KINEMA4D: KINEMATIC 4D WORLD MODELING FOR SPATIOTEMPORAL EMBODIED SIMULATION

Anonymous authors
Paper under double-blind review

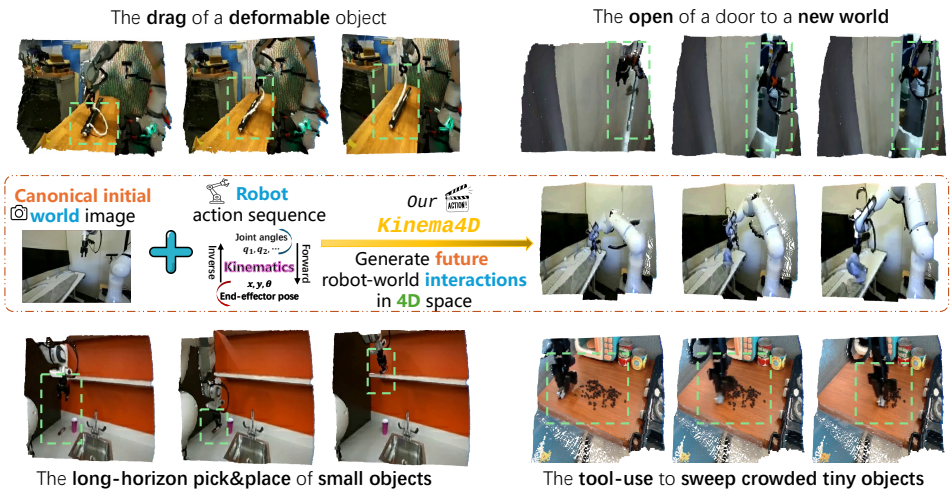

Figure 1: We propose **Kinema4D**, a new *action-conditioned 4D generative embodied simulator*. Given an initial world image with a robot at a canonical setup space, and an action sequence, our method generates future robot-world interactions in 4D space. It simulates physically plausible and geometrically consistent interactions between complex robot actions and diverse objects across various spatial constraints, providing a new foundation for next-generation embodied simulation.

ABSTRACT

Simulating robot-world interactions is a cornerstone of Embodied AI. Recently, a few works have shown promise in leveraging video generations to transcend the rigid visual/physical constraints of traditional simulators. However, they primarily operate in 2D space or are guided by static environmental cues, ignoring the fundamental reality that robot-world interactions are *inherently 4D spatiotemporal events* that require *precise* interactive modeling. To restore this 4D essence while ensuring the precise robot control, we introduce Kinema4D, a new action-conditioned 4D generative robotic simulator that disentangles the robot-world interaction into: i) *Precise* 4D representation of *robot controls*: we drive a URDF-based 3D robot via kinematics, producing a precise 4D robot control trajectory. ii) *Generative* 4D modeling of *environmental reactions*: we project the 4D robot trajectory into a pointmap as a spatiotemporal visual signal, controlling the generative model to synthesize complex environments' reactive dynamics into synchronized RGB/pointmap sequences. To facilitate training, we curated a large-scale dataset called Robo4D-200k, comprising 201,426 robot interaction episodes with high-quality 4D annotations. Extensive experiments demonstrate that our method effectively simulates physically plausible and geometry-consistent dynamics that faithfully mirror diverse real-world cases. It shows potential *zero-shot* transfer capability, providing a new foundation for next-generation embodied simulation.

1 INTRODUCTION

In the field of Embodied AI, the ability to roll out robot trajectories within a world environment is pivotal for scaling up robot demonstrations O'Neill et al. (2024), policy evaluation Li et al. (2024b); Team et al. (2025b), and reinforcement learning Ebert et al. (2018). Nevertheless, executing actions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

in the real world is prohibitively costly, potentially unsafe, and necessitates constant expert maintenance Brohan et al. (2022b); Khazatsky et al. (2024). Consequently, robotic simulation in virtual environments has emerged as a vital surrogate for real-world deployment.

While significant efforts have been dedicated to developing robust physical simulators Xiang et al. (2020); Makoviychuk et al. (2021); Mittal et al. (2023); Chen et al. (2024), these platforms often lack visual realism and rely on hand-crafted and pre-defined physical properties/rules. Such dependencies create a significant scalability bottleneck, particularly when attempting to synthesize new environments. Very recently, to circumvent these limitations, researchers have begun leveraging the intrinsic world dynamics captured by video generative models to synthesize robot-environment interactions across diverse scenes Zhu et al. (2025); Agarwal et al. (2025). By casting robot actions as conditional prompts for video synthesis, these methods directly simulate the visual outcome of adding robot controls to an environment, bypassing the need for predefined physical modeling required for traditional simulation. However, a critical gap remains. Current models primarily operate within a 2D pixel space Zhu et al. (2025); Agarwal et al. (2025), whereas robot-world interactions are *inherently 4D spatiotemporal events*. Without 4D constraints, physical interaction loses its most fundamental grounding. Although a few latest works Zhen et al. (2025); Liu et al. (2026) explored 4D simulation, they primarily rely on high-level linguistic instructions to represent robot control. Such semantic representations, while intuitive, *lack the precise guidance* essential for high-fidelity 4D world modeling. As a result, existing methods struggle to simulate precise interactive effects—such as material deformations or occluded object dynamics (see Fig. 4 and 5).

To bridge this gap and advance this problem, we propose Kinema4D, a new action-conditioned 4D generative robotic simulator. Our core motivation is to restore robot-world interactions to their *4D spatiotemporal essence*, while ensuring *precise robot control*. This motivation is further grounded in two synergetic insights that support our design philosophy—disentangling the simulation into *robot control* and their *resultant environmental changes*:

i) *Precise* 4D representation of robot actions via kinematic control: We recognize the fact that robot action is a *precise physical certainty in 4D space* and should not be "guessed" by a generative model. A joint-angle sequence or an end-effector pose is merely an abstract vector until it is mapped onto a physical structure. Reflecting on this, our framework enables a bridge that is otherwise impossible in 2D: we drive a 3D-reconstructed, URDF-based robot model via explicit kinematics to produce a continuous 4D robot trajectory that is guaranteed to be kinematically correct, providing the high-granularity spatiotemporal information that serves as the *causal driver* for any interaction.
ii) *Generative* 4D modeling of environmental reactions via controllable generation: While robot controls are deterministic, we argue that complex environments' dynamics require a *flexible generative modeling*. To this end, we project the previously derived 4D robot trajectory into a pointmap sequence—serving as a spatiotemporal visual signal that controls the generative model to offload the burden of kinematic modeling and focus on synthesizing the environment's *reactive dynamics*. Moreover, our architecture predicts synchronized RGB and pointmap sequences, effectively finishing a spatiotemporal reasoning task within a unified 4D space. In this paradigm, spatiotemporal awareness is enforced not merely as a control signal but as an intrinsic constraint throughout the generative process. This fosters a symbiosis between *precise control* and *flexible synthesis*.

To facilitate the training, we curated a large-scale dataset called Robo4D-200k, comprising 201,426 real-world and synthetic demonstrations with high-quality 4D annotations. We extensively benchmark our method using both video and geometric metrics, as well as policy evaluation. Experiments show that our Kinema4D generalizes to simulate physically plausible robot-world interactions that closely mirror diverse real-world dynamics, and shows potential zero-shot OOD transfer capability.

Our contributions are summarized as:

- Kinema4D, a new action-conditioned 4D generative robotic simulator that enjoys both spatiotemporal rigor and generative flexibility.
- Robo4D-200k, to our knowledge, the largest-scale 4D robotic dataset comprising 201,426 demonstrations with high-quality 4D annotations.
- A comprehensive evaluation scheme demonstrating that our method mirrors real-world dynamics, providing a new foundation for embodied simulation.
- Our code, dataset, and checkpoints will all be public upon acceptance.

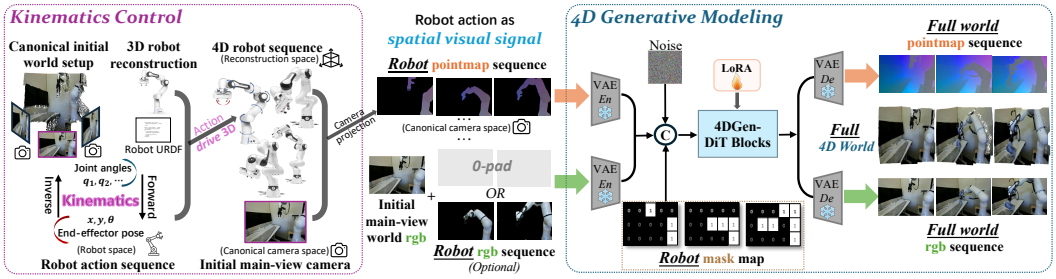

Figure 2: **Overview of our *Kinema4D*. *1) Kinematics Control***: Given a 3D robot with its URDF at initial canonical setup space, and an action sequence, we drive the 3D robot via kinematics to produce a 4D robot trajectory, which is then projected into a pointmap sequence. This process re-represents raw actions as a spatiotemporal visual signal. ***2) 4D Generative Modeling***: This signal and the initial main-view world image are sent to a shared VAE encoder, then fused with an occupancy-aligned robot mask and noise, which are denoised by a Diffusion Transformer Peebles & Xie (2023) to generate a full future 4D (pointmap+RGB) world sequence.

2 Related Work

Physical simulation. Physics-based robotic simulators Todorov et al. (2012); Xiang et al. (2020); Makoviychuk et al. (2021); Mittal et al. (2023) model interactions through explicit dynamics, requiring geometric assets and physical parameters. Real-to-sim approaches improve visual fidelity through reconstructed digital twins Li et al. (2024a); Abou-Chakra et al. (2024); Jiang et al. (2025a), but still depend on asset preparation and physical modeling. These requirements motivate learning environmental responses from data while retaining analytical robot kinematics.

