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 Jiaotong University, Shanghai, China
{leo_hao,22-wry, ranyi, ma-lz}@sjtu.edu.cn
[2] Shanghai AI Laboratory, Shanghai, China
Luyitas1217@gmail.com, xuxudong@pjlab.org.cn, pangjiangmiao@gmail.com
[3] The University of Hong Kong, Hong Kong, China
xihuiliu@eee.hku.hk

*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. Project page is at: egosimulator.github.io.

Keywords: Egocentric World Simulator · World Models · Generative Models

1 Introduction

The advances in video diffusion models [14,27,36] have significantly advanced the development of interactive world simulators [4, 5, 9, 22, 26], 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

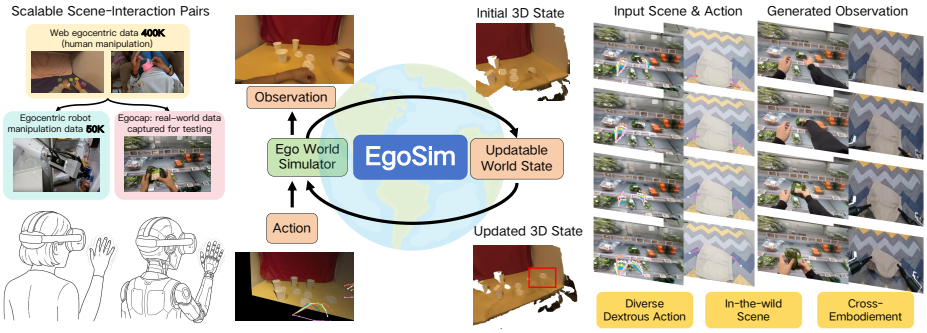

Fig. 1: Given a scene image and a sequence of actions, **EgoSim** generates temporally and spatially consistent egocentric observations and high-quality dexterous interactions. Egosim also persistently updates the 3D scene state for continuous simulation. We propose a data construction pipeline to leverage 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.

navigation, and typically render scenes from a third-person perspective, reducing 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 [13, 28, 32] 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 [28,32] 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 [13] 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 [6, 7, 28] or data collected from synthetic environments [12, 13], 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 [13], we also disentangle static 3D scenes from dynamic changes and formulate EgoSim as an ego-motion-action-aware observation simulation model gen-

erating 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 [37] or simply track moving objects [11], 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 [19] to estimate camera trajectories from the video, and employ HaMeR [23] 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-World data [2], it achieves promising performance on robotic arm manipulation [30]. 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 [14,27,36] has shifted world modeling from latent-space prediction [8] to high-fidelity visual simulation. Genie [4], The Matrix [5, 9], and Lingbot-World [26] 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 [25] generates single-frame hand-object interactions from hand mask inputs. Mask2IV [15] introduces a two-stage pipeline that first predicts interaction mask trajectories and then synthesizes trajectory-conditioned videos. SpriteHand [18] 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 [28] conditions on whole-body motion derived from synchronized ego-exocentric captures. Hand2World [32] 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 [13] 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.

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 [17, 19, 20, 29, 33] can recover dense geometry from single-view images. Several works have explored updating representations by considering object dynamics in video sequences. VIPE [11] fuses per-frame point clouds through decoupling moving objects to maintain a clean static scene. Spatia [37] 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 [24] 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 is 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 $\mathit{\Pi}(S_{k-1}; C_k)$ and synthesizing the dynamic observation residuals $\mathit{\Delta O}(H_k)$ induced by the hand action:

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

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).

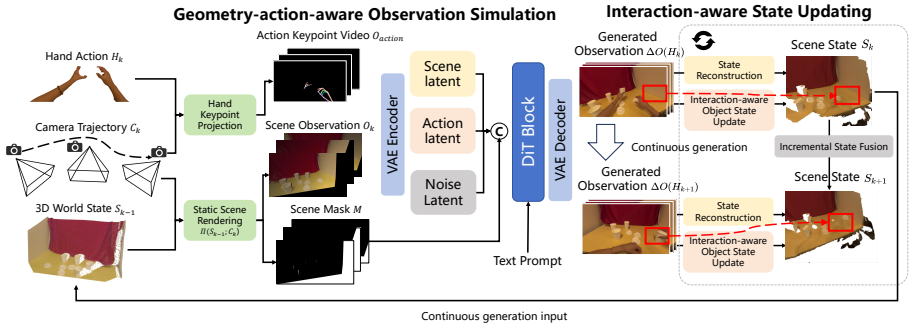

Fig. 2: Overview of EgoSim. Our framework enables continuous egocentric simulation via updatable world modeling. It consists of (1) a Geometry-action-aware Observation Simulation model that synthesizes action-conditioned visual dynamics O_k based on current scene geometry; and (2) an Interaction-aware 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:

