EgoSim: Egocentric World Simulator for Embodied Interaction Generation

Jinkun Hao[1*], Mingda Jia[2*], Ruiyan Wang[1], Xihui Liu[3], Ran Yi[1†], Lizhuang Ma[1†], Jiangmiao Pang[2], and Xudong Xu[2]

[1] Shanghai Jiao Tong University
[2] Shanghai AI Laboratory
[3] The University of Hong Kong

[*]Equal contribution [†]Corresponding author

Abstract. We introduce EgoSim, a closed-loop egocentric world simulator that generates spatially consistent interaction videos and persistently updates the underlying 3D scene state for continuous simulation. Existing egocentric simulators either lack explicit 3D grounding, causing structural drift under viewpoint changes, or treat the scene as static, failing to update world states across multi-stage interactions. EgoSim addresses both limitations by modeling 3D scenes as updatable world states. We generate embodiment interactions via a Geometry-action-aware Observation Simulation model, with spatial consistency from an Interaction-aware State Updating module. To overcome the critical data bottleneck posed by the difficulty in acquiring densely aligned scene–interaction training pairs, we design a scalable pipeline that extracts static point clouds, camera trajectories, and embodiment actions from in-the-wild large-scale monocular egocentric videos. Extensive experiments demonstrate that EgoSim significantly outperforms existing methods in terms of visual quality, spatial consistency, and generalization to complex scenes and in-the-wild dexterous interactions, while supporting cross-embodiment transfer to robotic manipulation. Codes and datasets will be open soon. The project page is at `egosimulator.github.io`.

Keywords: Egocentric World Simulator · World Models · Generative Models

1 Introduction

The advances in video diffusion models [19,32,41] have significantly advanced the development of interactive world simulators [6,8,13,25,31], revolutionizing how we model and interact with virtual environments, which demonstrates strong potential for applications in spatial intelligence, game engines, and embodied AI. Although existing world simulators can generate realistic videos and construct immersive scenes, they only support restricted forms of action control like navigation, and typically render scenes from a third-person perspective, reducing

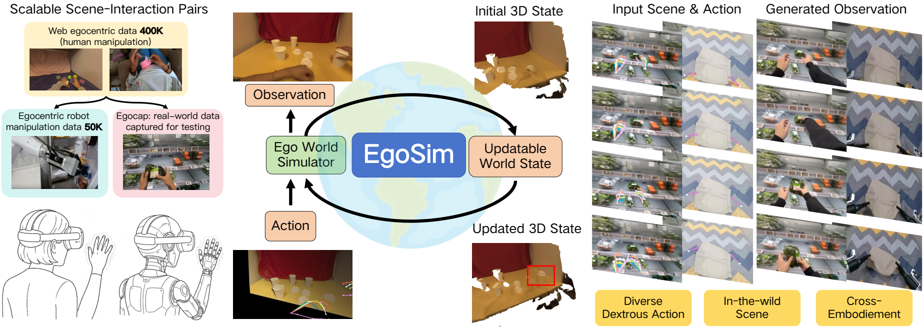


Fig. 1: Given a scene image and a sequence of actions, **EgoSim** generates temporally and spatially consistent egocentric observations and high-quality dexterous interactions. Egosim also persistently updates the 3D scene state for continuous simulation. We propose a data construction pipeline to leverage web-scale egocentric video data, and thus strengthen the generalization ability of Egosim with scalable scene-interaction pairs. EgoSim also exhibits strong few-shot adaptation capability to real-world scenarios and diverse robotic embodiments.

users to passive spectators. In contrast, humans and robots naturally function as active participants capable of performing far more diverse and flexible behaviors, such as fine-grained hand movements.

Recent studies [18, 33, 38] have begun to explore egocentric world simulators that support vivid interactions driven by human motions. Given an egocentric scene image or video as the world context, these methods typically take full-body or hand-only actions as conditions and synthesize corresponding interactive videos for world simulation. Despite achieving promising generation quality, such simulators still suffer from several critical limitations.

Some works [33, 38] leverage the implicit camera motion injection mechanism employed in video diffusion models, yet struggle to guarantee the 3D consistency of interaction videos and strict alignment between camera trajectories and generated frames. To circumvent this hurdle, DWM [18] explicitly decouples static 3D scenes from the dynamic changes induced by user actions. However, all of them ignore or fail to maintain a fundamental capability of any world simulator, consistent world state updating, especially during long-sequence generation. Furthermore, current world simulators are only trained on restricted data with aligned multiview videos [10, 11, 33] or data collected from synthetic environments [17, 18], which are orders of magnitude smaller than web-scale monocular egocentric videos. Undoubtedly, this severely limits the scalability of such models and consequently compromises their generalization ability.

In this paper, we present **EgoSim**, an egocentric world simulator as shown in Fig 1 that models 3D static scenes as *updatable* world states. Inspired by DWM [18], we also disentangle static 3D scenes from dynamic changes and formulate EgoSim as an ego-motion-action-aware observation simulation model generating egocentric interaction videos. In contrast to prior work, we first take the initial egocentric frame, remove hands via inpainting, and further reconstruct the point cloud of static scenes to construct the spatial condition of Egosim. Thanks to the editable nature of point clouds, EgoSim can conveniently memorize and update a persistent world state immediately after generating interaction observations from input actions, enabling continuous world simulation with supervision from a spatial-temporally persistent scene state. Specifically, we identify objects interacting with human or robotic arms and update their corresponding point clouds based on their latest observation changes. In contrast to existing approaches that either reconstruct only static scenes [42] or simply track moving objects [15], we first propose an Interaction-aware State Updating module that enables handling a wider range of interaction scenarios, including the manipulation of fixed articulated objects and multi-part assembly tasks.

To enable the scaling of training data, we propose an elegant data construction pipeline for processing massive in-the-wild monocular egocentric videos. Notably, EgoSim inherently requires *well-aligned paired data*, including a static 3D scene, camera trajectory, hand motion representations, and the egocentric video itself as the training target. Accordingly, given a monocular egocentric video, our pipeline first obtains the static 3D scene using the aforementioned procedure. We then leverage DepthAnything3 [23] to estimate camera trajectories from the video, and employ HaMeR [26] to extract hand keypoint sequences as motion conditions of EgoSim. It is noteworthy that we adopt hand keypoints rather than hand meshes, as this choice facilitates the adaptation of our model to any embodiment, such as robotic grippers.

Through extensive experiments, we validate the effectiveness of our proposed explicit and updatable world states. Given a scene image and dexterous hand actions, EgoSim not only synthesizes realistic interactive videos with great spatial consistency, but also successfully updates 3D scenes to maintain consistent world model transitions. Thanks to our well-designed data construction pipeline, EgoSim can be trained on web-scale monocular egocentric videos and thus exhibits strong generalization ability. Specifically, EgoSim performs favorably in complex scenes with multiple interactive objects and is capable of generating diverse, sophisticated manipulations, including deformable object handling and small object assembling, which are unattainable by existing approaches. Moreover, EgoSim demonstrates strong adaptation capability. After finetuning only 100 steps on AgiBot data [3], it achieves promising performance on robotic arm manipulation [36]. We further design EgoCap, a low-cost data collection device, and collect 50 hand interaction video clips in real-world application scenarios. Experimental results demonstrate that EgoSim can rapidly adapt to these real-world interactive videos.

2 Related Work

2.1 Interactive Video Generation

The rapid progress of video diffusion models [19, 32, 41] has shifted world modeling from latent-space prediction [12] to high-fidelity visual simulation. Genie [2,6], The Matrix [8,13], and Lingbot-World [31] generate controllable video streams for game environments and embodied AI. However, these methods support only coarse controls like directional commands or camera poses.

Meanwhile, several methods explore hand-object interaction video generation with finer-grained control signals. InterDyn [1] conditions on binary hand masks via a ControlNet-like branch to probe the implicit physical knowledge of pretrained video models. CosHand [29] generates single-frame hand-object interactions from hand mask inputs. Mask2IV [20] introduces a two-stage pipeline that first predicts interaction mask trajectories and then synthesizes trajectory-conditioned videos. SpriteHand [22] renders hands over static backgrounds with autoregressive generation for real-time interaction. However, these methods operate on monocular 2D signals without explicit 3D scene grounding, and none of them maintains a persistent world state for continuous simulation.

2.2 Egocentric World Simulators

Egocentric world simulators generate action-conditioned videos from a first-person perspective. PlayerOne [33] conditions on whole-body motion derived from synchronized ego-exocentric captures. Hand2World [38] distills a bidirectional video diffusion model into a causal autoregressive generator for monocular streaming synthesis. Both methods encode the scene state implicitly without explicit 3D representations, which limits spatial consistency under large viewpoint changes. Moreover, PlayerOne relies on synchronized ego-exocentric video pairs for training, which are difficult to scale. DWM [18] decouples the static 3D scene from action-induced dynamics by conditioning on rendered point maps and hand meshes, thereby improving spatial consistency. However, its scene is reconstructed once and never updated after interactions, and its training relies on synthetic environments or paired captures that are equally difficult to scale. EgoSim maintains an explicit, updatable scene state, enabling precise condition-following ability for both action and geometric inputs, even under large-scale and drastic view changes. EgoSim also benefits from large-scale real scene-interaction pairs in training data while generating complex and in-the-wild dexterous embodiment-object interactions.

2.3 World Models with Scene States

A fundamental requirement for a world simulator is to memorize and update the environment state from its generated observations. Recent 3D reconstruction methods [23, 35] can recover dense geometry from single-view images. Several works have explored updating representations by considering object dynamics in video sequences. VIPE [15] fuses per-frame point clouds through decoupling moving objects to maintain a clean static scene. Spatia [42] discovers utilizing motion-aware scene states as geometric prior to enhance video generation. However, they overlook other object-embodiment interactions that are more complex than simple motion. WristWorld [27] reconstructs coarse 4D scene points for enhancing the spatial consistency of robot world models. In contrast, EgoSim performs fine-grained interaction-aware state sensing, which explicitly tracks and updates complex object interactions. This interaction-aware state serves as a more appropriate spatial prior for interactive world simulators.

3 Method

EgoSim operates as a closed-loop world simulator that cyclically alternates between visual observation generation and 3D state updates to ensure temporal and physical consistency as illustrated in Figure 2. In this section, we first formalize the simulation process (Sec. 3.1). We then detail its two core components. The Geometry-action-aware Observation Simulation model responsible for state transitions (Sec. 3.2) and the Interaction-aware State Updating for long-term state persistence (Sec. 3.3).

3.1 Problem Formulation

We formulate the continuous world simulator by three core components: the environment **State (S)**, the interaction **Action (A)**, and the visual **Observation (O)**.

At any given stage k, the simulator maintains a 3D world state S_{k-1} representing the current scene state. The input action A_k is explicitly decoupled into a camera trajectory C_k and a hand interaction sequence H_k, expressed as $A_k = (C_k, H_k)$.

The simulation loop consists of two sequential operations. First, the model generates the intermediate visual observation O_k by rendering the static background $\Pi(S_{k-1}; C_k)$ and synthesizing the dynamic observation residuals $\Delta O(H_k)$ induced by the hand action:

$$O_k = \Pi(S_{k-1}; C_k) + \Delta O(H_k) \tag{1}$$

However, pure decoupled generation cannot preserve states across stages. To close the simulation loop, the system executes a 3D state update function $\mathcal{U}$ that extracts the latest physical layout from the generated observation O_k and persistently applies it back to the 3D scene:

$$S_k = \mathcal{U}(S_{k-1}, O_k) \tag{2}$$

By performing this localized state update, we prevent manipulated objects (e.g., an opened door) from reverting to their original layouts in subsequent renderings. This formulation naturally motivates our two core modules include the Observation Simulation Model (Sec. 3.2) and the interaction-aware State Updating (Sec. 3.3).

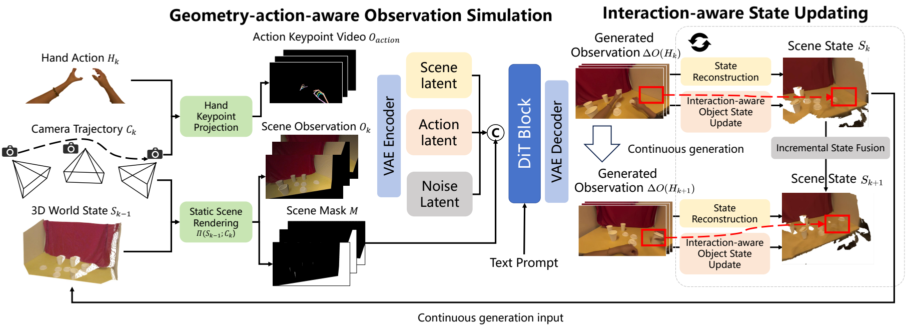


Fig. 2: Overview of EgoSim. Our framework enables continuous egocentric simulation via updatable world modeling. It consists of (1) a Geometry-action-aware Observation Simulation model that synthesizes action-conditioned visual dynamics O_k based on current scene geometry; and (2) an Interaction-aware state updating module that tracks point-based interaction-aware object states from generated observations and integrates them back into the persistent 3D world state to update S_k.