Embodied video-generation models. Embodied video models generate future observations conditioned on text Yang et al. (2024a); Zhou et al. (2024) or action embeddings Zhu et al. (2025); Agarwal et al. (2025); Guo et al. (2026). Such conditioning leaves robot kinematics implicit in the learned action-to-video mapping. ORV Yang et al. (2026) introduces 3D semantic occupancy as environmental guidance, while EVAC, AnchorDream, and VAP use 2D visual prompts Jiang et al. (2025b); Ye et al. (2025); Wang et al. (2025b). Closely related to our approach, BridgeV2W Chen et al. (2026) drives robots with URDF models and renders their trajectories as binary masks. Although these masks specify projected robot motion, they do not explicitly encode its metric 3D geometry. Kinema4D instead represents actions as camera-aligned robot pointmap sequences, providing explicit geometric control over time.

3D/4D world models. ParticleFormer Huang et al. (2025b) and PointWorld Huang et al. (2026) model dynamics through 3D particle trajectories, while Aether Team et al. (2025a) and 4DNex Chen et al. (2025) explore generative 4D world modeling. Embodied generators, including TesserAct Zhen et al. (2025), GWM Lu et al. (2025), iMoWM Zhang et al. (2025a), and Robo4DGen Liu et al. (2026), incorporate scene geometry but represent robot actions through language or latent tokens. Their geometric outputs therefore do not by themselves establish explicit kinematic grounding for control. Building on 4DNex, Kinema4D couples kinematics-derived robot pointmaps with joint RGB and world-pointmap generation. This grounds both action conditioning and generated outcomes in 4D space, enabling the model to predict environmental reactions to geometrically specified robot motion.

3 Our Approach

Fig. 2 illustrates our architecture, composed of two components: Kinematic Control (Sec. 3.1) and 4D Generative Modeling (Sec. 3.2).

3.1 Kinematics Control

The core of our framework lies in transforming abstract robot actions into a precise 4D representation, via a progressive process.

3D robot asset acquisition. To ground the robot in 4D space, we first establish its geometric entity. For standardized robots, we utilize factory-provided 3D CAD meshes. For unknown platforms, we implement a reconstruction pipeline to recover a textured robot mesh $\mathcal{C}_{recon}$; details are provided in supplementary material.

We first capture orbital videos of the robot and sample representative frames, then employ Grounded-SAM2 Ravi et al. (2024); Liu et al. (2023b) to segment the robot in the initial frame using a robust prompt (*e.g.*, `\the robot arm"`), followed by SAM2 Ravi et al. (2024) to propagate masks across the sequence via video tracking.

To enable articulation, we establish a digital twin alignment: the joint anchor points from the robot's URDF model $\mathcal{M}$ are mapped to their corresponding coordinates of $\mathcal{C}_{recon}$ within the canonical reconstruction space. This ensures that the analytical kinematic chain can directly drive the reconstructed mesh segments.

Kinematics-driven 4D robot trajectory expansion. Given the aligned robot model $\mathcal{M}$ within $\mathcal{C}_{recon}$, we transform input actions $\mathbf{a}_{1:T}$ into full-body 4D trajectories. Our framework handles two primary control modalities:

- *End-effector control.* When actions are provided as Cartesian poses $\{\mathbf{T}_{ee,t}\}_{t=1}^{T}$, we employ an Inverse Kinematics (IK) solver to resolve the joint configurations at time t: $\mathbf{q}_t = \text{IK}(\mathbf{T}_{ee,t}, \mathbf{q}_{t-1}, \mathcal{M})$, where the previous state $\mathbf{q}_{t-1}$ serves as the seed to ensure temporal smoothness and prevent joint-space flipping.
- *Joint-space control.* When actions are directly provided as joint angles or velocities, $\mathbf{q}_t$ is obtained via direct mapping or integration.

For each time t, we perform Forward Kinematics (FK) to compute the 6-DoF poses for all K links within the reconstruction space: $\{\mathbf{T}_{k,t}^{recon}\}_{k=1}^{K} = \text{FK}(\mathbf{q}_t, \mathcal{M})$.

Spatial-visual projection. We select a primary viewpoint—typically a medial-frontal perspective—for projection and later generation, as single-view robot demonstration data is extensively accessible to ensure training scalability. To place the articulated robot within the initial main-view world image I_0, we utilize the extrinsic camera transformation $\mathbf{T}_{recon}^{cam} \in SE(3)$ of the selected main view, which is derived during the previous 3D robot reconstruction phase Chang et al. (2025).

With the camera transformation $\mathbf{T}_{recon}^{cam}$ and the previously computed link poses $\{\mathbf{T}_{k,t}^{recon}\}$ within the reconstruction space, the full-body trajectory is projected onto the image plane to generate the 4D robot pointmap $\mathbf{M}_{1:T} \in \mathbb{R}^{H\times W\times 3}$. For any point $\mathbf{x}$ on the surface of link k, its projected pixel coordinates (u, v) and depth z are determined by:

$$\begin{bmatrix} u \cdot z \\ v \cdot z \\ z \end{bmatrix} = \mathbf{K} \cdot \mathbf{T}_{recon}^{cam} \cdot \mathbf{T}_{k,t}^{recon} \cdot \mathbf{x}, \tag{1}$$

where $\mathbf{K}$ is the camera intrinsic matrix. The resulting pointmap $\mathbf{M}_{1:T}$ is pixel-aligned with the RGB grid, while its pixel values store the camera-space (x, y, z) coordinates.

Optionally, we project the textured trajectory to RGB sequence.

Table. 4 ablates the robustness to the noisy pointmap. More details of our reconstruction and projection are provided in the supplementary material.

3.2 4D Generative Modeling

Next, we utilize a 4D diffusion model to focus on synthesizing the environment's reactive dynamics in response to the robot control signal.

We build our framework upon Latent Diffusion Models (LDM) Rombach et al. (2022), extending them to the 4D spatiotemporal domain. Instead of pixel-space denoising, the model operates on a compressed latent space $\mathcal{Z}$ provided by a pre-trained VAE. The conditional denoising objective is detailed in the supplementary material.

Multi-modal latent construction. To prepare the multi-modal input for our generative backbone, we first align the temporal dimensions of the initially captured RGB world image I_0 and the robotic



Figure 3: **Samples from our *Robo4D-200k* dataset**. Our dataset provides a comprehensive data foundation by aggregating diverse real-world demonstrations, including DROID Khazatsky et al. (2024), Bridge Walke et al. (2023), and RT-1 Brohan et al. (2022a). We further incorporate the LIBERO Liu et al. (2023a) to synthesize a vast array of successful/failure cases. Each episode captures a *complete* robot-world interaction (*e.g.*, pick-and-place)—providing the continuous information necessary for robust reasoning. The 4D point clouds viewed from various camera frustums are shown here, demonstrating the spatial precision of our pseudo-annotations.

control signals. Specifically, I_0 is extended along the temporal axis via zero-padding or by concatenating the previously robot RGB sequence. Next, following the data formatting strategy in 4DNex Chen et al. (2025), we concatenate this input with the previously produced robot pointmap sequence $\mathbf{M}_{1:T}^{robot}$ along the width dimension. This unified spatiotemporal signal is then processed by a shared VAE to obtain input latents, which maps the heterogeneous inputs—RGB and pointmap—into a synchronized latent representation.

To further enforce pixel-level control, we introduce a guided mask $\mathbf{m} \in \{0,1\}^{T\times H\times W}$, where $\mathbf{m}_{t,i,j} = 1$ indicates the spatial occupancy of the robot (derived from $\mathbf{M}_{1:T}^{robot}$) and 0 indicates the area to be generated. In practice, rather than employing a binary hard-mask, we implement a soft strategy by setting the value of 10% occupied regions to 0.5 (ablated in Table. 4). This design choice ensures that the generative model retains the capacity to *refine* the robot's visual signal. By doing so, we mitigate the impact of noise introduced during the previous phase, enhancing the structural robustness and visual coherence.

We concatenate input latents, noisy latents, and robot masks channel-wise. This fused input ensures that the generative model focuses exclusively on synthesizing the environment's reactive dynamics while *strictly adhering to the robot's trajectory*.

4D-aware joint modeling. Our backbone is a Diffusion Transformer Peebles & Xie (2023) that predicts synchronized RGB and pointmap sequences. To preserve pixel-wise alignment across these modalities, we adopt a shared Rotary Positional Encoding (RoPE) Su et al. (2024) across both RGB and pointmap latents. Additionally, following 4DNex Chen et al. (2025), we distinguish these domains using learnable domain embeddings. These embeddings act as modality-specific signatures, allowing the Transformer to perform cross-modal reasoning—effectively using the robot pointmap as geometric anchors to guide the synthesis of the RGB environmental response.

4D sequence synthesis. The denoised latents are processed by the shared VAE Decoder to reconstruct the full-world pointmap/RGB sequence. By predicting the environment's pointmap $\mathbf{M}_{1:T}^{world}$, the model produces a simulation that is not only visually realistic but also geometrically rigorous, ultimately yielding a 4D world where every pixel's depth and motion are grounded in 3D space.

3.3 Robo4D-200k: A Large-Scale 4D Robotic Dataset

To begin with, we aggregate 2D RGB video streams from leading real-world robotic demonstration repositories, including DROID Khazatsky et al. (2024), Bridge Walke et al. (2023), and RT-1 Brohan et al. (2022a). We further incorporate the LIBERO Liu et al. (2023a) platform to synthesize an extensive array of both successful executions and critical failure modes.

