ObjectForesight: Predicting Future 3D Object Trajectories from Human Videos

Rustin Soraki[1], Homanga Bharadhwaj[2,*], Ali Farhadi[1,*], Roozbeh Mottaghi[1,*]

[1] School of Computer Science & Engineering, University of Washington
[2] The Robotics Institute, Carnegie Mellon University
rustin@cs.washington.edu

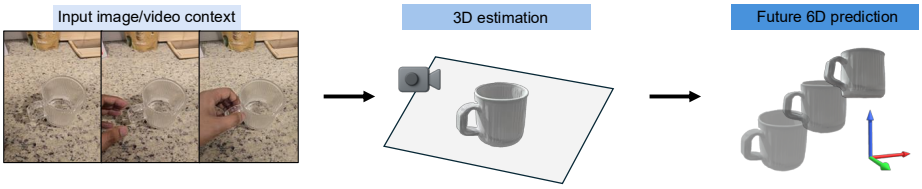


Fig. 1: We introduce **ObjectForesight**, a framework for predicting future 3D object trajectories from a video context of past motion. We first estimate the object's 3D shape and initial pose, and then predicts its future 6D poses over time. There are three key contributions: (1) introducing and formalizing the task of 3D object dynamics prediction from human videos, (2) a 3D object-centric dynamics model for future prediction of 6-DoF trajectories, (3) a large-scale dataset of 2+ million object-centric 3D trajectories. Video results are in the website objectforesight.github.io

Abstract. Humans can effortlessly anticipate how objects might move or change through interaction—imagining a cup being lifted, a knife slicing, or a lid being closed. We aim to endow computational systems with a similar ability to predict plausible future object motions directly from passive visual observation. We introduce **ObjectForesight**, a 3D object-centric dynamics model that predicts future 6-DoF poses and trajectories of rigid objects from short egocentric video sequences. Unlike conventional world/dynamics models that operate in pixel or latent space, ObjectForesight represents the world explicitly in 3D at the object level, enabling geometrically grounded and temporally coherent predictions that capture object affordances and trajectories. To train such a model at scale, we leverage recent advances in segmentation, mesh reconstruction, and 3D pose estimation to curate a dataset of 2+ million short clips with pseudo-ground-truth 3D object trajectories. Through extensive experiments, we show that ObjectForesight achieves significant gains in accuracy, geometric consistency, and generalization to unseen objects and scenes—establishing a scalable framework for learning physically grounded, object-centric dynamics models directly from observation.
objectforesight.github.io

Keywords: 3D Future Prediction · Object-Centric Motion Prediction · Manipulation Cues from Human Videos

1 Introduction

Humans possess an intuitive understanding of how the world around them can change through interaction. When we see a cup on a table, we can effortlessly imagine it being picked up, tilted, or placed elsewhere. Watching a hand reach toward a knife, we can anticipate the knife's motion and the transformation of the objects it touches. Such inferences go beyond recognizing what *is* — they reflect our ability to imagine what *can be*. This capacity to mentally simulate object interactions is central to intelligent behavior, allowing us to plan, predict, and act effectively in the physical world [23].

Our goal in this work is to endow computational systems with a similar capability: to infer and predict plausible *future* configurations of objects from passive visual observation. **We focus on the problem of predicting 3D object dynamics** — *learning how objects can move and interact in 3D space as a result of human actions, without directly modeling the human motion itself.* Rather than learning explicit manipulation trajectories or low-level control policies, we seek to model their *effects*: the diverse, physically coherent object motions that arise from everyday interactions.

To this end, we present **ObjectForesight**, a 3D object-centric forward dynamics model that learns to predict future 6-DoF trajectories of *rigid* objects from egocentric human videos. Given a sequence of RGB frames and an object mesh, **ObjectForesight** predicts a temporally coherent sequence of future object poses — effectively imagining how the object may move in the near future (Fig. 1). Operating in object-centric coordinates allows the model to generalize across varied objects, scenes, and manipulation styles, capturing the underlying semantics of object affordances.

For training **ObjectForesight**, a key challenge is data: There are no large-scale, clean, and physically grounded 3D interaction datasets. Existing robot datasets capture limited, scripted manipulations with explicit action supervision [33], while internet-scale human video corpora on their own, though rich and diverse, lack aligned 3D information such as object poses, camera geometry, or depth [14, 40]. To address this, we develop a scalable data curation pipeline that transforms passive human videos into structured 3D motion supervision. Specifically, we extract more than 2 million short clips (2–3 seconds each) from the EPIC-Kitchens dataset [9], automatically detecting hands [51] and identifying objects in contact using SAM [37]. We then recover 3D object meshes and poses with TRELLIS [47], and estimate camera motion and monocular depth using SpaTrackerv2 [49]. By expressing object poses relative to the first-frame camera coordinates, we effectively disentangle ego-motion from object motion. This process converts ordinary egocentric videos into a large-scale dataset of 3D object trajectories — the first at this level of scale, fidelity, and semantic diversity.

ObjectForesight integrates a Diffusion Transformer (DiT) [34] with a geometry aware 3D point encoder, PointTransformerV3 [46], to jointly reason about object motion and surrounding scene context. Given a short history of RGB frames with corresponding monocular depth maps and a mask of the object in

the anchor frame, the model encodes the local 3D geometry of the scene and the object's recent motion into a unified representation. Conditioned on this visual and spatial context, **ObjectForesight** predicts a distribution over future 6-DoF object poses through a denoising diffusion process. This formulation enables robust, multi-modal prediction of dynamically feasible and physically consistent object motions, maintaining geometric fidelity and temporal coherence across predicted trajectories.

In summary, *we introduce the task of predicting future 3D object dynamics from videos — a core capability for embodied visual reasoning, and build models and datasets towards this task.* Our key contributions are as follows:

- We introduce and formalize the task of **3D object dynamics prediction from human videos**, establishing a standardized setting for learning how objects move in the real world. This formulation enables models to leverage the vast amount of in-the-wild egocentric video data to learn physical interaction priors without requiring explicit action supervision.
- We propose **ObjectForesight**, a 3D object-centric dynamics model that predicts future 6-DoF trajectories of objects from short egocentric video snippets and monocular geometry.
- We construct a large-scale dataset of object-centric 3D trajectories from more than 2 million EPIC-Kitchens clips, using automatic object segmentation and pose estimation to recover high-quality 3D motion supervision from generic interaction videos.

Across extensive experiments in daily human activities, **ObjectForesight** produces accurate, stable, and physically coherent 6-DoF trajectories in diverse real-world scenes. The diffusion-based formulation outperforms autoregressive models and video-generation approaches, offering sharper long-horizon consistency and better multimodal prediction. These results show that large-scale observational data, combined with explicit 3D reasoning, provides a strong foundation for reliable and scalable object-centric motion forecasting.

2 Related Works

Extracting Representations from Human Videos. Large-scale egocentric datasets such as Something-Something [14], YouCook [10], EPIC-Kitchens [9], EGTEA [28], and Ego4D [15] have enabled learning rich representations of human–object interactions directly from video. Early work focused on recovering 3D hand and object poses [12, 19, 22, 39, 54] and reconstructing object geometry [20, 21, 25, 48], providing geometric supervision for understanding interaction. Advances in tracking and scene flow [11, 18, 24, 49] further enable dense motion estimation across time, while recent segmentation and reconstruction systems such as SAM [37], TRELLIS [47], and very recently SAM3D [8] make it possible to automatically extract 3D trajectories of objects from in-the-wild images and videos. Our work is closely related in that it leverages these advances to *curate a dataset of object-centric 3D trajectories at scale*, transforming ordinary

human videos into a resource for training predictive models of object dynamics. By building upon existing 3D pose estimation and reconstruction pipelines, we focus not on estimating geometry itself, but on learning how objects move and interact over time.