3.2 Geometry-action-aware Observation Simulation

The visual state transition is implemented through a geometry-action-aware video generation model acting as our Observation Simulation model. This model synthesizes the dynamic residuals driven by hand interactions, while ensuring strict adherence to the 3D environment.

Furthermore, to rigorously constrain the generation process, we formulate the conditioning signals by combining explicit scene geometry with action representations. The Scene Observation video O_k provides the background and accurate camera motion cues. Simultaneously, the interaction action is represented by a hand keypoint video O_{action}. Specifically, we extract 3D hand keypoints and project them onto the 2D observation plane using the corresponding camera intrinsics and extrinsics. This perspective projection enforces depth-dependent foreshortening and accurate geometry, anchoring the generated interactions in 3D space. We intentionally employ core joint keypoints rather than dense hand meshes, as keypoints serve as a universal, cross-embodiment action representation that facilitates generalization from human hands to robotic end-effectors. Furthermore, because O_k inevitably contains unrendered regions due to occlusions or incomplete scanning, we introduce a binary mask video M to explicitly indicate these unobserved regions that require visual synthesis.

Training & Inpainting Prior. During training, we operate in the latent space of a pretrained video VAE. The denoising network ϵ_θ, instantiated as a Diffusion Transformer (DiT), processes the concatenated latent $z_{in}^{(t)} = \text{Concat}(z_t, z_{bg}, z_{hand}, M)$ to predict the Gaussian noise ϵ, where all conditions are spatially aligned with the noisy latent z_t:

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{z_0, t, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon - \epsilon_\theta \left(z_{in}^{(t)}, t \right) \right\|_2^2 \right] \tag{3}$$

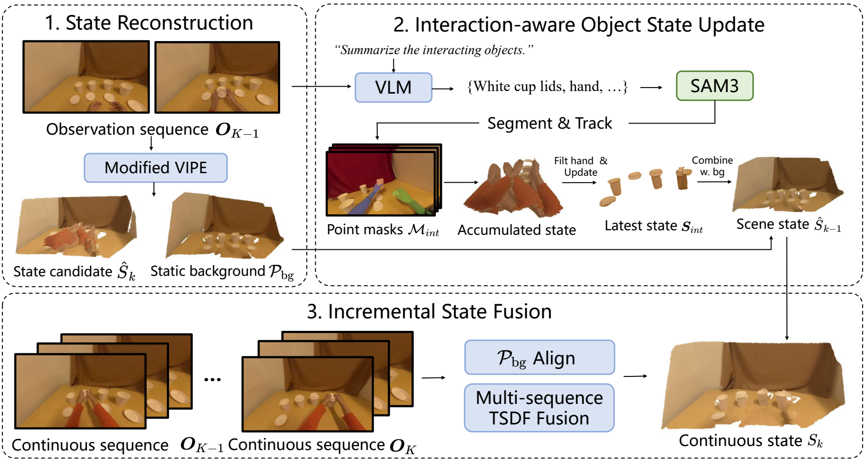

1. State Reconstruction
Observation sequence $\boldsymbol{O}_{K-1}$
Modified VIPE
State candidate $\hat{S}_k$
Static background $\mathcal{P}_{bg}$
2. Interaction-aware Object State Update
"Summarize the interacting objects."
VLM
{White cup lids, hand, ...}
SAM3
Segment & Track
Point masks $\mathcal{M}_{int}$
Accumulated state
Filt hand & Update
Latest state $\boldsymbol{S}_{int}$
Combine w. bg
Scene state $\hat{S}_{k-1}$
3. Incremental State Fusion
Continuous sequence $\boldsymbol{O}_{K-1}$
Continuous sequence $\boldsymbol{O}_{K}$
$\mathcal{P}_{bg}$ Align
Multi-sequence TSDF Fusion
Continuous state S_k

Fig. 3: Implementation details of Interaction-aware State Updating. State Reconstruction (left) builds a point cloud from observations via modified VIPE. Object State Update (top-right) identifies and tracks interactive objects with a VLM and SAM3, compositing their latest geometry into the scene. Incremental State Fusion (bottom) aligns and merges consecutive states via TSDF fusion.

We furthur initializing the DiT with pre-trained inpainting weights to address incomplete observations by hallucinating unmodeled regions in M. Furthermore, it acts as an identity function with a generative prior [18], encouraging the model to preserve the known background O_k and only activate generation in regions conditioned by actions in O_{action}.

3.3 Interaction-aware State Updating

State Reconstruction. To maintain a consistent 3D scene with updatable object states for continuous simulation, we design a training-free module to reconstruct the state candidate $\hat{S}_k$ by estimating and fusing segment-level point clouds from the K-th incoming observation sequence $\{O_0, O_1, \dots, O_{n-1}\} \subset \boldsymbol{O}_{K-1}$ with n frames, as an editable 3D environment representation. Inspired by VIPE [15], our pipeline first estimates camera intrinsics via DepthAnything3 [23] and extracts instance masks via SAM3 [5] guided by text phrases of interaction-related objects. Per-frame depths and camera poses are predicted and aligned using a dual-pass DROID-SLAM with multi-view depth alignment to resolve scale ambiguity. Building on this, the pipeline detects interactive objects via Grounding DINO [24] and SAM3 [5], and decouples them from static objects during reconstruction.

Interaction-aware Object State Update. To register and update the dynamic states $\boldsymbol{S}_{int}$ of interactive objects, we first employ a vision-language model [40]

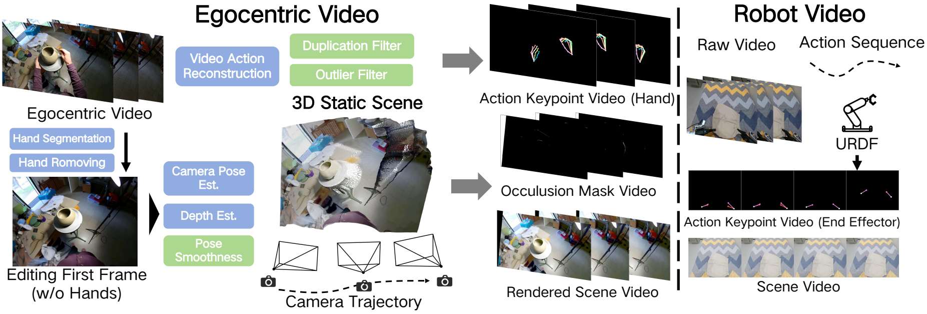


Fig. 4: Overview of our scalable data processing pipeline. For both human egocentric videos and robotic manipulation videos, we automate the extraction of aligned triplets: static 3D scene point clouds, precise camera trajectories, and dynamic interaction sequences represented as spatial action keypoints.

to identify object phrases involved in embodiment interactions. An open-vocabulary tracking pipeline based on SAM3 [5] then estimates a 3D point mask $\mathcal{M}_{int}^{i}$ for each interactive object $S_{int}^{i} \in \boldsymbol{S}_{int}$ through a hierarchical filtering process, we tag embodiment-centric instances as subject references, select mask candidates by IoU overlap, and refine them with depth consistency to suppress false positives. During reconstruction, point clouds excluding $\mathcal{M}_{int}$ are fused into a static background $\mathcal{P}_{\text{bg}}$, while interactive objects are tracked across frames and their latest-frame geometry is composited into $\mathcal{P}_{\text{bg}}$ to form the final $\hat{S}_k$.

Incremental State Fusion. During continuous generation, the scene state S_k of generated observation sequence $\boldsymbol{O}_K$ is obtained by incrementally fusing S_{k-1} and $\hat{S}_k$. We first align the coordinate systems of the two states via the Sim3 Umeyama algorithm computed from the overlapping camera poses C_k and C_{k-1}, then apply TSDF fusion to merge their point clouds, with overlapping regions updated from $\hat{S}_k$. Interactive objects retain their geometry from the last observed frame, yielding an up-to-date scene representation.

4 Scalable Data Pipeline for Interactive World Simulators

Training EgoSim requires well-aligned quadruplets: a static 3D scene, camera trajectory, embodiment-action sequence, and the corresponding interaction video. We design a fully automated pipeline (Sec. 4.1) that extracts such quadruplets from large-scale monocular egocentric and robotic videos, and further introduce EgoCap (Sec. 4.2), an intrinsics-free capture pipeline for low-cost real-world data acquisition with uncalibrated smartphones.

4.1 Automated Data Processing Pipeline

3D Static Scene Initialization. To acquire a clean, interaction-free 3D background, we extract the first frame of each egocentric video clip. We employ

SAM3 [5] to generate masks for hand regions, which are subsequently removed using the Qwen-Image-Editing [39] model. Finally, DepthAnything3 [23] provides monocular depth estimation, allowing us to unproject this 2D image into 3D space to form the initial scene point cloud.

Camera Trajectory Estimation and Static Rendering. To establish spatial geometric constraints, we process the original monocular video using DepthAnything3 [23] to extract per-frame camera parameters, yielding the rotation matrix, translation vector, and intrinsic matrix. By rendering the initial point cloud from the perspective of this estimated camera trajectory, we produce the rendered scene video that serves as a geometry-consistent reference for all subsequent frames.

Interaction Action Extraction and Rendering. We extract a universal, cross-embodiment action representation. For human hands, we employ HaMeR [26] to estimate pose sequences. We also apply duplication filters and outlier filters to enhance the annotation quality. For robotic end-effectors, we compute precise 3D action representations using recorded joint trajectories and robot URDF files. Using the extracted camera parameters, these 3D keypoints are projected onto the 2D camera plane. This operation explicitly defines the spatial action condition.

Dataset Construction. For training Egosim, we use our pipeline constructs a large-scale dataset with high-quality scene-interaction pairs. More details can be found in Table 5 in our supplementary.

4.2 EgoCap: Real-World Data Collection Pipeline

Traditional collection of aligned paired data relies on rigidly calibrated multi-camera arrays, which are expensive and hard to scale. We introduce EgoCap, an intrinsic-free data collection pipeline that uses uncalibrated smartphones to produce viewpoint-aligned paired data at low cost.

Uncalibrated Scene Scanning. In the first step, a user scans the environment with an uncalibrated mobile device. Our system builds a global 3D Gaussian Splatting map using the ARTDECO [21] streaming reconstruction framework, with a dynamic intrinsic self-calibration module that eliminates the need for prior calibration, yielding a scale-consistent scene point cloud.

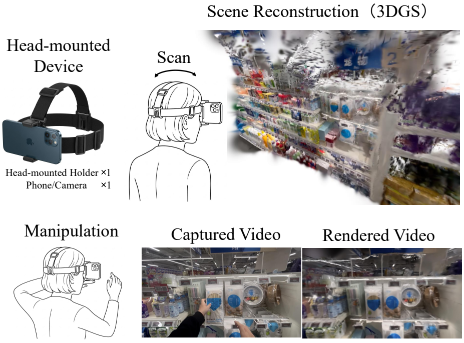


Fig. 5: The EgoCap pipeline. An uncalibrated head-mounted smartphone first scans the scene to build a 3DGS map, then records the ego-view interaction. Relocalizing against the map yields viewpoint-aligned paired data.

Dynamic Interaction Recording and Relocalization. The user then records the egocentric interaction with the same device. Despite unknown focal lengths, occlusions, and motion blur, a dense matching localizer continuously tracks visual features against the pre-built map to recover the 6-DoF camera trajectory.

Continuous Trajectory Optimization and Aligned Rendering. To suppress localization noise and frame discontinuities, we interpolate translations with cubic splines and rotations with SLERP, then globally smooth the trajectory using a Savitzky-Golay filter. The refined trajectory is then used to re-render the 3D scene, producing a geometry-consistent reference video aligned with the interaction viewpoint.

5 Experiments

5.1 Experimental Setup

Datasets. We process 240K video clips from the EgoDex [14] dataset and 160K video clips from the EgoVid [37] dataset. The EgoDex subset features fine-grained interactions accompanied by precise action labels. Conversely, the EgoVid subset consists of in-the-wild interaction data. For evaluation, we randomly select 100 videos from each dataset to form our test set, ensuring these videos remain entirely unseen during training. All video clips used for training are uniformly processed to a length of 61 frames at 16 FPS.

Training Details Our backbone is initialized from Wan-2.1-Fun-14B-InP [32], generating 832 × 480 videos of 61 frames at 16 FPS. To incorporate action conditioning, we expand the DiT input channels to 52. We fully fine-tune the DiT for 4,000 steps using AdamW with a learning rate of 1×10^{-5} and a per-GPU batch size of 4 across 8 NVIDIA H200 GPUs. All other components including T5, VAE, and CLIP remain frozen. Inference uses Flow Matching with 50 steps and a CFG scale of 1.0. Further details are in the supplementary.

Baselines. We compare against four methods: (1) **Wan 2.1-14B-InP** [32], a mask-guided video inpainting model; (2) **InterDyn** [1], which injects hand mask video into Stable Video Diffusion via ControlNet; (3) **Mask2IV** [20], which conditions on both hand and object segmentation masks via latent channel-wise concatenation, enabling simultaneous modeling of hand and object dynamics; and (4) **CosHand** [30], which generates each frame independently from a reference image and per-frame hand masks. All baselines are extended to 61 frames for fair comparison.

