STREAMING 4D HAND INTERACTION FORECASTING WITH AN AUTOREGRESSIVE VIDEO MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a streaming 4D interaction forecaster that turns an autoregressive video model into a structured predictor of future hand motion and contact. Given recent egocentric visual and state observations, the model predicts future interaction as a joint state of bimanual articulated hands, egocentric camera motion, and joint-wise hand-object proximity. This structured representation captures both motion and interaction cues in explicit geometry, providing a forecasting space that is more directly usable than generated video alone. We realize this representation with a 4D state decoder that extracts structured interaction trajectories from intermediate denoising features of an autoregressive video model. The decoder uses current hand and camera states to ground its predictions in the present scene, while incoming observations continually update the forecast. Trained on egocentric clips spanning lab-collected 3D data and diverse in-the-wild videos, our method improves articulated hand forecasting over prior baselines under matched observation and state access, while additionally predicting future hand-object proximity cues. We further demonstrate contact-guided interaction in physics simulation using the predicted hand motion and proximity cues as control references.

1 INTRODUCTION

Anticipating how human hands will move and interact with objects is important for understanding and acting in dynamic environments (Chen et al., 2026b; Li et al., 2025a; Liu et al., 2026a). A useful forecast should capture not only where the hands will move, but also how they articulate, how the egocentric viewpoint changes, and where contact with the scene is likely to occur. Generative video models (Yang et al., 2025; Team Wan et al., 2025; Agarwal et al., 2025; HaCohen et al., 2024) are a promising source of such forecasts because they learn rich priors over how human interactions evolve over time. Yet their predictions are primarily expressed as future visual content (Chen et al., 2025). For downstream interaction reasoning and control, it is often more useful to expose these predictive priors as explicit geometric states that describe future motion and interaction.

Recent work has shown that intermediate representations of video generative models contain useful structure beyond rendered pixels, enabling motion and action readout before video synthesis is complete (Ma et al., 2026b; Pai et al., 2025; Li et al., 2026a; Zheng et al., 2026; Ding et al., 2026; Ang et al., 2026). Other approaches recover geometric trajectories from generated video (Chen et al., 2025) or reconstruct hand motion and hand-object interactions from observed videos (Zhang et al., 2025; Ye et al., 2026b). In this work, we aim to expose the predicted visual future as a structured interaction trajectory that captures how articulated hands move, how the viewpoint evolves, and where interaction with the scene occurs. As the interaction unfolds, this forecast should also adapt to newly observed visual and state information.

To this end, we introduce a streaming 4D interaction forecaster built on an autoregressive video model. Rather than treating predicted video as the only representation of the future, we decode its predictive features into a joint trajectory of articulated bimanual hands, egocentric camera motion, and joint-wise hand-object proximity. This provides explicit geometric and interaction cues over the future horizon while retaining the predictive prior of the video backbone. As new observations arrive, they replace the previously generated continuation in the model context, allowing the structured forecast to be revised with the evolving interaction.

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.

We realize this representation with a 4D state decoder that extracts structured interaction trajectories from intermediate denoising features of the video backbone. The decoder combines predictive visual features with structured state queries across space and time. Observed hand and camera states condition these queries and ground the decoded future in the latest available state. For hand motion, state-relative output further anchors the trajectory to the supplied current wrist while preserving the decoded articulation and orientation.

The resulting formulation supports receding-horizon forecasting over an unfolding interaction. At each update, newly observed video and states are incorporated into the autoregressive context and a new structured future is decoded. In addition to forecasting articulated motion, the model predicts joint-wise proximity to objects, providing contact cues aligned with the hand trajectory. Beyond continuous forecasting on recorded egocentric video, we demonstrate contact-guided interaction in physics simulation, where predicted hand motion provides a motion reference and contact cues guide the downstream controller.

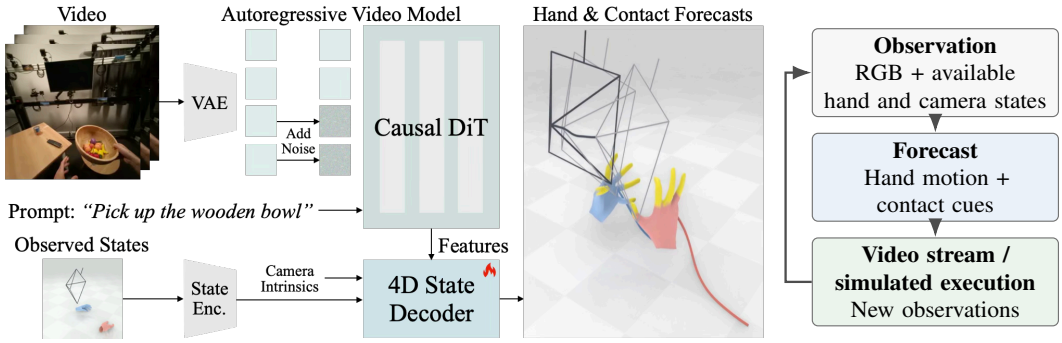
Our contributions are threefold. (1) We introduce a streaming 4D interaction forecasting framework that jointly represents articulated hand motion, egocentric camera motion, and hand-object proximity, and continually updates this forecast from incoming observations. (2) We develop a state-grounded 4D decoder that extracts structured interaction trajectories from the predictive features of an autoregressive video model while conditioning on observed hand and camera states. (3) We demonstrate improved articulated hand and proximity forecasting across four egocentric motion-capture corpora and further demonstrate streaming forecasting on recorded egocentric video and contact-guided interaction in physics simulation.

2 Related Work

Egocentric hand and interaction forecasting. Egocentric forecasting has progressed from predicting coarse hand trajectories and interaction regions to modeling increasingly detailed 3D hand motion. Earlier work forecasts hand trajectories, interaction hotspots, or uncertainty over future motion (Liu et al., 2022; Bao et al., 2023; Ma et al., 2025; 2026a), while more recent methods predict articulated hand pose (Hatano et al., 2025; Choi et al., 2026; Shi et al., 2026). VITRA and Being-H0 further model future hand motion from egocentric observations based on VLMs, providing strong baselines for articulated forecasting (Li et al., 2025a; Luo et al., 2025), while SFHand supports autoregressive forecasting with representations transferable to manipulation policies (Liu et al., 2026a). In parallel, short-term anticipation benchmarks such as Ego4D predict interacted objects and time to contact (Grauman et al., 2022). These approaches typically focus on either future hand geometry or interaction events. Our formulation brings the two together by forecasting articulated 3D hand motion alongside joint-wise hand-object proximity.

World action models and structured motion readout. Video generative models are increasingly used not only to synthesize future observations, but also as predictive representations for control and structured motion. One line of work recovers actions or motion from generated video (Du et al., 2023; Ko et al., 2024; Li et al., 2026b; Dharmarajan et al., 2025), while another couples visual prediction directly with action generation (Li et al., 2025b; Zhu et al., 2025; Ye et al., 2026a; Li et al., 2026a). Some approaches decode task-relevant quantities directly from intermediate video-model features: Video Prediction Policy predicts robot actions from an early denoising pass (Hu et al., 2025), while MotionWAM, DreamTraj, and Adaptive-WAM decode humanoid motion, object trajectories, and driving trajectories (Zheng et al., 2026; Ding et al., 2026; Ang et al., 2026). These methods establish video models as useful predictive backbones beyond pixel generation, but their structured readouts are generally specialized to a particular action or motion space. We instead decode a unified future interaction state and update it repeatedly as new observations arrive.

Hand reconstruction and contact estimation. A separate body of work recovers detailed hand geometry and interaction from observed images and video. HaMeR, HaWoR, and Dyn-HaMR estimate articulated hands and hand motion under moving cameras (Pavlakos et al., 2024; Zhang et al., 2025; Yu et al., 2025), while ViDiHand and ACE-Ego-Hand show that video diffusion representations provide strong features for hand reconstruction (Wang et al., 2026; Liu et al., 2026b). These methods primarily reconstruct observed motion rather than forecast future interaction. Our decoder builds on the geometric readout design of ACE-Ego-Hand, extending it to predictive decoding with



(a) Model overview **(b) Forecast updates**

Figure 1: **Streaming structured interaction forecasting.** (a) Conditioned on available hand and camera states, our 4D state decoder extracts articulated hand motion, egocentric camera motion, and hand-object proximity from observation and forecast features of an autoregressive video model. (b) As new observations arrive, they replace the previous predicted continuation and update the structured forecast. Simulated execution additionally feeds back executed states and rendered observations.

observed-state conditioning, camera motion, and future proximity. Contact has also been modeled explicitly: HACO estimates dense hand-surface contact (Jung & Lee, 2025), while VideoManip refines reconstructed hand-object interactions for downstream policy learning (Chen et al., 2026a). In our setting, proximity is forecast together with future hand motion and does not require object geometry as forecaster input.

3 METHOD

We build a streaming 4D interaction forecaster on an autoregressive video model. Given recent egocentric observations, the video backbone predicts a future visual continuation and exposes intermediate denoising features over the observation and forecast windows. A 4D state decoder extracts from these features a structured interaction trajectory comprising articulated bimanual hands, egocentric camera motion, and joint-wise hand-object proximity. Available hand and camera states condition the decoder, grounding the structured forecast in the current scene. As new observations arrive, they replace the previously generated continuation in the autoregressive context, and the model produces an updated forecast. Figure 1 summarizes the pipeline and the streaming cycle.

3.1 PROBLEM FORMULATION

At the last observed frame t, the forecaster F receives an egocentric visual history $\mathcal{O}_{\leq t}$, camera intrinsics $\mathbf{K}$, available observed hand and camera states $\mathcal{A}_{\leq t}$, and a language instruction ℓ. Given a stochastic future-latent initialization $\boldsymbol{\epsilon}$, the model forecasts a structured 4D interaction trajectory over the next H frames:

$$\hat{\mathcal{S}}_{t+1:t+H} = F\left(\mathcal{O}_{\leq t}, \mathbf{K}, \mathcal{A}_{\leq t}, \ell; \boldsymbol{\epsilon}\right), \tag{1}$$

where $\boldsymbol{\epsilon}$ denotes the stochastic initialization of the future video latent introduced in Sec. 3.2.

