MotionForesight

Re-purposing Video Models for Future 3D Scene-Flow Prediction

Homanga Bharadhwaj* Yash Jangir*
(* authors contributed equally)
Department of Computer Science, Johns Hopkins University
hbharad2@jhu.edu yjangir@jhu.edu

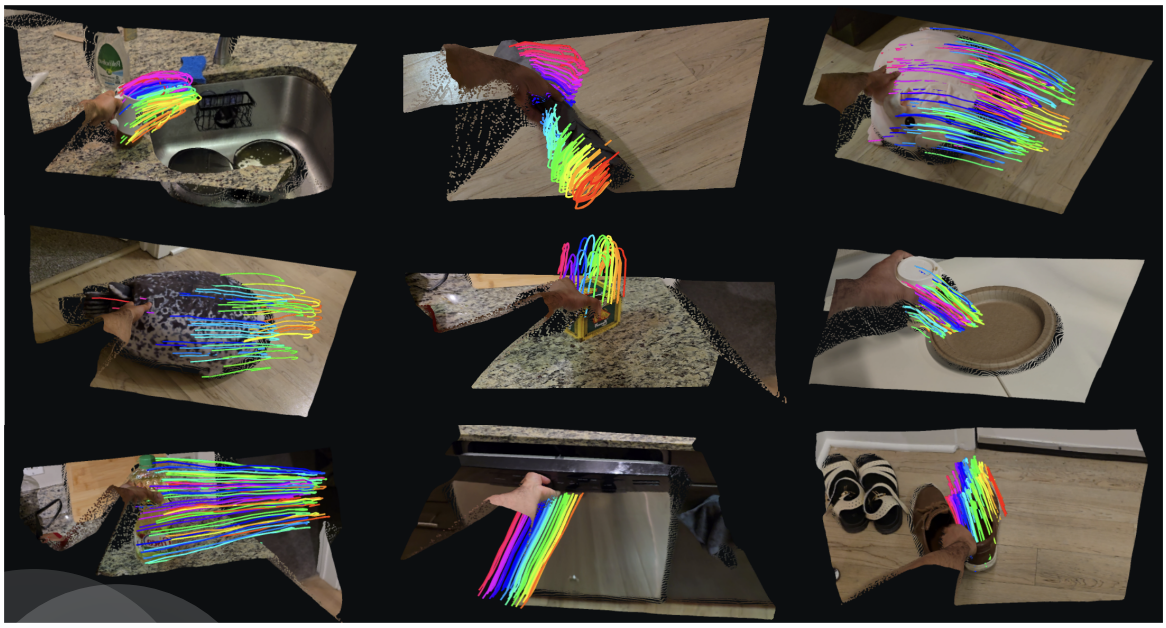

Figure 1: MotionForesight forecasts plausible motion in diverse everyday manipulation scenarios

Abstract. Humans can infer how objects are likely to move from passive observation: a cup may be lifted, a drawer may slide, and a lid may rotate shut. Such predictions expose the physical consequences of interaction needed to act in the real world. We study how to learn this anticipation from ordinary monocular videos of human-object interaction. Given a short observed video context, MotionForesight predicts future 3D trajectories for points on the manipulated object. This casts interaction prediction as object-centered 3D motion forecasting without any assumptions on the object properties. Our key insight is that video prediction models already encode rich priors about how objects move during human interactions. We redirect these priors from pixel prediction toward future 3D scene flow. We start from a dense 3D tracker built on a pretrained video model, generate pseudo-ground-truth tracks from complete clips, and train the forecaster using only the observed frames. We replace future RGB and geometry with learned mask latents and train a lightweight adapter to turn the retrospective tracking representation into a forward predictor, while freezing the large video and tracking components. Using just 40K human videos and no auxiliary inputs such as language, MotionForesight generalizes across diverse out-of-distribution objects, environments, viewpoints, and interactions. It also outperforms substantially larger models that use over a million training videos. These results show that we can efficiently re-purpose video priors into explicit geometric forecasts for embodied intelligence. motionforesight.github.io

1 Introduction

"One of the most fundamental properties of thought is its power of predicting events."
—Kenneth Craik, *The Nature of Explanation*

A central component of embodied intelligence is not only recognizing what has happened, but *anticipating what will happen next*. When people watch a hand approach a mug, a knife press into a fruit, or a cabinet door begin to move, they often infer the likely future configuration of the object before the motion is complete. This ability goes beyond visual prediction by reflecting an understanding of object affordances, contact, constraints, and goals. Studies in cognitive science have long emphasized that perception is tied to action possibilities [1], that motion supports structured scene interpretation [2], and that humans often reason by mentally simulating physical futures [3]. The classic rational-imitation result of Gergely et al. [4] further suggests that infants do not simply do as we do, but interpret observed actions in light of goals and constraints. **Modeling interaction effects** is therefore crucial: what matters for observational learning is how the object will move given a specific context, not the exact hand motion that produced it.

In this paper, we tackle the problem of *forecasting motion* in the form of *future 3D scene flow* for manipulated objects from casual monocular human videos. 3D scene flow provides a general representation for inferring motion in scenes: instead of committing to a category-specific state such as a rigid pose, it describes how points in the 3D scene move over time. Our setting differs from related directions along two main axes: the information used to specify the future and the representation used to describe motion. Some methods rely on language or action grounding to predict either sparse 3D point trajectories [5] or rigid 6-DoF object-manipulation trajectories [6]. Others predict future rigid 6-DoF object poses directly from visual observations [7]. Although pose representations are compact for rigid objects, they cannot naturally describe articulated parts, deformable surfaces, or local nonrigid motion. Some prior works predict future 2D point tracks for robot manipulation [8–10], whereas our target is explicit future metric 3D motion rather than image-plane tracks in a downstream control setting. We focus on dense, reference-anchored 3D scene flow because it provides a general interface for interaction dynamics: it is metric rather than image-plane, object-centered rather than embodiment-specific, and not restricted to a single rigid pose parameterization. In principle, the same representation can describe rigid objects, articulated parts, deformable surfaces, and local object motion. *Learning this prediction from everyday videos is therefore a compelling problem*: casual videos are noisy and unconstrained, but they contain broad evidence about how objects move when people use them.

Our key insight is that everyday human videos already describe the motion of common objects we want to predict, while modern video models are likely to encode useful priors about how such motion unfolds. A model trained on large video corpora must represent temporal regularities such as contact, affordance, object persistence, and plausible short-horizon dynamics, even if its native output is a future image or video latent. We ask whether these priors can be redirected toward a more explicit geometric target. Rather than generating future pixels and recovering motion afterward, MotionForesight directly predicts the future 3D trajectory field for object-attached points. The model uses the observed context to infer the likely evolution of the interaction, but its output remains a compact, metric motion representation that can be consumed by downstream planning or embodied reasoning systems. This direct geometric prediction is also computationally attractive for dynamic robot manipulation: it avoids spending inference time rendering future RGB frames and then running a separate tracking or reconstruction pipeline to recover the motion that the robot actually needs. By predicting future 3D motion directly, the model exposes an actionable representation with lower overhead than a generate-then-track alternative.