Evaluation Metrics. We utilize a multi-dimensional metric suite to evaluate both visual quality and spatial controllability. For Video Quality, we measure Peak Signal-to-Noise Ratio (PSNR, ↑), Structural Similarity Index Measure (SSIM, ↑), and Learned Perceptual Image Patch Similarity (LPIPS, ↓). To evaluate spatial consistency, we adopt the metrics proposed in [38]. Specifically, Depth-ERR (↓) is utilized to measure 3D structural consistency by calculating the error between the depth maps of the generated video and the ground truth. Cam-ERR (↓) is employed to evaluate viewpoint consistency by computing the discrepancy in per-pixel Plücker ray embeddings.

Table 1: Quantitative comparison of EgoSim against baselines on the EgoDex and EgoVid test sets.

Method	EgoDex (Tabletop Scene)					EgoVid (In-the-wild)				
	PSNR ↑	SSIM ↑	LPIPS ↓	Depth-ERR ↓	Cam-ERR ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Depth-ERR ↓	Cam-ERR ↓
Wan-2.1-14B-InP	17.998	0.447	0.708	42.335	0.0300	11.754	0.430	0.503	34.470	0.0174
Mask2IV	20.622	0.814	0.299	38.339	0.0181	12.311	0.414	0.571	34.413	0.0175
CosHand	17.119	0.776	0.386	56.524	0.0499	11.805	0.408	0.600	39.373	0.0516
InterDyn	22.250	0.830	0.255	44.345	0.0226	14.612	0.466	0.484	38.180	0.0308
EgoSim (Ours)	**25.056**	**0.896**	**0.170**	**8.888**	**0.0013**	**16.684**	**0.509**	**0.421**	**19.260**	**0.0105**

Table 2: Quantitative comparison of continuous generation on EgoDex.

Model	PSNR ↑	SSIM ↑	LPIPS ↓	Depth-ERR ↓	Cam-ERR ↓
EgoSim	**25.056**	**0.896**	**0.170**	**8.888**	**0.0013**
EgoSim (Continuous)	19.165	0.835	0.220	10.943	0.0017

Evaluation Settings. We evaluate EgoSim under two settings. In the **Normal** (single-clip) setting, the model receives a ground-truth first frame together with the 3D state rendered from the ground-truth point cloud, and generates a single 61-frame video clip. All baselines are evaluated under this setting. In the **Continuous Generation** setting, only the first ground-truth frame of the entire task is provided. The model first generates a 61-frame clip, after which the Interaction-aware State Updating module (Sec. 3.3) reconstructs and updates the scene state from the generated observation; the updated state is then rendered as the spatial condition for generating the second clip, yielding a 121-frame sequence in total. We randomly select 20 full task segments from EgoDex for this evaluation.

5.2 Qualitative Results

Normal Settings. Figure 6 provides a detailed comparison of EgoSim against the baseline methods on two unseen test scenarios. In the *scooping ice* task (Figure 6a), EgoSim accurately simulates physical state changes and complex depth relationships, whereas InterDyn suffers from severe depth ambiguity and spatial misalignment. Mask2IV struggles to preserve the identity of hands and manipulated objects, leading to distorted interactions. CosHand, as an image generation method, produces plausible single frames but lacks temporal coherence across the sequence. In the in-the-wild *watercolor painting* task (Figure 6b), EgoSim synthesizes realistic interaction semantics and seamlessly hallucinates unobserved background regions into a coherent visual space, while Wan2.1-14B-InP acts as an identity mapping outside the mask, merely reconstructing incomplete point cloud inputs without capturing hand-object dynamics. Notably, none of the baselines can faithfully follow egocentric head motion, whereas EgoSim maintains consistent viewpoint transitions throughout the simulation.

Additionally, to validate the real-world data collection pipeline EgoCap, we capture 50 clips in a supermarket environment focusing on shelf interactions, allocating 30 for training and 20 for testing. The model is initialized from our

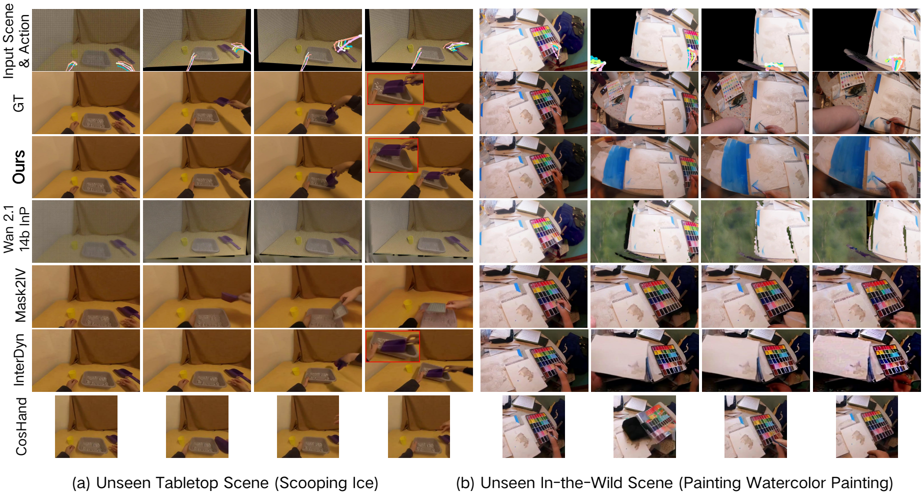

Input Scene & Action
GT
Ours
Wan 2.1 14b InP
Mask2IV
InterDyn
CosHand
(a) Unseen Tabletop Scene (Scooping Ice)
(b) Unseen In-the-Wild Scene (Painting Watercolor Painting)

Fig. 6: Qualitative comparison of EgoSim against state-of-the-art baselines on unseen test environments. (a) Inference on Egodex. (b) Inference on Egovid.

main checkpoint and fine-tuned for only 50 steps. As shown in Figure 8, even with this minimal fine-tuning data, EgoSim successfully synthesizes physically plausible hand-object interactions and consistent backgrounds on unseen test scenes, demonstrating efficient adaptation to novel real-world environments.

Continuous Generation. Figure 7 visualizes four continuous generation sequences. Notably, the 3D states of interacted objects are correctly updated and registered within the global scene after each generation stage. For instance, in the top examples, the white cup lids are correctly closed onto the corresponding cups during the interaction process. Similarly, in the bottom sequences, breads placed on plates in the simulated observations are accurately collected and maintained in the updated states. These results demonstrate that the closed-loop state update mechanism enables EgoSim to propagate interaction effects across clips, producing contextually coherent long-horizon simulations.

5.3 Quantitative Comparison

Normal Settings. Table 1 presents our quantitative evaluation under the single-clip setting across both the EgoDex and EgoVid test sets. The results confirm the substantial superiority of EgoSim in both visual quality and spatial controllability.

In terms of video quality, EgoSim significantly outperforms all baselines on both datasets. On the tabletop scenes of the EgoDex dataset, EgoSim achieves a PSNR of 25.056 and an SSIM of 0.896, demonstrating a substantial improvement over the mask-based InterDyn baseline. A similar trend is observed on the challenging, in-the-wild EgoVid dataset, where our method maintains the highest visual fidelity despite the complex background dynamics.

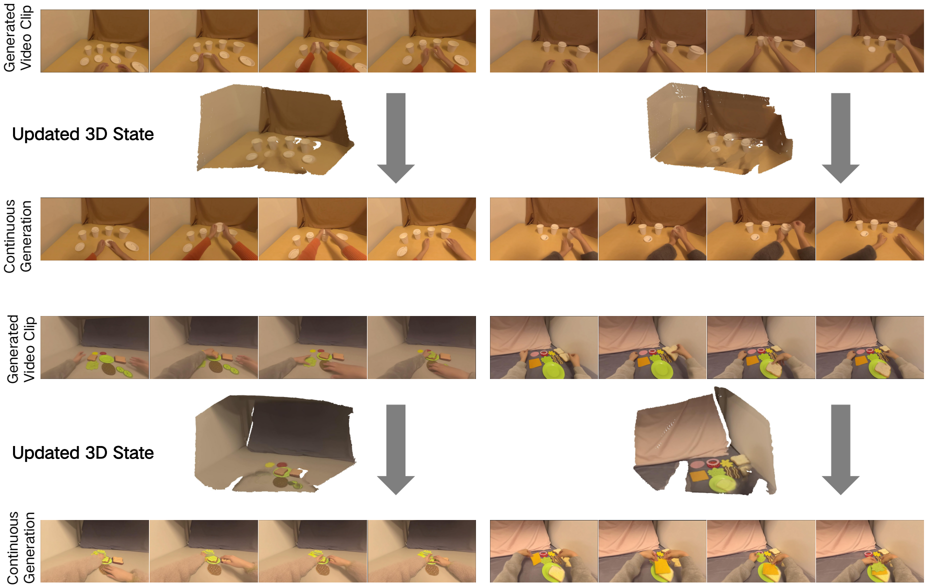


Fig. 7: More simulated observations under the continuous generation setting of EgoSim, EgoSim updates dynamic interaction-aware states from the rather generated sequence and uses it as the spatial condition for simulating the following sequences.

Crucially, in the spatial controllability metrics, EgoSim also outperforms the baselines. Conventional architectures like InterDyn frequently fail to decouple camera motion from hand motion, leading to severe background drifting and phantom occlusions, as reflected by their high Depth-ERR and Cam-ERR scores. By explicitly rendering point clouds along the true camera trajectory, EgoSim provides the generator with an unambiguous spatial anchor. This results in an overwhelming reduction in spatial errors. For instance, on the EgoDex dataset, our Depth-ERR is reduced to 8.888 compared to InterDyn with 44.345, and Cam-ERR is reduced by over an order of magnitude. These results unequivocally demonstrate that our spatial conditioning effectively anchors the 3D space, making EgoSim a highly spatially consistent, physically grounded simulator.

Continuous Generation. Table 2 reports the results under the continuous generation setting on EgoDex, where each 121-frame sequence is generated as two consecutive clips with an intermediate 3D state update.

As presented in Table 2, both video quality and spatial consistency remain robust. EgoSim (Continuous) achieves a PSNR of 19.165, an SSIM of 0.835, and an LPIPS of 0.220. While there is a marginal drop compared to the single-clip setting, as Depth-ERR increases from 8.888 to 10.943 and Cam-ERR from 0.0013 to 0.0017, this is primarily due to cumulative error from simulated artifacts and inherent noise within the updated states.

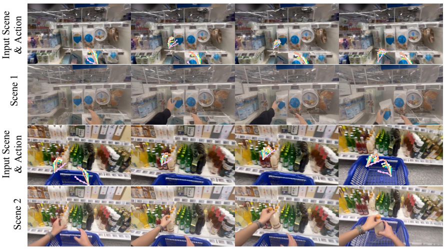


Fig. 8: Real-world generalization of EgoSim across diverse in-the-wild scenes.

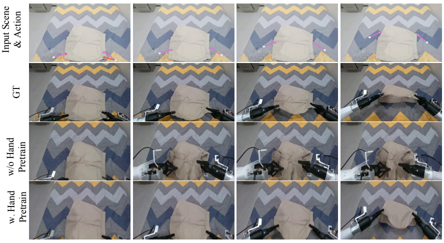


Fig. 9: Cross-embodiment simulation on AgiBot. *w/ pretrain* model simulates complex physical dynamics accurately.

Table 3: Ablation study on EgoDex.

Model	PSNR ↑	SSIM ↑	LPIPS ↓	Depth-ERR ↓	Cam-ERR ↓
w/o trajectory	23.380	0.845	0.244	10.238	0.0015
w/o mask	23.988	0.886	0.186	14.124	0.0022
EgoSim (Ours)	**25.056**	**0.896**	**0.170**	**8.888**	**0.0013**

Table 4: Cross-embodiment eval.

Setting	PSNR ↑	SSIM ↑	LPIPS ↓
w/o hand pretrain	15.180	0.662	0.350
w/ hand pretrain	**18.670**	**0.725**	**0.282**

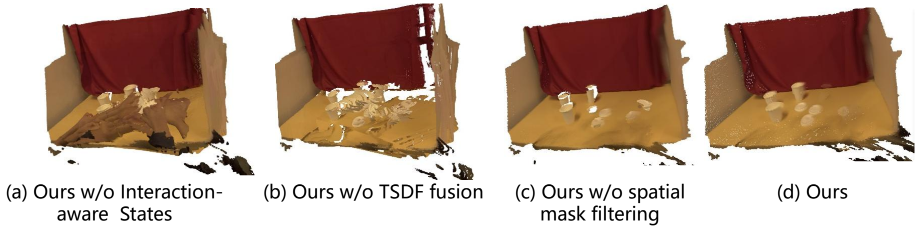
5.4 Ablation Study

To rigorously validate our architectural design choices, we conduct ablation studies on the fine-grained EgoDex dataset. The quantitative results are presented in Table 3.

Effect of Camera Trajectory Rendering. To verify the necessity of explicitly rendering point clouds along the true camera trajectory, we evaluate the *w/o trajectory* variant. By duplicating the initial static frame across all time steps, this setting ignores camera ego-motion. As shown in Table 3, this causes a significant drop in generation quality, with PSNR falling to 23.380. Without explicit physical constraints from the moving 3D point cloud, the model struggles to accurately capture background parallax and hallucinates changing scene geometry.