Predicting Manipulation Cues from Human Videos. Another line of research focuses on predicting or reasoning about manipulation cues and affordances from human videos. Classical works in affordance learning [5, 13, 29–32] study how objects are grasped, where contact occurs, how hands move in the future [3,7,29] or which parts of an object afford specific actions. More recent approaches [4, 40, 41] learn to anticipate manipulation outcomes or future contact regions, connecting perception to physical reasoning. Such methods primarily operate in 2D or intermediate feature space, forecasting human or object-centric cues that signal future interactions. Our work shares the goal of extracting predictive signals from human videos but differs in focus. Rather than predicting contact maps or categorical actions, we aim to learn a continuous model of *3D object dynamics*—how objects themselves move in space as a result of human interactions. By grounding prediction in SE(3) pose space and explicit geometry, we extend affordance learning toward physically coherent, object-centric reasoning about future motion.

World Models and Trajectory Representations. Building models of how the world evolves in response to interaction has long been a core challenge in both computer vision and robotics. Recent efforts in visual world modeling have primarily focused on learning predictive representations either at the pixel level through video generation [42,43] or in latent spaces through representation learning [1,17,26]. While such approaches capture temporal dependencies, they often lack explicit 3D grounding and object-level motion prediction. In contrast, our work develops an *explicit 3D object trajectory model* that operates in SE(3) space. Instead of predicting future pixels or abstract latent codes, our method explicitly models object evolution in 6-DoF pose space, and unlike implicit language conditioning [50], conditions explicitly on predicted object geometry and past motion context, offering a physically grounded representation well-suited for integration into robotic manipulation frameworks [4, 27]. Future prediction has been explored in the self-driving scenarios (e.g., [16]), but they focus on largely independent agents and external objects in a different dynamics regime.

3 Method

We aim to learn a forward dynamics model that predicts future 3D poses of rigid objects from passive human videos. The task involves inferring plausible 6-DoF trajectories conditioned on observed object geometry, local scene context, and a short history of object motion. Since no dataset exists for this setting, we construct a large-scale dataset of 3D object trajectories from egocentric human activity videos using off-the-shelf vision models (Sec. 3.2). We then train a diffusion-based transformer model (Sec. 3.3) that learns to sample diverse, physically consistent future trajectories conditioned on visual and geometric context.

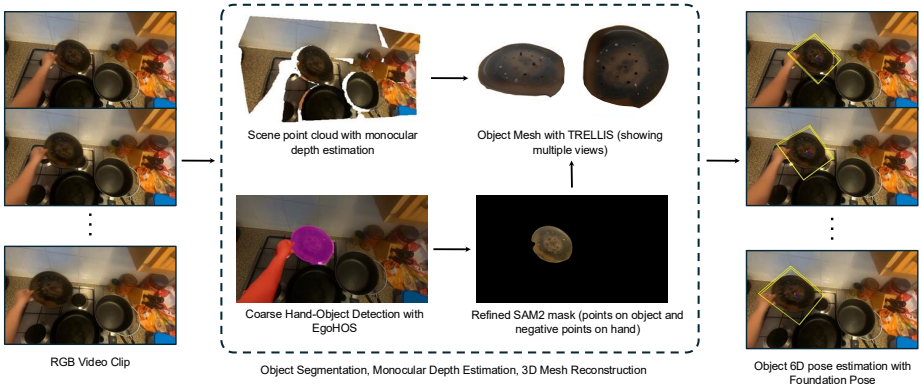


Fig. 2: Data curation pipeline from egocentric video to 3D object trajectories. Starting from EPIC-Kitchens action segments, we detect hands and objects, refine masks, and filter for clear manipulations. We then reconstruct an object mesh, recover metric depth and camera geometry, and do 6-DoF pose estimation and tracking. Sliding windows over these tracks yield short, clean, anchor-frame–canonicalized 6-DoF trajectories used to train **ObjectForesight**.

3.1 Overview

ObjectForesight tackles the problem of predicting future 3D object motion from short windows of egocentric video. Given C observed frames and a prediction horizon of H, the goal is to model a distribution over the next H future 6-DoF poses of a manipulated object. All frames in the window are expressed in the anchor-frame (first frame of the prediction horizon) camera coordinates, allowing us to isolate true object motion from ego-motion. In our default setting, we use C=3 and H=8.

Formally, we observe images $\mathcal{I}_{1:C}$ and their corresponding object poses

$$\mathbf{P}_{1:C} = [\mathbf{p}_1, \ldots, \mathbf{p}_C], \qquad \mathbf{p}_t \in \mathrm{SE}(3),$$

where each pose token $\mathbf{p}_t = [x_t, y_t, z_t, \mathbf{r}_{t,6\mathrm{D}}]$ contains translation and a continuous 6D rotation representation [52]. Depth from the anchor frame is backprojected to form a point cloud $\mathbf{X}$, and normalized object bounding boxes $\mathbf{B}_{1:P}$ provide coarse spatial cues. The forecasting target is the future sequence

$$\mathbf{P}_{\text{future}} = [\mathbf{p}_{t_a}, \ldots, \mathbf{p}_{t_a+H-1}], \quad t_a = C{+}1.$$

ObjectForesight contributes both *data* and *modeling*: (i) a large-scale pipeline that converts raw egocentric videos into metrically grounded, anchor-frame–canonicalized 6-DoF trajectories; and (ii) a geometry-aware diffusion model that predicts future object motion from these trajectories.

Our predictive architecture combines a context-conditioned geometry encoder over the anchor-frame point cloud with a Diffusion Transformer (DiT) temporal backbone. The encoder conditions point features on the recent motion context (FiLM) and pools them into an object-centric scene embedding $\mathbf{z}_{\text{geom}}$,

while the DiT models a distribution over future pose sequences conditioned on $\mathbf{z}_{\text{geom}}$ and an explicit pose-token prefix.

The model operates in a depth-normalized pose space for stability, and uses a cosine noise schedule with v-parameterized denoising. At inference, DDIM sampling produces smooth, diverse, and physically coherent 3D trajectories.

3.2 Data Curation: From Egocentric Video to 3D 6-DoF Object Trajectories

Our curation pipeline converts in-the-wild egocentric videos into clean, metrically grounded trajectories of hand-manipulated objects (Fig. 2). Starting from EPIC-Kitchens action segments, we apply a sequence of automatic extraction and quality gates to recover temporally coherent 6-DoF poses. We summarize the key stages below.

Action segment prefiltering. We begin from annotated single-activity segments and discard clips longer than 10 seconds to limit drift and ensure short, interaction-centric windows.

Hand–object discovery with EgoHOS. For each remaining clip, we run EgoHOS [51] to segment hands and candidate manipulated objects frame-wise. Frames without hands or without any object hypotheses are removed. This yields per-frame masks for (i) active hand(s) and (ii) plausible manipulated objects.

Robust object masks with temporal consensus. We initialize SAM2 [37] using point prompts derived from EgoHOS masks and propagate a single object instance through the clip. Positive prompts come from the interior of the EgoHOS object mask; negative prompts are drawn from the hand mask, the other-hand object (if present), and a thin ring around the object boundary. To mitigate occasional EgoHOS failures, we form *temporal consensus prompts*, intersections of masks over a small temporal window, which bias SAM2 toward temporally stable shapes. Newly proposed SAM2 masks must have low IoU with the active tracks to prevent duplication. The result is a temporally smooth, occlusion-resilient object mask sequence.

VLM gating for manipulation and view quality. We apply a two-stage VLM-based filter using InternVL3 [53]. First, at the video level, we check whether the highlighted object is actually moved by hand; static objects are discarded. Second, at the frame level, we crop around the object and evaluate visibility (no blur, limited occlusion). Frames passing this test form the set of *clean views*.

Object 3D reconstruction from clean views. TRELLIS [47] reconstructs a 3D object mesh from clean views. The mesh is *not* used during **ObjectForesight** training; it only serves as a geometric template for model-based pose estimation under occlusion.