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

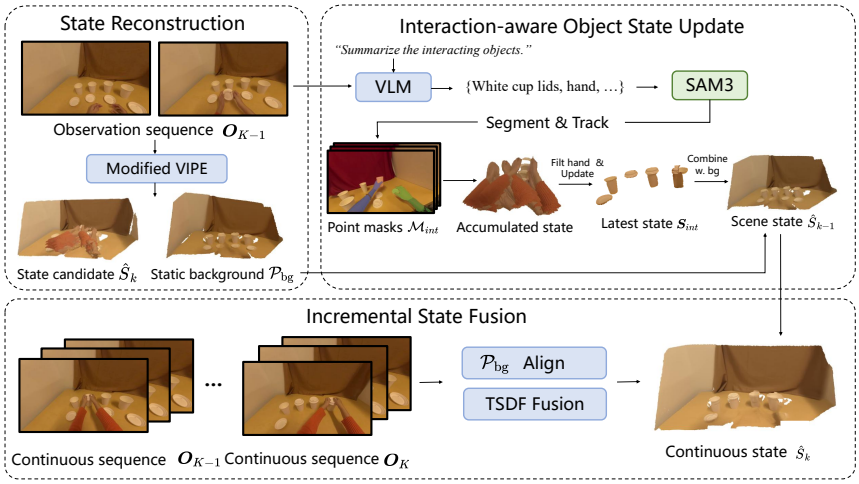

Fig. 3: Implementation details of Interaction-aware State Updating. State Reconstruction (left) builds a point cloud from observations via modified VIPE [11]. Object State Update (top-right) identifies and tracks interactive objects with a VLM and SAM3, compositing their latest geometry into the scene. Incremental State Fusion (bottom) aligns and merges consecutive states via TSDF fusion.

We further initialize 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 [13], 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 $\{O_0, O_1, \ldots, O_{n-1}\} \subset \boldsymbol{O}_{K-1}$ with n frames, as an editable 3D environment representation. Inspired by VIPE [11], our pipeline first estimates camera intrinsics via DepthAnything3 [19] and extracts instance masks via SAM3 [3] 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 [21] and SAM3 [3], 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 [35]

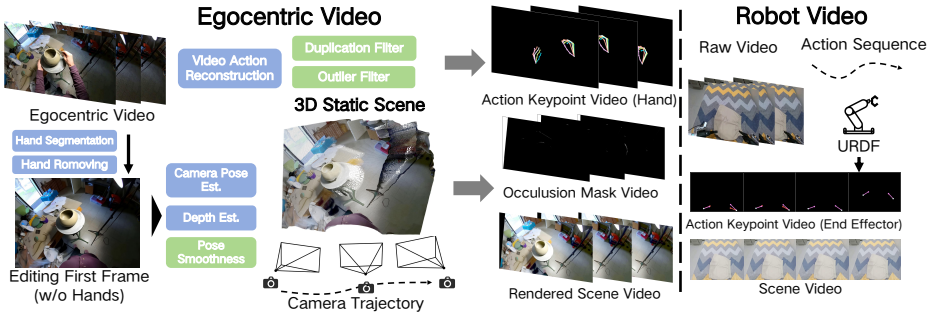

Fig. 4: Overview of our scalable data processing pipeline. For both human egocentric videos and robotic manipulation videos, we automate the extraction of aligned triplets: static 3D scene point clouds, precise camera trajectories, and dynamic interaction sequences represented as spatial action keypoints.

to identify object phrases involved in embodiment interactions. An open-vocabulary tracking pipeline based on SAM3 [3] 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 [3] to generate masks for hand regions, which are subsequently removed using the Qwen-Image-Editing [34] model. Finally, DepthAnything3 [19] 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 [19] 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 [23] 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 to construct a large-scale dataset with high-quality scene-interaction pairs. More details can be found in Table 3 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 [16] streaming reconstruction framework, with a dynamic intrinsic self-calibration module that eliminates the need for prior calibration, yielding a scale-consistent scene point cloud.