For each future frame $f \in \{t+1, \ldots, t+H\}$, the state jointly represents two articulated hands, camera motion, and joint-wise hand–object proximity:

$$\mathcal{S}_f = \left(h_f^L, h_f^R, \mathbf{T}_f, \boldsymbol{c}_f^L, \boldsymbol{c}_f^R\right), \qquad h_f^a = \left(\boldsymbol{t}_f^a, \mathbf{R}_f^a, \boldsymbol{\theta}_f^a, \boldsymbol{\beta}^a\right), \tag{2}$$

for hand $a \in \{L, R\}$.

We represent each hand using MANO (Romero et al., 2017) parameters: translation $\boldsymbol{t}_f^a \in \mathbb{R}^3$ in the camera coordinates of frame f, global orientation $\mathbf{R}_f^a \in \mathrm{SO}(3)$, local finger articulation $\boldsymbol{\theta}_f^a \in \mathrm{SO}(3)^{15}$, and shape $\boldsymbol{\beta}^a \in \mathbb{R}^{10}$. Shape is shared across the forecast window, while translation, orientation, and articulation vary over time. The camera transform $\mathbf{T}_f \in \mathrm{SE}(3)$ maps geometry from frame f into the reference camera frame at t. We additionally predict scalar hand-existence and visibility probabilities.

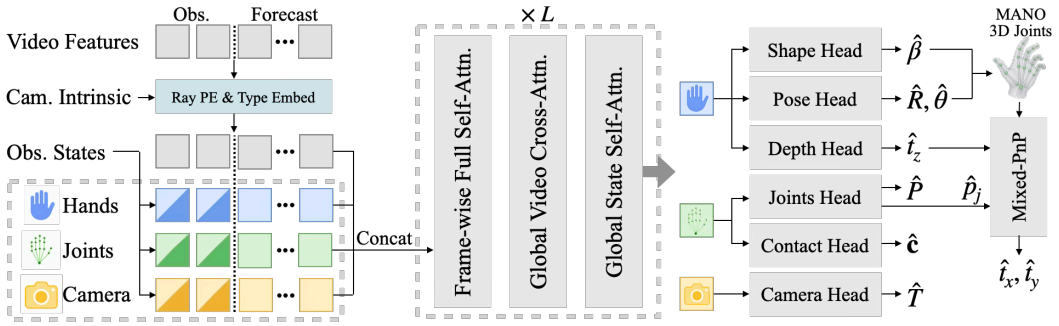


Figure 2: **4D state decoder.** State-conditioned hand, joint, and camera queries interact with observation and forecast features across space and time to extract articulated hand motion, camera motion, and joint-wise proximity.

The proximity cue $\boldsymbol{c}_f^a \in [0,1]^{21}$ assigns each hand joint a probability of lying within $\delta = 2\,\mathrm{cm}$ of an object surface. We derive these supervision labels from annotated hand–object geometry during training; object geometry is not provided to the forecaster at inference.

3.2 Decoding 4D Interaction Trajectories

We realize the forecaster F with an autoregressive video backbone G_ϕ and a structured state decoder D_ψ. The backbone exposes intermediate features for both observed and forecast blocks during denoising. Conditioned on available hand and camera states, the decoder extracts the 4D interaction trajectory directly from these features without RGB decoding (Figure 2).

Predictive video features. We encode the observed video history $\mathcal{O}_{\le t}$ into a clean latent context $\boldsymbol{z}^{\mathrm{obs}}$ and initialize the future latent block with Gaussian noise $\boldsymbol{\epsilon}$. The observed block is processed once as clean context at $\tau = 0$, and its cached representation conditions all subsequent denoising calls. We denote its intermediate features by

$$\mathbf{Z}_t^{\mathrm{obs}} = G_\phi^{\mathrm{feat}}\left(\boldsymbol{z}^{\mathrm{obs}}, \ell, 0\right). \tag{3}$$

These observed features are shared across denoising readouts.

The backbone then iteratively denoises the future block conditioned on the cached observed context and language instruction ℓ. Let $\boldsymbol{z}^{\mathrm{fut},(k)}$ denote the future latent block entering the k-th denoising call, with $\boldsymbol{z}^{\mathrm{fut},(1)} = \boldsymbol{\epsilon}$, and let τ_k denote the corresponding diffusion timestep. We extract forecast-block features as

$$\mathbf{Z}_t^{\mathrm{fut},(k)} = G_\phi^{\mathrm{feat}}\left(\boldsymbol{z}^{\mathrm{fut},(k)}; \boldsymbol{z}^{\mathrm{obs}}, \ell, \tau_k\right), \tag{4}$$

where the semicolon indicates conditioning on the cached observed context. Thus, τ_k applies only to the forecast block, while $\mathbf{Z}_t^{\mathrm{obs}}$ remains fixed across readouts. Readout settings are detailed in Appendix A.1.

The observed and forecast features are projected into the decoder hidden dimension and augmented with viewing-ray encodings derived from the camera intrinsics $\mathbf{K}$, associating each spatial token with its viewing direction in camera coordinates. A learned type embedding distinguishes observed and forecast features, while temporal organization follows the backbone's latent representation. We denote the resulting video tokens at readout k by $\mathbf{X}^{(k)}$. The structured forecast is then

$$\hat{\mathcal{S}}_{t+1:t+H} = D_\psi\left(\mathbf{X}^{(k)}, \mathbf{K}, \mathcal{A}_{\le t}\right). \tag{5}$$

The decoder also produces states for the observed frames. We use $\hat{\mathcal{S}}_{t+1:t+H}$ to denote the forecast portion of its output; decoded observed states are used for observed-state reconstruction and for grounding the future trajectory below.

State-conditioned spatiotemporal decoding. To recover frame-level interaction states from temporally compressed video features, we introduce two hand queries, 42 joint queries, and one camera query at each output frame. These queries represent hand pose and articulation, joint geometry and

proximity, and viewpoint motion, respectively. Let $\boldsymbol{q}_{f,m}$ denote the learned query for state slot m at frame f. Available hand and camera states condition these queries before visual information is aggregated. For observed frames $f \leq t$,

$$\tilde{\boldsymbol{q}}_{f,m} = \boldsymbol{q}_{f,m} + v_{f,m} A_m(a_{f,m}), \tag{6}$$

where A_m projects the observed quantity $a_{f,m}$ and $v_{f,m}$ indicates its availability. Hand queries receive pose and translation parameters, joint queries receive 3D joints and 2D keypoints, and camera queries receive the camera transform and focal parameters. Missing states leave the learned query unchanged, supporting full state histories, current-state-only conditioning, and partial observations. Future queries receive no observed-state addition.

Let $\mathbf{Q}_1$ be the stacked state queries after Eq. 6, and let $\mathbf{X}_1 = \mathbf{X}^{(k)}$. For decoder block l, we denote the incoming state and video tokens by $\mathbf{Q}_l$ and $\mathbf{X}_l$. We denote self-attention by $\mathrm{SA}(\cdot)$ and cross-attention by $\mathrm{CA}(\mathbf{Q}, \mathbf{X})$, with queries from $\mathbf{Q}$ and keys and values from $\mathbf{X}$. Each block performs

$$\begin{aligned} \left[\mathbf{Q}_l^{\mathrm{loc}} \,;\, \mathbf{X}_l^{\mathrm{loc}}\right] &= \mathrm{SA}_{\mathrm{frame}}([\mathbf{Q}_l \,;\, \mathbf{X}_l])\,, \\ \mathbf{Q}_l^{\mathrm{vis}} &= \mathrm{CA}_{\mathrm{video}}\left(\mathbf{Q}_l^{\mathrm{loc}}, \mathbf{X}_l^{\mathrm{loc}}\right), \\ \mathbf{Q}_{l+1} &= \mathrm{SA}_{\mathrm{state}}\left(\mathbf{Q}_l^{\mathrm{vis}}\right), \\ \mathbf{X}_{l+1} &= \mathbf{X}_l^{\mathrm{loc}}, \end{aligned} \tag{7}$$

where $[\,\cdot\,;\,\cdot\,]$ denotes concatenation along the token dimension. Frame-wise attention jointly updates state and video tokens grouped at the same latent time, gathering local spatial evidence. Video cross-attention then reads across the full observed and forecast windows, while state self-attention exchanges information among hand, joint, and camera queries across time. Per-stream feedforward layers are omitted from Eq. 7 for clarity.

Geometric readout and grounding. Prediction heads map the state queries from the final decoder layer to hand parameters, camera motion, and proximity. Hand queries predict global and finger rotations, optical depth, existence, and visibility, while temporally pooled hand features produce one shape vector per hand. Joint queries predict wrist-relative 3D joints and joint-wise proximity probabilities, and camera queries predict $\mathbf{T}_f$. We obtain 2D joint locations through spatial attention: in the final decoder layer, one frame-wise attention head restricts each joint query to video tokens at the corresponding latent frame. A soft-argmax over its attention weights and normalized grid-cell centers yields the 2D keypoint (Liu et al., 2026b).

Articulation determines hand geometry but not its camera-frame placement. We recover hand translation with a mixed-PnP scheme that combines learned depth with geometric alignment (Liu et al., 2026b). Writing $\boldsymbol{t} = (t_x, t_y, t_z)^\top$ with frame and hand indices suppressed, the hand query fixes t_z through the predicted optical depth of the MANO translation origin. With this depth fixed, we align the MANO joints from the predicted pose and shape with the predicted 2D keypoints under intrinsics $\mathbf{K}$, solving for (t_x, t_y) by closed-form linear least squares. If too few correspondences are valid or reprojection error is high, we instead place the wrist along its predicted viewing ray at the same depth. Both paths preserve articulation and depth, while visual joint evidence determines image-plane placement.