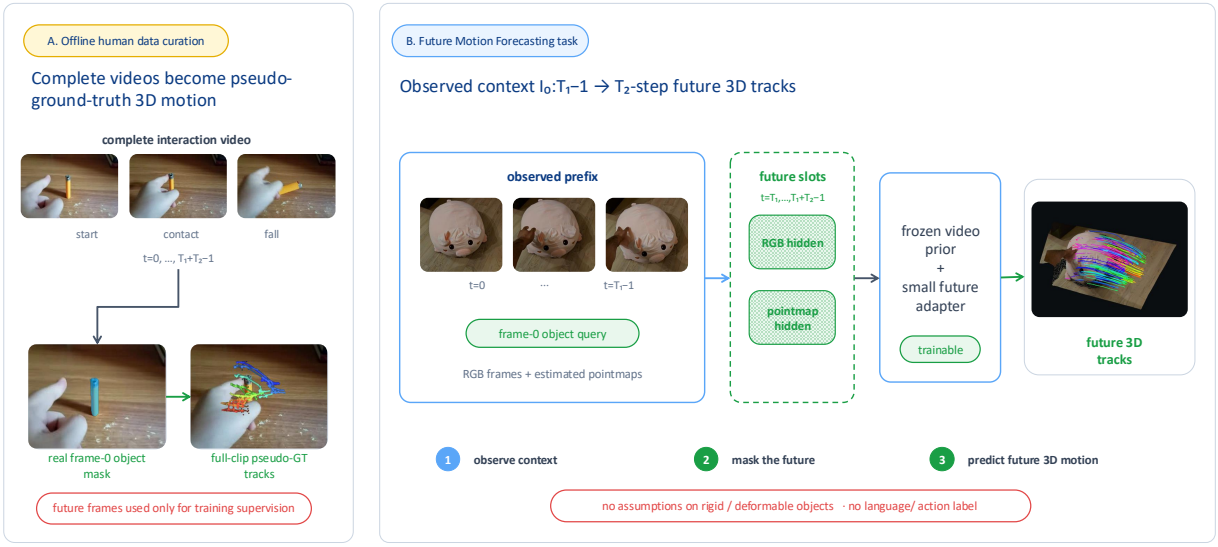

Figure 2: MotionForesight predicts future reference-anchored 3D tracks from passive human video. Given observed RGB frames and pointmaps obtained through estimated monocular depth, for the first T_1 frames, the model forecasts the future motion of the manipulated object for the next T_2 steps. The output is an explicit 3D trajectory field over points on the manipulated object and does not require a rigid-body or category-specific motion parameterization.

Concretely, MotionForesight repurposes TrackCraft3R [11], a feed-forward dense 3D tracker built around a video DiT [12]. TrackCraft3R is retrospective: it observes all frames, encodes RGB frames and reconstructed pointmaps into geometry latents, repeats a first-frame query latent across time, and predicts reference-anchored tracking pointmaps. We convert this tracker into a forecasting model by hiding the future frames. For timestamps after the observed prefix, the unavailable RGB and pointmap latents are replaced by learned mask latents, while the frame-0 query latent and temporal RoPE interface are preserved. The transformer therefore receives observed 3D context, a reference-frame object query, and future time indices, and it predicts residual track latents that decode into future 3D tracks. We keep the base video model, the original TrackCraft3R adapter, and the VAE encoders/decoders frozen, and train only a fresh low-rank adapter, I/O projections, the prediction head, and the mask latents. Pseudo-ground-truth future tracks are generated by running dense 3D tracking on complete human interaction clips, then training the model with those future observations removed.

Key point. MotionForesight uses passive monocular human videos to learn *object-centered future 3D motion*. This formulation abstracts away from the human's motor command and from the appearance of future pixels by focusing on how points on the manipulated object are likely to move.

In summary, we claim three contributions. First, we formalize future 3D scene-flow prediction from casual RGB interaction videos as a reference-anchored tracking-pointmap forecasting task. Second, we present a data curation pipeline that converts human videos into pseudo-ground-truth future tracks using dense 3D tracking and object masks, while masking future frames at training time. Third, we introduce a minimal modification to TrackCraft3R [11] that re-purposes it from a point track extraction model to a future point track prediction model. We show that this surprisingly simple and compute-efficient recipe trained on just 40k RGB human videos achieves compelling motion forecasting results in generic real-world scenes.

2 Related Work

Visual forecasting and world models. Building models of how the visual world evolves has long been a central problem in computer vision and robotics. Prior work predicts future pixels or latent states for planning, control, and representation learning [13–16]. More recent approaches scale this idea through video diffusion, predictive visual representations, and learned latent-action models [17–23]. These methods capture useful temporal regularities, but their predictions are typically represented as RGB frames, video features, or action-conditioned latent states without explicit object-level geometry. Our work builds on the temporal priors learned by video models but differs in its output representation. Rather than generating future pixels or abstract features, MotionForesight predicts the future metric 3D motion of points on the manipulated object.

Inferring interaction cues from human videos. Large-scale datasets such as Something-Something [24], YouCook [25], EPIC-Kitchens [26], EGTEA [27], and Ego4D [28] have enabled models to learn human–object interaction priors directly from video. One line of work predicts future action labels, active objects, contact regions, hand trajectories, and interaction hotspots [29–36]. Other work uses large-scale activity and contact observations to learn how people interact with objects [37, 38]. More recent methods forecast language- or context-conditioned hand trajectories [39, 40], while robot-learning approaches predict 2D point tracks, masks, or visual tokens as embodiment-agnostic plans [8, 9, 41]. These methods extract actionable cues from human videos, but their predictions largely remain semantic, hand-centric, or tied to the image plane. MotionForesight instead predicts continuous 3D trajectories for points in the scene, directly representing the physical consequence of an interaction without requiring language/action labels.

3D geometric extraction and forecasting. Another line of work models interaction through explicit geometry. Early approaches estimate 3D hand and object poses from visual observations [42–46] or recover 6-DoF object pose from RGB and RGB-D inputs [47–50]. Recent systems provide complementary tools for video segmentation and single-image 3D reconstruction: SAM 2 [51] propagates masks through video, while TRELLIS [52] and SAM 3D [53] reconstruct 3D geometry from images. Future geometry prediction has also been studied through LiDAR, point-cloud, and agent-trajectory forecasting in autonomous-driving and mobile-robot settings [54–56]. Closer to manipulation, methods predict language-conditioned 6-DoF trajectories [6], transfer human trajectories to robot embodiments [57], or forecast object poses, point motion, and scene flow under language, goal, RGB-D, or robot-action conditioning [7, 58, 59]. In contrast, MotionForesight predicts dense, reference-anchored dense 3D tracks from monocular RGB videos, allowing the same representation to describe rigid, articulated, and deformable motion. Concurrent with our work, MolmoMotion predicts a closely related output: future metric 3D trajectories, but formulates goal-conditioned forecasting from a short RGB history and a language description of the intended action, and predicts the motion of just 8 points on an object [5]. It is trained on MolmoMotion-1M, an action-described, object-grounded corpus constructed from 1.16M diverse videos, whereas MotionForesight is trained on 40K monocular SSv2 human-interaction videos and receives no language or action input. Thus, the two works share a point-trajectory output representation but study distinct supervision regimes: language-grounded intent from a broad video corpus for sparse points versus passive visual anticipation from observed interaction for dense 3D flow forecasting.

Scene flow and point tracking. Scene flow and point tracking provide the motion representation underlying our approach. Classical and learned scene-flow methods estimate dense 3D motion between observed frames, while recent trackers recover long-range correspondences through camera motion, occlusion, and nonrigid deformation [11, 60–65]. These methods are primarily retrospective:

they estimate where points moved after observing the relevant frames. We use this capability both to construct pseudo-ground-truth supervision and to define the output interface of our model. We then turn retrospective tracking into forecasting by hiding future observations and asking a tracking-adapted video backbone to predict where reference-frame object points will move before those frames become available.