ST-V2 Xiao et al. (2025) is used to reconstruct high-quality, pixel-aligned 4D trajectories for real datasets. For synthetic data from LIBERO, we directly leverage the native, noise-free depth parameters to ensure absolute ground-truth precision.

Table 1: **Quantitative comparison of video generation metrics**. ↑ and ↓ respectively indicates higher and lower is better. **xxx** denotes the best and xxx indicates the 2nd-best. "Action" means the representation of robot controls, where "Emb." is token embedding. "Output" represents the modality of generation output.

Method	Action	Output	PSNR↑	SSIM↑	L2$_{latent}$↓	FID↓	FVD↓	LPIPS↓
UniSim [ICLR'24] Yang et al. (2024a)	Text	RGB	19.32	0.681	0.2120	32.3	153.2	0.175
IRA-Sim [ICCV'25] Zhu et al. (2025)	Emb.	RGB	20.21	0.813	0.1722	25.2	126.0	0.135
Cosmos [arXiv'25] Agarwal et al. (2025)	Emb.	RGB	20.39	0.787	0.1935	27.1	113.4	0.110
EVAC [arXiv'25] Jiang et al. (2025b)	Emb.+2D	RGB	20.88	0.832	0.1896	29.3	122.0	0.150
ORV [CVPR'26] Yang et al. (2026)	Emb.+3D	RGB	19.45	0.790	0.2002	30.1	130.1	0.143
Ctrl-World [ICLR'26] Guo et al. (2026)	Emb.	RGB	21.03	0.803	0.1533	**24.9**	112.8	0.122
TesserAct [ICCV'25] Zhen et al. (2025)	Text	4D	19.35	0.766	0.1911	29.5	120.3	0.158
Ours	**4D**	**4D**	**22.50**	**0.864**	**0.1380**	25.2	**98.5**	0.105

Table 2: **Quantitative comparison of geometric metrics**. "(temp)" denotes the self-temporal consistency across frames. The threshold of F-Score is set to @0.01.

Method	CD-L_1↓	CD-L_1 (temp)↓	CD-L_2↓	CD-L_2 (temp)↓	F-Score↑	F-Score (temp)↑
TesserAct [ICCV'25] Zhen et al. (2025)	0.0836	**0.0067**	0.0130	0.0008	0.2896	0.9523
Ours	**0.0479**	0.0074	**0.0077**	**0.0002**	**0.4733**	**0.9686**

We applied a manual verification to prune low-quality captures and reconstruction artifacts. For very-long demonstrations, we applied uniform temporal downsampling to yield 49-frame sequences per episode. Following this pipeline, we established *Robo4D-200k*, comprising 201,426 high-fidelity episodes. More details of our dataset can be found in supplementary material.

4 Experiments

4.1 Setting

Implementation Details. We build our framework upon the WAN 2.1 Wan et al. (2025) base model (14B parameters), utilizing the 4D-aware pre-trained weights from 4DNex Chen et al. (2025). We employ Low-Rank Adaptation (LoRA) Hu et al. (2022) for parameter-efficient fine-tuning of the Diffusion Transformer Peebles & Xie (2023). We replace the text embedding with the VAE latents of robot sequences. For all other configurations, we follow the protocol established in the official 4DNex implementation. We use a stratified 2% subset of Robo4D-200k for validation. More implementation details, ablations of text embeddings, and benchmark details are provided in the supplementary material.

Baselines. UniSim Yang et al. (2024a) uses text instructions, while IRASim Zhu et al. (2025), Cosmos Agarwal et al. (2025), and Ctrl-World Guo et al. (2026) use action embeddings. EVAC Jiang et al. (2025b) and ORV Yang et al. (2026) incorporate 2D angle arrows and 3D environmental occupancy grids, respectively. TesserAct Zhen et al. (2025) generates 4D outputs (RGB, depth, and surface normals) conditioned on text instructions. We prioritize models with open-sourced implementations to ensure a fair comparison.

Metrics. We evaluate video quality using PSNR Hore & Ziou (2010), SSIM Wang et al. (2004), Latent L2 loss, FID Heusel et al. (2017), FVD Ranftl et al. (2020), and LPIPS Zhang et al. (2018). For TesserAct and our method, we additionally use Chamfer Distance and F-Score@0.01 to assess geometric accuracy against ground truth and temporal consistency across frames ("temp").

4.2 Main Results

Quantitative results. The quantitative evaluation of video synthesis quality is summarized in Table. 1. Our Kinema4D is uniquely characterized by its ability to precisely represent actions and synthesize outcomes within a unified 4D space. This advantage allows our framework to achieve either the leading or second-best performance across all metrics. The geometric fidelity of the generated 4D sequences is detailed in Table. 2. While TesserAct Zhen et al. (2025) exhibits slightly better performance in temporal CD-L_1, our approach outperforms TesserAct across all other met-

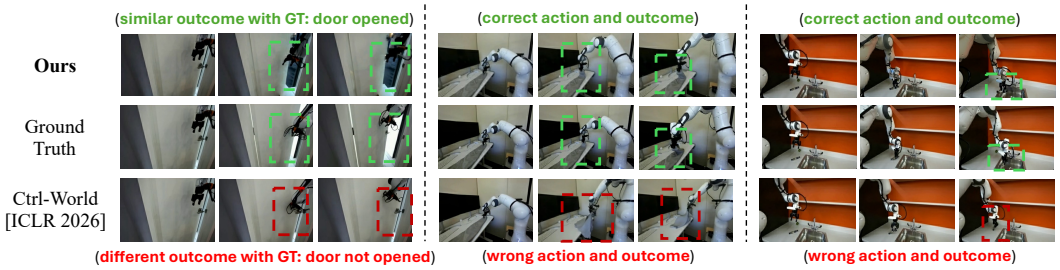

Figure 4: **Qualitative comparison of 2D video synthesis** between our Kinema4D and Ctrl-World Guo et al. (2026). Our method achieves superior fidelity. In contrast, Ctrl-World exhibits distorted actions and unrealistic environmental transitions.

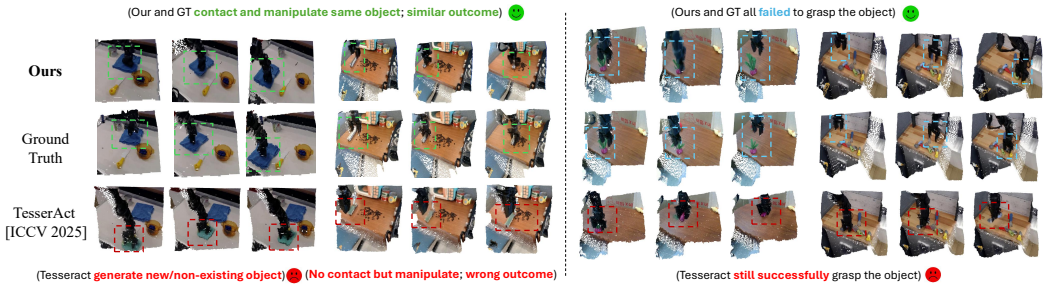

Figure 5: **4D qualitative comparison between our Kinema4D and TesserAct Zhen et al. (2025).** Unlike TesserAct that hallucinate outcomes, our Kinema4D precisely reflects Ground-Truth executions, including "near-miss" failure cases. For example, in the bottom left corner example, our model *correctly interprets the spatial gap* between the gripper and the plant, *even when their* ***RGB textures overlap in 2D views***.

rics. When evaluated against absolute ground-truth rather than mere self-consistency, our method shows a significant margin of superiority.

Qualitative results. We compare our method with the best baseline that outputs 2D videos (*i.e.*, Ctrl-World Guo et al. (2026)), focusing on the qualitative results under 2D video modality. As shown in Fig. 4, our Kinema4D demonstrates a superior ability to synthesize high-fidelity sequences that strictly adhere to input action trajectories while generating physically consistent environmental responses that are highly similar to the ground truth. In contrast, as Ctrl-World is limited to 2D video synthesis and relies on implicit robot embeddings, it often produces distorted robot kinematics and unrealistic environmental changes.

In Fig. 5, we present a 4D qualitative comparison between our framework and TesserAct Zhen et al. (2025). As TesserAct relies on text instructions, it struggles to align with the actual robot actions, and often yields incorrect environmental outcomes. Instead, by utilizing precise 4D controls, our Kinema4D demonstrates the ability to simulate ***both successful*** *rollouts and* ***subtle execution failures***. ***Most notably***, even when the RGB textures of the gripper and the object appear to "contact" from a 2D perspective, our 4D-aware framework correctly interprets the spatial gap and simulates a 'near-miss' or a failed grasp.

In a pilot experiment, we integrate a WAN-compatible accelerated variant from TurboDiffusion Zhang et al. (2025b) and fine-tune it for two days, achieving a $152\times$ speedup with PSNR, FID, and CD-L_1 of 21.48, 26.0, and 0.0503, respectively. More animated results are presented in the supplementary material.

4.3 Policy Evaluation

We investigate whether Kinema4D can support policy evaluation by accurately predicting the environmental outcomes of robot action sequences. Our evaluation covers both controlled simulation and real-world environments with complex physical interactions.

Table 3: Policy evaluation on three pick-and-place setups in simulation and real-world environments. Real-world results include zero-shot OOD evaluation and evaluation after fine-tuning on 100 real trajectories. The first two rows report actual and predicted success rates, respectively; rows marked "Gap" report absolute differences from Ground Truth. "–" indicates results not reported.