Effect of Mask Constraints. Feeding an all-black mask (*w/o mask*) still yields a high PSNR of 23.988, proving EgoSim acts as an *identity function with a generative prior* that intelligently synthesizes interactions without explicit region guidance. Nonetheless, providing accurate visibility masks (Ours) optimally blends dynamic interactions with hallucinated unobserved regions, achieving the best PSNR of 25.056.

Effect of Updatable Interaction-aware 3D States. We validate this mechanism and ablate its key components by visualizing the post-interaction point clouds from a novel *third-person perspective* (Figure 10). This explicitly proves that the physical displacement of objects is permanently registered in 3D space. Without interactive object filtering (Figure 10a), the static point cloud becomes corrupted by dynamic elements like hands, leading to ghosting artifacts. Removing TSDF fusion (Figure 10b) produces fragmented and noisy surface reconstructions that fail to provide a reliable geometric anchor. Similarly, omitting



(a) Ours w/o Interaction-aware States (b) Ours w/o TSDF fusion (c) Ours w/o spatial mask filtering (d) Ours

Fig. 10: Qualitative ablation of Interaction-aware State Updating. Our full model (d) maintains a clean, coherent 3D representation.

spatial mask filtering (Figure 10c) introduces floating artifacts and erroneous projections during dynamic interactions. By integrating all these components, our full pipeline (Figure 10d) maintains a clean, coherent 3D representation.

5.5 Robotic Simulation with Egocentric Pretraining

We further deploy EgoSim as a visual dynamics model for bimanual robotic manipulation using the AgiBot-World [3] dataset. Since the dataset is captured with a static camera, we erase the robotic arms from the initial frame via image editing to establish a clean background, and construct the static conditioning video by duplicating this frame. For the action condition, we directly project the robotic end-effector kinematics into the 3D keypoint space. We randomly sample 50k interaction clips for training and reserve 150 non-overlapping clips for evaluation.

As shown in Table 4, pretraining on large-scale egocentric human hand data before finetuning on AgiBot data (*w/ hand pretrain*) substantially outperforms training from scratch (*w/o hand pretrain*) across all metrics, boosting PSNR from 15.180 to 18.670 and reducing LPIPS from 0.350 to 0.282. Qualitatively (Figure 9), in the complex shirt-folding task, the from-scratch model struggles to generate correct cloth deformations, whereas the pretrained model accurately simulates the physical dynamics between robotic end-effectors and the environment. These results demonstrate that egocentric human interaction data provides generalizable dynamics priors, enabling EgoSim to serve as an effective visual dynamics model for robotic manipulation.

6 Conclusion

In this work, we introduce EgoSim, a closed-loop egocentric world simulator that bridges the gap between passive video generation and active embodied participation. By anchoring visual generation to an underlying 3D point cloud and explicitly maintaining an Updatable 3D Memory, EgoSim achieves faithful spatial conditioning and long-horizon state persistence, eliminating the structural drift common in prior world models. To overcome the critical bottleneck of acquiring perfectly aligned static-dynamic paired data, we develop a scalable data construction framework that extracts aligned triplets from web-scale videos, complemented by our EgoCap tool for low-cost, real-world capture. Extensive experiments demonstrate that EgoSim excels in complex, multi-object interactions and successfully transfers to bimanual robotic manipulations.

Limitations: While our pipeline successfully leverages in-the-wild data, monocular depth and camera pose estimations can occasionally fail in heavily occluded or highly dynamic environments, leading to imperfect point cloud initializations. Future work will explore integrating robust multi-view priors and physics-based contact constraints to further enhance simulation fidelity.

References

1. Akkerman, R., Feng, H., Black, M.J., Tzionas, D., Abrevaya, V.F.: Interdyn: Controllable interactive dynamics with video diffusion models. In: CVPR. pp. 12467–12479 (2025)
2. Bruce, J., Dennis, M.D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al.: Genie: Generative interactive environments. In: ICML (2024)
3. Bu, Q., Cai, J., Chen, L., Cui, X., Ding, Y., Feng, S., Gao, S., He, X., Hu, X., Huang, X., et al.: Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv preprint arXiv:2503.06669 (2025)
4. Cai, X., Qiu, R.Z., Chen, G., Wei, L., Liu, I., Huang, T., Cheng, X., Wang, X.: In-n-on: Scaling egocentric manipulation with in-the-wild and on-task data (2025), `https://arxiv.org/abs/2511.15704`
5. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts (2025), `https://arxiv.org/abs/2511.16719`
6. DeepMind, G.: Genie: Generative interactive environments. Web page (2026), `https://deepmind.google/models/genie/`
7. Duan, Z., Xia, J., Zhang, Z., Zhang, W., Zhou, G., Gou, C., He, Y., Chen, F., Zhang, X., Liu, L.: Liveworld: Simulating out-of-sight dynamics in generative video world models (2026), `https://arxiv.org/abs/2603.07145`
8. Feng, R., Zhang, H., Yang, Z., Xiao, J., Shu, Z., Liu, Z., Zheng, A., Huang, Y., Liu, Y., Zhang, H.: The matrix: Infinite-horizon world generation with real-time moving control. arXiv preprint arXiv:2412.03568 (2024)
9. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S.K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E.Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell,

T., Nishiyasu, T., Price, W., Puentes, P.R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G.M., Fuegen, C., Ghanem, B., Ithapu, V.K., Jawahar, C.V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H.S., Rehg, J.M., Sato, Y., Shi, J., Shou, M.Z., Torralba, A., Torresani, L., Yan, M., Malik, J.: Ego4d: Around the world in 3,000 hours of egocentric video (2022), `https://arxiv.org/abs/2110.07058`

10. Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Kumar, A., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In: CVPR. pp. 19383–19400 (2024)
11. Guo, Z., Hou, Y., Wang, P., Gao, Z., Xu, M., Li, W.: Ft-hid: A large-scale rgb-d dataset for first- and third-person human interaction analysis. Neural Computing and Applications **35**(2), 2007–2024 (2023)
12. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 **2**(3), 440 (2018)
13. He, X., Peng, C., Liu, Z., Wang, B., Zhang, Y., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., et al.: Matrix-game 2.0: An open-source real-time and streaming interactive world model. arXiv preprint arXiv:2508.13009 (2025)
14. Hoque, R., Huang, P., Yoon, D.J., Sivapurapu, M., Zhang, J.: Egodex: Learning dexterous manipulation from large-scale egocentric video. arXiv preprint arXiv:2505.11709 (2025)
15. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., Ren, J., Xie, K., Biswas, J., Leal-Taixe, L., Fidler, S.: Vipe: Video pose engine for 3d geometric perception. In: arXiv (2025)
16. Jia, M., Zhao, L., Li, G., Zheng, Y.: Orchestrating the symphony of prompt distribution learning for human-object interaction detection (2024), `https://arxiv.org/abs/2412.08506`
17. Jiang, N., Zhang, Z., Li, H., Ma, X., Wang, Z., Chen, Y., Liu, T., Zhu, Y., Huang, S.: Scaling up dynamic human-scene interaction modeling. In: CVPR. pp. 1737–1747 (2024)
18. Kim, B., Kim, T., Lee, J., Joo, H.: Dexterous world models (2025), `https://arxiv.org/abs/2512.17907`
19. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
20. Li, G., Zhao, B., Yang, J., Sevilla-Lara, L.: Mask2iv: Interaction-centric video generation via mask trajectories. In: AAAI (2026)
21. Li, G., Ren, K., Xu, L., Zheng, Z., Jiang, C., Gao, X., Dai, B., Pu, J., Yu, M., Pang, J.: Artdeco: Towards efficient and high-fidelity on-the-fly 3d reconstruction with structured scene representation. arXiv preprint arXiv:2510.08551 (2025)
22. Li, Z., Lyu, H., Shi, J., Zeng, Y., Fan, M., Zhang, H., Liang, C.: Spritehand: Real-time versatile hand-object interaction with autoregressive video generation. arXiv preprint arXiv:2512.01960 (2025)
23. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
24. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)

25. Mao, X., Li, Z., Li, C., Xu, X., Ying, K., He, T., Pang, J., Qiao, Y., Zhang, K.: Yume-1.5: A text-controlled interactive world generation model. arXiv preprint arXiv:2512.22096 (2025)
26. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing hands in 3D with transformers. In: CVPR (2024)
27. Qian, Z., Chi, X., Li, Y., Wang, S., Qin, Z., Ju, X., Han, S., Zhang, S.: Wrist-world: Generating wrist-views via 4d world models for robotic manipulation (2025), `https://arxiv.org/abs/2510.07313`
28. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM TOG **36**(6), 245:1–245:17 (Nov 2017)
29. Sudhakar, S., Liu, R., Van Hoorick, B., Vondrick, C., Zemel, R.: Controlling the world by sleight of hand. In: European Conference on Computer Vision (ECCV) (2024)
30. Sudhakar, S., Liu, R., Van Hoorick, B., Vondrick, C., Zemel, R.: Controlling the world by sleight of hand. In: ECCV. pp. 414–430. Springer (2024)
31. Team, R., Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., Chen, Y., Liu, J., Cheng, Y., Yao, Y., Zhu, J., Meng, Y., Zheng, K., Bai, Q., Chen, J., Shen, Z., Yu, Y., Zhu, X., Shen, Y., Ouyang, H.: Advancing open-source world models (2026), `https://arxiv.org/abs/2601.20540`
32. Team, W., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
33. Tu, Y., Luo, H., Chen, X., Bai, X., Wang, F., Zhao, H.: Playerone: Egocentric world simulator (2025), `https://arxiv.org/abs/2506.09995`
34. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: Geocalib: Learning single-image calibration with geometric optimization (2024), `https://arxiv.org/abs/2409.06704`
35. Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J.: Moge-2: Accurate monocular geometry with metric scale and sharp details (2025), `https://arxiv.org/abs/2507.02546`
36. Wang, R., Liu, Q., Deng, Y., Liu, G., Liu, Z., Jia, K.: Eva: Aligning video world models with executable robot actions via inverse dynamics rewards (2026), `https://arxiv.org/abs/2603.17808`
37. Wang, X., Zhao, K., Liu, F., Wang, J., Zhao, G., Bao, X., Zhu, Z., Zhang, Y., Wang, X.: Egovid-5m: A large-scale video-action dataset for egocentric video generation. arXiv preprint arXiv:2411.08380 (2024)
38. Wang, Y., Ouyang, W., Wei, T., Dong, Y., Shen, Z., Pan, X.: Hand2world: Autoregressive egocentric interaction generation via free-space hand gestures (2026), `https://arxiv.org/abs/2602.09600`
39. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin,

W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)

40. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
41. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
42. Zhao, J., Wei, F., Liu, Z., Zhang, H., Xu, C., Lu, Y.: Spatia: Video generation with updatable spatial memory (2025), `https://arxiv.org/abs/2512.15716`

Supplementary

This supplementary material provides comprehensive technical details, extended experimental evaluations, and additional results to support the main paper.

1. **More Visual Comparison** (Section 1): We provide additional qualitative comparisons.
2. **More Details of Interaction-aware State Updating** (Section 2): This section details the multi-stage pipeline for trajectory estimation, hierarchical interaction tracking, and multi-frame point cloud fusion.
3. **Data Pipeline Details** (Section 3): This section offers an in-depth description covering data annotation and cleaning pipelines, the development of the unified skeleton representation, and a comparative analysis with existing egocentric benchmarks.
4. **EgoCap Details** (Section 4): This section elaborates on further technical specifics regarding the EgoCap Pipeline.
5. **More Experiment Details** (Section 5): This section provides training configurations, hardware specifications, and baseline implementation protocols as concluding content.
6. **Additional Visualization Results** (Section 6): This section presents additional visualization results. We also provide more visualizations in the HTML page in our supplementary.