3 MotionForesight

MotionForesight aims to predict the future motion of objects being manipulated in a scene from a short video context. Toward this goal, we pose a geometric version of the video-prediction problem. Instead of predicting future images, we predict the future motion of object points that are already visible during the observed interaction. This choice is motivated by embodied settings, where a predictive model must capture changes in scene geometry more precisely than changes in texture or appearance. The physical consequence of an interaction is often better represented by where an object moves in 3D than by how the subsequent RGB frames look.

Concretely, given T_1 observed RGB frames, we predict the future 3D tracks of points on the manipulated object for the next T_2 time steps. At inference, we observe only the video prefix $I_{0:T_1-1}$. We estimate a pointmap $P_t \in \mathbb{R}^{H\times W\times 3}$ for each observed frame and sample query points $\mathcal{Q}$ from the object mask in the reference frame $a = 0$. Our goal is to predict the future 3D locations of these reference-frame points:

$$\hat{X}_t(q) \in \mathbb{R}^3, \qquad q \in \mathcal{Q}, \quad t = T_1, \ldots, T_1 + T_2 - 1. \tag{1}$$

During training, we have access to videos of length $T = T_1 + T_2$. We use the complete videos only offline to extract pseudo-ground-truth 3D tracks; the forecasting model itself receives only the first T_1 RGB frames and their estimated geometry. We express the observed geometry and future tracks in the last-observed camera frame, $t = T_1 - 1$, to reduce apparent motion caused by the camera. Our default setting uses $T_1 = 7$, $T_2 = 15$, and $T = 22$. The model receives no auxiliary inputs like language instructions or action labels.

3.1 Data curation: from RGB human videos to 3D object trajectories

We construct pseudo-ground-truth supervision from approximately 40K human-object interaction videos from the Something-Something V2 dataset. Because these videos do not provide metric 3D trajectories or temporally consistent object masks, we process each clip using an offline segmentation, reconstruction, and tracking pipeline.

The manipulated object is not always visible in the first frame, so we select an intermediate anchor frame in which it is sufficiently clear. We use the object name from the dataset annotation as a cue for mask extraction, initialize Segment-Anything [66] on the anchor frame, and propagate the resulting mask backward and forward through the clip. Query points are densely sampled within the object mask and retained wherever valid object support is available.

We estimate monocular depth for every frame using DepthAnything3 [67], recover camera motion, and combine both estimates to construct temporally aligned pointmaps. We then run TrackCraft3R [11] on the complete clip to obtain reference-anchored 3D trajectories for the sampled object points. The resulting tracks are transformed into the last-observed camera coordinate frame and stored with their validity masks. The complete clip is used only for offline pseudo-label generation; during training, future RGB frames and pointmaps are never provided to the forecasting model.

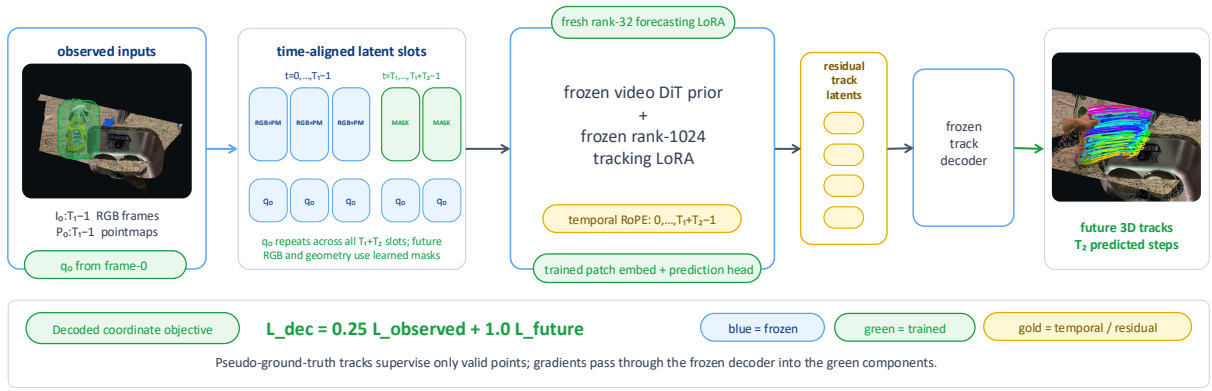

Figure 3: Architecture. We preserve TrackCraft3R's dual-latent construction. Observed frames produce standard RGB and pointmap geometry latents. Future geometry slots are filled by learned mask latents. The frame-0 track/query latent is repeated across all timestamps, and temporal RoPE assigns the target future time. A frozen video DiT with a small fresh LoRA predicts future residual-track latents, which the frozen track decoder converts into future tracking pointmaps.

3.2 Observed visual-geometry context

The first modeling step converts the observed video prefix into the latent sequence consumed by the video transformer. For each observed frame I_t, we pair the RGB image with its estimated pointmap P_t. The RGB and geometry streams are encoded separately and concatenated into a visual-geometry latent,

$$c_t = \left[E^{\mathrm{rgb}}(I_t); E^{\mathrm{pm}}(P_t)\right], \qquad t < T_1. \tag{2}$$

Here, E^{rgb} is the RGB VAE encoder and E^{pm} is the pointmap VAE encoder. Pointmap normalization statistics are computed using only the observed prefix because future geometry is unavailable at inference time.

The two streams provide complementary evidence to the model. The RGB stream captures appearance, contact, and human-object interaction cues, while the pointmap stream provides metric scene structure and observed 3D motion. Together, they allow the transformer to reason about which object is moving, how it is constrained, and how its visible motion is likely to continue into the future.

3.3 A point-track interface for a video prior

A pretrained video model contains a useful temporal prior, but its native output is not a metric 3D trajectory. We therefore start from TrackCraft3R [11], a dense 3D tracking model built on top of a video DiT, Wan2.1 [12]. TrackCraft3R provides an already-learned interface between video latents and reference-anchored 3D point tracks.

For a fully observed tracking clip, TrackCraft3R uses two time-aligned latent streams. The first is the context stream c_t defined above (Section 3.2), which varies with each video frame. The second is a reference, or query stream that repeats the reference-frame state at every timestamp:

$$r_t = \left[E^{\mathrm{rgb}}(I_a); E^{\mathrm{pm}}(P_a)\right], \qquad a = 0, \qquad t = 0, \ldots, T-1. \tag{3}$$

Repeating the query stream turns every temporal output slot into the same geometric question: *where is each reference-frame point at time t?* The video transformer receives the context and query latents together with temporal RoPE and predicts a residual-track latent $\hat{z}_t^{\Delta}$. A frozen track

decoder maps this latent to a 3D residual, which is added to the reference point:

$$\hat{X}_t(q) = X_0(q) + D^{\text{track}}(\hat{z}_t^{\Delta})(q). \tag{4}$$

Each output therefore has a direct geometric interpretation in terms of the predicted 3D position of a reference-frame object point. The pointmaps P_t are obtained by unprojecting per-frame depth and transforming the resulting 3D points into the reference-camera coordinate system. Thus, all geometry inputs share a common frame—an essential property that enables the model to produce reference-anchored tracks. TrackCraft3R uses the video DiT as a single-step latent regressor at a fixed regression timestep rather than as an iterative diffusion sampler; MotionForesight inherits this deterministic single-pass interface when converting tracking into future forecasting.