Observed-state conditioning informs the decoded motion but does not enforce exact coordinate agreement. When the current wrist is available, we therefore anchor the future trajectory to it. Let $\hat{\boldsymbol{J}}^a_{f,j}$ be the decoded joint position mapped into the reference camera frame, $\hat{\boldsymbol{J}}^a_{t,w}$ the decoded wrist at the current frame, and $\boldsymbol{J}^a_{t,w}$ the observed wrist. For $f \in \{t+1, \ldots, t+H\}$,

$$\tilde{\boldsymbol{J}}^a_{f,j} = \hat{\boldsymbol{J}}^a_{f,j} + \boldsymbol{J}^a_{t,w} - \hat{\boldsymbol{J}}^a_{t,w}. \tag{8}$$

The same offset is applied to every joint and future frame of a hand, preserving articulation and orientation while aligning the trajectory origin with the current state. State conditioning and wrist rebasing therefore provide complementary forms of grounding: the former informs the decoded motion, while the latter enforces agreement with the current trajectory origin.

3.3 Training

Datasets. We train on egocentric human video from ARCTIC, HOT3D, OakInk2, and TACO (Fan et al., 2023; Banerjee et al., 2025; Zhan et al., 2024; Liu et al., 2024), together with EgoDex and

VITRA-1M (Hoque et al., 2025; Li et al., 2025a). The first four provide annotated hand, camera, and object geometry, from which we derive joint-to-object proximity labels. EgoDex and VITRA provide estimator-derived hand and camera trajectories without object supervision. We mix the data sources through weighted sampling and apply each supervision term only where the corresponding annotation is available. The 81-frame clips provide both observed histories and future continuations, while VITRA's shorter 25-frame clips contribute only to observed-video decoding. Details are further provided in Appendix A.4.

Objective. We train the 4D state decoder with the video backbone fixed, constructing backbone features from recorded futures at the same noise levels used at inference. We optimize a multi-task objective over hand geometry, camera motion, and proximity:

$$\mathcal{L} = \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{cam}}\mathcal{L}_{\text{cam}} + \lambda_{\text{prox}}\mathcal{L}_{\text{prox}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}. \tag{9}$$

The geometric losses supervise wrist-relative and camera-frame joints, translation, 2D keypoints, and hand rotations. Camera motion is supervised in the reference camera frame, while proximity uses binary cross-entropy against the joint-to-object labels. Auxiliary losses supervise shape, existence, visibility, and temporal smoothness. We vary the availability of observed state anchors during training so that the decoder can operate with different amounts of state information. Appendix A provides the full loss definitions and weights.

3.4 Streaming Inference

At inference time, we apply the forecaster in a receding horizon as new observations arrive. At each forecast origin t, the model takes the most recent visual history $\mathcal{O}_{\leq t}$ together with available hand and camera states $\mathcal{A}_{\leq t}$ and produces a structured forecast $\hat{\mathcal{S}}_{t+1:t+H}$. After new observations become available, the forecast origin advances and the model predicts a new future from the updated context. Importantly, generated future frames are never treated as observations in subsequent updates. Instead, newly observed frames replace the corresponding portion of the previous predicted continuation, keeping each forecast grounded in actual visual and state feedback. The retained observation history provides temporal context, while the latest hand and camera states anchor the new structured trajectory.

The same update rule supports two inference settings. For recorded-video forecasting, new RGB frames and states are read directly from the recorded sequence. For simulated execution in a physics simulator, a short prefix of the predicted trajectory is executed, and the resulting rendered observations and executed states are fed back into the next forecasting step. Thus, the forecaster itself is unchanged across the two settings; only the source of incoming observations differs. Each update requires encoding the current observation window and predicting a new future latent block with the autoregressive video backbone before decoding the structured trajectory. We report implementation details and the runtime of this recurring computation in Appendix E.

4 Experiments

We evaluate our structured interaction forecasts along three axes: articulated hand motion, future hand-object proximity, and continuous forecasting as new observations arrive. Quantitative evaluation uses held-out sequences from four egocentric motion-capture datasets, while qualitative experiments examine streaming forecasts on recorded video and their use as motion and contact references in physics simulation.

Datasets. We evaluate on 291 clips from ARCTIC, HOT3D, OakInk2, and TACO (Fan et al., 2023; Banerjee et al., 2025; Zhan et al., 2024; Liu et al., 2024), weighting the four datasets equally in aggregate metrics. Each clip provides synchronized egocentric video with annotated hand and camera states, while object geometry supports evaluation of hand-object proximity. We evaluate three forecast origins per clip, at frames 28, 36, and 44. A forecast point is scored at a given horizon only when both hands have valid annotations at the origin and throughout that horizon. Unless otherwise stated, the model observes 29 RGB frames and forecasts up to 32 future frames; quantitative evaluation is performed at 16 and 32 frames, corresponding to 0.53 s and 1.07 s at the datasets' 30 Hz frame rate.

Table 1: **Articulated hand forecasting.** Longer visual context improves single-forecast accuracy at both short and long horizons. Sample-set metrics reveal a different trade-off across methods.

Method	Frames	0.53 s (16 frames)				1.07 s (32 frames)			
		Avg↓	Best10 Avg↓	Final↓	ES↓	Avg↓	Best10 Avg↓	Final↓	ES↓
Current hand state given									
Persistence	1	5.23	–	8.94	6.65	8.61	–	14.10	11.01
VITRA	1	10.32	8.24	19.04	9.71	–	–	–	–
Ours	1	5.94	5.69	9.33	6.63	9.05	8.73	14.07	10.42
Ours	29	**4.59**	**4.18**	**7.69**	**4.87**	**7.48**	**6.91**	**12.38**	**8.25**
No hand state input									
Being-H0	1	9.15	6.34	12.87	**5.67**	12.51	**8.82**	18.03	**8.12**
Ours	1	**6.53**	**6.28**	**9.74**	7.09	**9.54**	9.21	**14.45**	10.75

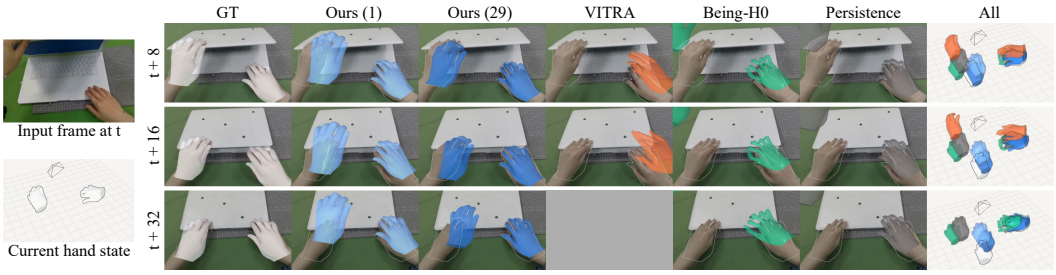


Figure 3: **Hand articulation forecasting comparison.** Each block shows the input frame and supplied current hand state (left), forecasts at $t{+}8$, $t{+}16$, and $t{+}32$ with ground-truth outlines overlaid, and a 3D overlay of all methods (right).

Baselines and metrics. We compare articulated hand forecasting against persistence, VITRA (Li et al., 2025a), and Being-H0 (Luo et al., 2025). Persistence holds the annotated current hand pose fixed over the forecast horizon and serves as a zero-motion reference. VITRA predicts future articulated hand motion from a single egocentric frame, the current hand state, and a language instruction, whereas Being-H0 predicts future hand motion from a single frame and task instruction without hand-state input. We therefore evaluate our model under corresponding one-frame and state-access settings, in addition to our default 29-frame configuration. For comparisons without state input, trajectories are aligned to the annotated current wrist at evaluation. Training exposure differs across methods, so these comparisons match inference inputs rather than training data.

We evaluate both wrists and ten fingertips in the reference camera frame of the last observation, without future scale fitting. **Avg** is the mean Euclidean error over all future frames and keypoints, while **Final** measures endpoint error. For stochastic forecasts, both metrics are computed per sample and averaged over ten trajectories. **Best10 Avg** measures sample-set coverage by selecting the complete trajectory with the lowest average error among ten samples. We additionally report the fair energy score (**ES**), a distributional metric that accounts for both agreement with the recorded future and diversity across samples. Full metric definitions and aggregation details are provided in Appendix B.

4.1 Structured Interaction Forecasting

Articulated hand forecasting. Table 1 compares future articulated hand motion across the four evaluation corpora. Our default 29-frame model achieves the strongest single-forecast Avg and Final errors at both forecast horizons. Compared with the one-frame setting, using visual history substantially improves forecasting accuracy, showing the benefit of observing how the interaction evolves before predicting its continuation. Under matched one-frame inputs, our model improves over VITRA when the current hand state is provided and over Being-H0 without state input. Our advantage is strongest in single-forecast accuracy, which is particularly relevant when a downstream system consumes one trajectory at a time. Being-H0, in contrast, achieves stronger distributional sample quality, with lower ES at both horizons and lower Best10 Avg at 1.07 s. Figure 3 qualitatively compares the predicted trajectories over time.

Table 2: **Future hand-object proximity forecasting.** Our model predicts future proximity states and contact events from the same structured interaction forecast.

(a) Future contact state

Method	Acc.↑	AP↑
GT-state persistence	0.84	0.91
Estimated-state persistence	0.80	0.94
Ours	**0.90**	**0.97**

(b) Future contact events

Event	$P\uparrow$	$R\uparrow$	$\Delta t\downarrow$ (s)
Onset	0.85	0.91	0.15
Release	0.46	0.27	–

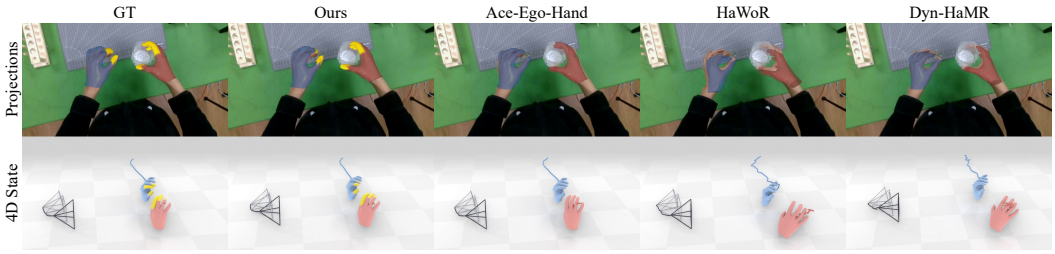


Figure 4: **Observed-state reconstruction.** Our decoder recovers globally consistent hand and camera motion from observed egocentric video while preserving articulated hand structure.