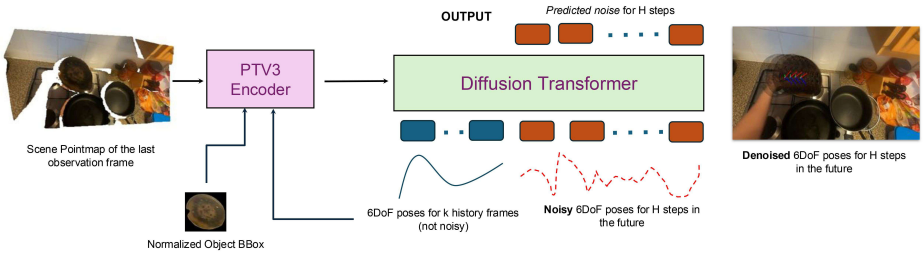


Fig. 3: Model architecture. Given past pose tokens and their normalized bounding boxes, we summarize motion context with anchor-query attention and use it to guide object-centric pooling in a PointTransformerV3 encoder, producing a geometry-aware scene embedding. A diffusion transformer (DiT, AdaLN-Zero) then denoises future depth-normalized pose tokens, conditioned on the scene embedding and an explicit prefix of past pose tokens. This design allows **ObjectForesight** to generate diverse, physically coherent, and temporally smooth 3D motion predictions.

Model-based 6-DoF pose with metric depth and amodal masks. SpaTrackerV2 [49] provides metric depth and camera geometry. We use DiffusionVAS [6] to complete amodal object masks. Pose initialization and tracking use FoundationPose [45] with three modifications for egocentric video:

(i) *Metric scale estimation.* TRELLIS meshes lack scale; we estimate scale by comparing masked depth points to mesh radii across neighboring frames (robust weighted median), then refine via depth–silhouette alignment.

(ii) *Multi-view initialization.* We pick up to five clean views, run FoundationPose initialization, and refine each using depth alignment and silhouette consistency. We choose the best by FoundationPose score, with an IoU-based override. Low-IoU cases are discarded.

(iii) *Bidirectional tracking with re-registration.* From the best initialization we track forward and backward. If projection IoU drops below 0.1, we trigger local re-registration using the current mask. This produces temporally coherent pose tracks with explicit re-registration events.

Trajectory slicing and final quality control. We slide a window of length $C+H$ along each track. A window is kept if it lies within a single registration segment and maintains stable projection IoU (no drop > 0.1). All poses are re-expressed in the anchor-frame camera coordinates to remove ego-motion.

Outcome. This automatic pipeline enforces (i) manipulation validity (VLM gating), (ii) mask fidelity (SAM2 with temporal consensus and amodal completion), and (iii) metric, temporally coherent poses (FoundationPose with depth, geometry, and re-registration). The result is a large collection of short, clean, object-centric trajectories suitable for training multi-modal 3D dynamics models.

3.3 Predicting Future Trajectories in 3D

Our forecasting model learns to generate diverse and physically coherent future pose sequences conditioned on the current geometric and motion context. It combines a geometry-aware encoder with a diffusion-based transformer operating on object-centric, depth-normalized 9D pose tokens. An outline of the model is depicted in Fig. 3.

Scene and Context Encoding. Given an anchor-frame point cloud $\mathbf{X} \in \mathbb{R}^{N\times 3}$, conditioning pose tokens $\mathbf{P}_{1:t_a} \in \mathbb{R}^{t_a\times 9}$, and corresponding normalized bounding boxes $\mathbf{B}_{1:t_a} \in \mathbb{R}^{t_a\times 4}$, our goal is to construct a compact representation that summarizes both the recent motion and the 3D scene structure. Here N is the number of points sampled from the anchor-frame depth map. We use C pre-anchor context frames, so the anchor index is $t_a = C+1$ in the $C + H$ window.

For each frame k in the conditioning sequence, we form $[P_k B_k] \in \mathbb{R}^{13}$ and project it into a 64D context space. We then pool the conditioning sequence with attention: the anchor token queries all conditioning tokens, and we add a sinusoidal embedding of the relative time to the anchor with a learnable scale. This yields a single context vector $\mathbf{ctx} \in \mathbb{R}^{64}$.

We feed the point cloud $\mathbf{X}$ into a PointTransformerV3 encoder [46]. Each point is represented by its anchor-camera coordinates and its coordinates in the estimated anchor object frame, enabling object-centric reasoning. We also provide the encoder with $\mathbf{ctx}$ which conditions the point cloud features on it using feature-wise linear modulation (FiLM) [35]. We then pool point features into a global scene embedding $\mathbf{z}_{\text{geom}} \in \mathbb{R}^{512}$ using an object-centric attention head that matches point features to a query derived from $\mathbf{ctx}$ and biases weights toward points near the object. $\mathbf{z}_{\text{geom}}$ is then used as a conditioning signal, injected using AdaLN-Zero [34] inside the DiT blocks.

Tokenization of Pose Sequences. We operate on object-centric pose tokens expressed in the anchor-frame camera coordinates. Each pose $\mathbf{p}_t = [x_t, y_t, z_t, \mathbf{r}_{t,6\text{D}}]$ is reparameterized into a depth-normalized token:

$$\mathbf{y}_t = [u_t, v_t, s_t, \mathbf{r}_{t,6\text{D}}], \quad u_t = \tfrac{x_t}{z_t},\ v_t = \tfrac{y_t}{z_t},\ s_t = \log z_t,$$

which reduces the dynamic range of translation and improves numerical stability. For the future horizon of length H, we form $\mathbf{Y}_{\text{future}} = [\mathbf{y}_{t_a}, \ldots, \mathbf{y}_{t_a+H-1}]$. We then apply channel-wise standardization using statistics $(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \mathbb{R}^9$ estimated over the first training batches (and fixed thereafter).

We apply the same depth reparameterization and standardization to the context-frame poses, yielding normalized context tokens $\tilde{\mathbf{P}}_{1:t_a}$. These tokens are embedded and prepended as a prefix to the future sequence inside the transformer, giving the DiT access to the full conditioning history.

Forward Diffusion Process and Cosine Schedule. Let $\tilde{\mathbf{Y}}_0 \in \mathbb{R}^{H\times 9}$ be the clean normalized future sequence for the batch. Following the standard diffusion framework, we define a forward noising process with a cosine β-schedule and

sample diffusion timesteps τ uniformly from $\{0, \ldots, T-1\}$ (we use T=1000). The DiT processes the noised sequence $\tilde{\mathbf{Y}}_\tau$ as a length-H sequence of 9D tokens, conditioned on the diffusion-timestep embedding, the geometric embedding $\mathbf{z}_{\text{geom}}$, and the normalized context tokens $\tilde{\mathbf{P}}_{1:t_a}$.

Tokens are embedded into a latent sequence and augmented with learned absolute positions, a token-type embedding (context vs. future), and a signed anchor-relative time embedding. Conditioning is injected via AdaLN-Zero where a lightweight MLP combines timestep and scene embeddings into per-layer normalization modulations and gated residuals within each transformer block.

v-Parameterization with p2 Weighting. Instead of predicting the noise $\boldsymbol{\epsilon}$ directly, we adopt v-parameterization, which stabilizes training across timesteps. We train with an SNR-weighted regression loss (p2 reweighting [38]) and additionally apply horizon-aware weighting that linearly increases toward later forecast steps (from 1 to 3 across the horizon). During training and DDIM sampling, we reconstruct $\hat{\tilde{\mathbf{Y}}}_0$ from the predicted $\mathbf{v}_\theta$ using the standard closed-form relation.