3.4 Turning tracking into forecasting

The original tracking model is retrospective: it observes the complete video and explains where the reference points moved. We convert it into a forecasting model by hiding all future observations. For observed timestamps, the context stream contains the actual visual-geometry latents,

$$\tilde{c}_t = \left[E^{\text{rgb}}(I_t); E^{\text{pm}}(P_t)\right], \qquad t < T_1. \tag{5}$$

For future timestamps, no RGB image or pointmap is available. We therefore replace both components of the future context stream with learned mask latents,

$$\tilde{c}_t = \left[m^{\text{rgb}}; m^{\text{pm}}\right], \qquad t \geq T_1. \tag{6}$$

The RGB mask latent m^{rgb} and pointmap mask latent m^{pm} are shared across future timestamps and broadcast spatially. They do not encode a particular future appearance or geometry. Instead, they indicate that the corresponding temporal slot is unobserved. Temporal Rotary Position Embedding (RoPE) [68, 69] assigns each slot a distinct time index, allowing the transformer to distinguish near- and long-horizon predictions even though the learned unknown-token content is shared.

We repeat the reference query stream across all T timestamps. Consequently, the model receives the observed visual-geometry context, the same reference-object query at every timestamp, and the temporal index of each requested future. It must then infer where the corresponding object points will move. This formulation directly exposes future 3D scene flow without first generating RGB frames and subsequently applying a separate reconstruction or tracking pipeline.

3.5 Forecasting supervision and objective

The curated full-clip trajectories provide pseudo-ground-truth targets $X_t(q)$, but the forecasting model receives only the observed prefix, the reference query, and the masked future slots. Future RGB frames and future pointmaps are never included in the model input.

The implementation supports both dense pointmap supervision and sparse query-point supervision. We express both using a common query-point interface. For dense clips, object points are sampled from the propagated reference-frame mask and their targets are read from the dense tracking pointmaps. For sparse clips, we use the native tracked points and their associated validity values. Let v_{tq} indicate whether point q has valid supervision at timestamp t.

The default objective is a decoded coordinate-space loss,

$$\mathcal{L}_{\text{dec}} = \lambda_{\text{obs}} \frac{\sum_{t<T_1} \sum_q v_{tq} \left\| \hat{X}_t(q) - X_t(q) \right\|_2^2}{\sum_{t<T_1} \sum_q v_{tq} + \epsilon} + \lambda_{\text{fut}} \frac{\sum_{t\geq T_1} \sum_q v_{tq} \left\| \hat{X}_t(q) - X_t(q) \right\|_2^2}{\sum_{t\geq T_1} \sum_q v_{tq} + \epsilon}. \tag{7}$$

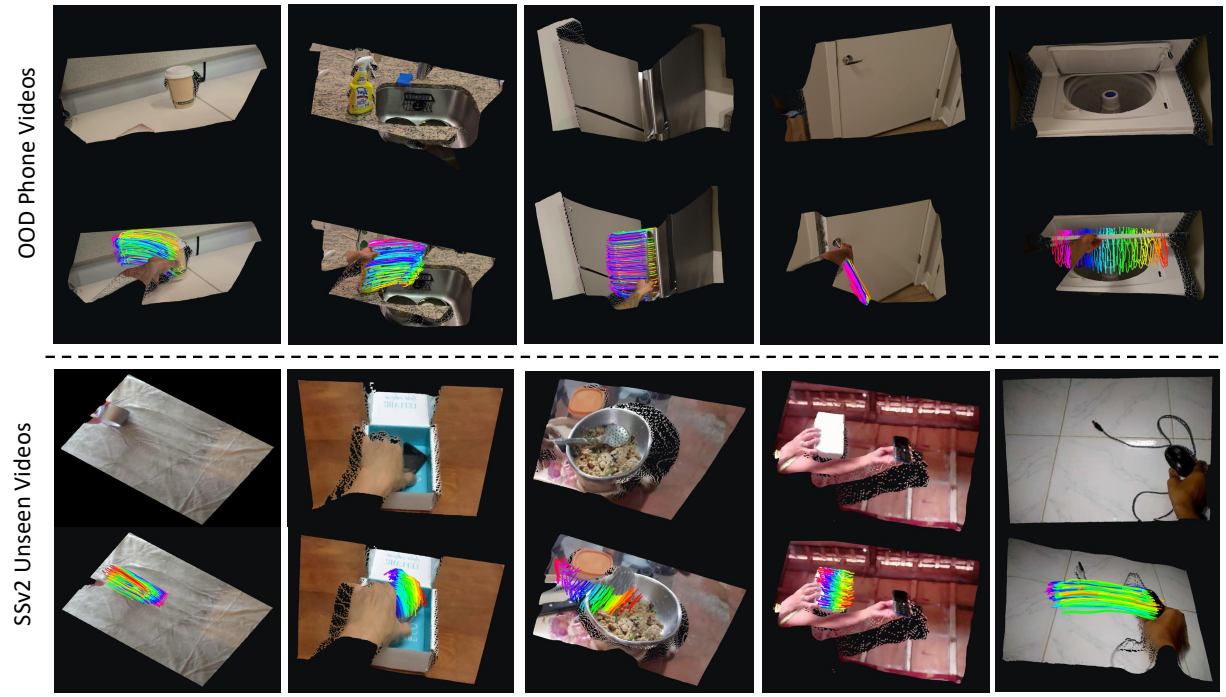

Figure 4: Qualitative results for future 3D track predictions. From only the observed video prefix, MotionForesight predicts plausible object motion including lifting, translation, rotation, constrained sliding, and local nonrigid motion. The visuals show pointmaps of the first and last observed frames, and the future predicted 3D tracks overlaid on the last observed frame. Detailed videos in motionforesight.github.io

We weight future prediction more heavily than observed-frame reconstruction. The observed term primarily stabilizes the inherited tracking interface, while the future term trains the model to extrapolate object motion beyond the available evidence. The loss back-propagates through the frozen track decoder, but the decoder parameters themselves are not updated.

4 Experiments

We organize our experiments around four research questions:

1. **Re-purposing video priors:** Can a pretrained video backbone be effectively adapted for future 3D track forecasting?
2. **Scaling with human video:** How does forecasting improve with more (both quantity and diversity) human-interaction videos?
3. **Out-of-distribution generalization:** How well does the learned forecasting recipe transfer beyond the fine-tuning distribution?
4. **Role of auxiliary supervision:** How does our approach, trained without language or action labels, compare with methods that use such information?

4.1 Datasets and baselines

We train MotionForesight on 40K human-object interaction videos from Something-Something V2 (SSv2) [24]. The model does not receive SSv2 action labels, language instructions, or auxiliary action annotations during either training or inference. Text is used only by the offline preprocessing pipeline, when needed, to identify the manipulated object.

Table 1: Comparison on SSv2 and OOD phone videos. All methods use the same observed interval and prediction horizon. Rows below the horizontal rule receive a ground-truth action description and therefore use more test-time information than MotionForesight. Note that MolmoMotion is trained with 1M videos from diverse sources. MotionForesight is trained with 40k RGB videos from SSv2.

		SSv2 unseen clips (150)			OOD phone videos (50)		
Method	Additional input	ADE ↓	FDE ↓	PWT ↑	ADE ↓	FDE ↓	PWT ↑
MotionForesight (ours)	None	**4.47**	**6.23**	**76**	**9.31**	**14.88**	**54**
MolmoMotion, no language	Null action field	5.66	8.90	70	9.50	16.05	53
Video generation + tracks	None	11.20	12.58	40	13.82	17.65	32
MolmoMotion, with language	Action description	5.93	9.38	68	9.94	17.16	51
Video generation + tracks	Action description	11.99	13.57	44	13.63	16.71	29

We evaluate on 150 held-out SSv2 videos and 50 independently recorded phone videos. The phone videos in home and office scenes contain previously unseen objects, environments, viewpoints, and capture conditions, providing an out-of-distribution (OOD) evaluation. The complete clip is used only offline to construct pseudo-ground-truth trajectories; every forecasting method observes the same first T_1 frames and predicts the following T_2 frames.