1 More Qualitative Comparison

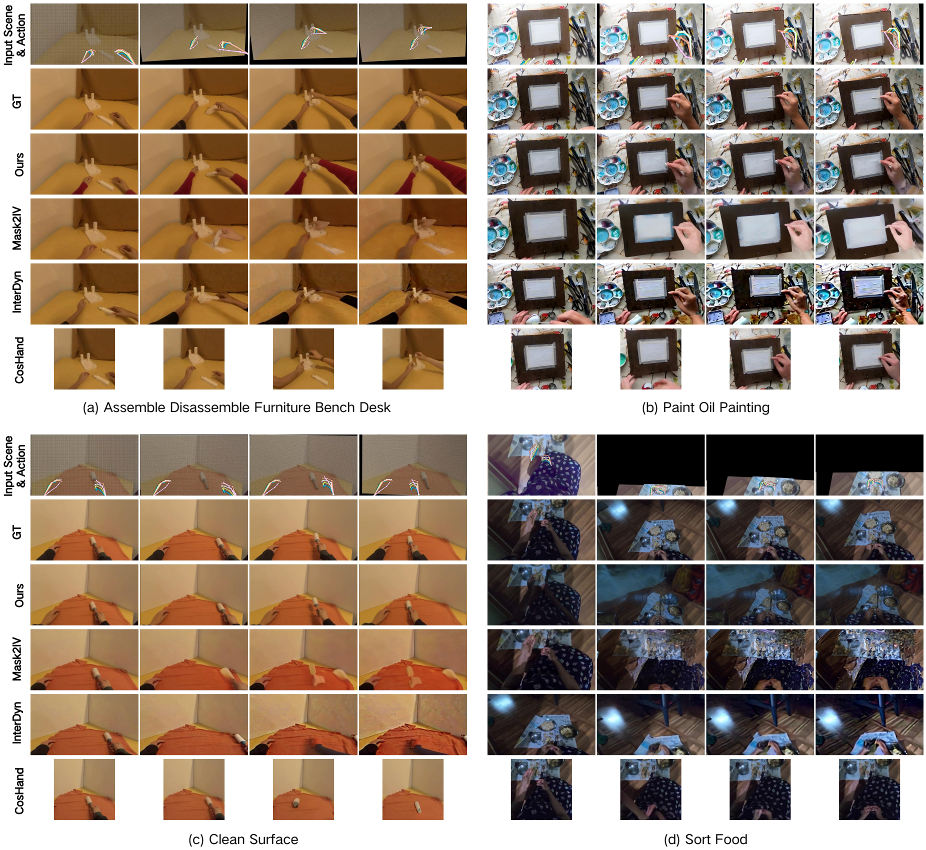
We provide additional qualitative comparisons across four diverse scenarios in Fig. 11. Compared to Mask2IV, InterDyn, and CosHand, our method consistently generates more realistic hand-object interactions and better preserves scene consistency throughout the video.

2 Interaction-aware State Updating.

The Interaction-aware State Updating pipeline processes raw observation sequences and outputs synchronized instance-level interaction metadata within a reconstructed 3D scene state.

2.1 State Reconstruction

The pipeline begins with a pre-processing stage modified from VIPE [15]. In this stage, the camera intrinsics are firstly estimated using the GeoCalib [34]. To establish an initial spatial understanding, we extract instance masks using SAM3 [5] guided by text phrases of objects interacted with embodiments predicted via Grounding-DINO [24]. The detection thresholds for bounding boxes and text are set to 0.35 and 0.5, respectively. The prompt of Grounding-DINO



(a) Assemble Disassemble Furniture Bench Desk

(b) Paint Oil Painting

(c) Clean Surface

(d) Sort Food

Fig. 11: More comparisons of Egosim with additional baselines.

can refer to Figure 19, the masks of interaction objects will be tracked during the state construction along each observation sequence.

At the same time, this pipeline predicts and aligns per-frame depths and camera poses following VIPE [15] We employ a dual-pass DROID-SLAM system. During the first pass, the system tracks sparse features and selects keyframes when the tracking threshold exceeds 4.0. These keyframes receive initial metric depth from DepthAnything3. In the second pass, a temporal interpolation mechanism assigns 6-DoF camera poses and intrinsic attributes to all non-keyframe observations. Finally, a multiview depth processor aligns the predicted depth maps across different perspectives to resolve scale ambiguity.

2.2 Interaction-aware Object State Update

Based on the aligned geometry and continuous interaction instance tracking, we introduce a five-stage hierarchical filtering process to accurately detect and update interaction states.

First, an initial semantic tagging is performed. Instances containing human-centric phrases, such as *hands*, *arms*, and *person* [16], are immediately marked as interaction references. Second, for other general objects, we evaluate their spatial proximity to the human reference. During the frame-by-frame tracking process, any object whose bounding box exhibits an IoU greater than 0.15 with the hand bounding box is considered a potential interactive candidate.

To eliminate false positives caused by 2D projection overlap from background objects, we introduce a strict depth-based refinement. An object is confirmed as interactive only if the median depth difference between the object mask and the human reference mask is less than 0.15m. Objects exceeding this threshold are filtered out. Furthermore, to guarantee temporal completeness, our system performs a retrospective check to update the interaction state of the last observed frame for every object. These dynamic states, including interaction status and depth distance, are serialized into a point-based updatable state to support continuous simulation.

2.3 Incremental State Fusion

The final phase constructs a global 3D scene state by fusing multi-frame point clouds and removing redundant geometry. To handle overlapping observations and sensor noise, we implement a Truncated Signed Distance Function (TSDF) fusion.

The system utilizes a scalable TSDF Volume for point subsampling with a voxel size of 0.003m, and a maximum depth truncation of 3.0m. During integration, multi-frame observations are weighted and averaged within the voxels, which automatically handles overlap and smooths depth noise.

We employ an incremental fusion strategy with explicit overlap detection. For the fusion of the accumulated states and constructed states from a new observation sequence, the points are projected into the camera coordinate system of the previous state. The points are integrated only if their depth discrepancy is greater than 0.05m or if their color difference is significant. Non-interactive objects preserve their geometry from their last observed frame to maintain scene completeness. Finally, we apply a statistical outlier removal filter with 20 neighbors and a standard deviation ratio of 2.0 to extract a clean, global point cloud artifact.

2.4 Dynamic-View Simulation with Persistent 3D States

Recent world models always fail to model the out-of-view object dynamics [7]. EgoSim could continuously generate embodiment interactions with out-of-view state dynamics. As shown in Figure 12, Egosim updates the state of green Crayon Shin-chan in an in-the-wild real-life scene, though the camera view turns around and leaves the toy, it could still successfully generate it when the view goes back, showing potential in generating controllable and spatio-temporally consistent mobile content.

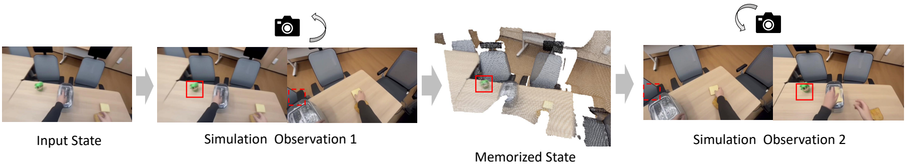


Fig. 12: Egosim with updatable 3D states, cached seen environments, and enables spatial-consistency generation under large-scale dynamic view changes.

3 Data Pipeline Details

3.1 Introduction

World simulators [14, 38] serve as the cognitive foundation for intelligent systems, aiming to internalize the underlying structure and causal dynamics of the physical environment to enable effective reasoning and planning. A high-fidelity world model must not only maintain a consistent representation of static scenes but also accurately predict the dynamic consequences induced by specific embodiment motion and actions, such as mobile manipulation and dexterous hand manipulations. However, transitioning from general video generation [32] to faithful world simulation remains a formidable challenge.

Developing a world model capable of such precision necessitates a fundamental shift in data requirements. It demands a new category of training data that explicitly decouples the static environment from dynamic interactions, providing ground-truth for camera trajectories, depth, and embodiment skeletons simultaneously.

Existing egocentric datasets [3, 14, 18, 37] primarily face a trade-off between structural precision and data diversity. On one hand, synthetic datasets [18] provide perfectly aligned pairs of embodiment interactions and static scene renderings under identical trajectories, yet they suffer from a sim-to-real gap and limited diversity in interaction dynamics. On the other hand, large-scale real-world datasets collected from the web or wearable devices offer rich diversity but are often noisy, as they contain distracting body parts, inconsistent camera parameters, low-quality observations, extreme camera and embodiment movements, and especially lack precise spatial and action annotations.

To bridge this gap, we propose a scalable and general-purpose data processing pipeline along with a novel Large-scale Heterogeneous Egocentric Manipulation Dataset. Our dataset is distinguished by the following key characteristics.

(a) Embodiment-agnostic and Source-independent. Unlike prior works limited to specific robots or human setups, our pipeline treats varied data from diverse web videos and open-source datasets to in-the-wild captures as a unified stream. By decoupling the interaction from specific physical embodiments, our dataset achieves true heterogeneity, enabling models to learn universal manipulation priors that transcend specific hardware or subjects.

Table 5: Comparison of the scalable egocentric dataset of EgoSim with recent open-source egocentric manipulation datasets and datasets collected by compared world simulators.

Dataset	Year	Domain	Type	Clear scene	Camera	Depth	Action	Embodiment	# Clips
Ego4D [9]	2022	Mobile	Real	✘	✘	✘	✘	Human	3.8M
Egovid [37]	2024	Mobile	Real	✘	Limited	✘	✘	Human	5M
Agibot-world [3]	2025	Tabletop	Real	✘	✔	✘	Joint	Robot	1M
Egodex [14]	2025	Tabletop	Real	✘	✔	✘	Skeleton	Human	338K
DWM [18]	2025	Mobile	Sim & Real	✔	✔	✘	mesh	Human	110K
Hand2World [38]	2026	Tabletop	Real	✔	Limited	✘	mesh	Human	$< 8K$
Ours	2026	Mobile	Real	✔	✔	✔	**Unified Skeleton**	**Mixed**	**500K (Scalable)**

(b) Rich Dexterous Interactions and Mobility. We significantly expand the scope of egocentric data beyond simple tabletop tasks with restricted camera motions. Our dataset encompasses dexterous manipulation with complex hand-object interactions, as well as large-scale camera locomotion. This variety ensures the model can perceive and predict outcomes in dynamic, non-static environments.

(c) High-quality Multi-modal Annotations. To provide the precise supervision required for world simulating, each clip is processed to include depth maps, camera trajectories, and action skeletons. Crucially, we provide clean static scene reconstructions by computationally removing and inpainting the ego-body and foreground occlusions, paired with high-quality automated text captions.

Our dataset serves as a robust foundation for training world models capable of generalizing to complex, real-world manipulation and locomotion tasks.

3.2 Data Annotation Pipeline

We provide a comprehensive description of the dataset collection pipeline of EgoSim. This pipeline consists of four primary stages, including clip standardization, static scene reconstruction, action representation, and prompt generation. The codes of the data annotation pipeline will be released soon.

Clip standardization. We construct our training corpus by integrating diverse human and robotic manipulation datasets. We construct training clips into a uniform format of 61 frames per clip at 16 fps. For Human Datasets, we utilize EgoDex [14] and EgoVid [37]. Specifically, for EgoDex, we directly segment the raw footage into clips. For EgoVid, we adopt their original clips and convert them to our format. For robot datasets, we incorporate a subset of the Agibot-World-Beta dataset from task indices 327 to 475, focusing on 30 daily tabletop manipulation tasks. We segment 100K high-quality clips from 25,507 provided center-camera videos. The detailed task distribution is summarized in Table 6.

Static scene reconstruction. To decouple dynamic interactions from the environment, we employ a mask-inpainting strategy to obtain clean and static scene backgrounds.

(a) Egocentric body-part Masking. We first employ SAM3 [5] to detect and mask ego-body parts, like hands and arms, using the phrases $\{$*hand, arm, person*$\}$, generating precise red masks. For robotic datasets where SAM3 struggles with mechanical textures, we directly inpaint and discard their arms.

(b) Embodiment Inpainting. We utilize the Qwen-Image-Edit [40] model to remove the masked regions. For human data, we use the prompt, *Discard and remove the hand and arms according to the red masks of this image, keep the other part the same.* For robotic data, we specifically address interaction dynamics with the prompt, *Remove and discard the robot arms and grippers in this image, and fill the discarded area with suitable background.*

(c) Spatial Geometry Estimation. To reconstruct the 3D structure, we leverage DepthAnything3 [23] to predict camera trajectories, camera intrinsics, first-frame point clouds, and per-frame depth maps. By rendering the static scene with the first-frame point clouds and predicted camera trajectories, we could obtain the videos of point-based spatial conditions for EgoSim.

(d) Unified Action Representation. A key challenge in enhancing robots with generative world simulators is the embodiment gap between human hands and robot grippers. We propose a unified skeleton representation to achieve embodiment-agnostic action modeling. For human actions, we utilize a unified 21-keypoint MANO [28] skeleton to represent their hand actions. For EgoDex, we transform their ARKit-style 26-keypoint skeleton annotations into 21-keypoint MANO skeletons by deleting the redundant hand keypoints. For datasets lacking hand annotations, we utilize the Hamer [26] model to extract hand poses, as shown in Figure 15. For robot actions, we synthesize skeleton-based annotations using end-effector (EE) poses, joint states, and URDF of corresponding robot embodiments. To maintain consistency with human hand structures, we map the EE status to a simplified skeleton comprising the index finger and thumb, while simultaneously registering the opening state of the grippers, as shown in Figure 18.

(d) Prompt Generation. To provide EgoSim with high-level semantic guidance, we generate descriptive prompts for each clip. For human data, we employ the Qwen3VL-8B [39] model to generate detailed scene and action comprehensions, following the descriptive style established in EgoVid [37]. For robot data, given the structured nature of robotic tasks, we directly utilize the provided task descriptions as the ground-truth text prompts.

3.3 Data Cleaning Pipeline

Real-world egocentric data often contains distracting body parts, inconsistent camera parameters, and extreme jitter. We implement a robust cleaning pipeline to enhance spatial-temporal consistency.