De-normalization and Pose Decoding. We recover physical 9D poses by reversing the standardization and depth reparameterization applied during preprocessing. The network output is first de-standardized as

$$\hat{\mathbf{Y}}_0 = \hat{\tilde{\mathbf{Y}}}_0 \odot \boldsymbol{\sigma} + \boldsymbol{\mu}, \tag{1}$$

where $\odot$ is elementwise multiplication. Each resulting token $\hat{\mathbf{y}}_t = [\hat{u}_t, \hat{v}_t, \hat{s}_t, \hat{\mathbf{r}}_{t,\text{6D}}]$ is then converted back to Cartesian translation coordinates by inverting the log-depth and normalized-coordinate transforms:

$$\hat{z}_t = \exp(\hat{s}_t), \qquad \hat{x}_t = \hat{u}_t\, \hat{z}_t, \qquad \hat{y}_t = \hat{v}_t\, \hat{z}_t.$$

This yields the final 9D pose token $[\hat{x}_t, \hat{y}_t, \hat{z}_t, \hat{\mathbf{r}}_{t,\text{6D}}]$.

Losses. The primary training objective is the v-prediction MSE,

$$\mathcal{L}_v = \mathbb{E}\left[w(\tau) \, \left\| \mathbf{v}_\theta(\tilde{\mathbf{Y}}_\tau, \tau) - \mathbf{v}_\tau \right\|_2^2 \right],$$

where $\mathbf{v}_\tau = \sqrt{\bar{\alpha}_\tau}\, \boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_\tau}\, \mathbf{Y}_0$ is the v-parameterization target and $w(\tau) = (1 + \text{SNR}(\tau))^{-\gamma}$ is a P2-style weight that downweights low-noise timesteps.

Because we predict poses in anchor-frame camera coordinates, we can additionally supervise the decoded SE(3) trajectory directly. For each future step k, we convert the predicted 6D rotation to $\hat{\mathbf{R}}_k \in \text{SO}(3)$ and measure translation error $\|\mathbf{t}_k - \hat{\mathbf{t}}_k\|_2$ and rotation error via the geodesic distance $\text{d}_{\text{geo}}(\mathbf{R}_k, \hat{\mathbf{R}}_k)$. Averaging both over the prediction horizon gives the auxiliary pose loss

$$\mathcal{L}_{\text{aux}} = \mathbb{E}\big[\bar{\alpha}_\tau \big(\lambda_R\, \overline{d}_{\text{geo}} + \lambda_{\text{trans}}\, \overline{e}_{\text{trans}}\big)\big],$$

where $\overline{d}_{\text{geo}}$ and $\overline{e}_{\text{trans}}$ are the horizon-averaged rotation (radians) and translation errors, respectively. The expectation is over training samples and sampled

diffusion steps; λ_R and λ_{trans} balance the two error magnitudes; and the factor $\bar{\alpha}_\tau$ (near 1 for clean samples, near 0 for noisy ones) downweights steps where the $\hat{\tilde{\mathbf{Y}}}_0$ reconstruction is unreliable.

To encourage smooth trajectories we add SE(3) velocity and acceleration losses on consecutive pose increments (also weighted by $\bar{\alpha}_\tau$). Defining translation increments $\Delta\mathbf{t}_k = \mathbf{t}_{k+1} - \mathbf{t}_k$, rotation increments $\Delta\mathbf{R}_k = \mathbf{R}_k^\top \mathbf{R}_{k+1}$, and letting Δ^2 denote second differences:

$$\mathcal{L}_{\text{vel}} = \overline{\|\Delta\mathbf{t}_k - \Delta\hat{\mathbf{t}}_k\|_2^2 + \mathrm{d}_{\text{geo}}(\Delta\mathbf{R}_k, \Delta\hat{\mathbf{R}}_k)^2},$$

$$\mathcal{L}_{\text{acc}} = \overline{\|\Delta^2\mathbf{t}_k - \Delta^2\hat{\mathbf{t}}_k\|_2^2 + \mathrm{d}_{\text{geo}}(\Delta^2\mathbf{R}_k, \Delta^2\hat{\mathbf{R}}_k)^2},$$

where $\overline{\cdot}$ averages over valid horizon steps k. Finally, a small depth-floor penalty discourages degenerate predictions with extremely small depth:

$$\mathcal{L}_{z_{\min}} = 0.01\,\mathrm{ReLU}(z_{\min} - \hat{z}_t).$$

The complete objective is

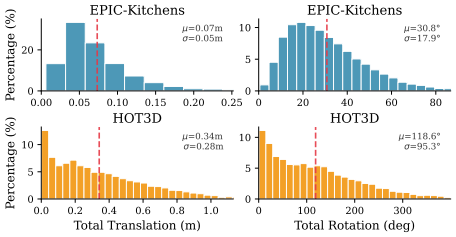
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{v}} + \mathcal{L}_{\text{aux}} + \mathcal{L}_{z_{\min}} + \lambda_{\text{vel}}\,\mathcal{L}_{\text{vel}} + \lambda_{\text{acc}}\,\mathcal{L}_{\text{acc}},$$

where $\mathcal{L}_{\text{v}}$ operates in normalized token space while all other terms are computed on the decoded pose sequence (after de-normalization and depth decoding). We set $\lambda_R = 2.0$, $\lambda_{\text{trans}} = 20.0$, $\lambda_{\text{vel}} = 0.5$, and $\lambda_{\text{acc}} = 0.1$.

Diffusion Sampling At inference time we sample from pure Gaussian noise using deterministic DDIM with S=50 evenly spaced denoising steps. At each step the model predicts $\mathbf{v}_\theta$ conditioned on $(\mathbf{z}_{\text{geom}}, \tilde{\mathbf{P}}_{1:t_a})$, from which we reconstruct $\hat{\tilde{\mathbf{Y}}}_0$ and apply the DDIM update. After the final step, the result is the predicted future trajectory $\hat{\mathbf{P}}_{\text{future}}$ in the anchor frame.

Why Diffusion? Diffusion-based modeling is well suited to 3D interaction dynamics. Given identical 3D conditioning, multiple future motions can be plausible (e.g., a mug can be picked up, slid, or rotated). Our DiT captures this inherently one-to-many nature while encouraging temporally smooth, physically plausible trajectories. In our experiments, it yields more plausible and geometrically consistent predictions than an autoregressive transformer baseline trained on the same object-centric representation.

Summary. By combining an object-centric 3D scene encoder with an AdaLN-Zero conditioned diffusion transformer over depth-normalized pose tokens, **ObjectForesight** learns a rich conditional distribution over future object motion. The architecture explicitly leverages metric geometry, camera coordinates, and pose history to generate accurate, diverse, and physically plausible 6-DoF trajectories in real-world egocentric scenes.



(a)

Step Name	Number
Action Segments	76,885 vids
Selected Vids (hands, $\leq$ 10s)	72,046 vids
SAM2 Tracks	229,102 tracks
Filtered Tracks	112,057 tracks
TRELLIS Models	71,296 models
Objects with Pose Tracks	59,174 tracks
Pre-Filter Trajectories	3,065,568 trajs
Post-Filter Trajectories	2,073,109 trajs

(b)

Fig. 4: Dataset overview. (a) Motion statistics of the curated 3D object trajectories and HOT3D dataset, showing total translation and rotation for all objects. (b) Summary of each stage in our automatic data curation pipeline as applied to EpicKitchens videos.

4 Experiments

Our experiments aim to answer three questions:

1. Is the curated dataset of object-centric 3D trajectories reliable and diverse?
2. Are the predicted future object motions plausible and physically consistent?
3. Does the model generalize beyond the distribution of curated scenes?

4.1 Datasets

We use our curated dataset from EpicKitchen for the main experiments. We also conducted further experiments with the HOT3D-Clips [2] dataset that includes larger object motion and precise groundtruth but collected in a lab setting. Motion statistics, along with the mean and standard deviation, of the object translation and rotation in the trajectories from both datasets can be seen in Fig. 4a.