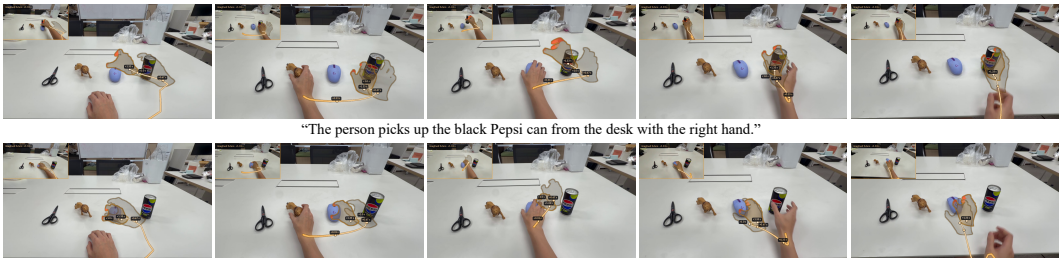


Figure 5: **Streaming forecasting on recorded video.** As new observations arrive, the predicted interaction trajectory conditioned on the prompt adapts to the evolving scene.

Hand-object proximity forecasting. We evaluate future proximity at two levels in Table 2. (a) Future contact state measures whether each hand is in contact over the forecast horizon, where contact is defined by any joint lying within 2 cm of an object surface. Our model achieves strong hand-level accuracy and AP relative to persistence baselines. (b) Future contact events evaluate transitions into and out of contact using onset/release precision and recall together with timing error on detected events. Onset is predicted reliably, while release remains substantially more challenging. Because proximity is predicted per joint alongside the articulated trajectory, the forecast provides spatially localized interaction cues rather than only a hand-level event; the per-joint cue itself is validated on observed video against HACO (Jung & Lee, 2025) in Appendix C. Full definitions are provided in Appendix B.6.

4.2 Qualitative Results

Observed-state reconstruction. Figure 4 qualitatively compares the structured geometry recovered from observed egocentric video. Our decoder recovers articulated hand shape, global placement, and camera-relative motion from the observed sequence. These examples complement the quantitative reconstruction results in Appendix C and verify that the same decoder used for forecasting produces meaningful states on observed frames.

Streaming forecasting. Figure 5 illustrates how the structured interaction forecast evolves as new observations arrive. We follow the same recorded egocentric sequence under two fixed language instructions and repeatedly update the forecast from the latest visual and state observations. The predicted hand trajectories and proximity cues adapt as the observed interaction unfolds. When the requested instruction differs from the recorded continuation, the forecast should be interpreted as an instruction-conditioned interaction proposal rather than a prediction of the recorded future.

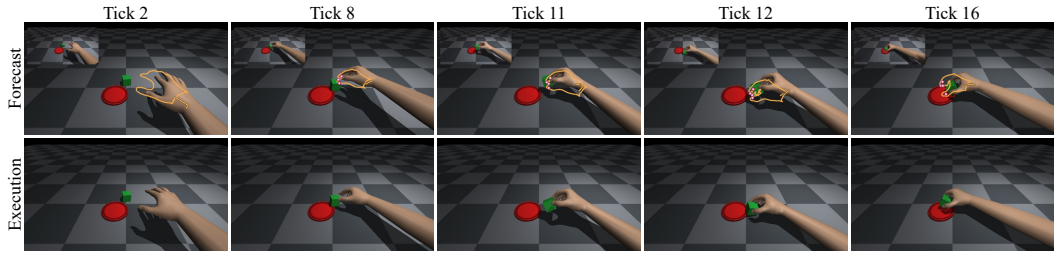


Figure 6: **Closed-loop simulated execution.** Predicted hand motion and proximity cues are repeatedly converted into control actions, executed in simulation, and updated from the resulting observations.

Table 3: **Forecast grounding.** State conditioning improves articulation, while wrist rebasing improves global hand placement.

Cond.	Rebase	Avg↓	Final↓	Artic.↓
×	×	9.33	13.19	4.05
✓	×	8.90	13.01	**3.45**
×	✓	7.82	12.47	4.05
✓	✓	**7.49**	**12.37**	**3.45**

Table 4: **Denoising readout.** Different readouts reveal a trade-off between single-forecast accuracy and distributional sample quality.

Readout	Avg↓	Final↓	ES↓	Contact AP↑
1	**7.5**	**12.4**	8.25	**0.969**
2	8.1	13.6	8.11	0.965
3	8.5	14.2	8.05	0.959
4	8.5	14.4	**7.98**	0.955

Closed-loop execution in a physics simulator. We further examine whether the structured forecast can serve as a reference for downstream interaction. In physics simulation, the predicted articulated hand trajectory provides a motion reference, while joint-wise proximity guides contact-aware refinement through model predictive control (MPC). After executing a short motion prefix, the resulting rendered observations and executed hand and camera states are fed back to the forecaster to produce the next interaction trajectory. Figure 6 shows this closed-loop process and demonstrates how the forecast can be consumed by a downstream controller. This experiment illustrates the forecast interface in closed loop rather than serving as a manipulation benchmark. The MPC formulation and execution protocol are described in Appendix E.4.

4.3 Ablation Studies

Forecast grounding. We first isolate the two mechanisms used to ground the structured forecast in the observed state. Table 3 shows that state conditioning and state-relative wrist rebasing play complementary roles. Conditioning primarily improves wrist-relative articulation, whereas rebasing accounts for most of the reduction in global trajectory error by anchoring the forecast to the observed wrist. Combining both gives the strongest overall Avg and Final errors, supporting the distinction between conditioning the decoded motion and correcting its global placement.

Denoising readout. Table 4 compares structured forecasts extracted at different denoising stages. The first readout achieves the strongest single-forecast hand accuracy and future-contact prediction, motivating its use for quantitative forecasting. Later readouts obtain lower ES despite higher point error, revealing a different trade-off between single-forecast accuracy and distributional sample quality. Streaming visualization and simulated execution instead use the final readout so that the structured trajectory accompanies the completed video rollout; this choice does not imply superior forecasting or control performance.

5 Conclusion

We presented a streaming 4D interaction forecaster that exposes the predictive representations of an autoregressive video model as structured trajectories of articulated hands, egocentric camera motion, and hand-object proximity. A state-grounded 4D decoder extracts these trajectories from predictive video features, while incoming visual and state observations continually update the forecast as an interaction unfolds. Experiments across four egocentric motion-capture datasets demonstrate improved articulated hand and proximity forecasting and highlight the complementary roles of state conditioning and trajectory grounding. Qualitative results further show streaming forecasting on recorded video and the use of predicted motion and proximity as references for contact-guided interaction in physics simulation. Limitations are provided in Appendix G.

AI USE STATEMENT

We used generative AI tools (large language model assistants) to assist with code implementation and testing, experimental design and parameter selection, and manuscript editing and presentation. All AI-assisted code and analyses were reviewed and verified by the authors, and all scientific claims, experimental choices, and manuscript content were critically evaluated and approved by the authors. The authors take full responsibility for the final content of this work, including any text, code, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Architecture, decoder design, training-feature construction, loss weights, and optimization settings are provided in Appendix A, with backbone feature-extraction and denoising-readout details in Appendix A.1. Appendix B specifies the evaluation set, forecasting metrics, camera metrics, and proximity/event protocols. Streaming updates, runtime measurements, and simulated execution are described in Appendix E. All training and evaluation datasets are publicly available. We will release the code, trained structured decoder, and evaluation clip lists.

REFERENCES

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.

Sining Ang, Yuguang Yang, and Yan Wang. Adaptive-WAM: Quality-Guided Early-Exit Planning from Intermediate Video-Diffusion Features. *arXiv preprint arXiv:2608.06008*, 2026.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In *CVPR*, 2025.

Wentao Bao, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, and Yu Kong. Uncertainty-aware state space transformer for egocentric 3D hand trajectory forecasting. In *ICCV*, 2023.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025.

Hongyi Chen, Tony Dong, Tiancheng Wu, Liquan Wang, Yash Jangir, Yaru Niu, Yufei Ye, Homanga Bharadhwaj, Zackory Erickson, and Jeffrey Ichnowski. Dexterous Manipulation Policies from RGB Human Videos via 3D Hand-Object Trajectory Reconstruction. *arXiv preprint arXiv:2602.09013*, 2026a.

Mingfei Chen, Yifan Wang, Zhengqin Li, Homanga Bharadhwaj, Yujin Chen, Chuan Qin, Ziyi Kou, Yuan Tian, Eric Whitmire, Rajinder Sodhi, Hrvoje Benko, Eli Shlizerman, and Yue Liu. EgoMAN: Interaction-structured reasoning for egocentric 3D hand trajectory prediction. In *ECCV*, 2026b.

Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, Yuyang Zhao, Huizi Mao, Ying-Cong Chen, Enze Xie, Xiaojuan Qi, and Song Han. LongLive-2.0: An NVFP4 Parallel Infrastructure for Long Video Generation. *arXiv preprint arXiv:2605.18739*, 2026c.

Jaeyoung Choi, Hyeondong Kim, Yujin Kim, and Daehee Park. EggHand: A multimodal foundation model for egocentric hand pose forecasting. *arXiv preprint arXiv:2605.07642*, 2026.

Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2Flow: Bridging Video Generation and Open-World Manipulation with 3D Object Flow. *arXiv preprint arXiv:2512.24766*, 2025.

Tongsheng Ding, Zhen Luo, Yixuan Yang, Boyu Wang, Luyang Xie, Jinyu Yang, and Feng Zheng. DreamTraj: Generating 6-DoF Object Trajectories by Reading Unrendered Video Diffusion Latents. *arXiv preprint arXiv:2608.00486*, 2026.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *CVPR*, 2023.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Masashi Hatano, Zhifan Zhu, Hideo Saito, and Dima Damen. The invisible EgoHand: 3D hand forecasting through EgoBody pose estimation. *arXiv preprint arXiv:2504.08654*, 2025.

Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

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 *ICLR*, 2022.

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 *ICML*, 2025.