All methods are evaluated using the same object query points, future timestamps, validity masks, and coordinate frame. Since MolmoMotion allows forecasting only 8 points at once, we do multiple passes through it. Following prior trajectory-forecasting work [5, 7, 40, 70], we report average displacement error (ADE), final displacement error (FDE), and the percentage of predictions within a fixed distance threshold (PWT). ADE and FDE are reported in centimeters; lower ADE/FDE and higher PWT(@5cm) are better.

4.2 Comparison with baselines

We compare MotionForesight with future video generation (with Wan-VACE) followed by our 3D track-extraction pipeline (inspired by prior works [17, 71, 72]) and with concurrent work MolmoMotion [5]. We evaluate both baselines without language (i.e. a setting similar to MotionForesight) and with a ground-truth action description (i.e. using more proviledged information than MotionForesight). For SSv2 we use the language annotations provided and on phone videos, the language descriptions are manually written.

Table 1 shows that MotionForesight performs best on every SSv2 and OOD metric, despite using neither language nor action labels. With the longer observed context in Table 2, MotionForesight substantially outperforms MolmoMotion: its ADE is 10.20 cm, compared with 11.90 cm without language and 12.57 cm with language, with corresponding gains in FDE and PWT@5cm. This advantage—despite MolmoMotion's much larger, language-paired training corpus [5]—suggests that strong geometric grounding is especially useful for converting richer observed context into metrically consistent future motion. Video generation followed by track extraction performs substantially worse than either geometry-aware method. This supports our central

Table 2: Long-context evaluation on OOD phone videos. The first 50% of each video is observed and the remaining 50% is forecast. All methods share the same split and evaluation inputs; the language-conditioned MolmoMotion variant and the Video generation model additionally receive the ground-truth action description in language where specified.

	OOD phone videos		
Method	ADE ↓	FDE ↓	PWT@5cm ↑
MotionForesight (ours)	**10.20**	**13.59**	**47.4**
MolmoMotion (no lang.)	11.90	16.27	41.3
Videogen + tracks (no lang.)	13.33	15.12	20.9
MolmoMotion (+ lang.)	12.57	18.77	37.1
Videogen + tracks (+ lang.)	12.91	14.55	17.7

Table 3: Scaling with human-interaction videos. We evaluate models trained on increasingly large, action-stratified subsets of SSv2 while holding all other settings fixed.

	SSv2 validation			OOD phone videos		
Training videos	ADE ↓	FDE ↓	PWT ↑	ADE ↓	FDE ↓	PWT ↑
1K	4.81	6.57	74	9.48	14.74	53
10K	4.72	6.38	73	**8.97**	**14.63**	52
40K	**4.47**	**6.23**	**76**	9.31	14.88	**54**

motivation: a pretrained video generator contains useful temporal priors, but visually plausible future pixels do not necessarily preserve point correspondence or metric 3D motion. Among methods that predict geometry explicitly, MotionForesight benefits from reasoning jointly over observed RGB, reconstructed pointmaps, and a tracking-adapted video representation, preserving scene structure while exploiting the backbone's motion prior.

Note that ADE, FDE, and PWT compare a prediction with one realized future, but interaction forecasting is inherently multimodal, and there could be multiple plausible future trajectories. A predicted forecast can therefore disagree with the recorded ground truth while remaining physically and semantically plausible. We consequently treat these metrics as complementary rather than exhaustive and provide side-by-side qualitative comparisons in Appendix Fig. 5 and the website.

4.3 Scaling with human-interaction videos

We train the same model on nested, action-template-stratified subsets containing 1K, 10K, and 40K SSv2 videos. Architecture, initialization, optimization, and evaluation sets are fixed; only the amount and diversity of fine-tuning data change. Quantitatively, the 40K model is strongest overall: it performs best on all three SSv2 metrics and achieves the highest OOD PWT, while the 10K model has slightly lower OOD ADE and FDE. Qualitatively, however, the 40K model is almost always better, producing more coherent and object-aligned future motion than the smaller-data variants. This mismatch is expected for a multimodal task: displacement to a single recorded future can favor a conservative trajectory even when another prediction is more plausible. Overall, the results indicate that greater data diversity improves the learned motion prior, although that gain is not always reflected monotonically by these metrics. Thus, we perform additional analyses in section 4.5, and qualitatively visualize the results in the project website motionforesight.github.io.

4.4 Qualitative results

Figure 4 shows that MotionForesight predicts smooth, spatially coherent motion conditioned on the observed interaction. The model captures lifting, translation, rotation, constrained sliding, changes in direction, and local nonrigid motion rather than simply extrapolating instantaneous velocity. It also produces plausible forecasts on independently recorded phone videos, despite differences in objects, environments, viewpoints, and capture conditions. This transfer suggests that adapting the pretrained video-and-tracking representation preserves useful motion priors beyond the SSv2 fine-tuning distribution. As expected, deterministic prediction remains challenging when the observed context admits several plausible futures; comparisons with the baselines are shown in Appendix Fig. 5. Please check the website for more detailed qualitative results in diverse scenarios motionforesight.github.io.

Table 4: Motion-conditional evaluation on SSv2 unseen clips. We use $\tau = 2\,\text{cm}$ and a three-frame smoothing window. Darker and lighter blue denote the best and second-best learned result for each metric. TVO/VVO/DQS are evaluated on the 108 clips with at least five ground-truth-moving points.

Method	TVO ↑	VVO ↑	MoveF1 ↑	MoveIoU ↑	DQS ↑	$\bar{r} \to 1$
MotionForesight (40K)	**0.231**	**0.175**	**0.618**	**0.448**	**0.326**	0.72
MotionForesight (10K)	0.167	0.127	0.582	0.388	0.237	0.57
MotionForesight (1K)	0.150	0.112	0.487	0.370	0.186	0.50
MolmoMotion (no language)	0.101	0.078	0.585	0.258	0.195	1.27
MolmoMotion (language)	0.122	0.089	0.586	0.269	0.225	1.46
Video generation + tracks (no language)	0.165	0.138	0.568	0.435	0.228	0.52
Video generation + tracks (language)	0.157	0.137	0.613	0.415	0.256	**0.90**

4.5 Motion-conditional dynamics analysis

Standard metrics such as ADE, FDE, and PWT remain useful for comparison with prior work, but they compare a deterministic prediction with only one realized future. When the observed interaction admits several plausible outcomes, they can favor a conservative prediction near the starting position. We therefore add motion-conditional diagnostics that explicitly evaluate whether the predicted dynamics of the motion agree with the realized interaction.

Trajectory-Vector Overlap (TVO) compares the predicted and ground-truth displacement vector at every future frame. It gives high credit only when a point moves in the correct direction, by the correct amount, and at the correct time; delayed, missing, or excessive motion is penalized. *Velocity-Vector Overlap (VVO)* applies the same overlap to frame-to-frame velocity. It complements TVO by responding more strongly to local changes such as acceleration, turns, stops, and reversals. *MoveF1* asks whether the model moves the correct object points beyond a small motion threshold. Its threshold-free companion, *MoveIoU*, compares the peak predicted and ground-truth excursion of every point and therefore also reflects motion magnitude. *DQS* combines trajectory fidelity and motion placement through the geometric mean of TVO and MoveF1. Finally, the motion ratio $\bar{r}$ diagnoses magnitude calibration: values below one indicate motion that is too timid, while values above one indicate overshooting. Full mathematical definitions are provided in Appendix B.