Curated EpicKitchen Dataset Details. We curate a large-scale collection of object-centric 3D motion trajectories from egocentric videos using an automated eight-stage pipeline (Fig. 4b). From **76K** EPIC-Kitchens action segments, we retain **72K** short clips ($\leq$10s) with visible hands to ensure the presence of interactions. Object masks and 2D tracks are obtained using SAM2, yielding **229K** raw tracks before quality filtering reduces them to **112K**. Using TRELLIS, we reconstruct **71K** object meshes and obtain **59K** pose-aligned tracks. Sliding-window extraction produces **3.06M** raw (3+8)-step 3D trajectories, which are further filtered to **2.07M** high-quality trajectories, each consisting of a 0.13s window, used for training and evaluation.

HOT3D-Clips. We also train and evaluate **ObjectForesight** on HOT3D-Clips to validate that the model can learn future 3D motion from cleaner trajectories. For the HOT3D experiments, we skip frames to convert the clips to 6 fps and then extract the same (3+8)-frame windows used in our main setting, making

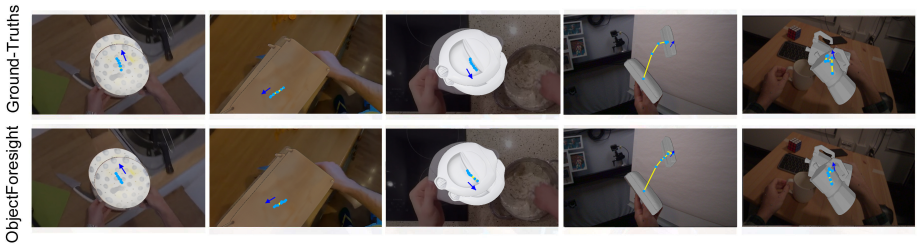


Fig. 5: Qualitative results from ObjectForesight. From the left, the first three columns are from the curated dataset from EpicKitchen and the final two columns are from the HOT3D-Clips dataset. For each sequence, we overlay 8 predicted poses on the last observed frame, illustrating the projected object centers, the transformed object mesh, and the general direction of the movement using an arrow. The images are zoomed in for clarity.

the trajectories 1.33s long. We also filter-out trajectories where the object is either stationary or the movement is negligible (less 0.01m translation or less than 5° rotation). This gives us **167K** high-quality trajectories with large object state change.

4.2 Baselines, Ablations, and Metrics

We compare four approaches. **ObjectForesight-DiT** is our diffusion transformer for multimodal trajectory prediction. **ObjectForesight-AR** is an autoregressive transformer variant that removes the diffusion formulation. **Constant Velocity** is a simple baseline that extrapolates future translation and rotation assuming the velocity observed in the context frames remains constant. Finally, **Video-generation (Luma Ray3)** is an off-the-shelf state-of-the-art video generator that synthesizes a short future clip from three context frames, from which we recover 6-DoF poses using our curation pipeline. Because this pipeline is computationally expensive, we evaluate it on 20 randomly selected validation videos with clear object visibility.

We report six trajectory-level metrics. For translation: **ADE** (average displacement error across all timesteps), **FDE** (displacement error at the final timestep), and **DES** (displacement error slope, capturing the per-timestep trend); all three are in meters. For rotation: **ARE** (average rotation error), **FRE** (final rotation error), and **RES** (rotation error slope); all three are in degrees. Additional ablations, such as varying the number of history frames, are provided in the supplementary material.

4.3 Qualitative Results

Fig. 5 shows that ObjectForesight produces smooth, physically plausible 6-DoF trajectories across diverse manipulation scenarios, capturing realistic interactions such as lifting, rotating, and placing objects while maintaining temporal

	ADE↓	FDE↓	DES↓	ARE↓	FRE↓	RES↓
Epic-Kitchens						
Constant Velocity	0.027	0.053	0.007	2.47°	5.60°	0.80°
ObjectForesight-AR	0.067	0.074	**0.002**	9.48°	12.58°	0.93°
ObjectForesight-DiT	**0.016**	**0.029**	0.004	**2.30°**	**4.82°**	**0.66°**
vs. Video Generation						
Luma AI Ray3	0.084	0.149	0.020	12.86°	20.90°	2.62°
ObjectForesight-DiT	**0.029**	**0.059**	**0.008**	**7.29°**	**13.98°**	**1.77°**
HOT3D-Clips						
Constant Velocity	0.136	0.280	0.04	38.70°	68.53°	9.85°
ObjectForesight-AR	0.055	0.082	0.007	9.80°	14.95°	1.55°
ObjectForesight-DiT	**0.021**	**0.026**	**0.003**	**8.92°**	**12.58°**	**1.16°**

Table 1: 3D trajectory forecasting on Epic-Kitchens and HOT3D-Clips. Lower is better for all metrics. ObjectForesight-DiT outperforms or matches ObjectForesight-AR across all metrics on both datasets, and substantially outperforms the Luma AI Ray3 video-generation baseline on Epic-Kitchens, highlighting the benefit of modeling motion directly in SE(3) rather than inferring it from synthesized frames. Note that the comparison with the video generation method is conducted on a subset of the dataset due to its computational complexity.

(a) Scene Encoders

	ADE↓	FDE↓	ARE↓	FRE↓
DGCNN [44]	0.0171	0.0297	2.333°	4.968°
PointNet++ [36]	0.0171	0.0298	2.357°	5.024°
SparseConv	0.0179	0.0295	2.700°	5.299°
No-Encoder	0.0174	0.0298	2.690°	5.380°
PTV3	**0.0165**	**0.0287**	**2.299°**	**4.816°**

(b) DiT Scaling

	ADE↓	FDE↓	ARE↓	FRE↓
6L-384D	0.0193	0.0311	4.242°	7.762°
8L-512D	0.0171	0.0294	2.802°	5.663°
12L-768D	**0.0165**	**0.0287**	**2.299°**	**4.816°**

Table 2: Ablation studies on Epic-Kitchens. (a) Scene encoder comparison with a fixed 12L-768D DiT. (b) DiT scaling with a fixed PTV3 encoder. Lower is better. L and D denote the number of layers and embedding dimension of the DiT, respectively.

coherence with the observed context. The examples span a range of motions, from simple translations of a plate, to the more complex rotation of a kettle, and the wiping action of the eraser on the board. For videos, refer to the supplementary material.

4.4 Quantitative Results

Table 1 reports all 6-DoF trajectory metrics on Epic-Kitchens and HOT3D-Clips. On Epic-Kitchens, ObjectForesight-DiT achieves the best translation and rotation accuracy overall, with an ADE of 0.016 m and an ARE of 2.30°. The autoregressive variant underperforms the Constant Velocity baseline on most metrics, suggesting that without the diffusion formulation the model struggles to capture the multimodal nature of future object motion. In contrast, ObjectForesight-DiT surpasses Constant Velocity on every metric, cutting ADE by 41% and FDE by 45%. On HOT3D-Clips, ObjectForesight-DiT again leads across all metrics

(0.021 m ADE, 8.92° ARE). The gap between Constant Velocity and the learned methods is considerably larger on this dataset, indicating that HOT3D-Clips contains more complex object motions that simple linear extrapolation cannot capture. In the video-generation comparison on Epic-Kitchens, ObjectForesight-DiT substantially outperforms Ray3 across all metrics (0.029 m vs. 0.084 m ADE; 7.29° vs. 12.86° ARE), reinforcing the benefit of predicting motion directly in SE(3) rather than inferring it from synthesized frames. We present these results in a separate section of the table, as this evaluation is conducted on a subset of the data due to the high computational cost and manual effort required by the video generation model.

We additionally ablate the scene encoder backbone and the DiT model scale (Table 2). For the encoder, we compare *PointTransformerV3* (PTV3), *DGCNN*, *PointNet++*, a simple sparse convolution network (*SparseConv*), and a variant with no scene encoding (*No-Encoder*). As shown in Table 2a, PTV3 achieves the best results across all metrics. SparseConv performs comparably to or worse than No-Encoder (e.g. 2.700° vs. 2.690° ARE), indicating that a poorly suited geometric backbone can negate the benefit of scene conditioning entirely, while a well-chosen encoder such as PTV3 provides meaningful gains. Table 2b further shows that scaling the DiT from 6L-384D to 12L-768D yields consistent improvements across metrics, reducing ARE from 4.242° to 2.299° and ADE from 0.0193 m to 0.0165 m, clearly confirming that the task benefits from increased model capacity.