Daniel Sungho Jung and Kyoung Mu Lee. Learning dense hand contact estimation from imbalanced data. In *NeurIPS*, 2025.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *ICLR*, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. *arXiv preprint arXiv:2601.21998*, 2026a.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, Yizhong Zhang, Xi Chen, Hao Chen, Lily Sun, Dong Chen, Jiaolong Yang, and Baining Guo. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *RSS*, 2025b.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning Video Models into Generalist Robot Policies. *arXiv preprint arXiv:2605.27817*, 2026b.

LightX2V Team. LightX2V Autoencoders: LightTAE for Wan2.2. Model repository, 2026. URL `https://huggingface.co/lightx2v/Autoencoders`. Accessed September 25, 2026.

Ruicong Liu, Yifei Huang, Liangyang Ouyang, Caixin Kang, and Yoichi Sato. SFHand: Learning embodied manipulation by streaming egocentric 3D hand forecasting. *arXiv preprint arXiv:2511.18127*, 2026a.

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022.

Yufei Liu, Xixi Wang, Hao Li, Ganlong Zhao, Kaitong Cai, Chengkai Jin, Chunxiao Liu, Jianbo Liu, Siyuan Huang, Xingang Pan, and Hongsheng Li. ACE-Ego-Hand: Repurposing Video Diffusion Models for Occlusion-Robust Egocentric 3D Hand Motion Recovery. *arXiv preprint arXiv:2608.20308*, 2026b.

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. TACO: Benchmarking generalizable bimanual tool-action-object understanding. In *CVPR*, 2024.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

Junyi Ma, Jingyi Xu, Xieyuanli Chen, and Hesheng Wang. Diff-IP2D: Diffusion-based hand-object interaction prediction on egocentric videos. In *IROS*, 2025.

Junyi Ma, Xieyuanli Chen, Wentao Bao, Jingyi Xu, and Hesheng Wang. MADiff: Motion-aware Mamba diffusion models for hand trajectory prediction on egocentric videos. *TPAMI*, 48(3): 3250–3267, 2026a.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026b.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

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

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 36(6):245:1–245:17, 2017.

Novin Shahroudi, Mihkel Lepson, and Meelis Kull. Evaluation of trajectory distribution predictions with energy score. In *ICML*, 2024.

Zhaofeng Shi, Heqian Qiu, Lanxiao Wang, Xiang Li, and Hongliang Li. Exo2EgoPose: Leveraging exocentric demonstrations for vision-language guided egocentric 3D hand pose forecasting. *arXiv preprint arXiv:2607.15890*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IROS*, 2012.

Yuxi Wang, Chengkai Jin, Yufei Liu, Wenqi Ouyang, Tianyi Wei, Zhiwei Zeng, Siyuan Huang, Zhiqi Shen, and Xingang Pan. The Surprising Effectiveness of Video Diffusion Models for Hand Motion Reconstruction. *arXiv preprint arXiv:2606.30308*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026a.

Yufei Ye, Jiaman Li, Ryan Rong, and C. Karen Liu. WHOLE: World-grounded hand-object lifted from egocentric videos. In *CVPR Findings*, 2026b.

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. Dyn-HaMR: Recovering 4D Interacting Hand Motion from a Dynamic Camera. In *CVPR*, 2025.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OakInk2: A dataset of bimanual hands-object manipulation in complex task completion. In *CVPR*, 2024.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-Space Hand Motion Reconstruction from Egocentric Videos. In *CVPR*, 2025.

Jia Zheng, Teli Ma, Yudong Fan, Zifan Wang, Shuo Yang, and Junwei Liang. MotionWAM: Towards Foundation World Action Models for Real-Time Humanoid Loco-Manipulation. *arXiv preprint arXiv:2606.09215*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *RSS*, 2025.

A Implementation Details

A.1 Backbone and Readout

We use Wan2.2-TI2V-5B (Team Wan et al., 2025) within the autoregressive LongLive-2.0 framework (Chen et al., 2026c), and extract intermediate features from transformer block 15 of 30. Eight observed latents condition eight future latents, corresponding to a 32-frame prediction horizon.

The future block is generated with the released four-step distribution-matching-distillation (DMD) schedule (Chen et al., 2026c). The shift-5 flow schedule uses timesteps $t \approx \{999, 938, 833, 625\}$, with corresponding noise levels $\sigma = t/1000$. We use these same noise levels when constructing decoder training features through $\mathcal{Q}_{\text{deploy}}$ (Appendix A.5).

Readout k denotes features extracted during the k-th denoising call. The first readout ($t \approx 999$) is taken before any update to the future latent block, whereas the final readout ($t \approx 625$) follows three latent updates. Quantitative forecasting uses the first readout, while streaming visualization and simulated execution use the final readout. Appendix D.1 describes the readout-selection procedure and sensitivity analysis.

A.2 Decoder

The 4D state decoder contains 12 transformer blocks with hidden width 1024, feedforward width 4096, and 16 attention heads. Backbone features have width 3072. At each output frame, the decoder maintains two hand queries, 42 joint queries, and one camera query, together with four register tokens per latent frame.

Video features are projected from the backbone width to the decoder width with a normalized linear layer and augmented with viewing-ray and observation/forecast type encodings. The viewing-ray encoding maps the azimuth and elevation of the unit ray through each spatial token center, computed from normalized camera intrinsics, through eight Fourier-feature octaves followed by a small zero-initialized MLP. The same spatial encoding is shared across latent frames. A two-entry zero-initialized learned embedding distinguishes observed and forecast latents. All forecast readouts share the same entry, and the decoder receives no denoising-timestep embedding.

A.3 Mixed-PnP Translation

Following ACE-Ego-Hand (Liu et al., 2026b), we recover metric hand translation by combining learned optical depth with a geometric solve for the two in-plane translation components. The hand query predicts ζ, which determines the depth of the MANO translation origin as

$$t_z = f_x \exp(\zeta), \tag{10}$$

where f_x is the normalized focal length.

Let $\boldsymbol{J}_j = (X_j, Y_j, Z_j)^\top$ denote a MANO joint after applying the predicted pose and shape, but before global translation. Its depth after translation is $z_j = Z_j + t_z$. The corresponding predicted 2D keypoint (u_j, v_j) defines the viewing direction

$$b_j^x = \frac{u_j - c_x}{f_x}, \qquad b_j^y = \frac{v_j - c_y}{f_y}, \tag{11}$$

where the keypoints and intrinsics use the same normalized image coordinates.

With t_z fixed, projection is linear in t_x and t_y. Minimizing reprojection error over valid correspondences yields

$$t_x = \frac{\sum_j m_j (b_j^x z_j - X_j)/z_j^2}{\sum_j m_j / z_j^2}, \qquad t_y = \frac{\sum_j m_j (b_j^y z_j - Y_j)/z_j^2}{\sum_j m_j / z_j^2}, \tag{12}$$

where m_j is a binary validity mask. We retain joints with depth greater than 0.05 m and 2D keypoints lying at least 0.02 inside the normalized image boundary. The solve changes only the in-plane translation; predicted depth, rotation, and articulation remain unchanged.

Table 5: **Training data.** Clip counts in the training split, label provenance per source, whether joint-to-object proximity labels exist, and the per-corpus sampling weight (drawn with replacement). VITRA contributes observation-only windows.

Dataset	Clips	Hand	Camera	Proximity	Weight
ARCTIC	4,663	annotated	annotated	✓	0.063
HOT3D	11,456	annotated	annotated	✓	0.135
OakInk2	22,204	annotated	annotated	✓	0.187
TACO	9,015	annotated	annotated	✓	0.136
EgoDex	91,664	estimated	estimated	–	0.120
VITRA	459,084	estimated	estimated	–	0.359

We accept the solution when at least six valid correspondences are available and the reprojection RMS is at most $\max(15 \text{ px}, 0.25d)$, where d is the predicted 2D keypoint bounding-box diagonal. Otherwise, we use a wrist-ray fallback with the same predicted depth and shape. Let $\boldsymbol{w}_\beta$ denote the MANO origin-to-wrist offset for the predicted shape. The wrist depth is $z_w = t_z + w_{\beta,z}$. We unproject the predicted 2D wrist keypoint at depth z_w and subtract $\boldsymbol{w}_\beta$ to recover the MANO-origin translation. Gradients propagate through the selected branch with a detached branch-selection gate.

A.4 Training Data

Table 5 summarizes the training sources, including the number of clips in each training split, the provenance of hand and camera labels, availability of object geometry for proximity supervision, and the sampling weight assigned to each corpus. Sampling is performed with replacement, and each loss term is applied only when its required annotations are available.

A.5 Training Features

Decoder training uses both observed-only windows and windows with annotated future continuations. For samples with a future continuation, the observed context remains clean while the recorded future latent is corrupted to match the noise levels encountered at deployment:

$$\boldsymbol{z}_\sigma = (1-\sigma)\boldsymbol{z}^{\text{gt}} + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \sigma \sim \mathcal{Q}_{\text{deploy}}. \tag{13}$$

Here, $\mathcal{Q}_{\text{deploy}}$ samples the denoising noise levels defined in Appendix A.1. This construction exposes the decoder to the same feature regimes used at inference while retaining supervision from the recorded future. At evaluation, the future continuation is withheld and generated by the video backbone.

For state conditioning, anchor availability is sampled independently for each training window: no state anchors with probability 0.2, only the last observation frame state with probability 0.4, and state anchors at every observed latent boundary (frames $0, 4, \ldots, 28$) with probability 0.4. Hand and joint anchors are additionally masked according to per-frame annotation validity.

A.6 Optimization

We optimize the decoder using the multi-task objective described in Sec. 3.3. Loss weights are 1.0 for wrist-relative forward-kinematic and joint-token supervision, 0.5 for camera-frame joints, 0.2 for wrist position, 0.1 for translation, rotation, 2D keypoints, reprojection, and camera pose, 0.05 for existence, acceleration, and proximity, 0.025 for visibility, and 0.01 for shape. Missing annotations are handled with per-loss validity masks.

Placement and 2D losses are linearly warmed up over the first 1500 optimization steps. We train with a learning rate of 2×10^{-4}, 500 learning-rate warmup steps, effective batch size 16, and a 40K-step cosine schedule. The video backbone remains fixed throughout decoder training.