Table 4 shows that MotionForesight (40K) leads all five motion-quality metrics, and its TVO and DQS improve monotonically from 1K to 40K training videos. Video generation followed by tracking obtains relatively strong motion placement but lower TVO and DQS, indicating that it can identify which points should move without preserving their metric trajectories as accurately. The 40K model still under-predicts total motion ($\bar{r} = 0.72$), leaving magnitude calibration as an important direction for improvement. Qualitative results of predictions shown in the project website further support these quantitative analyses motionforesight.github.io.

5 Discussion, Limitations, and Conclusion

In this work, we studied future 3D scene-flow prediction from short monocular videos of human–object interaction. We showed that a tracking-adapted video model can be converted from retrospective reconstruction into prospective prediction by masking future observations and training only a lightweight adapter, while preserving the pretrained video and tracking components. The resulting reference-anchored 3D tracks expose the metric variable needed for physical reasoning without restricting the object to a single rigid 6-DoF pose, allowing one representation to describe translation, rotation, articulated motion, and local deformation. Our quantitative, motion-conditional, and qualitative results show that the prediction model captures meaningful interaction dynamics, scales

with additional human video, and transfers to phone captures on home and office scenes.

Several challenges remain. The current model is deterministic and predicts only one future, although a short interaction prefix may support multiple physically plausible outcomes; both conventional displacement metrics and our motion-conditional diagnostics still compare that prediction with a single recorded trajectory. Supervision is also pseudo ground truth, so errors from monocular depth, camera estimation, segmentation, and tracking can propagate into both training and evaluation, while inference remains sensitive to errors in the estimated observed pointmaps. Finally, training is limited to 40K SSv2 manipulation clips. Although the OOD phone experiments demonstrate transfer across objects, environments, and viewpoints, they do not establish robustness to substantially different regimes such as head-mounted egocentric video, very large camera motion, longer-horizon interactions, or non-manipulation dynamics.

These limitations suggest clear directions for future work. Multi-hypothesis or probabilistic forecasting could represent alternative outcomes and expose calibrated uncertainty, paired with evaluation protocols that measure both plausibility and coverage across multiple valid futures. More accurate or jointly learned geometry, segmentation, and tracking could reduce dependence on a fixed pseudo-label pipeline, while broader training data could cover egocentric viewpoints, stronger camera motion, longer interactions, and more diverse physical processes. A further step is to connect the predicted 3D tracks to downstream planning and control, testing whether an object-centered motion prior learned from passive human video improves embodied decision making. Overall, MotionForesight provides evidence that large video priors can be efficiently redirected from rendering future appearance toward forecasting explicit, actionable 3D dynamics without language or action supervision.

References

[1] James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979. ISBN 9780395270493.

[2] Shimon Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979. ISBN 9780262210072.

[3] Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45): 18327–18332, 2013. doi: 10.1073/pnas.1306572110.

[4] György Gergely, Harold Bekkering, and Ildikó Király. Rational imitation in preverbal infants. *Nature*, 415(6873):755, 2002. doi: 10.1038/415755a.

[5] Jianing Zhang, Chenhao Zheng, Yajun Yang, Max Argus, Rustin Soraki, Winson Han, Taira Anderson, Chun-Liang Li, Shuo Liu, Jiafei Duan, Zhongzheng Ren, Jieyu Zhang, and Ranjay Krishna. MolmoMotion: Forecasting point trajectories in 3d with language instruction. *arXiv preprint arXiv:2606.18558*, 2026. doi: 10.48550/arXiv.2606.18558.

[6] Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Generating 6dof object manipulation trajectories from action description in egocentric vision. *arXiv preprint arXiv:2506.03605*, 2025. doi: 10.48550/arXiv.2506.03605.

[7] Rustin Soraki, Homanga Bharadhwaj, Ali Farhadi, and Roozbeh Mottaghi. ObjectForesight: Predicting future 3d object trajectories from human videos. *arXiv preprint arXiv:2601.05237*, 2026. doi: 10.48550/arXiv.2601.05237.

[8] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *Computer Vision – ECCV 2024*, volume 15134 of *Lecture Notes in Computer Science*, pages 306–324. Springer, 2024. doi: 10.1007/978-3-031-73116-7_18.

[9] Chuan Wen, Xingyu Lin, John Ian Reyes So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.092.

[10] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8802–8810. IEEE, 2025.

[11] Jisu Nam, Jahyeok Koo, Soowon Son, Jaewoo Jung, Honggyu An, Junhwa Hur, and Seungryong Kim. TrackCraft3R: Repurposing video diffusion transformers for dense 3d tracking. *arXiv preprint arXiv:2605.12587*, 2026. doi: 10.48550/arXiv.2605.12587.

[12] 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, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. doi: 10.48550/arXiv.2503.20314.

[13] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. doi: 10.48550/arXiv.1803.10122.

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

[15] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017.

[16] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[17] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3936–3951. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/bharadhwaj25a.html.

[18] 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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 24328–24346. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

[19] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. doi: 10.48550/arXiv.2508.00795.

[20] Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026. doi: 10.48550/arXiv.2601.05230.

[21] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

[22] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

[23] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations (ICLR)*, 2020.

[24] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "Something Something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017.

[25] Pradipto Das, Chenliang Xu, Richard F. Doell, and Jason J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[26] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[27] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[28] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022.

[29] Miao Liu, Siyu Tang, Yin Li, and James M. Rehg. Forecasting human-object interaction: Joint prediction of motor attention and actions in first person video. In *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 704–721. Springer, 2020. doi: 10.1007/978-3-030-58452-8_41.

[30] Zhifan Ni, Esteve Valls Mascaró, Hyemin Ahn, and Dongheui Lee. Human–object interaction prediction in videos through gaze following. *Computer Vision and Image Understanding*, 233: 103741, 2023. ISSN 1077-3142. doi: 10.1016/j.cviu.2023.103741.

[31] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. ContactGrasp: Functional multi-finger grasp synthesis from contact. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[32] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[36] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8688–8697, 2019.

[37] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database. Technical report, Carnegie Mellon University, 2008.

[39] Chen Bao, Jiarui Xu, Xiaolong Wang, Abhinav Gupta, and Homanga Bharadhwaj. HandsOnVLM: Vision-language models for hand-object interaction prediction. *arXiv preprint arXiv:2412.13187*, 2024. doi: 10.48550/arXiv.2412.13187.

[40] 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. Flowing from reasoning to motion: Learning 3d hand trajectory prediction from egocentric human interaction videos. *arXiv preprint arXiv:2512.16907*, 2025. doi: 10.48550/arXiv.2512.16907.

[41] Junbo Zhang and Kaisheng Ma. Mask2Act: Predictive multi-object tracking as video pre-training for robot manipulation. In *36th British Machine Vision Conference 2025, BMVC 2025*. BMVA, 2025. URL https://bmvc2025.bmva.org/proceedings/124/.

[42] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single RGB image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[45] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020. doi: 10.48550/arXiv.2008.08324.

[46] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single RGB images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[47] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. PVN3D: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[48] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[49] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[50] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.019.

[51] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

[52] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[53] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, et al. SAM 3D: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025. doi: 10.48550/arXiv.2511.16624.

[54] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1444–1454. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/mersch22a.html.

[55] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14673–14684, June 2024.

[56] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. ViP3D: End-to-end visual trajectory prediction via 3d agent queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[57] Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6991–7003, 2025.