5 Conclusion

We introduced the task of forecasting future 3D object motion directly from passive human videos, framing it as an object-centric SE(3) trajectory prediction problem. To support this task, we constructed a large-scale dataset through automated segmentation, tracking, monocular reconstruction, and pose alignment, yielding millions of metrically grounded trajectories across diverse everyday manipulations without requiring any manual annotation or motion capture. **ObjectForesight** combines monocular geometry, recent motion history, and local scene structure within a diffusion-based transformer to produce multimodal, physically consistent trajectory predictions. The diffusion formulation is central to this capability, allowing the model to capture the inherent uncertainty of future object motion rather than collapsing to a single deterministic output. Experiments show that **ObjectForesight** outperforms autoregressive and video-generation baselines across different metrics.

Our current formulation is limited to rigid objects and short prediction horizons. Natural extensions include adopting more expressive representations such as articulated kinematic models or learned deformation fields, exploring longer prediction horizons, and integrating with downstream planning or manipulation policies. More broadly, our results establish a foundation for scalable, object-centric 3D dynamics modeling and point toward richer predictive models of physical interaction.

References

1. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025) 4
2. Banerjee, P., Shkodrani, S., Moulon, P., Hampali, S., Han, S., Zhang, F., Zhang, L., Fountain, J., Miller, E., Basol, S., Newcombe, R., Wang, R., Engel, J.J., Hodan, T.: HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. CVPR (2025) 11
3. Bao, C., Xu, J., Wang, X., Gupta, A., Bharadhwaj, H.: Handsonvlm: Vision-language models for hand-object interaction prediction. Transactions on Machine Learning Research (2025) 4
4. Bharadhwaj, H., Mottaghi, R., Gupta, A., Tulsiani, S.: Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In: ECCV (2024) 4
5. Brahmbhatt, S., Handa, A., Hays, J., Fox, D.: Contactgrasp: Functional multi-finger grasp synthesis from contact. In: IROS (2019) 4
6. Chen, K., Ramanan, D., Khurana, T.: Using diffusion priors for video amodal segmentation. In: CVPR (2025) 7
7. Chen, M., Wang, Y., Li, Z., Bharadhwaj, H., Chen, Y., Qin, C., Kou, Z., Tian, Y., Whitmire, E., Sodhi, R., et al.: Flowing from reasoning to motion: Learning 3d hand trajectory prediction from egocentric human interaction videos. arXiv preprint arXiv:2512.16907 (2025) 4
8. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. arXiv preprint arXiv:2511.16624 (2025) 3
9. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. ECCV (2018) 2, 3
10. Das, P., Xu, C., Doell, R.F., Corso, J.J.: A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. CVPR (2013) 3
11. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. In: ICCV (2023) 3
12. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3d hand shape and pose estimation from a single rgb image. In: CVPR (2019) 3
13. Goyal, M., Modi, S., Goyal, R., Gupta, S.: Human hands as probes for interactive object understanding. CVPR (2022) 4
14. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. CVPR (2017) 2, 3
15. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. CVPR (2022) 3
16. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In: CVPR (2023) 4

17. Hafner, D., Lillicrap, T.P., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019) 4
18. Harley, A.W., You, Y., Sun, X., Zheng, Y., Raghuraman, N., Gu, Y., Liang, S., Chu, W.H., Dave, A., You, S., et al.: Alltracker: Efficient dense point tracking at high resolution. In: ICCV (2025) 3
19. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C.: Learning joint reconstruction of hands and manipulated objects. In: CVPR (2019) 3
20. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. CVPR (2020) 3
21. Hu, Y., Hugonot, J., Fua, P., Salzmann, M.: Segmentation-driven 6d object pose estimation. CVPR (2019) 3
22. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5 d heatmap regression. In: ECCV (2018) 3
23. Jeannerod, M.: Neural simulation of action: a unifying mechanism for motor cognition. Neuroimage **14**(1) (2001) 2
24. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: ECCV (2024) 3
25. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. ICCV (2017) 3
26. Kipf, T., van der Pol, E., Welling, M.: Contrastive learning of structured world models. In: ICLR (2020) 4
27. Li, K., Li, P., Liu, T., Li, Y., Huang, S.: Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In: CVPR (2025) 4
28. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. ECCV (2018) 3
29. Liu, S., Tripathi, S., Majumdar, S., Wang, X.: Joint hand motion and interaction hotspots prediction from egocentric videos. In: CVPR (2022) 4
30. Mo, K., Guibas, L.J., Mukadam, M., Gupta, A., Tulsiani, S.: Where2act: From pixels to actions for articulated 3d objects. CVPR (2020) 4
31. Mottaghi, R., Bagherinezhad, H., Rastegari, M., Farhadi, A.: Newtonian image understanding: Unfolding the dynamics of objects in static images. In: CVPR (2015) 4
32. Nagarajan, T., Feichtenhofer, C., Grauman, K.: Grounded human-object interaction hotspots from video. CVPR (2019) 4
33. Padalkar, A., Pooley, A., Jain, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Singh, A., Brohan, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models. arXiv preprint arXiv:2310.08864 (2023) 2
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. ICCV (2023) 2, 8
35. Perez, E., Strub, F., de Vries, H., Dumoulin, V., Courville, A.C.: Film: Visual reasoning with a general conditioning layer. In: AAAI (2018) 8
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017) 13
37. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: ICLR (2025) 2, 3, 6
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 9

39. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. arXiv preprint arXiv:2008.08324 (2020) 3
40. Shan, D., Geng, J., Shu, M., Fouhey, D.: Understanding human hands in contact at internet scale. In: CVPR (2020) 2, 4
41. De la Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., Beltran, P.: Guide to the carnegie mellon university multimodal activity (cmu-mmac) database (2009) 4
42. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: ICLR (2017) 4
43. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from static images using variational autoencoders. In: ECCV (2016) 4
44. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (2019) 13
45. Wen, B., Yang, W., Kautz, J., Birchfield, S.T.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: CVPR (2023) 7
46. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler, faster, stronger. In: CVPR (2024) 2, 8
47. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: CVPR (2025) 2, 3, 6
48. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv (2018) 3
49. Xiao, Y., Wang, J., Xue, N., Karaev, N., Makarov, Y., Kang, B., Zhu, X., Bao, H., Shen, Y., Zhou, X.: Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In: ICCV (2025) 2, 3, 7
50. Yoshida, T., Kurita, S., Nishimura, T., Mori, S.: Generating 6dof object manipulation trajectories from action description in egocentric vision. In: CVPR (2025) 4
51. Zhang, L., Zhou, S., Stent, S., Shi, J.: Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. In: ECCV (2022) 2, 6
52. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR (2019) 5
53. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025) 6
54. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: CVPR (2017) 3

6 Appendix

Contents in the supplementary

Please refer to our website https://objectforesight.github.io/ for detailed qualitative results and videos. In the subsequent sections of this appendix, we elaborate on our dataset curation of 3D object poses from monocular egocentric videos, provide additional ablation studies that complement the results in the main paper, and expand on how we use scene context in our method.

A Additional Details of the Data Curation Pipeline

We elaborate on the stages of the data curation pipeline summarized in Sec. 3.2, focusing on the heuristics, constraints, and cross-stage checks that improve data quality for pose trajectories.

A.1 Presence Filtering and Initialization

To mitigate false segmentations from EgoHOS, we aggregate interaction signals over each clip. We apply run-length smoothing to binary hand/object presence indicators using a threshold proportional to the clip length. This process fills brief detection gaps and eliminates false short-duration positives. The resulting smoothed signals serve two critical functions: they act as execution gates to ensure downstream modules run only when targets are reliably present, and they guide the SAM initialization towards temporally stable windows, reducing error propagation without introducing long-term drift.