A.7 Generator Adaptation

For forecasting, we adapt the video backbone to egocentric video using LoRA (Hu et al., 2022), while keeping the trained 4D state decoder fixed. The domain adapter uses rank 128 and scaling 128. Observed-state reconstruction uses the base backbone without this adapter. Importantly, the structured decoder is not retrained on features from the LoRA-adapted backbone.

B Evaluation Protocol

B.1 Evaluation Set

The evaluation set contains 291 clips: 34 from ARCTIC, 84 from HOT3D, 25 from OakInk2, and 148 from TACO. We evaluate three forecast origins per clip, at frames 28, 36, and 44, yielding up to 873 forecast points. A forecast point is retained for a given horizon only when both hands have valid annotations at the origin and throughout the full horizon. Aggregate results weight the four corpora equally by averaging per-corpus means.

B.2 Hand Forecasting

We evaluate two wrists and ten fingertips in the camera frame of the last observation, without future scale fitting. For a forecast horizon of H frames, **Avg** is the mean Euclidean error over all 12 keypoints and all future frames, while **Final** measures the same error at frame $t + H$.

For stochastic forecasts, Avg and Final are first computed independently for each of ten samples and then averaged across samples. These metrics therefore measure the expected error of drawing a single forecast rather than the error of an averaged trajectory. They are not proper distributional scores and can favor concentrated predictions.

Best-K measures sample-set coverage through oracle selection. For each subset of K samples, we select one complete two-hand trajectory using the recorded future: Best-K Avg selects the trajectory with the lowest Avg error, while Best-K Final selects the trajectory with the lowest endpoint error. Reported Best-K scores average over all K-subsets of the ten generated samples.

Persistence holds the annotated current hand pose fixed throughout the forecast horizon. For state-relative evaluation, each predicted hand trajectory is translated so that its decoded wrist at the forecast origin matches the annotated current wrist; articulation and orientation are unchanged. No forecasting result substitutes an estimated current state for the annotated state.

For samples $X_1, \ldots, X_K$, recorded trajectory Y, and trajectory distance d, we additionally report the fair energy score (Shahroudi et al., 2024):

$$\widehat{\mathrm{ES}}_{\mathrm{fair}} = \frac{1}{K}\sum_{r=1}^{K} d(X_r, Y) - \frac{1}{2K(K-1)}\sum_{r\neq s} d(X_r, X_s). \tag{14}$$

We define the trajectory distance as

$$d(X, Y) = \frac{\|X - Y\|_2}{\sqrt{12H}}, \tag{15}$$

where each trajectory stacks H future frames, 12 keypoints, and three coordinates. This gives an RMS-style whole-trajectory distance in meters and differs from Avg, which averages per-keypoint Euclidean distances. For a deterministic forecast, ES reduces to $d(X, Y)$. The fair estimator excludes self-pairs; lower is better.

B.3 Observed Geometry

All methods are evaluated on 61 observed frames. Baselines process the full sequence jointly, while our decoder uses overlapping windows spanning frames 0-28, 16-44, and 32-60. Hand outputs are cross-faded across overlapping windows; camera estimates use the first and last windows. State-conditioned rows receive observed states at frames 28, 44, and 60.

Following ACE-Ego-Hand, metrics with the suffix "-p" penalize missed on-screen hands by replacing them with a canonical hand at the camera origin, while +OOS additionally scores out-of-view hands. Image-plane EPE assigns the image diagonal to missed detections, whereas $\mathrm{EPE}^{\mathrm{det}}$ evaluates detected hands only. Temporal jitter is reported in mm/frame^2.

B.4 Camera Metrics

Camera trajectories are expressed relative to frame 60. ATE uses rigid alignment, ATE-S additionally estimates a global scale, and RPE measures frame-to-frame drift. Scale is summarized as the median

fitted factor within each corpus, and the overall value is the mean of the four corpus medians; a value of one indicates correct metric scale. Camera evaluation excludes EgoDex and VITRA because their camera trajectories are estimator-derived rather than annotated.

B.5 Observed Proximity

Observed proximity labels are computed from annotated hand geometry and object surfaces using a 2 cm threshold. HACO predicts dense hand-surface contact; we pool its vertex probabilities into 21 joint regions per hand for comparison with our joint-wise output. Pooling rules and decision thresholds are fixed using a disjoint calibration split.

B.6 Future Proximity and Events

Future proximity labels are computed from mean-shape MANO joint centers using the same 2 cm threshold to object surfaces. We evaluate two forecast seeds independently and average their scores over frames $t + 1$ through $t + 32$ at origins $t \in \{28, 36, 44\}$. A hand-frame is included only when labels are available and the hand is inside the image.

A hand is considered in contact when at least one joint is within the proximity threshold. We reduce the predicted joint-wise probabilities to a hand-level score by taking the maximum over joints. Binary metrics use a threshold of 0.5, while AP uses the continuous hand-level score. Hand-frames are pooled within each corpus before corpus-level aggregation.

For event evaluation, onset is defined for hands that are initially out of contact and release for hands that are initially in contact. The first predicted transition within the horizon defines the predicted event time. Precision and recall evaluate whether an event is detected; timing error is the absolute difference between predicted and recorded event times and is computed only when both occur. Missed events therefore reduce recall rather than contributing a timing error.

The recorded-state persistence baseline keeps the ground-truth current contact state fixed over the horizon, while estimated-state persistence holds the model's estimated current state fixed. Confidence intervals use 1000 recording-level bootstrap samples within each corpus. These labels measure geometric proximity rather than physical contact force or grasp stability.

C Observed-State Reconstruction

Although our primary task is future interaction forecasting, the same decoder also produces structured states for observed frames. We evaluate this observed-state readout to verify its geometric and proximity predictions independently of forecasting. We compare articulated hand and camera recovery against video-based reconstruction methods, and joint-wise proximity against a dedicated contact estimator. Evaluation protocols and metric definitions are provided in Appendix B, and extended qualitative comparisons appear in Appendix F.

C.1 Hand and Camera Recovery

Table 6 evaluates hand and camera reconstruction over 61 observed frames. Hand metrics cover detection, articulated pose, image-plane and metric placement, and temporal stability, while camera metrics evaluate recovered viewpoint motion. Full metric definitions and handling of missed or out-of-view hands are provided in Appendix B.3.

C.2 Observed Joint Proximity

Table 7 evaluates the decoder's joint-wise proximity output on observed video using annotated hand and object geometry. Our readout substantially outperforms HACO (Jung & Lee, 2025) in joint-level AP across the evaluation corpora. Unlike the future-proximity evaluation, these labels use annotated hand geometry directly rather than mean-shape MANO joints.

Table 6: **Observed-video reconstruction.** Hand and camera recovery on shared held-out ARCTIC s05 and four-corpus means. K: intrinsics; S: supplied hand state (grey rows, excluded from bolding).

Method	Input	F1↑	3D pose (mm)			Placement			Jitter↓	Camera		
			MPJPE-p ↓	PA-p ↓	+OOS ↓	EPE px↓	GO deg↓	CT cm↓		Rot deg↓	Tr cm↓	ATE-S cm↓
ARCTIC s05: held out by every method												
HaWoR	K	96.3	32.6	7.4	71.4	38.2	24.8	7.2	20.79	**1.4**	26.4	2.3
Dyn-HaMR	K	90.2	42.3	10.9	106.9	86.4	32.5	20.7	25.56	5.8	38.6	3.2
ACE-Ego-Hand	K	**100.0**	**12.5**	**6.6**	**21.5**	**10.6**	**10.6**	4.9	**4.51**	–	–	–
Ours	K	97.6	17.4	7.6	24.9	29.8	15.9	**4.2**	7.49	2.2	**2.3**	**0.9**
Ours + state	K+S	97.6	13.8	6.2	22.2	29.5	12.1	3.8	7.79	2.3	2.3	0.9
Mean over four motion-capture corpora												
HaWoR	K	97.2	25.8	7.9	38.5	35.9	19.6	7.0	16.20	**1.1**	9.8	0.9
ACE-Ego-Hand	K	**100.0**	**13.0**	**6.5**	**15.8**	**8.6**	**8.8**	4.0	**2.69**	–	–	–
Ours	K	98.8	15.0	**6.5**	17.8	21.2	12.2	**3.2**	5.38	1.6	**1.9**	**0.7**
Ours + state	K+S	98.8	11.1	4.9	14.0	20.2	8.8	2.9	5.42	1.6	1.9	0.7

Table 7: **Observed joint proximity.** Joint-level AP using true-geometry proximity labels.

Method	ARCTIC	Mean
Chance	0.23	0.23
HACO	0.72	0.58
Ours	**0.85**	**0.89**

Table 8: **Generator adaptation.** Paired comparison of the stock and ego-adapted generators with the structured decoder fixed: state-relative single-forecast errors in cm over 291 clips at the 1.07 s horizon. ΔFinal is the paired ego−stock endpoint difference with a 95% recording-level bootstrap interval.

Readout	Generator	Avg↓	Final↓	ΔFinal [95% CI]
First	stock	7.49	12.37	–
First	ego	7.48	12.38	$+0.00\ [-0.20, +0.20]$
Final	stock	9.81	16.58	–
Final	ego	8.54	14.37	$-2.21\ [-2.58, -1.80]$

D Forecasting Analyses

D.1 Readout Selection

We select the first denoising readout for quantitative forecasting using a separate set of EgoDex clips outside the main evaluation. On this selection set, the first readout gives the lowest state-relative endpoint error among the evaluated readouts. Table 4 shows the same trend on the main evaluation: the first readout gives the strongest single-forecast hand accuracy and future-contact prediction, while later readouts achieve lower ES. We therefore use the first readout for quantitative forecasting. Streaming visualization and simulated execution use the final readout so that the structured trajectory is aligned with the completed video rollout.

D.2 Generator Adaptation

We evaluate the effect of the egocentric LoRA while keeping the structured decoder fixed (Table 8). Adaptation leaves the first readout used for quantitative forecasting essentially unchanged, while substantially improving the final readout. This suggests that egocentric adaptation primarily affects the generated future as denoising progresses rather than the earliest predictive representation consumed by our default forecaster.