[58] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Fei-Fei Li. PointWorld: Scaling 3d world models for in-the-wild robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20765–20779, June 2026.

[59] Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025. doi: 10.48550/arXiv.2512.13644.

[60] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas. FlowNet3D: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, June 2019. doi: 10.1109/CVPR.2019.00062.

[61] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10061–10072, 2023. doi: 10.1109/ICCV51070.2023.00923.

[62] Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas J. Guibas. AllTracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5253–5262, 2025.

[63] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *Computer Vision – ECCV 2024*, volume 15120 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2024. doi: 10.1007/978-3-031-73033-7_2.

[64] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. DELTA: Dense efficient long-range 3d tracking for any video. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.24211.

[65] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6726–6737, 2025.

[66] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. doi: 10.1109/ICCV51070.2023.00371.

[67] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. doi: 10.48550/arXiv.2511.10647.

[68] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *Computer Vision – ECCV 2024*, pages 289–305. Springer, 2024.

[69] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

[70] Neerja Thakkar, Shiry Ginosar, Jacob Walker, Jitendra Malik, Joao Carreira, and Carl Doersch. Forecasting motion in the wild. *arXiv preprint arXiv:2604.01015*, 2026.

[71] Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. Novaflow: Zero-shot manipulation via actionable flow from generated videos. *arXiv preprint arXiv:2510.08568*, 2025.

[72] Yuxuan Kuang, Sungjae Park, Katerina Fragkiadaki, and Shubham Tulsiani. Dex4d: Task-agnostic point track policy for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2602.15828*, 2026.

A Implementation Details

Here we discuss implementation details about the method, and additional details about the experiment setup. We also showcase additional results to help understand our method and the baselines.

A.1 Default model configuration

Table 5: Default model configuration. The method starts from a pretrained video DiT, uses a learned point-track latent interface as the geometric output representation, and trains only a small future adapter. Language is held at a null context and is not an input.

Component	Setting
Input horizon	7 observed frames, 15 predicted future frames, 22 total frames
Input signals	RGB context frames plus estimated observed pointmaps; no language, action, robot state, or future-frame input
Reference points	Object query points sampled from the frame-0 object mask
Coordinate frame	Last-observed camera frame for the default camera-subtracted model; reference object points still originate from frame 0
Video backbone	Wan2.1 T2V DiT with temporal RoPE; used as a single-step latent regressor
Track interface	Tracking-adapted point-track latent interface, instantiated with TrackCraft3R encoders/decoders and frozen rank-1024 tracking LoRA
Frozen components	Base video DiT, released tracking LoRA, RGB VAE, pointmap VAE, track decoder, visibility decoder, and null text context
Trainable adapter	Fresh rank-32 LoRA on self-attention projections (q, k, v, o)
Other trainable parameters	Future RGB mask latent, future pointmap mask latent, patch embedding, and output head
Trainable parameter count	12.19M
Resolution	320×576
Training loss	Decoded coordinate-space MSE at query points, scaled by 10 in implementation; future weight 1.0, observed weight 0.25

A.2 Data and pseudo-label generation

The main training corpus is built from human-object interaction videos. For each raw video, preprocessing selects a 22-frame clip centered around an interaction. The pipeline searches for visible hand-object interaction, advances to an anchor frame near contact, segments the manipulated object, and writes a fixed-length clip. Monocular depth and camera estimates are then computed for the clip, producing pointmaps and camera transforms. Finally, the dense 3D tracking stack is run on the complete clip to produce pseudo-ground-truth future tracks.

The distinction between label construction and predictor input is important. The full clip is used offline to obtain supervision, but the forecasting model never receives future frames as input. During training and inference, the context slots for $t < K$ contain observed RGB and pointmap latents, while the future slots for $t \geq K$ contain only learned mask latents.

The loader presents dense and sparse supervision through a common query-point interface. Dense tracking outputs are converted into sampled object query points from the reference mask. Sparse tracking outputs use their native valid query points. In both cases, the loss is evaluated only at query points with valid pseudo-ground-truth supervision.

A.3 Coordinate frame and normalization

The default model expresses pointmaps and track targets in the camera coordinate frame of the last observed image, $t = T_1 - 1$. Let $\bar{P}_t$ denote a pointmap in its original camera coordinate frame, and

let $G_{t \to T_1 - 1}$ denote the estimated transform into the last-observed camera frame. The model uses

$$P_t = G_{t \to T_1 - 1} \bar{P}_t. \tag{8}$$

For observed frames, this transformation is applied before pointmap encoding. For future frames, the transformed tracks are used only as training targets. At inference time, no future geometry is available.

This coordinate choice removes a large part of camera motion from the prediction problem. The model still predicts the future 3D position of reference-frame object points, but the coordinates are expressed relative to the last observed camera rather than the original frame-0 camera. Earlier dense-only variants used frame-0 coordinates; the last-observed camera frame is the default because it makes the target more object-motion-centered.

Pointmap normalization statistics are computed from observed pointmaps only. This keeps training consistent with inference, where the model has access to no future pointmaps. The same observed-prefix statistics are used to normalize the observed pointmap latents and to define the scale of the decoded coordinate loss.

A.4 Training objective details

The default training objective is the decoded coordinate-space loss in Eq. 7. We use

$$\lambda_{\text{obs}} = 0.25, \qquad \lambda_{\text{fut}} = 1.0.$$

The coordinate MSE is multiplied by 10 in the implementation. The future term is the primary objective, while the observed term keeps the adapted model aligned with the inherited tracking interface.

The track decoder is frozen, but gradients are allowed to pass through it into the trainable future adapter, patch embedding, output head, and mask latents. To fit the decoded loss at 320×576, the implementation decodes one frame at a time and uses gradient checkpointing. This makes the decoded metric loss feasible without updating the large video or tracking components.

A.5 Forecasting pass

At inference time, the model receives only the observed prefix. The forward pass is:

1. Estimate observed pointmaps $P_{0:K-1}$ from the RGB context frames.
2. Transform the observed pointmaps into the last-observed camera frame.
3. Encode observed RGB frames and pointmaps into visual-geometry latents.
4. Replace all future RGB and pointmap context latents with learned mask latents.
5. Repeat the reference-frame query latent across all T time steps.
6. Run the frozen video DiT with the frozen tracking adapter and the trained future adapter.
7. Decode residual-track latents with the frozen track decoder.
8. Add the decoded residuals to the reference point positions to obtain future 3D tracks.

A.6 Evaluation metrics

For held-out validation clips with pseudo-ground-truth tracks, we evaluate future average displacement error and final displacement error over valid object query points. Let $\mathcal{T}_{\text{fut}} = \{K, \dots, T-1\}$. The future ADE is

$$\text{ADE} = \frac{\sum_{t \in \mathcal{T}_{\text{fut}}} \sum_q v_{tq} \left\| \hat{X}_t(q) - X_t(q) \right\|_2}{\sum_{t \in \mathcal{T}_{\text{fut}}} \sum_q v_{tq} + \epsilon}. \tag{9}$$

The future FDE is computed at the final predicted frame:

$$\text{FDE} = \frac{\sum_q v_{T-1,q} \left\| \hat{X}_{T-1}(q) - X_{T-1}(q) \right\|_2}{\sum_q v_{T-1,q} + \epsilon}. \tag{10}$$

Both metrics are reported in metric 3D space. For the OOD phone videos, we evaluate against pseudo-ground-truth tracks extracted from each complete clip and supplement the metrics with qualitative comparisons. Because each clip records only one of several plausible futures, ADE and FDE should not be interpreted as complete measures of forecast quality: a prediction can differ from the recorded trajectory while remaining physically plausible and consistent with the observed interaction.