Dynamic Interaction Recording and Relocalization. The user

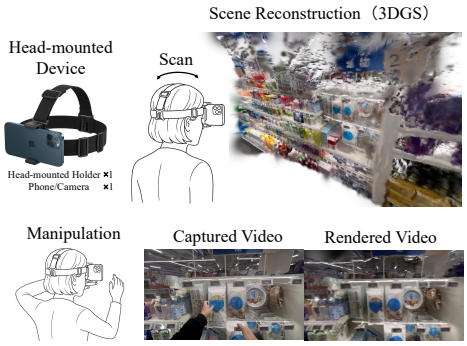

Fig. 5: The EgoCap pipeline. An uncalibrated head-mounted smartphone first scans the scene to build a 3DGS map, then records the ego-view interaction. Relocalizing against the map yields viewpoint-aligned paired data.

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 [10] dataset and 160K video clips from the EgoVid [31] 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 [27], 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** [27], a mask-guided video inpainting model; (2) **InterDyn** [1], which injects hand mask video into Stable Video Diffusion via ControlNet; (3) **Mask2IV** [15], 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** [25], 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 [32]. 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)	<u>19.165</u>	<u>0.835</u>	<u>0.220</u>	<u>10.943</u>	<u>0.0017</u>

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

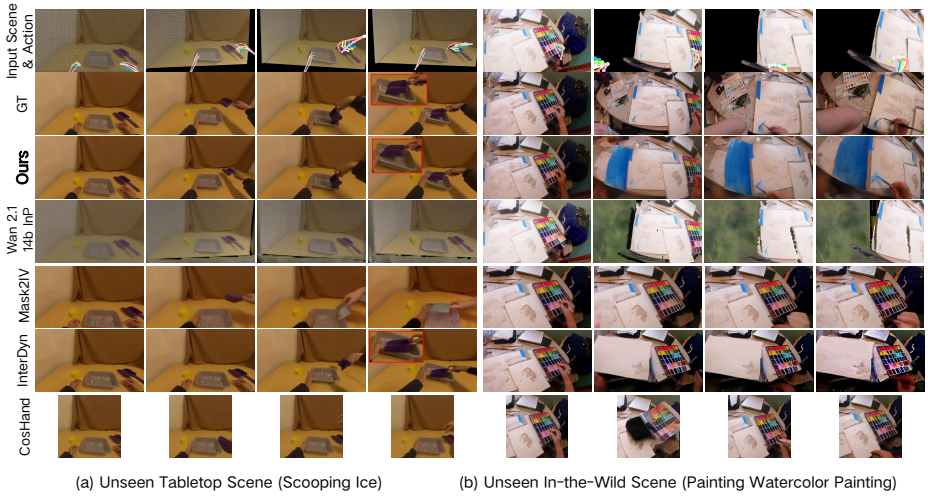

Fig. 6: Qualitative comparison of EgoSim against state-of-the-art baselines on unseen test environments. (a) Inference on Egodex. (b) Inference on Egovid.

main checkpoint and fine-tuned for only 50 steps. As shown in Figure 7, 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.

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.

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

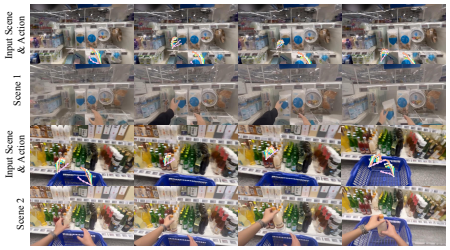

Fig. 7: Real-world result of EgoSim across diverse in-the-wild scenes.

Fig. 8: 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	16.36	0.69	0.31
w/ hand pretrain	**18.67**	**0.72**	**0.28**

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.

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.

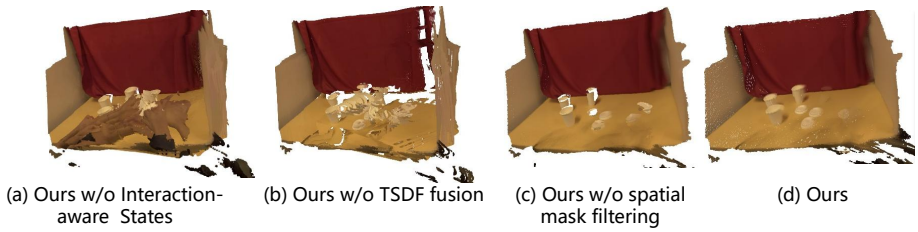

Fig. 9: Qualitative ablation of Interaction-aware State Updating. Our full model (d) maintains a clean, coherent 3D representation.

