GEM-4D: Geometry-Enhanced Video World Models for Robot Manipulation

Kaichen Zhou[1,2,*] Yuzhen Chen[1,*] Fangneng Zhan[2]
Hang Hua[4] Grace Chen[1] Xinhai Chang[2] Ao Qu[2]
Yilun Du[1] Zhuang Liu[3] Paul Pu Liang[2,†] Mengyu Wang[1,†]

[1]Harvard University
[2]Media Lab and EECS, MIT
[3]Princeton University
[4]MIT-IBM Watson AI Lab

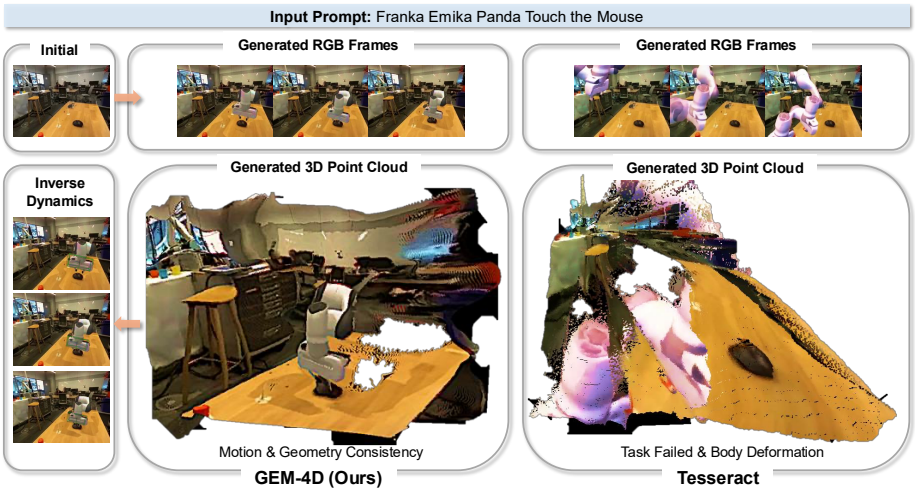

Fig. 1: Given an instruction and an initial observation, our model predicts future frames while preserving geometric consistency. Compared with the Tesseract [69] baseline (right), our approach produces more structurally coherent scene evolution.

Abstract. Video world models can generate realistic futures from a single instruction, but they often fail to track the same physical points consistently across time. As a result, the generated videos appear plausible, yet lack the physical grounding required for reliable action execution, such as robot manipulation. We present GEM-4D, a geometry-grounded video world model that resolves this limitation by injecting dense 4D correspondence supervision distilled from a pretrained geometry foundation model into the video generative backbone during training. This supervision enables the video world model to jointly capture appearance and geometric structure while retaining a single-stream architecture with no additional inference cost. We further introduce an inverse dynamics module that converts correspondence-consistent video rollouts into

[*]Equal contribution as first authors. [†]Joint supervision.

executable robot trajectories, enabling direct deployment in both real-world and simulated manipulation. GEM-4D achieves state-of-the-art performance on both video prediction and geometric consistency across both simulation and realistic scenarios and improves real-world manipulation success from 61% to 81%. Additional results are available at the `https://gem-4d.github.io/`.

1 Introduction

General-purpose robotic manipulation requires policies that can generalize across diverse scenes, objects, tasks, and even embodiments. Vision-language-action policies [4, 28] have shown strong progress by directly mapping observations and language instructions to actions, but such generalization often depends on large-scale robot demonstration data and embodiment-specific training or fine-tuning. Video world models [1, 14, 15, 20, 60, 69, 74, 75] offer a complementary path. By predicting task-conditioned future observations from the current scene and instruction, they can provide a more general visual representation of how an interaction may unfold, potentially reducing reliance on task- and embodiment-specific action supervision.

However, generic video generation models [47, 64] primarily emphasize visual realism, whereas using a video world model for robot planning requires more than photorealistic prediction. To derive executable actions from predicted futures, a robot must recover precise object and end-effector motions [2]. This requires the generated video to preserve reliable inter-frame correspondences, such that pixels corresponding to the same 3D surface point follow physically consistent trajectories across time [22].