Evaluator	**Simulation**			**Real-world (OOD)**		
	1	2	3	1	2	3
Ground Truth	0.48	0.38	0.80	0.34	0.46	0.78
Ours	0.56	0.46	0.84	0.60	0.76	0.90
Ctrl-World Guo et al. (2026) (Gap)	0.14	0.13	0.09	0.10	0.11	0.08
IRASim Zhu et al. (2025) (Gap)	–	–	–	0.32	0.40	0.16
Ours (Gap)	0.08	0.08	0.04	0.26	0.30	0.12

Setup and implementations. In LIBERO Liu et al. (2023a), we execute a Diffusion Policy Chi et al. (2025) to obtain action rollouts, then use Kinema4D to synthesize the resulting environmental responses. Robot pointmaps are obtained directly from native meshes and camera parameters, avoiding reconstruction and calibration errors. For real-world evaluation, we execute the policy on a physical robot and obtain robot pointmaps through our reconstruction pipeline (Sec. 3.1). The real-world OOD evaluation is strictly zero-shot, without fine-tuning on the target setup. For both domains, we evaluate three scenarios with 50 runs each.

Results. Table 3 reports success rates and absolute success-rate gaps. In simulation, Kinema4D achieves gaps of 0.04–0.08, indicating close agreement with actual executions. This is consistent with the inclusion of LIBERO data in training. Real-world OOD evaluation yields larger gaps of 0.12–0.30, highlighting the challenges of zero-shot transfer. Nevertheless, Kinema4D achieves lower gaps than Ctrl-World Guo et al. (2026) and IRASim Zhu et al. (2025) in all three scenarios under the same zero-shot setting. In a separate in-domain experiment, fine-tuning Kinema4D on 100 real trajectories further reduces the real-world gaps to 0.16, 0.14, and 0.08. Our model still tends to overestimate success, possibly reflecting the high proportion of successful training trajectories (approximately 73%), which may bias generation toward successful outcomes.

The qualitative results in Fig. 6 illustrate two aspects of our model. First, consistent with Fig. 5, it predicts near-miss failures even when the gripper and object overlap in 2D views, reflecting their separation in 3D space. Second, it produces coherent environmental responses despite noisy robot pointmaps from real-world reconstruction, demonstrating robustness to imperfect control signals. Additional simulation results and evaluation details are provided in the supplementary material.

Policy transfer. We further investigate whether Kinema4D-generated trajectories can provide useful training data for policy improvement. Following Ctrl-World Guo et al. (2026), we conduct a pilot experiment by rephrasing task instructions and sampling actions from a perturbed $\pi_{0.5}$ policy. Kinema4D generates 50 trajectories per setup, of which 10–20 are retained through human assessment. We then fine-tune the policy on the selected simulated trajectories for 2,000 steps, achieving an average success-rate improvement of **32.3%** on novel instructions and objects. These preliminary results suggest the potential of simulated trajectories for improving policy generalization.

4.4 Ablation Studies and Analysis

We provide comprehensive ablation analysis and report results in Table. 4.

Different representations of robot controls. We replace our robot pointmap conditions with other types of robot control representations, including text instructions ("text"), robot binary masks ("binary"), token embeddings ("emb."), robot RGB ("RGB") and robot RGB+pointmap ("RGB+pm") sequence. Our pointmap representation yields the 2nd-best results. Interestingly, using RGB+pointmap sequence yields marginal improvements, as RGB data potentially introduces more noise and overfitting risks. Instead, pointmap already captures the essential spatial information required for precise control.

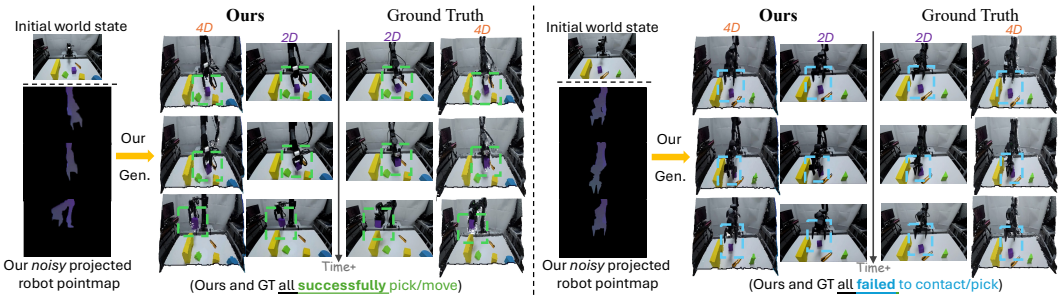

Figure 6: **Qualitative comparison between real-world Ground Truth (GT) and our method for zero-shot OOD evaluation**. To intuitively compare with GT, we also reconstruct the GT using Xiao et al. (2025) and present both 4D and 2D results. Our results strongly align with real outcomes, accurately synthesizing both successful rollouts and "near-miss" failures. For instance, in the right figure, our model *correctly interprets the spatial gap* between the gripper and the cube, *even when their **RGB textures overlap in 2D views***. Besides, the *noisy* robot pointmap produced by our reconstruction pipeline is shown, proving the robustness of our framework.

Table 4: Quantitative results of ablation studies.

Metrics	Ours	text	binary	emb.	RGB	RGB+pm	single	2D-out	w/o mask	0%	20%	50%	70%	remove	gaus.	trans.	rot.
PSNR↑	22.50	19.89	21.47	20.89	21.53	**22.98**	21.26	20.07	21.03	21.10	22.04	21.75	21.83	22.48	21.98	21.87	22.34
FID↓	25.2	28.8	27.5	26.3	25.8	25.7	26.7	27.4	26.8	26.1	25.2	**25.0**	26.0	26.0	25.3	25.6	26.3
CD-L_1↓	0.0479	0.0750	0.0639	0.0528	0.0677	0.0495	0.0581	0.0712	0.0510	0.0528	**0.0433**	0.0455	0.0463	0.0499	0.0501	0.0513	0.0483

Why not generate RGB videos first, then reconstruct? We trained our method to purely output RGB videos (no pointmap, "2D-out") and then reconstruct the produced RGB videos using ST-v2 Xiao et al. (2025). This significantly degrades the performance, proving the necessity of 4D awareness throughout the generative process.

The robot mask map. As in Sec. 3.2, we introduced a robot occupancy mask and softly set 10% occupied regions to 0.5. Here we trained our method without robot mask or using different soft mask ratio. Our method performs stably well under different ratios, yet discarding the mask or setting 0% regions to 0.5 degrades the performance, showing the necessity of generative refinement.

The robustness to the robot pointmap noise. We disturb the produced robot pointmap by randomly removing (5%), adding gaussian noise ("gaus."), translating (± 5 pixel coordinates, "trans.") and rotating ($\pm 5°$, "rot."). Our framework is robust to the noise of robot pointmap control, potentially due to the prior of the initial world image and the refinement of generative modeling.

5 Conclusion

We presented Kinema4D, a novel framework that shifts the paradigm of robotic simulation to 4D spatial-temporal reasoning. By integrating a kinematics-driven grounding stage with a diffusion transformer-based generative pipeline, we effectively decouple deterministic robot motion from stochastic environmental reactions. Experimental results demonstrate that Kinema4D can generalize to simulate diverse real-world dynamics. By providing a 4D world, Kinema4D paves the way for scalable, high-fidelity, and complex embodied simulations.

Limitations. Since the environmental dynamics are learned through statistical synthesis rather than governed by explicit physical constraints (such as rigid-body dynamics or friction coefficients), our method may occasionally produce behaviors that violate conservation laws or exhibit penetration artifacts (see the failure cases in the supplementary material), leaving a future research direction of incorporating physical laws into our model.

AI use statement

We used generative AI tools primarily to assist with manuscript writing and editing, including language refinement, organization, and checks of clarity, consistency, and LaTeX formatting. We also

used Qwen3-VL-plus to generate episode-level language annotations, as described in the supplementary material. AI-assisted content is subject to author review and verification. We take full responsibility for the final content of this work, including its technical claims, experimental results, and AI-assisted text and annotations.

ETHICS STATEMENT

Kinema4D is intended to support research on robotic simulation and policy evaluation. Robo4D-200k combines existing robotic demonstrations with simulated data; the use and redistribution of these resources are subject to the source datasets' licenses and privacy requirements. Our user study evaluates the geometric fidelity and temporal consistency of 4D annotations. Generated rollouts may overestimate policy success or fail to capture fine-grained physical interactions. Consequently, predictions from our simulator should be independently validated before being used to guide real-world robot deployment, particularly in settings where failures could cause harm.

REPRODUCIBILITY STATEMENT

The main paper and supplementary material document our method and experimental procedures to support reproducibility. The main paper describes the kinematic control pipeline and 4D generative architecture, while the supplementary material provides implementation details, training configurations, dataset construction and annotation procedures, evaluation settings, and real-world deployment protocols (Sec. A.2). Additional qualitative results and ablation studies support the assessment of the reported findings. We will release our code, dataset, and model checkpoints upon acceptance to further facilitate reproduction and future research.

REFERENCES

Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Suenderhauf. Physically embodied gaussian splatting: A realtime correctable world model for robotics. In *CoRL*, 2024.

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022b.

Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation. *arXiv preprint arXiv:2510.23306*, 2025.

Yixiang Chen, Peiyan Li, Jiabing Yang, Keji He, Xiangnan Wu, Yuan Xu, Kai Wang, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. Bridgev2w: Bridging video generation models to embodied world models via embodiment masks. *arXiv preprint arXiv:2602.03793*, 2026.

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4dnex: Feed-forward 4d generative modeling made easy. *arXiv preprint arXiv:2508.13154*, 2025.

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *IJRR*, 2025.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *ICLR*, 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, 2010.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. *arXiv preprint arxiv:2511.19971*, 2025.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers arXiv:2508.10934*, 2025a.