Outlier Detection and Correction The raw camera trajectories predicted by Depthanything3 [23] sometimes exhibit sudden jumps. For each frame i, we evaluate its deviation from adjacent frames $i \pm 1$ based on translation (t) and rotation (q). A frame is flagged as an outlier if:

$$||t_i - t_{i\pm1}|| > \delta_{trans} \quad \text{and} \quad \theta(q_i, q_{i\pm1}) > \delta_{rot} \tag{4}$$

where $\delta_{trans} = 0.1m$ and $\delta_{rot} = 50°$. Outliers are repaired using the linear mean for translation and the normalized average of the neighborhood $[i-2, i+2]$ for rotation.

Pose Kalman Smoothing To suppress high-frequency noise, we employ a Pose Kalman Filter that includes both translation and rotation smoothing.

Translation Smoothing. We define a 6D state vector $S_t = [x, y, z, v_x, v_y, v_z]^T$. We assume a constant velocity model for prediction and use the observed coordinates for updates. We set the process noise $Q = 1 \times 10^{-4}$ and measurement noise $R = 1 \times 10^{-3}$, ensuring the filter remains responsive to actual movement while suppressing jitter.

Rotation Smoothing. We maintain a 4D quaternion state q. The prediction step assumes $q_{pred} = q_{prev}$. After the update step using the observed quaternion, we perform re-normalization:

$$q_{final} = \frac{q_{updated}}{||q_{updated}||} \tag{5}$$

This ensures the rotation remains a valid unit quaternion throughout the sequence.

Action Filtering and Refinement While modern human hand estimation models [26] have achieved promising performance, they frequently exhibit instability when processing unconstrained in-the-wild egocentric videos. These models are particularly susceptible to motion blur, heavy occlusion, and low-resolution observations. To address these challenges and ensure high-fidelity action labels for EgoSim, we implement a robust filtering pipeline.

Frame-wise Spatial Filtering and De-duplication We process each frame independently to resolve redundant detections. For each frame i in the sequence, we derive 3D joints from MANO parameters including *global_ orient*, *hand_pose*, *betas*, and *cam_ t*. These 3D joints are projected onto the 2D image plane to obtain 21-keypoint coordinates for computing the hand bounding box B_i. We utilize an Intersection-over-Union (IoU) threshold to identify duplicates and employ a Breadth-First Search (BFS) to group overlapping detections, retaining only the instance with the highest confidence score for each hand side.

Temporal Outlier Detection To ensure temporal smoothness, we identify sudden spikes in hand motion that signify tracking failures. We evaluate the dynamics between the previous $(i-1)$, current (i), and next $(i+1)$ frames by calculating the translational velocity v_t in camera space through the following expression

$$v_t = \frac{||cam\_t_{curr} - cam\_t_{prev}||}{\Delta t} \tag{6}$$

A frame is classified as a temporal outlier if it exhibits a bidirectional jump where both the incoming velocity and outgoing velocity exceed predefined thresholds. Such frames are discarded to prevent the model from learning physically impossible motion discontinuities.

3.4 Dataset Statistics

To further evaluate the superiority of our proposed pipeline, we conduct a comprehensive comparison with recently released and widely-used egocentric manipulation datasets, as summarized in Table 5.

Limitations of Existing Benchmarks. Current egocentric datasets often face a trade-off between structural precision and data scalability. While foundational datasets like Ego4D [9] provide massive scale 3.8M clips, they lack essential 3D grounding, such as depth information and precise action labels. Conversely, recent efforts such as Agibot-world [3] and EgoDex [14] introduce structured action representations like joint angles or skeletons. However, these datasets are typically restricted to static tabletop environments, failing to generalize to complex mobile scenarios.

Scalability and Diversity. In contrast, our dataset is the first to achieve both large-scale diversity and high-fidelity structural alignment in mobile egocentric settings. By leveraging an automated processing pipeline on diverse web data, we provide 500K scalable clips that encompass broad mobile domains, involving significant locomotion and varied camera ego-motion with fully aligned camera trajectories and depth maps for every clip, ensuring a consistent physical grounding.

Bridging Human and Robot Actions. A key innovation of our dataset is the introduction of a Unified Skeleton representation, which addresses the long-standing challenge of data heterogeneity across different embodiments. We map diverse hand and robot arm configurations from multiple sources into a standardized five-finger skeleton format, as shown in Figure 15. To bridge the gap between human hands and robotic end-effectors, inspired by [4], we represent robotic grippers using the indices of the human thumb and index finger. This unified formulation allows our model to learn universal manipulation priors from a Mixed embodiment pool with potential in scaling up to more different embodiment types, as shown in Figure 18.

4 Detailed Implementation of EgoCap Pipeline

We detail the Egocap internal reconstruct-then-relocalize mechanisms, which robustly recover precise 6-DoF camera trajectories from uncalibrated videos.

Intrinsics-Free Scene Reconstruction. During the initial scanning phase, the system extracts frames at a fixed rate (20 FPS) and invokes COLMAP for automatic intrinsic self-calibration, jointly estimating focal length and lens distortion without prior device profiles. Building upon the ARTDECO [21] framework, the pipeline incrementally processes these frames to perform joint SLAM and 3D Gaussian Splatting (3DGS) optimization. This yields a scale-consistent global 3DGS map, which is subsequently converted into a standard 3D point cloud via opacity-based filtering to remove low-confidence primitives, providing a clean static scene representation.

Robust Camera Relocalization. To recover the interaction trajectory under severe occlusions and motion blur, the system samples the interaction video densely (30 FPS) and employs a dual-path relocalization architecture. The primary path utilizes the ARTDECO dense matching localizer to establish frame-to-map correspondences against the pre-built 3D map, independently recalibrating intrinsics to account for unknown focal lengths. If tracking success falls below 70% due to extreme motion, the system gracefully degrades to a fallback path. This path performs COLMAP-based offline registration and subsequently aligns the local trajectory to the unified scene coordinate system via a Sim(3) similarity transformation with outlier rejection.

Trajectory Smoothing and Aligned Rendering. Since unconstrained handheld capture inevitably introduces localization noise and frame discontinuities, we apply a multi-stage temporal refinement to the raw pose estimates. The system detects anomalous jump frames and applies cubic spline interpolation for translations alongside spherical linear interpolation (SLERP) for rotations, ensuring smooth spatial transitions while preserving unit quaternion constraints. A global Savitzky-Golay filter then suppresses high-frequency jitter. Based on this continuous 6-DoF trajectory, the system re-renders the initial 3DGS map to produce a geometry-consistent static background video, thereby forming the strictly aligned paired training data required by the EgoSim framework.

5 More Experiment Details

5.1 Training & Inference Details.

Foundation Model Pre-training. Our base video diffusion model is initialized from Wan2.1-Fun-14B-InP, generating output simulations at a resolution of 832×480 with 61 frames at 16 FPS. To accommodate hand action conditioning, we expand the Diffusion Transformer (DiT) input channels to 52. We apply full fine-tuning to the DiT for 4,000 steps, while all other components, including T5 text encoder, VAE, and CLIP image encoder, are remain frozen. We employ the AdamW optimizer with a learning rate of 1×10^{-5}. Training is conducted on a computing node equipped with 8 NVIDIA H200 GPUs, using a per-GPU batch size of 4.

Cross-Embodiment Fine-Tuning. For robotic manipulation evaluations on the Agibot dataset, we randomly select 50K clips from the processed 100K for training and 150 non-overlapping clips for testing. All clips are processed to 61 frames at 16 FPS. We explore two initialization settings. (1) *Without hand data pre-training*, which trains directly from the expanded Wan2.1-Fun-14B-InP model, and (2) *With hand data pre-training*, training from our EgoSim model pre-trained for 4,000 steps. Both variants are fine-tuned for an additional 200 steps utilizing identical hyperparameters as the main training phase.

Real-World Fine-Tuning. To validate our real-world data collection pipeline, EgoCap, we capture 50 clips in a supermarket environment, focusing on shelf interactions. Following the standard format of 61 frames and 16 FPS, we allocate 30 clips for training and 20 for testing. The model is initialized from our main 4,000-step checkpoint and fine-tuned for 50 steps. For this dataset, we adjust the per-GPU batch size to 1 while keeping other optimizer settings consistent.

Inference Settings. During the inference phase, all models across different datasets and evaluation settings employ the same generation hyperparameters. We use the Flow Matching sampling process with 50 inference steps. To preserve the generative prior and balance fidelity with action-conditioning adherence, the classifier-free guidance (CFG) scale is consistently set to 1.0.

Inference of Continuous Simulation. To enable the state updating along the dynamics of out-of-view contents and spatio-temporal consistency in continuous observation simulation, we propose and implement a novel updatable 3D interaction-aware state.

In the continuous generation setting, we formulate the updated 3D state, constructed from previously generated observation sequences, as the spatial 3D condition for simulating the subsequent sequence. We utilize the input actions H_{k+1} and camera trajectories C_{k+1} to render the accumulated stateS_k, thereby obtaining the rendered Scene Observation O_{k+1}. This rendering serves as a geometric anchor, ensuring that all previously modified object states are preserved and correctly projected into the new simulation cycle.

5.2 Baseline Details

InterDyn. InterDyn [1] extends Stable Video Diffusion (SVD) with a ControlNet branch. Given a single conditioning image $\mathbf{I}0$ and a temporally aligned sequence of hand-object segmentation masks $\{\mathbf{M}_t\}$ as control signals, the model generates a fixed-length video clip pf 14 frames in one forward pass via score-distillation sampling. To produce a 61-frame output for fair comparison with our method, we adopt an autoregressive inference strategy. The full mask sequence is pre-divided into consecutive non-overlapping segments of equal length. At each step, the model is conditioned on the last frame of the previously generated segment as the new starting image, together with the corresponding mask subsequence, to generate the next segment. The resulting segments are concatenated by discarding the first frame of each subsequent segment, which is identical to the last frame of the preceding segment, yielding a seamless 61-frame video.

Mask2IV. Mask2IV [20] is a latent diffusion model that jointly conditions on an initial frame and a mask video sequence. At each inference call, the initial frame is replicated T times to construct the input video tensor $\mathbf{V} \in \mathbb{R}^{1\times C\times T\times H\times W}$; the corresponding T-frame mask segment is encoded into the latent space and concatenated channel-wise with the replicated frame latent, forming a compound conditioning signal. A text prompt is additionally injected via cross-attention. The model generates T=16 new frames per call using DDIM sampling. To extend generation to 61 frames, we apply the same autoregressive chaining strategy: beginning from the first frame of the ground-truth video, we iteratively produce 16-frame segments, each time feeding the last predicted frame as the initial frame for the next segment, while the mask for segment i is sliced from the full-length mask video as frames $[i \cdot T,\ (i{+}1) \cdot T)$. Four segments are generated and concatenated with duplicate boundary frames removed.

CosHand. CosHand [30] generates videos in a non-autoregressive manner. Given a single reference frame from the video clip and a sequence of target hand masks specifying the desired hand position at each timestep, the model performs an independent DDIM diffusion sampling for each frame. The hand geometry is injected into the UNet via latent concatenation, while the semantic appearance of the reference frame is provided through CLIP features via cross-attention conditioning. In total, 61 frames are generated independently and arranged chronologically to form the final predicted video. Since each frame is generated in isolation without any temporal dependency on neighboring frames, the model lacks an explicit mechanism to ensure inter-frame consistency, which is one of its primary limitations compared to autoregressive video generation approaches.

6 More Visual Results

In this section, we provide additional qualitative results to further demonstrate the versatile simulation capabilities of EgoSim across diverse scenarios and embodiments. These results showcase our method's ability to handle dexterous manipulation, large viewpoint movements, deformable objects, and realistic object dynamics. We also provide more visualizations in the HTML page in our supplementary.

Dexterous Manipulation and Large Viewpoint Movements. Figures 13, 14, 15, and 16 present our inference results on the EgoDex and EgoVid datasets. Our method exhibits precise control and high visual fidelity in fine-grained, dexterous hand-object interactions, successfully maintaining the geometric consistency of manipulated objects without introducing severe structural distortions or blending artifacts. Furthermore, despite drastic camera ego-motion and large viewpoint shifts typical of in-the-wild recordings, EgoSim maintains stable and persistent background generation.

Cross-Embodiment Manipulation. Figure 18 presents the simulation results on the Agibot [3] dataset, verifying that our framework smoothly generalizes from human hands to robotic end-effectors. Guided by the extracted 3D Action keypoints, EgoSim generates physically plausible interactions between bimanual robotic grippers and target objects. Notably, our method effectively simulates the complex physical dynamics of deformable objects during robotic manipulation, correctly reflecting the state changes of items like garments and clothes under continuous interaction.

Real-World Object Dynamics with EgoCap. Figure 17 illustrates the inference results on real-world scenes collected using our custom EgoCap pipeline. Tested in unstructured environments such as supermarkets, EgoSim accurately simulates realistic object dynamics, like shifting items on a shelf, while ensuring the rendering remains perfectly aligned with the complex real-world backgrounds. This confirms the robustness of both our data collection pipeline and the generative simulation model in practical, open-ended scenarios.