Current video diffusion models [5, 39], trained largely with pixel- or latent-space reconstruction objectives, provide no explicit guarantee of such consistency [36, 58]. They can produce photorealistic videos in which rigid objects deform non-rigidly, contacts drift, and depth varies inconsistently across frames [68]—errors that may be visually subtle but can fundamentally break action extraction. This limitation is structural: reliable inter-frame correspondence depends on camera motion, scene depth, and object motion, whereas standard pixel- or latent-space generation losses do not explicitly constrain these factors. Existing remedies address this limitation only partially. Explicit 4D-supervised methods [69, 75] add predictions such as RGB, depth, and surface normals, thereby constraining certain geometric quantities. However, they require large-scale annotations and still do not provide a unified correspondence signal that jointly accounts for camera motion, scene depth, and object motion.

Our key observation is that the geometric factors governing inter-frame correspondence are already encoded by modern 4D geometry foundation models. Models such as PAGE-4D [72], Depth Anything V3 [33], VGGT [49] and VGGT-omega [50] estimate dense geometry and camera motion from video, so their intermediate representations capture depth, viewpoint change, and object motion in a unified form. Rather than supervising correspondence explicitly, we distill

these geometry-aware representations into the video backbone [65]. This feature-level supervision encourages the generative model to encode the camera, depth, and motion structure needed for consistent inter-frame correspondence, yielding predicted futures that are more reliable for robot action extraction.

We present GEM-4D, a world model that operationalizes this principle via feature-level distillation. During training, a video diffusion transformer is paired with a geometry branch that predicts representations from a pretrained 4D geometry foundation model, conditioned on intermediate video features. This forces the backbone to internalize correspondence-consistent geometric structure without modifying its output space or adding parameters. The coupling is asymmetric and efficient: the geometry branch reads from video features but never writes back, and is discarded entirely at inference, yielding a single-stream generator with zero additional cost. We further introduce an inverse dynamics module that converts correspondence-consistent rollouts into executable 6-DoF end-effector trajectories using off-the-shelf vision foundation models, closing the loop from language instruction to real-world manipulation without task-specific training.

GEM-4D achieves state-of-the-art performance on both video prediction and geometric consistency across both simulation and realistic scenarios and improves real-world manipulation success from 61% to 81%. Our contributions are summarized as follows:

1. **Principle.** We formalize the connection between geometry foundation model representations and inter-frame correspondences, showing that geometry supervision acts as a representation-level regularizer that encourages the video backbone to encode correspondence-consistent structure.
2. **Architecture.** We introduce GEM-4D, a dual flow-matching framework that distills 4D geometry features into a video backbone via asymmetric conditioning, thereby achieving correspondence-aware generation at zero additional inference cost.
3. **System.** We close the loop from world model to robot control: an inverse dynamics module extracts executable trajectories from generated rollouts, achieving 81% success on real-world Droid tasks (+20 points over the strongest baseline) and 63–82% success on RLBench.

2 Related Work

2.1 Video World Models for Embodied Control

Learning world models that predict future observations has become a central paradigm for embodied decision-making [1, 3, 8, 9, 12, 14, 15, 18, 19, 21, 23, 32, 40, 41, 59, 60, 63, 74]. A common approach treats video generation as a planning substrate: UniPi [15] generates future frames using video diffusion and recovers actions via inverse dynamics, while Uni-Sim [63] and Genie [7] leverage world models to synthesize interaction data for downstream policy learning. These methods demonstrate strong generalization across tasks and environments, but operate purely in pixel space and provide no mechanism to enforce geometric