Table 9: **Forecasting by motion magnitude.** State-relative first readout with the ego-adapted generator, binned by recorded 32-frame wrist displacement. Distances are in cm; cosine scores motion direction, and Final wins compare endpoint error against persistence. N counts correlated hand-origin pairs.

Motion bin	N	Mean wrist motion		Cosine	Avg error↓		Final wins (%)
		Recorded	Predicted		Ours	Persistence	
< 3	439	1.2	4.5	—	3.56	**1.50**	14
3-10	555	6.2	6.0	0.44	**4.94**	5.51	59
10-20	459	14.5	8.6	0.61	**7.80**	10.49	82
> 20	293	29.3	12.4	0.64	**13.60**	18.81	85

Table 10: **Dependence on predictive forecast features.** Forecast-block features are replaced with features from another clip while observation features, state anchors, and intrinsics remain unchanged. Errors are in cm on 80 clips, two forecast origins, and two seeds.

Readout	Forecast features	Frame $+1 \downarrow$	$\text{Avg}_{16} \downarrow$	$\text{Avg}_{32} \downarrow$
First	matched clip	**1.6**	**5.0**	**8.2**
First	other clip	10.4	11.2	12.9
Final	matched clip	**1.7**	**6.4**	**10.6**
Final	other clip	12.4	15.2	17.6

D.3 Motion Magnitude and Persistence

Table 9 analyzes forecasting performance as a function of recorded wrist displacement. Persistence is strongest for nearly stationary hands, while our forecast becomes increasingly advantageous as motion magnitude grows. For the largest motions, the model still tends to underestimate displacement, indicating a conservative motion bias. We also evaluate a damped constant-velocity baseline tuned on EgoDex; the selected velocity weight is zero, reducing it to persistence.

D.4 Impact of Forecast Features

We test whether the structured decoder uses the forecast-block representation by replacing forecast features with layout-compatible features from another clip while keeping the observation features, state anchors, and intrinsics fixed. Table 10 shows that mismatched forecast features substantially degrade prediction at both the first and final readouts. This confirms that the decoder relies on clip-matched forecast features rather than only on the observed context and supplied states.

E Streaming and Runtime

E.1 Observation Updates

Each streaming update forecasts 32 future frames from the most recent 29 observed frames. Updates occur after eight new observations arrive, so the next context retains 21 frames from the previous window and appends eight new frames. The current implementation re-encodes the full observation window at every update. Generated future frames are replaced with newly observed RGB frames and available hand and camera states which are then used to advance the forecast.

E.2 Lightweight Video Autoencoder

For streaming forecast, we replace the full video autoencoder with the 11.4M-parameter `lighttaew2_2` tiny autoencoder (TAE) (LightX2V Team, 2026). It uses 48 latent channels, $16\times$ spatial compression, $4\times$ temporal compression, and Wan's channel normalization. Its encoder preserves the latent format expected by the video backbone, allowing the generator and 4D state decoder to remain unchanged. The TAE also accelerates optional RGB decoding for visualization, although structured hand and proximity readout does not require decoded video. Quantitative evaluations use the full video autoencoder. TAE is used only for the streaming deployment.

Table 11: **Live model request runtime.** Mean seconds per update over 48 eager requests during simulated execution, including the first. RGB display decoding is optional; controller and simulator time are excluded.

Component	Mean time (s)
TAE encoding: 29 observed frames	0.008
Video generation: context, four steps, cache update	0.802
Hand/proximity decoding	0.123
TAE RGB display decoding	0.018
Request overhead	0.076
Model request total	**1.027**

Table 12: **Runtime ablations.** Seconds at 352×640 on an RTX PRO 6000 Blackwell. (a) Component means over 15 calls. (b) Warm in-process means over nine calls with TAE. Requests include hand and RGB decoding; startup is excluded.

(a) Video autoencoder

Operation	VAE	TAE
Context encoding	0.915	**0.008**
RGB display decoding	5.604	**0.028**

(b) Backbone compilation with TAE

Measurement	Eager	Compiled
Video generation	0.818	**0.488**
Model request	0.990	**0.659**

E.3 Model Runtime

Table 11 reports the runtime of the model request during simulated execution at BF16 and 352×640 resolution on an RTX PRO 6000 Blackwell. The request includes observation encoding, video generation, structured decoding, optional RGB decoding, and request overhead.

Table 12 separately measures the effects of the lightweight autoencoder and backbone compilation. The TAE substantially reduces observation encoding and RGB decoding time, while compilation reduces the cost of the video-generation request. Even with these optimizations, model latency remains above the eight-frame update interval. We therefore use "streaming" to denote causal forecast updates from incoming observations rather than real-time throughput.

E.4 Simulated Execution

We demonstrate one way to consume the structured forecast in closed loop using MuJoCo (Todorov et al., 2012). A calibrated fixed camera observes an articulated MANO hand with a floating wrist and 45 finger coordinates. At each update, the forecast provides a state-relative hand-motion reference over the 32-frame horizon. Model predictive control (MPC) optimizes bounded wrist and finger residuals around this reference, executes the first eight frames, and returns the resulting rendered observations and executed states to the next forecasting step.

Each simulated candidate is scored by

$$J = J_{\text{model}} + J_{\text{track}} + J_{\text{contact}} + J_{\text{smooth}} + J_{\text{penetration}} + J_{\text{force}}. \tag{16}$$

J_{model} penalizes deviations from the decoded trajectory, while J_{track} measures tracking under simulated dynamics. J_{smooth} regularizes changes in the control residual, and $J_{\text{penetration}}$ and J_{force} discourage undesirable contact.

Predicted joint-wise proximity enters the contact term through the per-finger score $p_{f,d} = \max_{j \in d} c_{f,j}$. For finger-to-object surface gap $g_{f,d}$,

$$J_{\text{contact}} = \frac{w_c}{H} \sum_{f=1}^{H} \frac{\sum_d p_{f,d} \left(\sqrt{1 + (g_{f,d}^{+}/d_0)^2} - 1 \right)}{\max(1, \sum_d p_{f,d})}, \tag{17}$$

where $g^{+} = \max(g, 0)$. We use $w_c = 24$ and $d_0 = 4\,\text{cm}$. The controller uses simulator geometry and dynamics, but does not use a target object pose, lift reward, or scripted grasp trajectory. This controller is specific to the simulation demonstration and is not required for streaming or recorded video forecasting.

F Additional Qualitative Results

F.1 Reconstruction Comparison

Figures 7 and 8 extend the observed-state reconstruction comparison of Figure 4 across all four evaluation corpora and an in-the-wild VITRA clip. For each clip, we show projected hand overlays together with 3D views containing the recovered hands, camera frusta, and wrist trajectories. Our decoder additionally predicts joint-wise proximity (yellow), which the reconstruction baselines do not provide.

F.2 Forecast Comparison

Figure 9 extends the forecasting comparison of Figure 3 with additional sequences. Each block shows the input frame and the supplied current hand state, forecasts at three horizons for our one- and 29-frame configurations, VITRA, Being-H0, and persistence with ground-truth outlines overlaid, and a 3D overlay of all methods. VITRA is evaluated only through $t+16$.

F.3 Simulated Execution

Figure 10 shows three additional closed-loop episodes with the forecast and the executed simulator state at selected updates. Contact-guided MPC and the execution protocol are described in Appendix E.4.

G Limitations and Future Work

Forecasting. Our structured forecast is ultimately bounded by the predictive quality of the underlying video model. Errors, hallucinations, or prompt non-adherence in the predicted future can propagate to the decoded hand motion, camera motion, and proximity cues. Proximity supervision is derived largely from motion-capture corpora with rigid object geometry, which may limit generalization to thin, deformable, or geometrically ambiguous objects. Extending the structured state with explicit object tracking, together with supervision from more diverse object geometries and interaction types, is a natural direction for future work.

Streaming and efficiency. Our use of "streaming" refers to causal forecast updates as new observations arrive rather than real-time execution. Each update currently re-encodes the observation window and generates a new future block, so model latency remains longer than the observation-update interval.

Interaction and downstream control. The structured forecast predicts geometric proximity rather than physical contact, and therefore does not directly model forces, grasp stability, or object dynamics. This is reflected in weaker release prediction and limits how directly the proximity cue can be used for manipulation. The simulated controller also relies on privileged geometry and dynamics and is intended only to demonstrate how predicted motion and proximity can be consumed downstream. Extending the representation toward richer contact and object dynamics, and evaluating closed-loop execution on physical systems, are natural directions for future work.