Suning Huang, Qianzhong Chen, Xiaohan Zhang, Jiankai Sun, and Mac Schwager. Particleformer: A 3d point cloud world model for multi-object, multi-material robotic manipulation. In *CoRL*, 2025b.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025a.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025b.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *CORL*, 2024b.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025a.

Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9080–9090. IEEE, 2025b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023b.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation. In *ICLR*, 2026.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *ICCV*, 2025.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IRAL*, 2023.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *ICRA*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *CVPR*, 2022.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 2024.

Aether Team, Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *ICCV*, 2025a.

Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025b.

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

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025a.

Yuang Wang, Chao Wen, Haoyu Guo, Sida Peng, Minghan Qin, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Precise action-to-video generation through visual action prompts. In *ICCV*, 2025b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *ICCV*, 2025.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *ICLR*, 2024a.

Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, Lifu Wang, Zhuo Chen, Sicong Liu, Yuhong Liu, Yong Yang, Di Wang, Jie Jiang, and Chunchao Guo. Tencent hunyuan3d-1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293*, 2024b.

Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Chen Zhaoxi, Minghan Qin, Ding Yikang, Xin Jin, Hang Zhao, and Hao Zhao. Orv: 4d occupancy-centric robot video generation. In *CVPR*, 2026.

Junjie Ye, Rong Xue, Basile Van Hoorick, Pavel Tokmakov, Muhammad Zubair Irshad, Yue Wang, and Vitor Guizilini. Anchordream: Repurposing video diffusion for embodiment-aware robot data synthesis. *arXiv preprint arXiv:2512.11797*, 2025.

Chuanrui Zhang, Zhengxian Wu, Guanxing Lu, Yansong Tang, and Ziwei Wang. imowm: Taming interactive multi-modal world model for robotic manipulation. *arXiv preprint arXiv:2510.09036*, 2025a.

Jintao Zhang, Kaiwen Zheng, Kai Jiang, Haoxu Wang, Ion Stoica, Joseph E Gonzalez, Jianfei Chen, and Jun Zhu. Turbodiffusion: Accelerating video diffusion models by 100-200 times. *arXiv preprint arXiv:2512.16093*, 2025b.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. In *ICCV*, 2025.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: learning compositional world models for robot imagination. In *ICML*, 2024.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. In *ICCV*, 2025.

A Appendix

Supplementary Material for Kinema4D

Outline

This supplementary document is arranged as follows:
(1) Sec. A.1 provides more qualitative results;
(2) Sec. A.2 elaborates more technical details and discussions of our framework implementation, dataset, and real-world deployment;
(3) Sec. A.3 discusses more ablation studies and analysis.

A.1 More Qualitative Results

fig. VII shows more 2D qualitative comparison with Ctrl-World Guo et al. (2026), fig. VIII visualizes more 4D qualitative comparison with TesserAct Zhen et al. (2025), fig. IX illustrates our qualitative results for policy evaluation in simulation platform, and fig. X provides extensive qualitative results of our method with Ground Truth as the reference. These results again demonstrate the effectiveness of our Kinema4D.

We also provide a demo video, ***demo.mp4***, for our work. In this video, a teaser introduction is provided from the beginning to 1:20. 1:20 $\sim$ 2:40 illustrates our framework, 2:40 $\sim$ 3:46 describes our dataset acquisition, 3:50 $\sim$ 7:00 provides quantitative and animated qualitative results for the main comparison, 7:00 $\sim$ 8:18 shows animated qualitative results for policy evaluation, 8:20 $\sim$ 8:30 visualizes extensive animated results.

A.2 More Technical Details

A.2.1 Framework Implementations.

Deployment of Kinematic Control. In the initial Kinematics Control phase, we implement a multi-view reconstruction-projection pipeline. For individual robot reconstruction, we capture sparse orbit videos via a smartphone and sample 20 representative frames. We employ Grounded-SAM2[1] Liu et al. (2023b); Ravi et al. (2024) to segment the target object in the initial frame using the prompt `\the object on the desk"`, a strategy that proved highly robust across varying

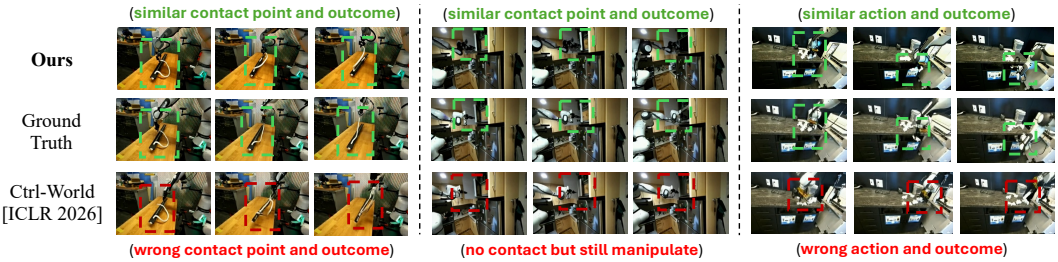

Figure VII: **Qualitative comparison of 2D video synthesis** between our Kinema4D and Ctrl-World Guo et al. (2026). Our method achieves superior fidelity. In contrast, Ctrl-World exhibits distorted actions and unrealistic environmental transitions.

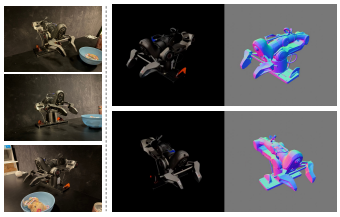

Figure XI: Reconstruction.

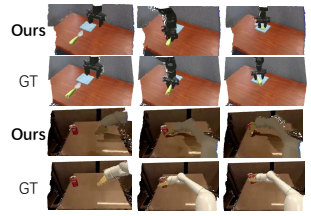

Figure XII: Simpler-Env.

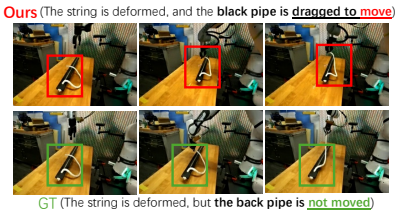

Figure XIII: Failure case.

lighting conditions. These segments are then propagated throughout the sequence using the temporal tracking capabilities of SAM2[2] Ravi et al. (2024). Given these masked multi-view images, we utilize ReconViaGen[3] Chang et al. (2025) for pose-free high-fidelity shape synthesis. Compared to traditional Structure-from-Motion (SfM) pipelines like COLMAP Schonberger & Frahm (2016); Schönberger et al. (2016) or recent feed-forward models like VGGT Wang et al. (2025a), ReconViaGen generates more complete meshes with significantly fewer holes in unobserved regions—a critical advantage under sparse, unconstrained viewpoints. Furthermore, it outperforms commercial-grade alternatives such as Hunyuan3D Yang et al. (2024b) in our specific laboratory setting. This pipeline produces a high-quality textured mesh and precise camera poses in approximately ***15*** *seconds* on a single NVIDIA A100 GPU, facilitating the rapid construction of robot pointmap sequences in real-world deployment.

Throughout both training and inference, we normalize each robot pointmap to a $[0, 1]$ range based on the extrema (minimum and maximum values) across the entire pointmap sequence prior to VAE encoding. This sequence-level normalization facilitates affine-invariant 4D control, encouraging the framework to prioritize relative spatial geometry and inter-object relationships over absolute coordinate values. By mapping diverse robot trajectories into a canonical latent space, this approach enhances the model's ability to generalize across different scales and workspace configurations.

Dependency on robot assets and kinematic information. Our pipeline requires basic kinematic information from the robot's URDF to articulate its geometry according to the input actions. Such information is generally accessible for deployable robotic systems. For standardized robots, factory-provided CAD meshes can be used directly. When these assets are unavailable, our reconstruction pipeline obtains the robot geometry from multi-view images captured through handheld smartphone orbit scans. This protocol supports our real-world deployment without requiring factory CAD assets.

Reconstruction quality and spatial alignment. We assess the quality of the reconstructed robot geometry by comparing it with the factory CAD model. Using ReconViaGen Chang et al. (2025) with handheld multi-view inputs, the reconstruction achieves a CD-L_1 of 0.0352. This geometric comparison complements the pointmap-noise ablation in Table 4, which evaluates the tolerance of the generative model to imperfect control signals. Figure XI visualizes the reconstructed robot and

[1] https://github.com/IDEA-Research/Grounded-SAM-2
[2] https://github.com/facebookresearch/sam2
[3] https://github.com/estheryang11/ReconViaGen