A.2 Robust 2D Tracking

We augment the standard SAM2 tracking pipeline with a multi-stage regularization protocol that promotes temporal stability and suppresses duplicate object instances.

Point Sampling Strategy. To initialize and guide the model, we employ a robust sampling strategy. Positive points are sampled from the segmented object mask. To prevent mask leakage into the surrounding context, we explicitly sample negative points from three regions: detected hand masks, other object masks (if present), and a dilated background band surrounding the target object's mask.

Temporal Stability and Consensus. To mitigate per-frame segmentation noise, we construct a short-window consensus mask. When individual frame proposals are noisy, this consensus serves as a high-confidence positive prior. Furthermore, we apply mild morphological opening and closing to eliminate isolated speckles and smooth boundaries.

Trajectory Linking and De-duplication. We associate object components across frames using greedy Intersection over Union (IoU) matching. To handle

brief occlusions or detection failures, we permit a small gap tolerance in the temporal sequence. Tracks that fail to meet a minimum length requirement are discarded as noise. Simultaneously, we perform de-duplication within each video clip. If a new object proposal overlaps with an existing active track above a defined IoU threshold within a short temporal window, it is rejected. This ensures that the system maintains unique, distinct identifiers for each object instance.
Initialization. When multiple seeds are available, we prioritize candidates with the largest temporally stable area. Propagation is executed bidirectionally to maximize tracking duration.

A.3 Quality Filtering and Selection

We implement a two-stage filtering protocol to ensure only viable candidates reach the reconstruction stage.
Manipulation Gate. We employ a strict video-level gate using InternVL3 to filter out static or irrelevant objects. This module operates on object-highlighted visual summaries derived from the input track, rather than raw frames. Only tracks exhibiting active manipulation are retained.
Clean-View Selection. For the remaining valid tracks, we categorize frames into *Partial/Invalid* (occluded, blurred, or insufficient resolution) and *Clean* (unambiguous shape). Only clean frames are selected for geometry estimation. Input crops include a context margin to preserve local semantic cues.

A.4 Reconstruction Preparation

We prepare the data for 3D reconstruction through a sequence of filtering and completion steps.
Frame Selection and Background Removal. For the TRELLIS model, we select optimal "clean" frames based on foreground area size, excluding statistical outliers to maximize geometric consistency. Background clutter is masked out to isolate the object on a neutral canvas, enhancing texture and shape recovery.
Amodal Mask Generation. Separately, we use Diffusion-VAS to generate amodal masks. The segmentation masks contain holes or cutouts wherever the object is blocked by hands or other interactions. Diffusion-VAS corrects this by estimating the complete, physical shape of the object, filling in the missing regions. This ensures that we recover the full object silhouette, which is essential for accurate pose estimation and tracking in later steps of the pipeline.

A.5 Pose Estimation and Tracking

We adapt FoundationPose to recover robust 6DoF object trajectories, utilizing camera intrinsics, extrinsics, and dense depth maps provided by SpaTrackerV2. We add the following specific safeguards:
Scale Estimation and Locking. To handle monocular scale ambiguity, we lock the mesh diameter after the initial depth-to-mesh alignment. Subsequent

residuals are normalized by this fixed diameter to ensure consistent error scoring across objects of varying sizes.

Initialization Stress-Test. To prevent tracking failures from the start, we do multi-view initialization of object pose. Each potential initial frame undergoes a brief "refine-and-validate" optimization loop that jointly minimizes depth alignment error and maximizes silhouette consistency. Initial views yielding high depth alignment errors or silhouette inconsistencies are rejected.

Bidirectional Tracking and Re-registration. Tracking proceeds bidirectionally (forward and backward) from the optimal seed, with the estimator explicitly re-centered at the anchor frame before each pass. To detect and correct drift, we compute a suite of complementary consistency terms at every step:

- **Silhouette Metrics:** We monitor Intersection over Union (IoU) with specific penalties for *overflow* (mesh projection exceeding the mask) and *underfill* (mesh projection failing to cover the mask).
- **Geometric Residuals:** We track the error between the rendered mesh depth and the observed sensor depth.
- **Motion Monitors:** We apply conservative thresholds on rotation and translation deltas to flag physically implausible jumps.

Re-registration is triggered by compounded evidence from these metrics, allowing the system to curb drift under heavy occlusions or rapid egocentric motion.

B Window-Size Ablation Studies

In this section, we conduct a series of ablation studies to evaluate the contribution of different values of context length (C) and prediction horizon (H) and validate the design choices of our proposed framework.

B.1 Ablation Study on Context Length

C	ADE↓	FDE↓	ARE↓	FRE↓
1	0.026	0.038	7.97°	12.36°
2	0.021	0.033	7.61°	12.14°
3	**0.016**	**0.029**	**2.30°**	**4.82°**
5	0.025	0.035	7.69°	11.68°
10	0.027	0.038	8.09°	12.21°

Table 3: Ablation studies on the number of context frames. We evaluate the impact of context length C on pose prediction accuracy with a fixed prediction horizon of $H = 8$.

To determine the optimal temporal receptive field for our method, we conducted an ablation study on the number of input context frames C. We evaluated the model's performance by varying $C \in \{1, 2, 3, 5, 10\}$ while maintaining a fixed prediction horizon of $H = 8$. To ensure a consistent evaluation benchmark across

all configurations, the validation set was constructed using the maximum context length ($C = 10$). For models trained with shorter contexts, we trimmed the input sequences accordingly, ensuring that all models predicted the exact same target frames based on the appropriate historical window. The results of this experiment are summarized in Table 3.

As illustrated in Table 3, we observe that increasing the context information initially improves prediction accuracy. The performance improves significantly as C increases from 1 to 3, with $C = 3$ achieving the lowest error rates across the majority of metrics, including an ADE of 0.018 and an ARE of 7.03°. This suggests that a context of three frames provides sufficient historical information to effectively capture the object's immediate trajectory and rotational dynamics.

However, increasing the context length beyond this point ($C = 5$ and $C = 10$) results in a performance degradation. For instance, at $C = 10$, the ADE regresses to 0.027, and the ARE increases to 8.09°. We attribute this decline to two primary factors. First, longer context sequences are more susceptible to accumulated noise, which can distract the model from the most relevant recent motion cues. Second, an excessively long history may cause the model to overfit to past trajectories, hindering its ability to generalize to dynamic changes in pose movements or sudden shifts in direction. Consequently, we adopt $C = 3$ as the default setting for our main method.

B.2 Ablation Study on Prediction Horizon

	Eval @ $H = 4$		Eval @ $H = 8$		Eval @ $H = 16$		Eval @ $H = 32$	
Train H	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
4	0.016	0.023	-	-	-	-	-	-
8	**0.098**	**0.016**	**0.016**	**0.029**	-	-	-	-
16	0.015	0.020	0.022	0.034	0.034	0.055	-	-
32	0.018	0.022	0.023	0.031	**0.032**	**0.049**	**0.050**	**0.083**

Table 4: Translation Error Analysis. Comparison of ADE and FDE across models trained with different horizon lengths (H). Lower is better. Missing values (-) indicate the model cannot predict to that horizon.

	Eval @ $H = 4$		Eval @ $H = 8$		Eval @ $H = 16$		Eval @ $H = 32$	
Train H	ARE↓	FRE↓	ARE↓	FRE↓	ARE↓	FRE↓	ARE↓	FRE↓
4	1.51°	2.39°	-	-	-	-	-	-
8	**1.10°**	**1.95°**	**2.30°**	4.82°	-	-	-	-
16	1.46°	2.25°	2.60°	4.97°	5.11°	8.98°	-	-
32	1.87°	2.40°	2.77°	**4.68°**	**4.94°**	**8.46°**	**9.51°**	**15.67°**

Table 5: Rotation Error Analysis. Comparison of ARE and FRE across models trained with different horizon lengths (H). Lower is better.