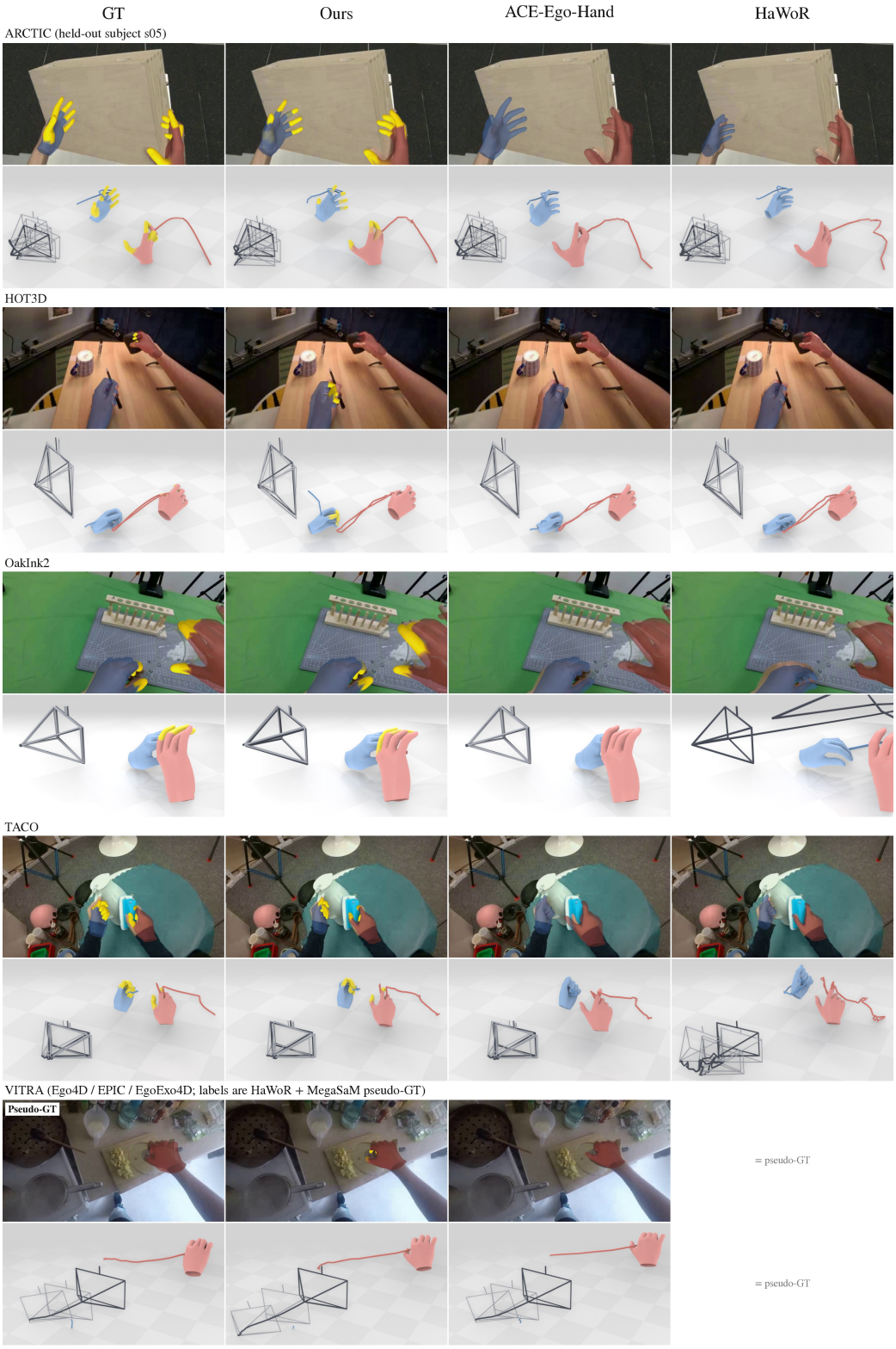


Figure 7: **Observed-state reconstruction comparisons.** Each corpus block shows image overlays (top) and 3D views with camera frusta and wrist trajectories (bottom) for GT, our decoder, ACE-Ego-Hand, and HaWoR. Yellow joints mark ground-truth and predicted proximity; among the compared methods, only our decoder predicts joint-wise proximity alongside geometry.

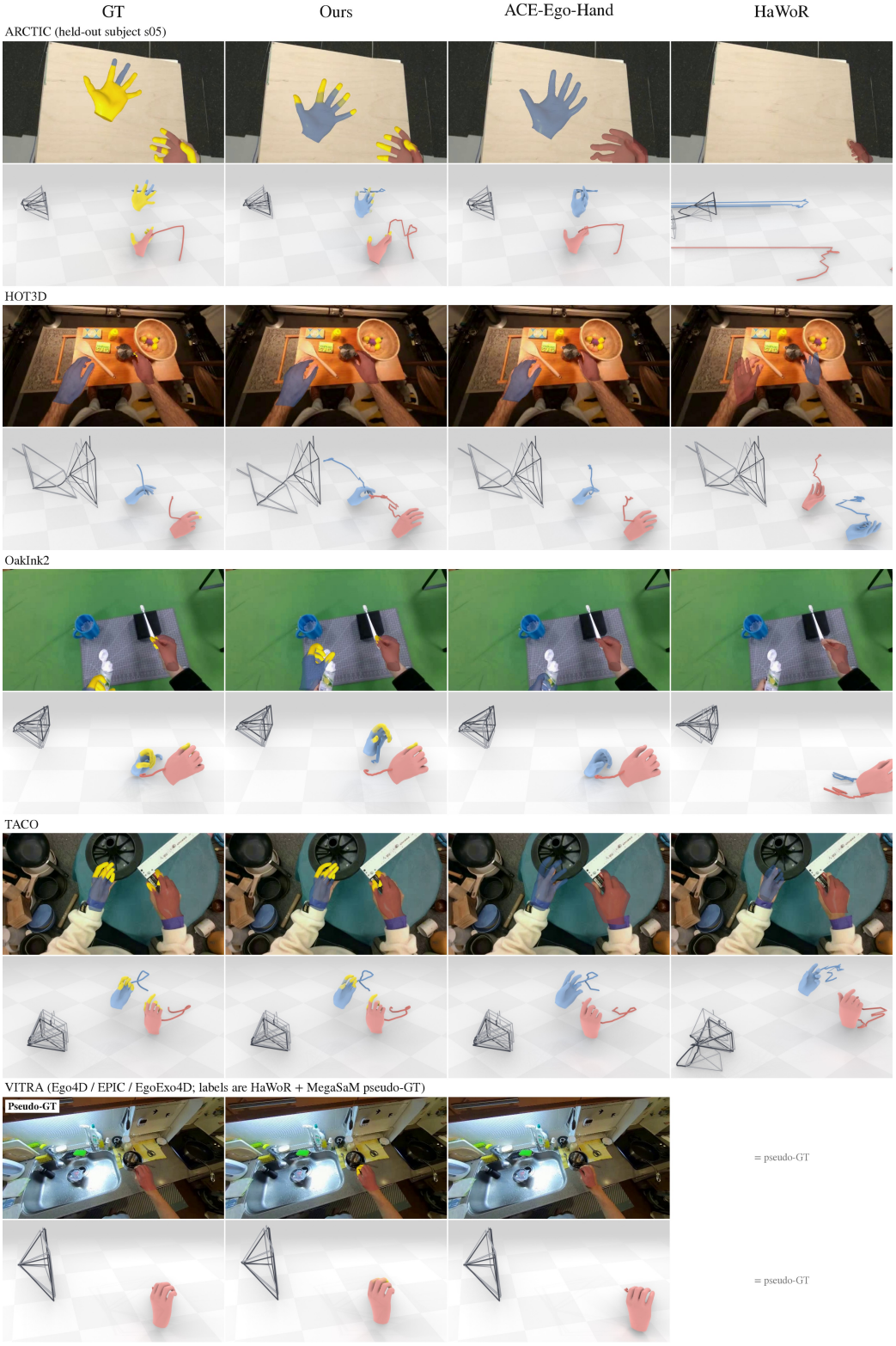


Figure 8: **Observed-state reconstruction comparisons.** Each corpus block shows image overlays (top) and 3D views with camera frusta and wrist trajectories (bottom) for GT, our decoder, ACE-Ego-Hand, and HaWoR. Yellow joints mark ground-truth and predicted proximity; among the compared methods, only our decoder predicts joint-wise proximity alongside geometry.



Figure 9: **Hand articulation forecasting comparisons.** Each block shows the input frame and supplied current hand state (left), forecasts at $t{+}8$, $t{+}16$, and $t{+}32$ for our one- and 29-frame configurations, VITRA, Being-H0, and persistence with ground-truth outlines overlaid, and a 3D overlay of all methods (right). VITRA is evaluated only through $t{+}16$.

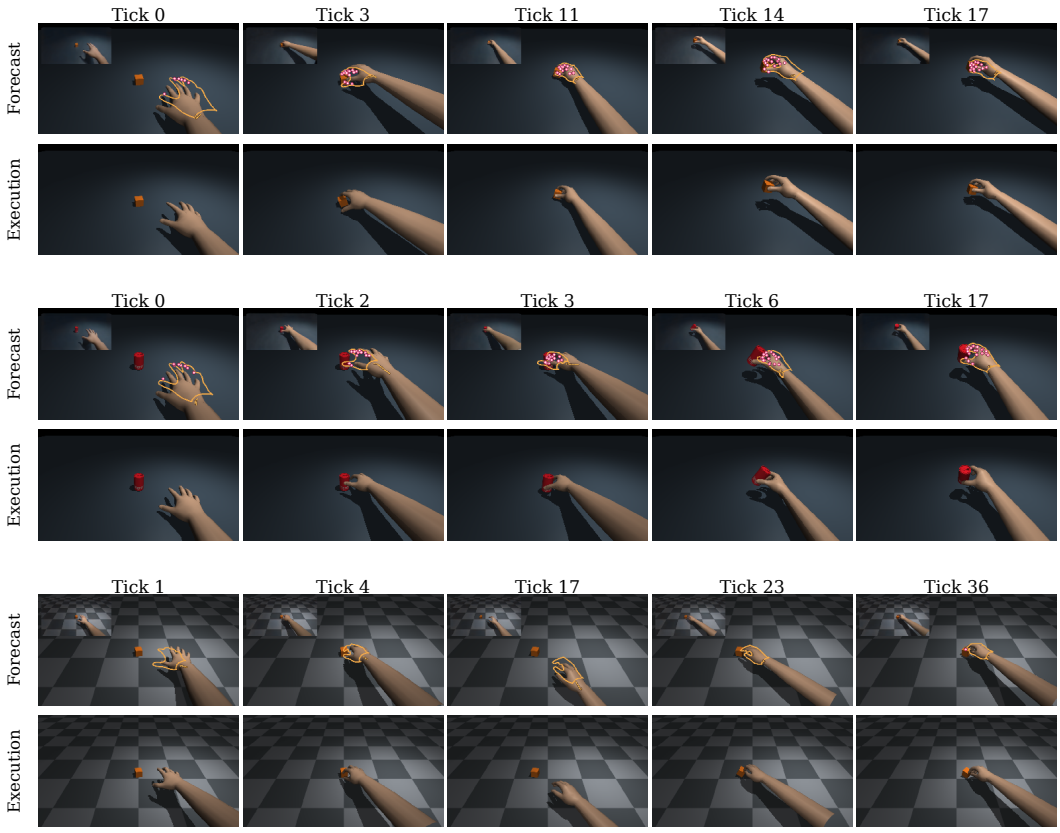


Figure 10: **Additional simulated-execution episodes.** Three episodes, each showing the structured forecast (top) and the executed simulator state (bottom) at selected updates. At each update, contact-guided MPC refines the forecast reference, the simulator executes an eight-frame prefix, and the rendered observations and executed states condition the next forecast.