{\text{hand}}$ ↓	PSNR ↑	SSIM ↑
Vision-only baseline	0.4905	0.3817	**14.98**	**0.5271**	0.5024	0.3992	13.83	**0.4751**
DTWM (ours)	**0.4654**	**0.3536**	14.89	0.5180	**0.4777**	**0.3733**	**13.95**	0.4705
Difference	-0.0252	-0.0281	-0.09	-0.0091	-0.0247	-0.0259	$+0.12$	-0.0045

F Qualitative results

Generated frames are shown at the 384×512 resolution produced by the model.



Figure 8: **Further in-distribution examples show the same behavior as Figure 3: the hands stay in place with the reading and drift or dissolve without it.** As Figure 3.

G Contact and force dynamics of the corpus

This appendix measures how the tactile signal changes over the frames the model predicts, and whether its course over the observed frames carries information about what happens next. All statistics are computed on 12,000 training clips in the evaluation layout of Appendix D: the observation ends at frame 12, the first predicted chunk covers frames 13 to 28 and the second frames 29 to 44.

Definition. Contact and force are measured on the pressure taxels of each hand, excluding the fifteen flexion channels (Appendix B). A taxel is pressed when its normalized pressure exceeds 0.05, and a hand is in contact when more than 2% of its taxels are pressed. The force of a hand is the sum of its taxel values, the observed trend is the change of that force from frames 0 to 3 to frames 9 to 12 relative to its value at the end of the observation, and a hand is released within a chunk when it is in contact at frame 12 and out of contact at some frame of the chunk.

The force changes far more often than the contact state. Within the first predicted chunk the contact state of a hand changes in 17% of cases, and that of the two hands taken together in only 5%. The force of a hand in contact, in contrast, changes by more than 20% in 46% of cases and by more than 50% in 22%, and the center of pressure moves by more than two cells in 48% (Table 14). Over the second chunk all of these rates are higher. A binary contact state therefore describes only a small part of how the interaction evolves over the predicted frames.