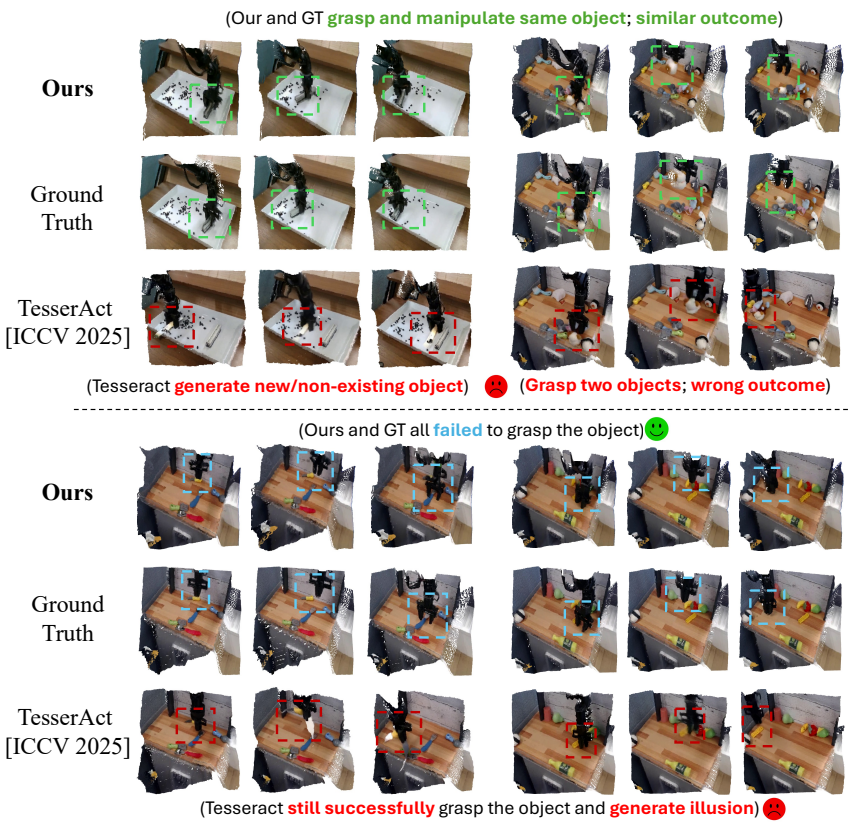

Figure VIII: **4D qualitative comparison between our Kinema4D and TesserAct Zhen et al. (2025)**. Unlike TesserAct that hallucinate outcomes, our Kinema4D precisely reflects Ground-Truth executions, including "near-miss" failure cases. For example, in the two bottom examples, our model *correctly interprets the spatial gap* between the gripper and the objects, *even when their **RGB textures overlap in 2D views***.

the factory CAD reference. Specify the alignment, scale normalization, and point sampling used to compute CD-L1. A direct comparison with the noise-ablation threshold requires a consistent error definition and projection setting.

The projected robot pointmap is pixel-aligned with the RGB grid, while its pixel values store the camera-space (x, y, z) coordinates. This transformation maps the robot trajectory from the reconstruction space into the target camera coordinate system, aligning the robot occupancy with the environmental background. More accurate reconstruction could further improve the quality of these control signals.

Details of 4D Generative Modeling. In the second phase of 4D Generative Modeling, we employ Parameter-Efficient Fine-Tuning (PEFT) via LoRA Hu et al. (2022), initializing weights from 4DNeX[4] Chen et al. (2025). The base model, 4DNeX, was originally fine-tuned on the Wan2.1 Wan et al. (2025) 14B image-to-video architecture. To mitigate the distribution shift between RGB and pointmap latents, we adopt the latent normalization strategy from 4DNeX. Specifically, we compute the empirical statistics of the pointmap domain in the latent space over 5,000 random training samples, yielding a mean $\mu = -0.17$ and a standard deviation $\sigma = 1.36$. These values serve as constant normalization terms for the XYZ latents during both training and inference. The LoRA fine-tuning is configured with a rank of 64 to balance parameter efficiency with representational capacity. Training is conducted with a batch size of 32 using the AdamW optimizer, a peak learning rate of 2×10^{-5}, and a cosine warmup followed by constant decay. The model is trained for 5,000 iterations across 32 NVIDIA A100 GPUs, taking approximately 2 days, at a spatial resolution of

[4] https://github.com/3DTopia/4DNeX



Figure IX: **4D qualitative results of our Kinema4D for policy evaluation in simulation platform**. Our simulated results strongly align with actual executions, accurately synthesizing both successful rollouts and "near-miss" failures. For instance, in the bottom figures, our model correctly interprets the spatial gap between the gripper and the object, even when their RGB textures overlap in 2D views.

480×720 per modality, occupying 67GB of memory per GPU. While the original 4DNeX or WAN uses a text encoder to inject text conditions, we intentionally replace the text embedding with the VAE latents of robot sequences, which is discussed and ablated in section A.3 ("Text conditions"). During training and inference, our robot mask map is spatially and temporally downsampled according to the VAE latent transformation rules, ensuring exact alignment with the latent tensors. It is subsequently expanded to four channels (yielding a shape of $B \times 4 \times H_{latent} \times W_{latent}$), where non-zero values are restricted to robot-occupied positions, and is then channel-wise concatenated with both the noisy and input latents. In addition, we follow ControlNet Zhang et al. (2023) to add a 1×1 zero-initialized convolution layer on the robot latent, to eliminate random noise as gradients in the initial finetuning steps.

Robot-mask refinement. To retain the capacity to refine imperfect robot signals, we employ a soft occupancy mask: 10% of the robot-occupied regions are assigned a value of 0.5, while the remaining occupied regions retain a value of 1 and the background remains 0. This strategy allows the model to refine the robot's visual signal while preserving its spatial guidance, mitigating the impact of reconstruction noise. The mask is concatenated with the input and noisy latents to condition the generation of environmental responses, as evaluated in Table 4.

Underlying insight of joint modeling. By jointly modeling RGB and pointmap latents, the generative process is transformed into a spatiotemporal reasoning task. In addition to synthesizing visual appearance, the model learns to predict the world's 3D occupancy and deformations in response to the robot's movement. The robot pointmap provides geometric guidance, while joint RGB and pointmap prediction encourages consistency between the appearance and spatial structure of the generated world.

A.2.2 Dataset

Selection of the 4D annotation method. A significant challenge lies in lifting raw RGB videos from real-world datasets into the 4D space required for joint generative modeling. We explored several 3D/4D reconstruction frameworks, including MonST3R Zhang et al. (2024), MegaSaM Li et al. (2025a), VGGT/VGGT4D Wang et al. (2025a); Hu et al. (2025), ViPE Huang et al. (2025a), and ST-V2 Xiao et al. (2025). Among these methods, ST-V2 produced the most robust and temporally consistent pointmap sequences in our trials, particularly for robot manipulations involving rapid mo-

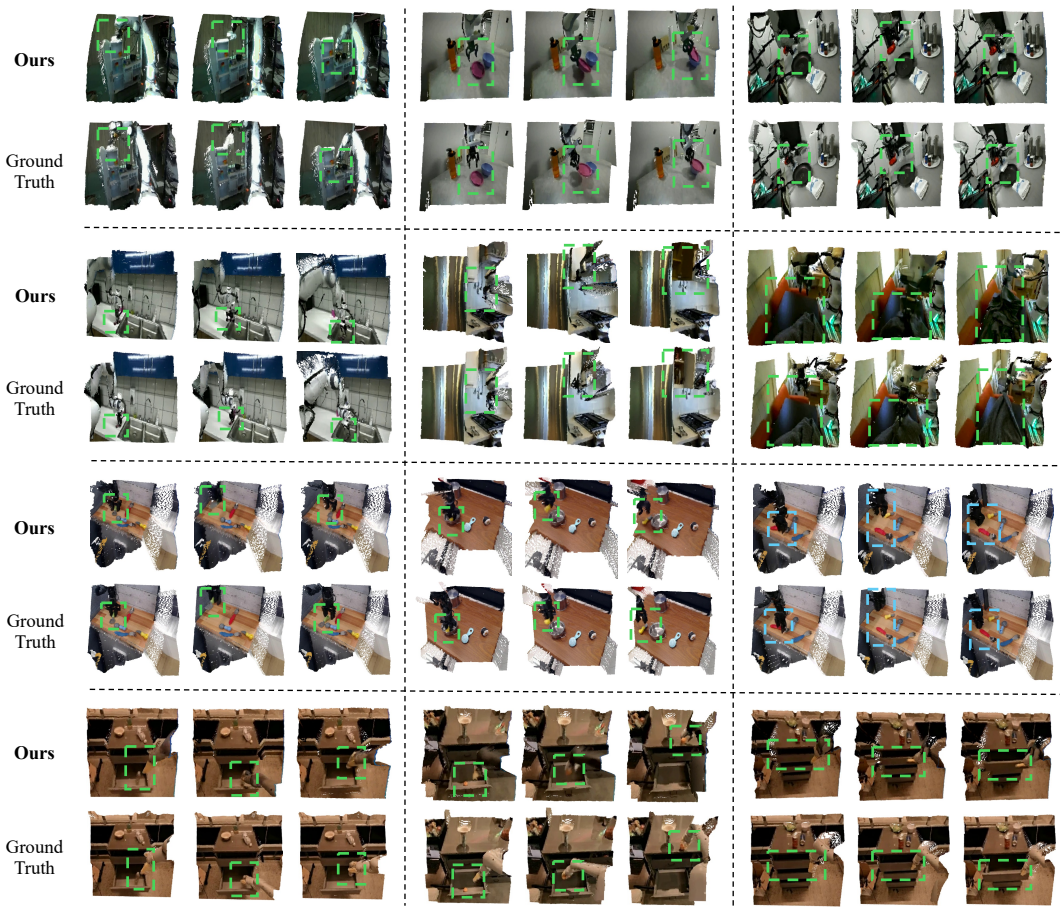

Figure X: **Extensive 4D qualitative results of our Kinema4D**, across diverse domains, environments, robot actions, and the manipulated objects. These results are *non-overlapping* with other figures. More animated results are provided in *demo.mp4*.

tion. We therefore use it to reconstruct pixel-aligned 4D trajectories for real-world demonstrations. For synthetic data from LIBERO Liu et al. (2023a), we instead obtain pointmaps from the native depth and camera parameters.