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 9). This explicitly proves that the physical displacement of objects is permanently registered in 3D space. Without interactive object filtering (Figure 9a), the static point cloud becomes corrupted by dynamic elements like hands, leading to ghosting artifacts. Removing TSDF fusion (Figure 9b) produces fragmented and noisy surface reconstructions that fail to provide a reliable geometric anchor. Similarly, omitting spatial mask filtering (Figure 9c) introduces floating artifacts and erroneous projections during dynamic interactions. By integrating all these components, our full pipeline (Figure 9d) 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 [2] 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. To ensure a controlled comparison, both settings use the same total number of training steps: *w/o hand pretrain* trains on robot data for 1,400 steps, while *w/ hand pretrain* first pretrains on egocentric human hand data for 1,200 steps and then finetunes on robot data for 200 steps.

As shown in Table 4, pretraining on large-scale egocentric human hand data before finetuning on AgiBot data (*w/ human pretrain*) substantially outperforms robot-only training (*w/o human pretrain*) across all metrics, boosting PSNR

from 16.36 to 18.67, improving SSIM from 0.69 to 0.72, and reducing LPIPS from 0.31 to 0.28. Indicating better spatial and object consistency in the generated robot videos.

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.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62595774, 72192821, 62302297, 62472282), the Fundamental Research Funds for the Central Universities (project number: YG2023QNA35), YuCaiKe [2023] Project Number: 231111310300. Thanks to Hongrui Zhu for providing support for the real-world robot experiments.

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. 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)
3. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025)
4. DeepMind, G.: Genie: Generative interactive environments. Web page (2026), https://deepmind.google/models/genie/

5. 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)
6. 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)
7. 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)
8. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 **2**(3), 440 (2018)
9. 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)
10. 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)
11. 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)
12. 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)
13. Kim, B., Kim, T., Lee, J., Joo, H.: Dexterous world models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 29663–29673 (2026)
14. 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)
15. Li, G., Zhao, B., Yang, J., Sevilla-Lara, L.: Mask2iv: Interaction-centric video generation via mask trajectories. In: AAAI (2026)
16. 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)
17. Li, J., Pang, B., Wang, P.S.: Joint point cloud upsampling and cleaning with octree-based cnns. Computational Visual Media (2026)
18. 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)
19. 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)
20. Lin, H., Wang, Y., Yin, B., Yang, X.: A review of learning based visual relocalization methods. Computational Visual Media (2025)
21. 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)
22. 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)

23. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing hands in 3D with transformers. In: CVPR (2024)
24. Qian, Z., Chi, X., Li, Y., Wang, S., Qin, Z., Ju, X., Han, S., Zhang, S.: Wristworld: Generating wrist-views via 4d world models for robotic manipulation. arXiv preprint arXiv:2510.07313 (2025)
25. 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)
26. Team, R., Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., et al.: Advancing open-source world models. arXiv preprint arXiv:2601.20540 (2026)
27. 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)
28. Tu, Y., Luo, H., Chen, X., Bai, X., Wang, F., Zhao, H.: Playerone: Egocentric world simulator. Advances in Neural Information Processing Systems **38**, 145235–145261 (2026)
29. 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. Advances in Neural Information Processing Systems **38**, 35928–35959 (2026)
30. 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. arXiv preprint arXiv:2603.17808 (2026)
31. 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)
32. Wang, Y., Ouyang, W., Wei, T., Dong, Y., Shen, Z., Pan, X.: Hand2world: Autoregressive egocentric interaction generation via free-space hand gestures. arXiv preprint arXiv:2602.09600 (2026)
33. Wu, T., Yuan, Y.J., Zhang, L.X., Yang, J., Cao, Y.P., Yan, L.Q., Gao, L.: Recent advances in 3d gaussian splatting. Computational Visual Media **10**(4), 613–642 (2024)
34. 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)
35. 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)
36. 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)
37. Zhao, J., Wei, F., Liu, Z., Zhang, H., Xu, C., Lu, Y.: Spatia: Video generation with updatable spatial memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4245–4257 (2026)