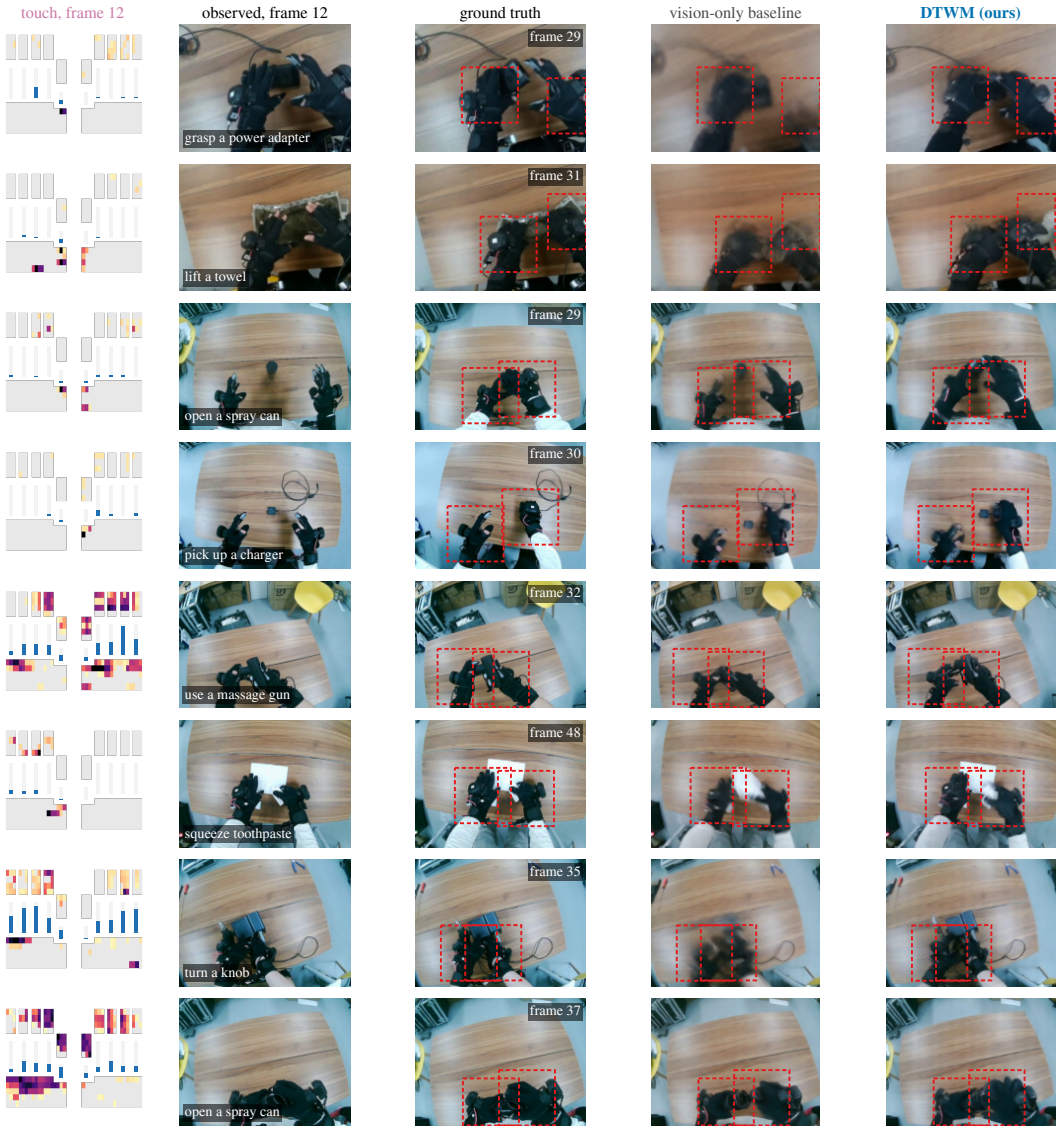


Figure 9: **The same behavior on further in-distribution examples**, continued from Figure 8.

The observed course of the force signals whether a grasp will change. We divide the hands in contact at frame 12 into thirds by their observed trend. A hand whose force holds steady is released within the first predicted chunk in 7% of cases, a hand whose force decays in 16%, and a hand whose force rises in 12%. A steady force thus indicates a grasp that persists, and a changing force, in either direction, a grasp about to change. This is not a consequence of the force level alone. Within each fifth of the distribution of the current force, a decaying or rising hand is released 1.2 to 4.7 times as often as a steady one. A logistic predictor of release that sees the current force and contact area reaches an area under the ROC curve of 0.723; adding the observed trend raises it to 0.735 (+0.012, 95% interval [+0.007, +0.018] over episodes, cross-validated with episodes as groups), and to 0.649 from 0.627 for the second chunk.

Relation to the tactile inputs of Table 2. The per-taxel contact input keeps where each hand is pressed but removes the force and therefore its trend, and the contact bit per hand removes both. The statistics above show that the removed information is predictive of how the interaction continues, which is consistent with the order of cost in Table 2. They describe the corpus and do not by themselves show which part of the reading the model uses.

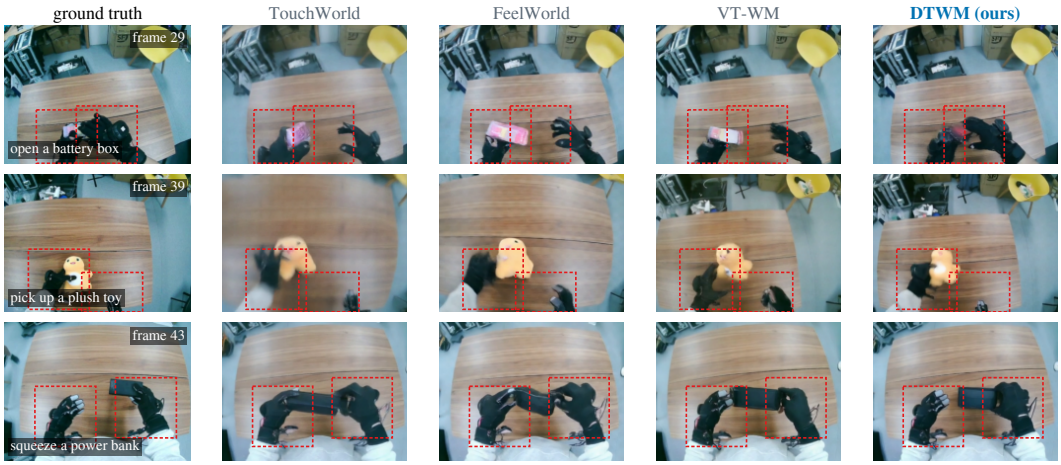


Figure 10: **On further clips, the hands of DTWM remain closest to the ground truth among the tactile models.** As Figure 4.

Table 14: **The force and its location change far more often than the contact state, and the observed force trend signals an imminent release.** Training clips in the evaluation layout; the second and third blocks use the hands in contact at the last observed frame.

	First chunk (frames 13 to 28)	Second chunk (frames 29 to 44)
Share of predicted chunks in which, relative to the last observed frame, …		
the contact state of the two hands taken together changes	5.0%	6.2%
the contact state of a hand changes	16.8%	21.0%
the force of a hand in contact changes by more than 20%	46.1%	60.6%
the force of a hand in contact changes by more than 50%	22.4%	36.4%
the center of pressure of a hand moves by more than two cells	47.8%	62.3%
Release within the chunk of a hand in contact at the last observed frame, by observed force trend		
decaying (lowest third)	16.3%	16.8%
steady (middle third)	6.6%	9.8%
rising (highest third)	11.6%	17.1%
Predicting release, area under the ROC curve		
current force and contact area	0.723	0.627
adding the observed force trend	0.735	0.649