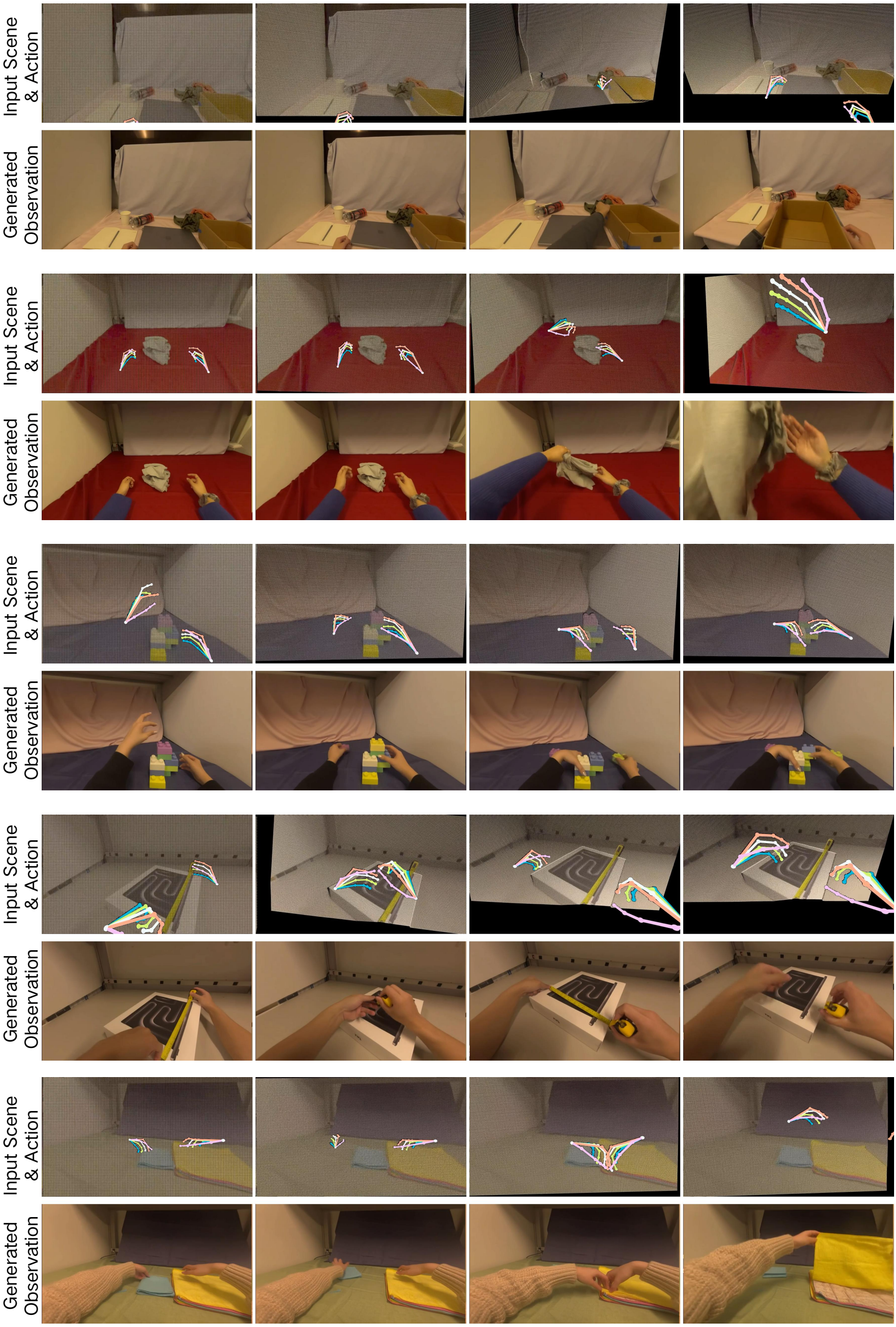


Fig. 13: Visualization of EgoSim simulating test samples of Egodex, highlighting the ability for generating difficult dexterous manipulations over soft, small, or assembled objects.

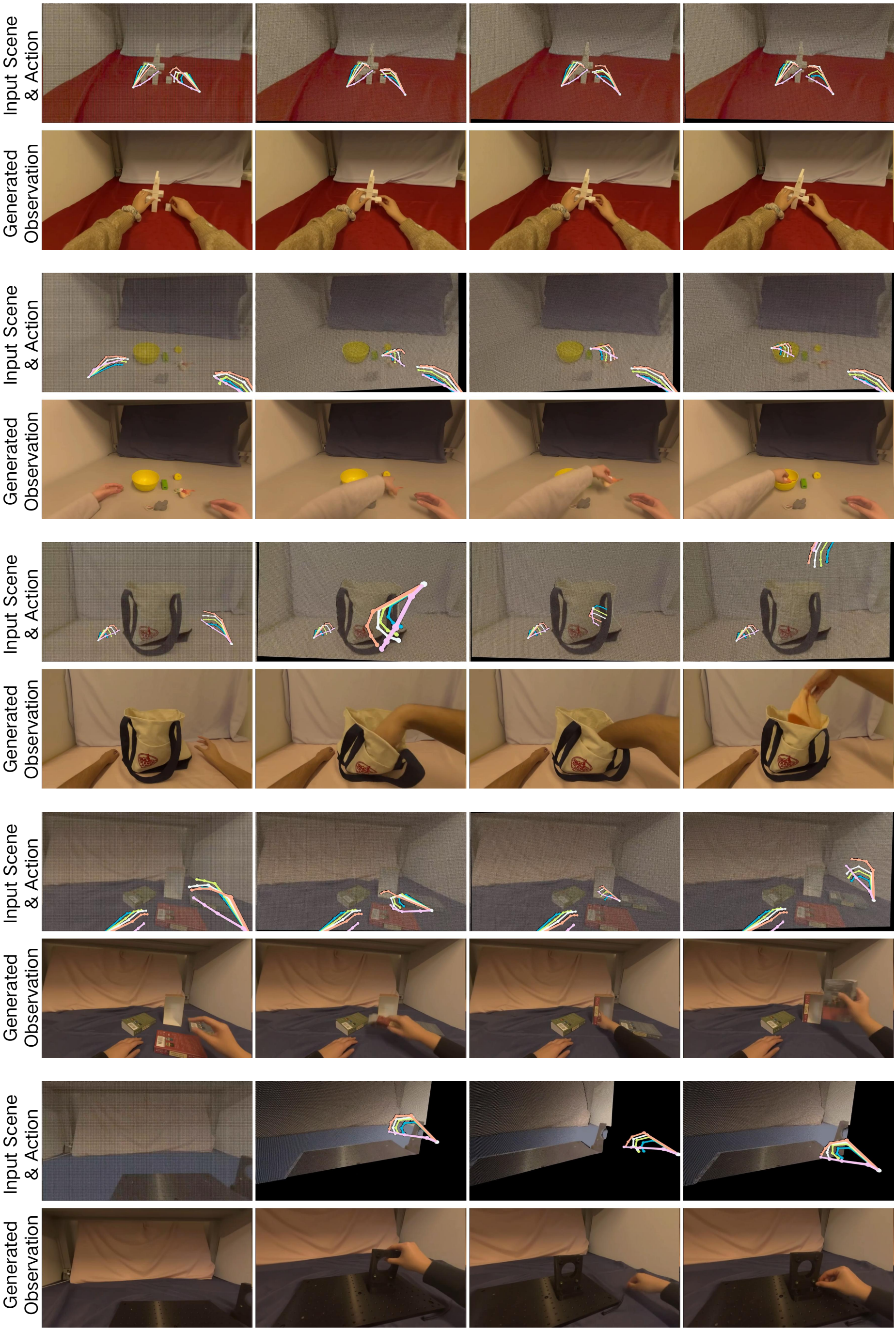


Fig. 14: Visualization of EgoSim simulating test samples of Egodex, highlighting the ability for generating difficult dexterous manipulations over soft, small, or assembled objects.

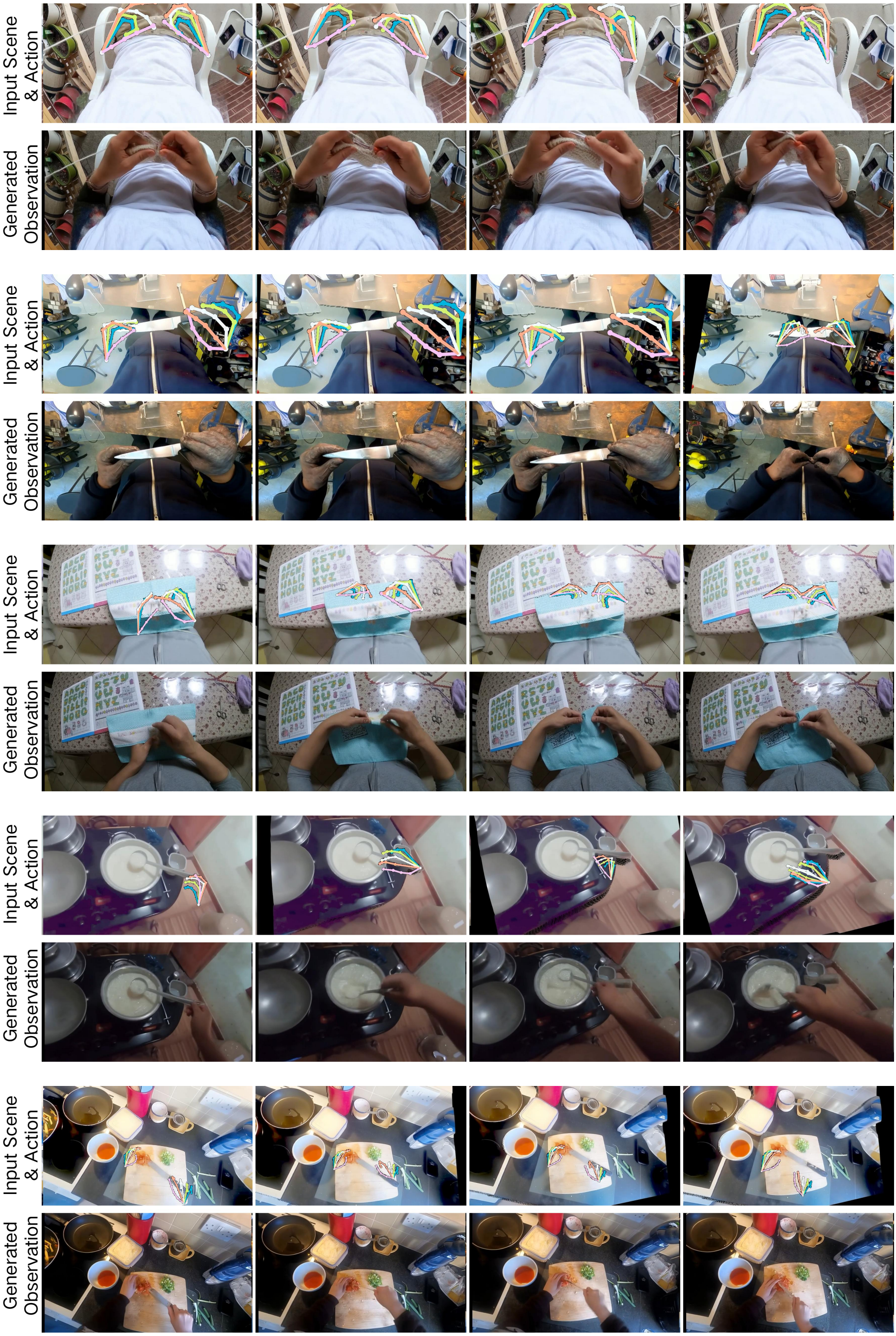


Fig. 15: Visualization of EgoSim simulating test samples of Egovid, highlighting the ability for generating dynamic views and motions.