Acquisition of LIBERO simulated data. Our data generation framework is built upon the MuJoCo Todorov et al. (2012) engine within the LIBERO Liu et al. (2023a) platform. We extend the OpenVLA Kim et al. (2024) dataset pipeline[5] by integrating MuJoCo's depth-buffer rendering to capture per-pixel depth maps alongside RGB sequences. These are subsequently back-projected into world-coordinate pointmaps using intrinsic and extrinsic camera parameters. To maintain a consistent data format, these pointmaps are stored as pseudo-RGB MP4 videos after undergoing per-sequence min-max normalization. To ensure high data variance, we perform procedural randomization of background objects within each scene. While the core dataset consists of successful demonstrations, we systematically synthesize failure cases to enhance the model's discriminative capabilities. Specifically, each successful trajectory is partitioned into three temporal segments, where Gaussian noise of varying intensities ($\sigma \in \{0.5, 0.8, 1.2\}$) is injected into the 6-DoF pose dimensions. This process yields nine distinct failure trajectories per demonstration, as the cumulative error causes the robot to deviate progressively from the target path. Notably, the gripper state remains unperturbed to ensure that the resulting failures stem from reaching or positioning errors rather than unnatural grasping artifacts.

[5] https://github.com/openvla/openvla

Extracting robot pointmap from 4D data annotations. Using either simulation platform for producing synthetic data or ST-v2 Xiao et al. (2025) for processing real-world data, we can easily gain the whole pointmap (including both the robot and environment) sequence at a main camera view. To further gain the robot-only pointmap as the control signal, we need to extract the robot area from the given pointmap. To realize this, we segment the robot's presence within RGB frames by utilizing SAM2 Ravi et al. (2024) with a robust semantic prompt (*e.g.*, "the robot arm"), and apply the segmented mask on both pointmap and RGB frames to gain robot-only sequences. By doing so, we maintain the automated property of our data processing pipeline, so that our framework can leverage scalable video data from diverse, in-the-wild sources without manual labeling overhead. During both the model training and inference, we use the processed robot pointmap sequence as the control signal.

Language annotations. As we find that many text instruction labels are incomplete in our original data resource Khazatsky et al. (2024); Walke et al. (2023); Brohan et al. (2022b). We utilize Qwen3-VL-plus[6] Bai et al. (2025) to generate the language descriptions for each episode. Two types of texts are generated, including the complete text (considering the whole robot-world dynamic interactions across the whole video) and the environment-centric descriptions (only considering the static environment at the initial frame).

Validation split. To ensure a rigorous and fair comparison, we constructed a unified validation set by sampling and aggregating 3200 episodes from the official validation episodes of all baselines. This evaluation set is sampled from our core data sources—2,000 from DROID, 1,000 from Bridge, 100 from RT-1, and 100 from LIBERO (50 successful and 50 failure modes)—preserving the proportional distribution of the training set. This composition ensures comprehensive coverage of complex robot-world interactions, ranging from articulated/deformable object manipulation and pick-and-place of objects. Crucially, all samples within this combined validation set were strictly excluded from our training set to prevent data leakage. This setup allows us to evaluate the generalization capability of our model across multiple data distributions simultaneously, whereas baselines are often specialized to a single domain.

The underlying logic behind 4D pseudo annotation. To facilitate efficient large-scale data synthesis, we employ ST-v2 Xiao et al. (2025) to generate pseudo-annotations directly from diverse robotic demonstration datasets. Our core design philosophy *prioritizes* ***scalability*** as the primary driver for building a *generalizable* robotic simulator. While we acknowledge that 4D reconstruction methods like ST-v2 may not achieve absolute, sub-millimeter geometric ground truth, they provide sufficiently high-fidelity relative spatial geometry. This level of accuracy is adequate for our generative framework to internalize the underlying spatial constraints and physical affordances of the environment (as verified in our qualitative results). By prioritizing the breadth of data over the exhaustive precision of individual samples, we enable the model to learn robust, generalizable motion priors that transcend the noise inherent in pseudo-labels.

Quantitative assessment of annotation quality. To evaluate the accuracy of our pseudo-annotations, we manually annotate 100 episodes to obtain ground-truth references. Against these references, Robo4D-200k achieves an AbsRel of 0.076 and a $\delta_{1.25}$ of 0.924. Moreover, 93% of the evaluated data satisfies $\mathrm{AbsRel} < 0.10$. These results support the geometric quality of the annotations while acknowledging that pseudo-annotations are not error-free. Our emphasis on scalable 4D annotation is also consistent with the use of external reconstruction methods in TesserAct Zhen et al. (2025).

User study on annotation reliability. We further conduct a user study involving 39 participants to assess geometric fidelity and temporal consistency. Robo4D-200k is preferred in 96.4% of the evaluated cases. This perceptual assessment complements the quantitative annotation-quality evaluation and provides additional evidence of the spatial and temporal coherence of our 4D annotations.

Frame number . Each raw video is sliced into 49 frames per sample and maintains a unified frequency of motion dynamics for stable training.

[6] https://github.com/QwenLM/Qwen3-VL

A.2.3 REAL-WORLD DEPLOYMENT

Reconstruction and projection. Our real-world evaluation platform utilizes a YAM Arm[7] (6-DoF collaborative manipulator). For comprehensive rollout recording and visual input, a primary RGB camera cam is positioned at an oblique top-down angle relative to the workspace $\mathcal{W}$, ensuring a clear view of the manipulation area. After capturing orbit videos of the workspace, the primary viewpoint (fixed camera) image is integrated with the orbit-scan frames and processed via ReconViaGen Chang et al. (2025) to produce 3D robot mesh and the camera pose of the primary viewpoint. To ensure the robot's digital twin accurately reflects the physical state, we perform a rigid-body transformation $\mathbf{T}_{recon}^{base}$ between the physical robot's mounting point at $\mathcal{F}_{base}$ and the reconstructed global mesh at $\mathcal{F}_{recon}$ via manual calibration. By retrieving the real-time joint encoders from the YAM Arm's URDF, we solve the Forward Kinematics (FK) to update the pose of each link mesh $\mathbf{x}_{base}$ at $\mathcal{F}_{base}$ and transform it into $\mathcal{F}_{recon}$: $\mathbf{x}_{recon} = \mathbf{T}_{recon}^{base} \cdot \mathbf{x}_{base}$.

Experimental setup. We designed three distinct pick-and-place scenarios, denoted as $\mathcal{S} \in \{\text{Easy, Medium, Hard}\}$, to evaluate the zero-shot generalization of our Kinema4D. Each scenario consists of $N = 50$ trials to ensure statistical significance. The task complexity is categorized by the density of distractor objects $\mathcal{D}$ and the spatial constraints $\mathcal{C}$ within the workspace $\mathcal{W}$. Critically, both the physical embodiment and the unseen laboratory environment are entirely out-of-distribution (OOD) relative to our training data. This rigorous setup evaluates the framework's ability to transcend the reality gap without any real-world fine-tuning or domain-specific adaptation.

Table V: Specifications of Real-World Evaluation Scenarios.

Scenario	Difficulty	Distractors ($\mathcal{D}$)	Constraints / Challenges
1. Dynamic	Hard	$n > 5$, randomized	Partial occlusion & narrow passages
2. Cluttered	Medium	$n \in [3, 5], r \leq 5$cm	Spatial proximity & collision avoidance
3. Canonical	Easy	$\emptyset$	Basic OOD texture grasping

Baseline comparison under zero-shot deployment. We additionally evaluate Ctrl-World Guo et al. (2026) and IRASim Zhu et al. (2025) under the same real-world OOD setting. We measure the absolute difference between the success rate predicted by each simulator and that observed in actual robot execution. Across the three scenarios, Ctrl-World yields gaps of $[0.34, 0.36, 0.18]$, while IRASim yields $[0.32, 0.40, 0.16]$. Kinema4D achieves smaller gaps of $[0.26, 0.30, 0.12]$ in all three scenarios, indicating closer agreement with real execution under this zero-shot setting.

Adaptation with real-world trajectories. Separately from the zero-shot evaluation, we fine-tune Kinema4D on 100 real-world trajectories to investigate the effect of in-domain adaptation. The resulting success-rate gaps decrease to $[0.16, 0.14, 0.08]$ across the three scenarios, compared with $[0.26, 0.30, 0.12]$ before adaptation. These results show that additional real-world data can reduce the discrepancy between generated and actual outcomes. This experiment adapts the generative simulator and is reported separately from the zero-shot results.

RGB and pointmap controls under real-world distribution shift. The control-modality ablation in Table 4 evaluates in-domain performance, where adding robot RGB signals can provide useful appearance information. To examine whether this benefit transfers to real deployment, we additionally compare RGB-only, RGB-plus-pointmap, and pointmap-only robot controls under the real-world OOD setting. These variants concern the robot control representation; the initial world RGB image remains part of the conditioning input.

RGB-plus-pointmap controls yield success-rate gaps of $[0.30, 0.30, 0.14]$, while RGB-only controls yield $[0.30, 0.32, 0.18]$. Our pointmap-only controls achieve gaps of $[0.26, 0.30, 0.12]$. We hypothesize that the appearance gap between training RGB signals and robot renderings used during real deployment introduces an additional transfer challenge. Pointmap controls reduce this dependence on appearance and provide explicit spatial guidance, resulting in more robust performance across the evaluated OOD scenarios.

[7] https://i2rt.com/collections/yam-arm

A.2.4 EVALUATION IN SIMPLER-ENV

Experimental setup. We further evaluate Kinema4D in Simpler-Env using both the RT and Bridge environments. For each environment, we select four tasks and conduct 10 trials per task. Kinema4D generates the environmental outcomes from the robot action sequences, while the remaining evaluation configurations are kept unchanged.

Results. Comparing success rates from the generated outcomes with those from the reference environment executions, we obtain average Pearson correlation coefficients of 0.91 and 0.94 for the RT and Bridge environments, respectively. These results provide additional evidence that Kinema4D can capture variations in policy success rates in another simulation benchmark. Figure XII presents qualitative examples from the Simpler-Env evaluation. Specify the evaluated tasks and policies, and the observations and averaging procedure used to compute the reported Pearson correlations.