We further analyze the impact of the prediction horizon H by training separate models with $H \in \{4, 8, 16, 32\}$ and a fixed input context length $C = 3$.

To ensure a fair comparison, the validation set is constructed using the maximum horizon ($H = 32$); for models with shorter output capabilities, we crop the ground truth sequences to match their respective prediction lengths (4, 8, or 16 frames). This setup allows us to evaluate how training on different temporal lengths affects performance at various evaluation horizons.

Table 4 and Table 5 summarize the results for translation (ADE/FDE) and rotation (ARE/FRE) errors, respectively. Columns indicate the evaluation horizon used, while rows represent the model's training configuration.

The results highlight a clear trade-off between short-term precision and long-term capability. Interestingly, the model trained with $H = 8$ outperforms the model trained with $H = 4$ when evaluated at the shorter horizon of $H = 4$ (e.g., ADE decreases from 0.0161 to 0.0098). This suggests that training on a slightly longer horizon encourages the network to learn more robust motion dynamics, acting as a form of regularization that benefits short-term accuracy.

However, blindly increasing the training horizon is not always beneficial. The model trained with $H = 32$ exhibits significantly higher error rates at shorter horizons ($H = 4, 8$) compared to the $H = 8$ model. This degradation likely stems from the optimization difficulty; the loss function for $H = 32$ is averaged over a long sequence where errors naturally accumulate, potentially diluting the gradients for earlier frames. Conversely, for long-term predictions ($H = 16$ and $H = 32$), the model explicitly trained on the larger horizon ($H = 32$) yields the superior performance. This is expected, as models trained with shorter horizons optimize for immediate accuracy and lack the supervisory signal required to maintain trajectory consistency over extended periods. Without the long-term loss component, these models suffer from severe error accumulation (drift) when extrapolating beyond their training window. The $H = 32$ model, by contrast, learns to model global temporal dependencies, effectively trading off some short-term precision for long-term stability.

C Scene Encoding Pipeline

We now describe how a raw depth observation is converted into the global scene embedding that conditions the temporal pose predictor. The pipeline proceeds in four stages: point cloud construction, per-point feature design, context-aware pooling, and conditioning injection into the diffusion transformer.

C.1 Point Cloud Generation and Preprocessing

Given a depth map and the associated camera intrinsics, we back-project each valid depth pixel into a 3D point in the camera coordinate frame using the standard pinhole model. The resulting points are then transformed into anchor-frame camera coordinates using the camera extrinsics at the anchor frame t_a. To obtain a fixed-size input, we first randomly subsample down to a coarse cap, then apply voxel-grid downsampling with a cell size of 0.005 m to ensure approximately uniform spatial coverage. If the number of surviving points falls

below a target count of N=4096, we restore it by interpolating between randomly selected point pairs. The output is a point cloud $\mathbf{X} \in \mathbb{R}^{N\times 3}$ with metric anchor-frame coordinates, independent of the original depth resolution or scene density.

C.2 Object-Centric Dual-Coordinate Features

Rather than supplying only anchor-frame XYZ coordinates as input features, we augment each point with its position expressed in the object's local reference frame. Given the 6-DoF object pose at the anchor frame, $\mathbf{T}_{\text{cam}}^{\text{obj}} \in SE(3)$, we compute the inverse $\mathbf{T}_{\text{obj}}^{\text{cam}} = (\mathbf{T}_{\text{cam}}^{\text{obj}})^{-1}$ and transform each point to obtain its object-centric coordinates. The resulting 6D per-point feature vector is:

$$\mathbf{f}_i = \left[x_i^{\text{cam}},\, y_i^{\text{cam}},\, z_i^{\text{cam}},\; x_i^{\text{obj}},\, y_i^{\text{obj}},\, z_i^{\text{obj}} \right], \tag{2}$$

where the first three components are the anchor-camera coordinates and the latter three encode the same point's position relative to the object's center and orientation. This dual representation provides the backbone with an explicit geometric prior: the object-frame channel enables it to distinguish points on the object surface from those in the surrounding scene without having to learn this invariance from data alone. These 6D features, together with the anchor-frame coordinates used for spatial indexing, are passed to a PointTransformerV3 backbone, which produces a per-point feature $\mathbf{h}_i \in \mathbb{R}^d$ for each input point.

C.3 Context Vector

To condition both the point cloud backbone and the downstream pooling on the object's recent motion history, we construct a context vector $\mathbf{ctx} \in \mathbb{R}^{64}$. For each frame k in the conditioning sequence $1{:}t_a$, we concatenate the 9D pose token $\mathbf{p}_k$ with the normalized bounding box $\mathbf{B}_k$ to form a 13D descriptor $[\mathbf{p}_k, \mathbf{B}_k] \in \mathbb{R}^{13}$. A shared linear layer projects each descriptor into a 64-dimensional embedding. The resulting sequence is aggregated via anchor-query cross-attention: the anchor token (frame t_a) serves as the query and attends to all t_a conditioning tokens, with sinusoidal positional encodings based on the relative temporal offset of each frame from the anchor. This produces a single 64D vector that summarizes the object's recent trajectory and visual extent. The context vector is supplied to PTv3's adaptive normalization layers (PDNorm) during backbone processing, and is reused in the pooling stage described next.

C.4 Object-Aware Attentive Pooling

After backbone processing, per-point features are linearly projected to the model's embedding dimension (d=768). We aggregate these into the global scene embedding $\mathbf{z}_{\text{geom}}$ using an *object-aware attentive pooling* mechanism that combines content-based and geometry-based cues:

- **Content score.** A query vector $\mathbf{q}$ is obtained by projecting **ctx** into the embedding space. The content logit for point i is the scaled dot product $s_i^{\text{cnt}} = \mathbf{h}_i^\top \mathbf{q} \,/\, \sqrt{d}$.
- **Distance score.** Each point's post-backbone coordinate is transformed into the object frame, and its distance from the object origin is computed. A small MLP with a learnable temperature τ maps this distance to a scalar bias: $s_i^{\text{dist}} = \text{MLP}\big(-\|\mathbf{x}_i^{\text{obj}}\| \,/\, \exp(\tau)\big)$.
- **Aggregation.** The final score $s_i = s_i^{\text{cnt}} + s_i^{\text{dist}}$ is normalized via softmax over all points within each sample, and the global feature is computed as the attention-weighted sum $\mathbf{z}_{\text{geom}} = \sum_i w_i \, \mathbf{h}_i$.

By combining semantic relevance (content score) with spatial proximity to the object (distance score), this pooling directs the encoder's attention toward the object and its immediate surroundings while retaining access to the broader scene context.

C.5 Scene Conditioning via AdaLN-Zero

The scene embedding $\mathbf{z}_{\text{geom}}$ is injected into the diffusion transformer via Adaptive Layer Normalization with zero initialization (AdaLN-Zero). Specifically, $\mathbf{z}_{\text{geom}}$ is projected and passed through a small MLP, then fused with the diffusion timestep embedding via a learned combination MLP to produce a single conditioning vector $\mathbf{c}_{\text{comb}} \in \mathbb{R}^{768}$. Each transformer block contains a per-layer MLP that maps $\mathbf{c}_{\text{comb}}$ to six modulation parameters: a scale γ, shift β, and gate α for both the self-attention and feed-forward sub-layers. The layer normalization in each block carries no learnable affine parameters; modulation is instead applied as $\hat{\mathbf{h}} = (1+\gamma)\,\text{LN}(\mathbf{h}) + \beta$, with the sub-layer output scaled by the gate α. All gate parameters are zero-initialized so that each block begins as an identity function, promoting stable early training. This design ensures that the scene geometry conditions the denoising process deeply and uniformly at every layer, rather than through a single additive bias.