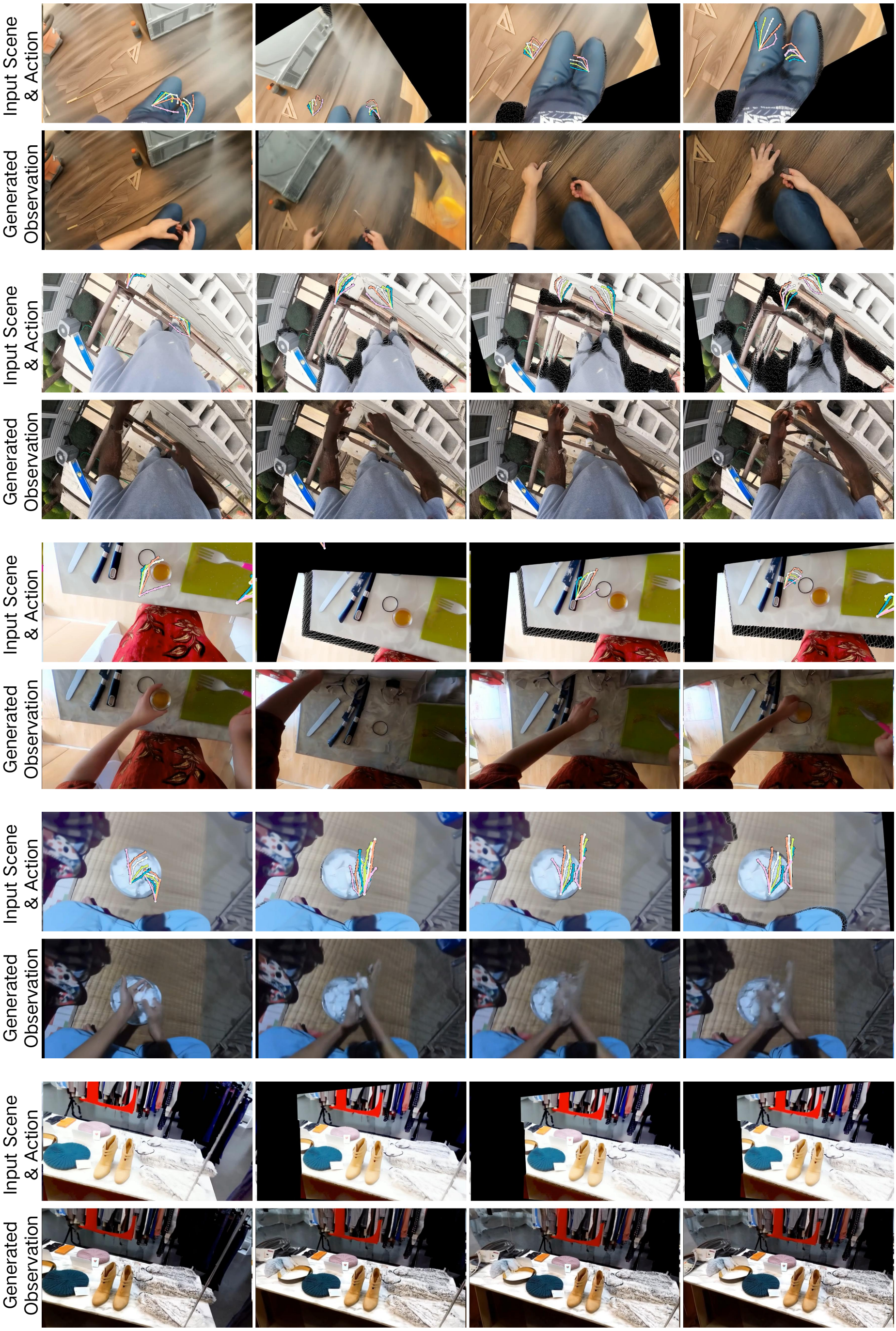


Fig. 16: Visualization of EgoSim simulating test samples of Egovid, highlighting the ability for generating dynamic views and motions.

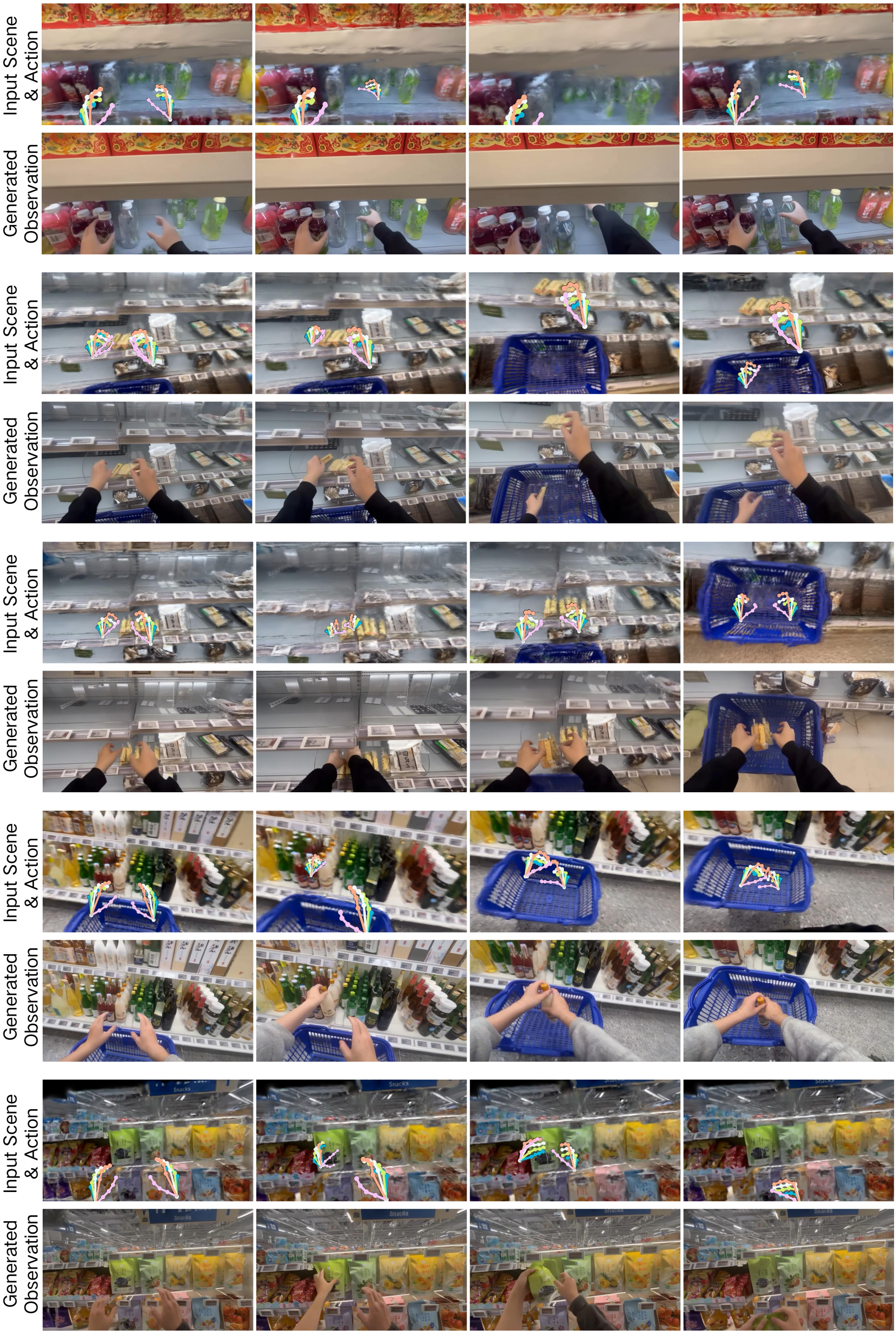


Fig. 17: Visualization of EgoSim simulating human interactions in in-the-wild real-life scenarios.

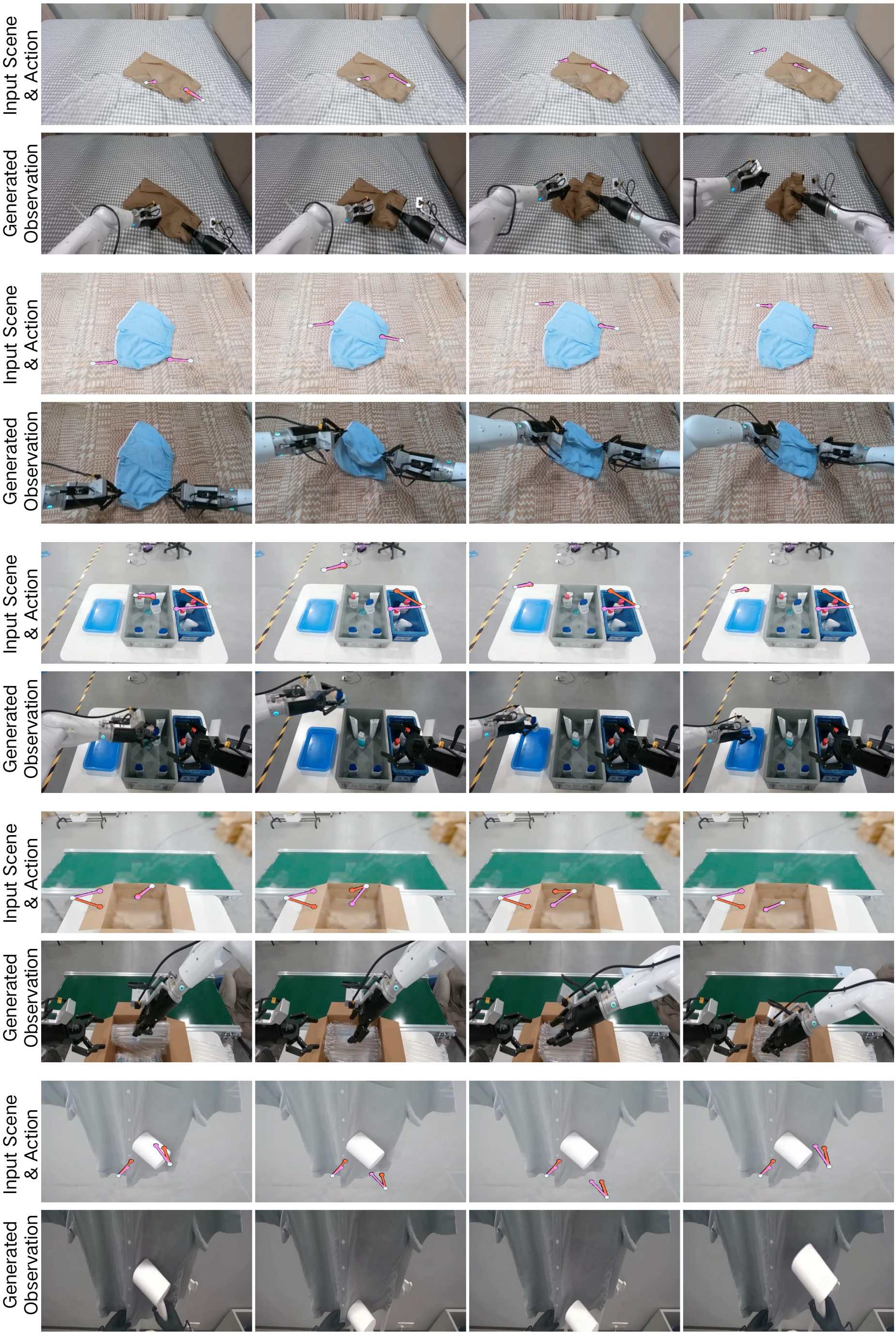
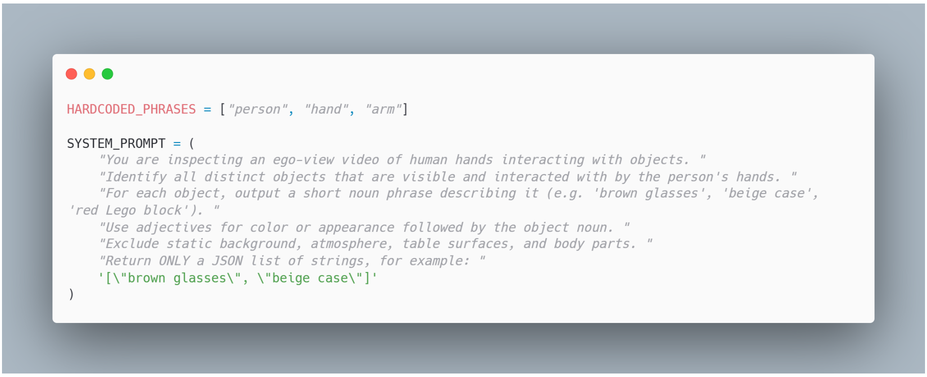


Fig. 18: Visualization of EgoSim simulating robot interactions.



```
HARDCODED_PHRASES = ["person", "hand", "arm"]

SYSTEM_PROMPT = (
    "You are inspecting an ego-view video of human hands interacting with objects. "
    "Identify all distinct objects that are visible and interacted with by the person's hands. "
    "For each object, output a short noun phrase describing it (e.g. 'brown glasses', 'beige case',
'red Lego block'). "
    "Use adjectives for color or appearance followed by the object noun. "
    "Exclude static background, atmosphere, table surfaces, and body parts. "
    "Return ONLY a JSON list of strings, for example: "
    '[\"brown glasses\", \"beige case\"]'
)
```

Fig. 19: Prompt used for detecting interaction objects by Grounding-DINO in Updatable Interaction-aware States.

Table 6: Statistics of Task Distribution for selected tasks from Agibot-world-beta [3].

Task ID	Task Description	# Clips
327	Pickup items in the supermarket	619
352	Open the fridge to get food	3,855
354	Pickup items in the supermarket	724
356	Packing in the supermarket	1,105
358	Toast bread	1,382
359	Sort in the warehouse	1,531
360	Packing in the supermarket	1,499
361	Flatten shorts	15
362	Fold shorts	25,260
365	Sort personal care products	1,370
366	Sort food	2,263
367	Take toast from toaster	616
374	Sort laundry and personal care products	124
375	Brew tea	2,150
377	Packing in e-commerce	4,832
380	Packing in e-commerce	3,967
384	Insert a book into the bookshelf	1,092
385	Pickup in the supermarket produce section	641
388	Pickup in the supermarket	371
389	Pickup in the supermarket	881
390	Checkout and scan barcode in the supermarket	4,016
398	Sort clothes	566
410	Water pouring in restaurant	2,302
414	Hang clothes with hanger	11,243
421	Pick up the item to wipe away the stain	5,631
422	Pack items for industrial logistics	9,948
424	Clear the countertop waste	1,363
440	Iron clothes	6,247
446	Transport table with another robot	293
475	Ironing clothes	4,094
Total		**100,000**