A.2.5 CLOSED-LOOP CONTROL IN A PUSH-T-STYLE ENVIRONMENT

Task setup and data preparation. We investigate whether Kinema4D can support closed-loop control through model-based planning in a Push-T-style manipulation environment, following the trajectory-ranking approach of IRASim Zhu et al. (2025). The task requires the robot to push an object toward a target configuration. To accommodate our visual conditioning pipeline, we introduce a fixed third-person camera that observes the robot and its workspace. During offline data collection, action sequences are executed in the environment and recorded as video sequences. We segment the robot region from these recordings to extract robot-specific visual signals for data preparation. This recording and segmentation procedure is performed offline; candidate evaluation during planning relies on model predictions.

Value function. Following the Push-T planning protocol of IRASim Zhu et al. (2025), we use a ResNet50 value model R to estimate the intersection over union (IoU) between the manipulated object and its target region from an observation. The value model is trained on rollouts collected by the proposal policy, including both successful and failed executions, using the corresponding ground-truth IoU as supervision. During planning, R evaluates the final RGB frame predicted for each candidate trajectory, providing a task-specific score for action selection.

Prediction and trajectory selection. At each planning step t, we obtain the current third-person observation I_t and sample $K = 50$ candidate action trajectories from the proposal policy. For each candidate, our kinematic control pipeline constructs the corresponding robot pointmap sequence from the proposed actions and the robot model, aligned with the third-person camera. Conditioned on I_t and this control sequence, Kinema4D predicts the resulting RGB and pointmap sequences. We rank the candidates by their predicted terminal IoU and select the highest-scoring trajectory:

$$k^* = \arg \max_{k \in \{1,\dots,K\}} R\Big(\hat{I}^{(k)}_{\text{final}}\Big), \qquad K = 50, \tag{2}$$

where $\hat{I}^{(k)}_{\text{final}}$ denotes the final RGB frame predicted for candidate k. The candidate robot controls are constructed from the proposed actions before execution, so trajectory selection does not require observing their actual future outcomes.

Closed-loop execution. We execute a short action segment from the selected trajectory and acquire a new observation from the environment. The planner then generates and evaluates a new set of candidates conditioned on the updated observation. Repeating this prediction–selection–execution cycle allows subsequent decisions to incorporate the actual object motion and contact outcomes, including deviations from previous predictions. This feedback mechanism closes the loop between model-based planning and environment interaction. The process continues until task termination.

Evaluation protocol. Following IRASim Zhu et al. (2025), the evaluation protocol uses 100 trials and reports the mean object–target IoU computed from actual environment executions. The predicted IoU from R is used for candidate selection, while evaluation scores are obtained independently from the environment. To assess the contribution of planning, the comparison includes the proposal policy without trajectory ranking and an IRASim-based planner. For a controlled comparison in the adapted third-person environment, both world-model-based planners use the same

Table VI: Quantitative results of more ablation studies.

Metrics	Ours	add env. pm	env. text	null text	depth
PSNR↑	**22.50**	21.93	20.97	22.31	21.04
FID↓	25.9	26.2	27.0	**25.6**	26.3
CD-L_1↓	**0.0479**	0.0420	0.0592	0.0501	0.0583

proposal policy, candidate budget ($K = 50$), value model, execution horizon, and evaluation initializations, with matched task-specific training data budgets.

Results. Kinema4D achieves a mean IoU of `0.983`, compared with `0.637` for the proposal policy without planning and `0.961` for the IRASim-based planner under the same evaluation protocol.

A.3 More Ablation Studies and Analysis

Why not also use the initial environmental pointmap as additional input? During the Kinematics Control phase, our primary objective is the extraction of precise 4D robotic trajectories. Consequently, we prioritize reconstructing the robot's geometry from masked multi-view images rather than the entire scene. This produces robot-centric pointmap sequences, which are fused with the initial world RGB frame as the conditioning input for generative modeling. While it is technically feasible to reconstruct the full static environment and provide an initial-frame world pointmap for enhanced 4D spatial grounding, our empirical results suggest this is unnecessary. As listed in table VI ("add env. pm"), incorporating the complete initial-frame pointmap does not yield performance gains. We attribute this to the pre-trained weights from 4DNeX Chen et al. (2025), which endow our framework with sufficient spatial inference capabilities derived from RGB signals alone.

To further validate this design, we extended this ablation to our real-world setup. By reconstructing the entire physical environment, we provided the model with an initial real-world environmental pointmap. However, this led to a significant performance degradation, with a larger discrepancy between the policy simulation and the ground truth. Specifically, it yields [0.70, 0.82, 0.96] success rates for three different real-world test setups, respectively. We hypothesize that full-scene reconstruction introduces aleatoric noise and sensor artifacts from the physical world, which compromises the robustness of the latent control signal. This ablation confirms that our original design—focusing on precise robot kinematics while relying on RGB for environmental context—optimizes the balance between spatial accuracy and system robustness.

Text conditions. Since our model is designed to simulate the deterministic physical outcomes of actual robotic actions that could be either successful or failed for a task execution, relying solely on high-level task instruction texts is infeasible. To investigate the impact of language conditioning, we evaluated three variants: Environment-centric text, which describes only the static scene geometry; Null conditioning ("null text"); and our proposed robot VAE latents.

As shown in table VI, we observed that incorporating environment-centric text ("env. text") slightly degrades performance, likely due to the modality gap between abstract linguistic descriptions and the requirement for precise 4D transitions. In contrast, our framework maintains high stability when using either null text ("null text") or Robot VAE latents ("Ours"). Ultimately, we opt for robot VAE latents as the primary conditioning signal, as they provide a dense, kinematically-informed representation that significantly enhances the fine-grained control over robot pointmap generation, surpassing the spatial ambiguity inherent in natural language.

Depth map as the spatial representation. While our framework utilizes pointmaps as the spatial representation, we evaluate the versatility of our model by training and testing on depth-map sequences. Leveraging our ST-v2 Xiao et al. (2025) annotation pipeline, high-fidelity depth information is derived directly from our existing data without requiring additional effort. However, empirical results reported in table VI ("depth") indicate that relying solely on depth maps leads to a performance degradation. We hypothesize that while depth captures distances from the camera, it lacks the explicit tri-coordinate geometric constraints (X, Y, Z) inherent in pointmaps. This weakens the model's ability to maintain precise metric control, particularly for off-axis manipulation tasks.

Table VII: Comparison of the efficiency and cross-domain transfer. While the baseline methods offer better efficiency, their generalization abilities are extremely worse than our method, considering both the actual quality and the self-temporal consistency.

Method	Time (min)	Memory (GB)	**PSNR↑**	**PSNR (temp)↑**	**FID↓**	**FID (temp)↓**	**CD-L_1↓**	**CD-L_1 (temp)↓**
Ours	15	56	**19.23**	**36.95**	**29.3**	**15.4**	**0.0621**	**0.0068**
Ctrl-World Guo et al. (2026)	**2.5**	**15**	12.14	19.32	38.5	49.5	-	-
TesserAct Zhen et al. (2025)	10	31	10.80	25.87	41.8	28.7	0.0928	0.0125

Discussion on efficiency and memory usage. We provide a comparison of computational efficiency and peak memory usage against the leading 2D-based baseline, Ctrl-World Guo et al. (2026), and the 4D-based framework, TesserAct Zhen et al. (2025). All evaluations were conducted on a single NVIDIA A100 GPU to ensure parity. We also report the results of cross-domain evaluation, which is trained on DROID Khazatsky et al. (2024) but tested on Bridge Walke et al. (2023). All the baseline implementations use their official codes. As summarized in table VII, although both TesserAct and Ctrl-World show better efficiency, they are limited in cross-domain ability.

Especially, Ctrl-World uses an auto-regressive architecture that offers superior throughput and a lower memory footprint. However, by relying on token-based action embeddings, it exhibits severe overfitting to the training dataset and lacks cross-embodiment capabilities due to the large gap between different embodiments' raw action spaces. Consequently, when deployed in unseen domains, it produces extremely inconsistent temporal frames (denoted by "(temp)" in table VII) and inferior visual quality due to the accumulated errors inherent in auto-regressive generation. As for TesserAct, it uses text instructions to represent action information, which lacks precise control and makes it fundamentally ineffective for the task-agnostic simulation.

In contrast, our framework prioritizes generative fidelity and structural consistency as a generalizable and robust foundation for 4D world simulation. We argue that for a foundation model, the priority must be the ***reliability*** *and* ***generalizability*** *of the physical transitions*. We leave efficiency optimizations, such as knowledge distillation and model quantization, for future work.

Success overestimation and failure modes. Our generated rollouts tend to overestimate policy success in real-world evaluation. One possible contributing factor is that approximately 73% of the training data consists of successful demonstrations, which may bias generation toward successful outcomes even when the executed actions do not establish the necessary physical contact. The spatial guidance provided by robot pointmaps helps mitigate this behavior: as illustrated in Figure 6, Kinema4D can synthesize near-miss failures when the gripper and object overlap in the image plane but remain separated in 3D. Nevertheless, reliably reproducing the full range of physical failures remains challenging.

Limitations of physical modeling. The physical plausibility of our framework refers to kinematics-grounded robot actions and spatially and temporally coherent environmental responses. Without explicit knowledge of object-specific physical properties, fine-grained dynamics remain difficult to reproduce precisely. Figure XIII illustrates representative physical failure cases.

Following the direction explored by WonderPlay Li et al. (2025b), future work could integrate explicit physics solvers, such as the Material Point Method (MPM), to provide additional physical constraints. Our spatiotemporal generative priors could then refine the resulting sequences, combining explicit physical modeling with flexible visual generation.