B Motion-Conditional Dynamics Metrics

The following diagnostics measure predicted motion relative to the last observed frame rather than absolute position. Let $O = T_1$ be the number of observed frames, and let $p_{t,i}, \hat{p}_{t,i} \in \mathbb{R}^3$ denote the ground-truth and predicted position of point i. We define independently anchored displacements

$$\Delta_{t,i} = p_{t,i} - p_{O-1,i}, \qquad \hat{\Delta}_{t,i} = \hat{p}_{t,i} - \hat{p}_{O-1,i}, \qquad t \geq O. \tag{11}$$

This removes constant offsets at the forecast boundary and isolates the subsequent dynamics. To reduce one-frame noise in the monocular pseudo-ground-truth tracks, both series are smoothed with a centered three-frame moving average: $g_{t,i} = \text{sm}(\Delta)_{t,i}$ and $h_{t,i} = \text{sm}(\hat{\Delta})_{t,i}$.

Moving points.. A point is considered moving if its peak excursion from the anchor exceeds τ:

$$e_i = \max_{t \geq O} \|g_{t,i}\|_2, \qquad \hat{e}_i = \max_{t \geq O} \|h_{t,i}\|_2, \qquad \mathcal{M} = \{i : e_i \geq \tau\}, \quad \hat{\mathcal{M}} = \{i : \hat{e}_i \geq \tau\}. \tag{12}$$

Peak excursion counts motion that later returns to its starting position. We use $\tau = 2\,\text{cm}$, which remains above the smoothed pseudo-track jitter floor while retaining 108 of 150 valid SSv2 clips for fidelity evaluation; results are also checked at $\tau \in \{1, 5, 10\}\,\text{cm}$. A clip is fidelity-eligible when $|\mathcal{M}| \geq 5$.

Trajectory-Vector Overlap.. For each ground-truth-moving point, TVO measures frame-aligned agreement in direction and magnitude:

$$\text{TVO}_i = \frac{\sum_{t \geq O} [\cos(h_{t,i}, g_{t,i})]_+ \min(\|h_{t,i}\|_2, \|g_{t,i}\|_2)}{\sum_{t \geq O} \max(\|h_{t,i}\|_2, \|g_{t,i}\|_2) + \epsilon}, \tag{13}$$

where $[x]_+ = \max(x, 0)$ and the cosine term is defined as zero if either vector has zero norm. Missing, delayed, excessive, or directionally incorrect motion leaves unmatched magnitude in the denominator. We average over moving points within each eligible clip and then average clips equally.

Velocity-Vector Overlap.. TVO compares displacement chords from the anchor, which can change slowly along a curved path. VVO applies the same overlap to smoothed temporal velocities,

$$v_{t,i} = g_{t,i} - g_{t-1,i}, \qquad \hat{v}_{t,i} = h_{t,i} - h_{t-1,i}, \qquad g_{O-1,i} = h_{O-1,i} = 0, \tag{14}$$

and is aggregated using the TVO equation with (h, g) replaced by $(\hat{v}, v)$. It therefore emphasizes whether a prediction turns, reverses, accelerates, or stops at the correct time.

Motion placement and magnitude.. MoveF1 is the point-set F1 score between $\hat{\mathcal{M}}$ and $\mathcal{M}$. If neither set contains a moving point, the score is one; if exactly one set is empty, it is zero. Its threshold-free companion compares peak excursions for all points:

$$\text{MoveIoU} = \frac{\sum_i \min(\hat{e}_i, e_i)}{\sum_i \max(\hat{e}_i, e_i) + \epsilon}. \tag{15}$$

We additionally report the clip-averaged motion ratio

$$r_c = \frac{\sum_{i,t \geq O} \|h_{t,i}\|_2}{\sum_{i,t \geq O} \|g_{t,i}\|_2 + \epsilon}, \qquad \bar{r} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} r_c, \tag{16}$$

where $\bar{r} = 1$ indicates calibrated total motion, values below one indicate under-prediction, and values above one indicate over-prediction. For a compact summary, we compute $\text{DQS}_c = \sqrt{\text{TVO}_c \, \text{MoveF1}_c}$ on each fidelity-eligible clip before averaging. We treat TVO and MoveF1 as the primary components because the composite can hide whether an error comes from trajectory fidelity or motion placement.

These diagnostics still compare against one realized future and therefore complement the qualitative evaluations of plausible alternative outcomes.

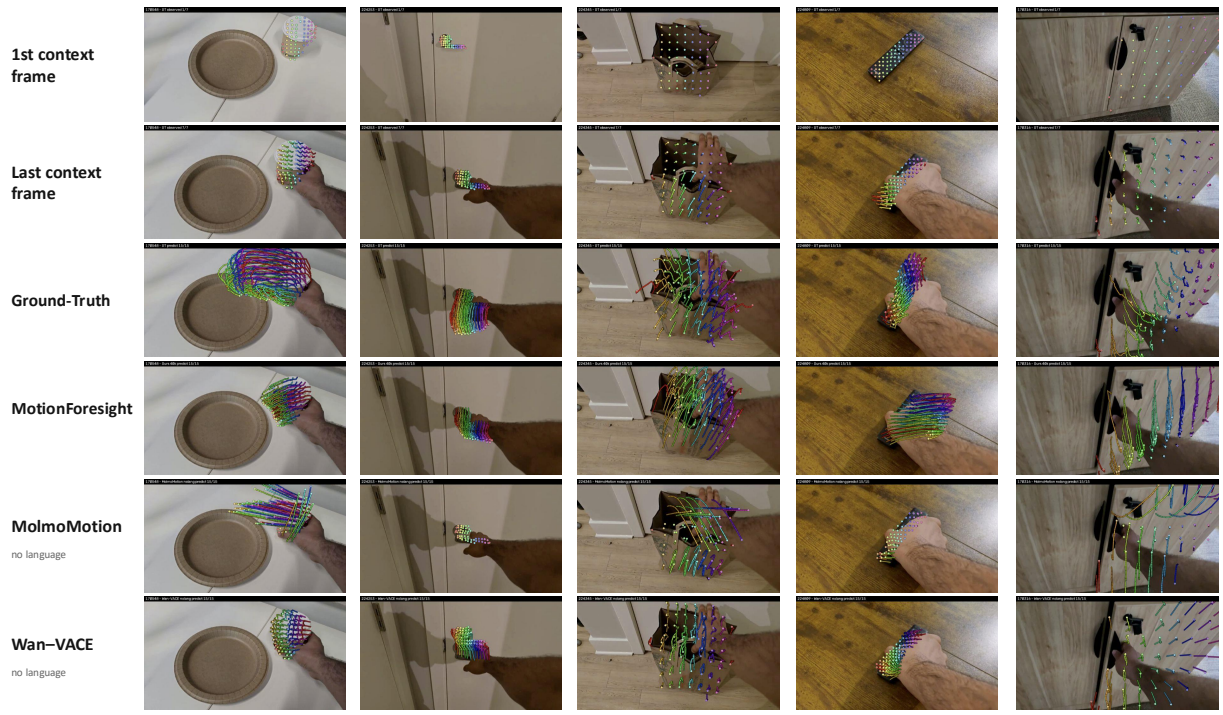

Figure 5: Qualitative comparison with baselines. We show the observed context, the recorded future tracks, and predictions from MotionForesight, MolmoMotion, and video generation followed by tracking. These examples complement the single-ground-truth trajectory metrics by revealing motion coherence and plausible alternative futures.