consistency in predicted futures. Recent work addresses this limitation by incorporating explicit geometric structure. TesserACT [69] jointly generates RGB, depth, and surface normals to enable spatiotemporal reconstruction and action prediction. 3DFlowAction [70] represents actions as 3D flow fields, grounding planning in scene geometry. WristWorld [41] synthesizes wrist-view observations through reconstruction, while RoboTransfer [35] enforces multi-view geometric constraints via cross-view feature interactions. Related approaches impose geometric consistency through pointmap alignment or multi-view supervision [37], and multi-modal world models such as iMoWM [66] and MinD [11] jointly model video and action generation. Despite their effectiveness, these approaches share a common limitation: they *modify the output space* of the video model to explicitly predict geometric quantities (e.g., depth, normals, or flow), which requires large-scale geometric annotations and constrains the expressive capacity of pretrained video backbones. In contrast, GEM-4D does not alter the output space. Instead, it distills geometric structure into the model's *internal representations* during training, enabling correspondence-aware generation while retaining a single-stream architecture with zero additional inference cost.

2.2 Feed-Forward 3D and 4D Geometry Models

Feed-forward geometry models infer dense 3D structure directly from visual observations, avoiding iterative optimization. DUSt3R [52] introduced this paradigm by predicting pixel-aligned pointmaps from image pairs, enabling unconstrained stereo reconstruction. MASt3R [31] improves geometric fidelity through local feature matching, while Fast3R [62] scales to multi-view reconstruction via global fusion. Spann3R [48] introduces a spatial memory mechanism for incremental reconstruction, and and π^3 [53] further removes reference-view bias through permutation-equivariant visual geometry learning. Extending these methods to dynamic scenes introduces additional challenges, as both camera and object motion must be disentangled. MonST3R [67] estimates temporally consistent pointmaps and recovers global geometry and camera poses via optimization, while DAS3R [61] accelerates this process. CUT3R [51] enables feed-forward dynamic reconstruction through large-scale training, and Easi3R [10] explores training-free generalization. PAGE-4D [72] further improves robustness by disentangling static and dynamic components via motion-aware masking. These models are typically used as standalone perception modules for geometry estimation. In contrast, GEM-4D repurposes them as *correspondence teachers*: their learned representations encode depth, camera motion, and scene dynamics, providing a supervision signal that enforces geometrically consistent inter-frame correspondences within a video world model. To our knowledge, this is the first work to use 4D geometry foundation models as training-time regularizers for video generation, rather than as direct predictors of geometric outputs.

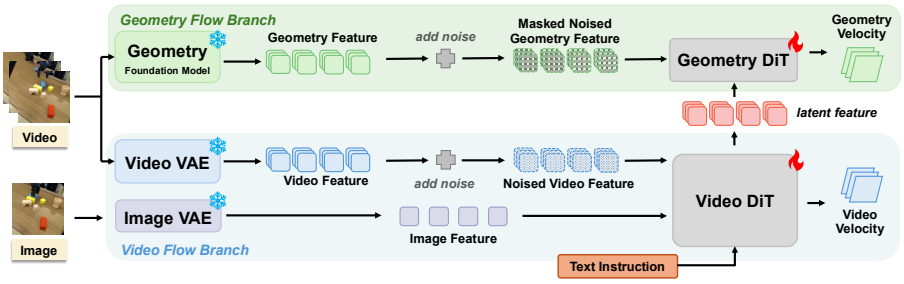

Fig. 2: GEM-4D training. During training, a video DiT predicts the velocity of the noised video latent, while its intermediate features guide a geometry DiT to predict geometry velocity. This coupled training enforces geometry-consistent generation. During inference, only the video branch is used for efficient generation.

3 GEM-4D

3.1 Problem Formulation

Given an initial observation $\mathbf{I}_0$ and a language instruction c, we learn a world model $M : (\mathbf{I}_0, c) \to \{\mathbf{I}_t\}_{t=1}^{N}$ that predicts future frames, and design a policy extraction module $P : \{\mathbf{I}_t\}_{t=0}^{N} \to \{\mathbf{a}_t\}_{t=0}^{N-1}$ that extracts executable actions from the predicted rollout. For P to succeed, the generated rollout must preserve *inter-frame correspondences*: pixels depicting the same 3D surface point must evolve consistently across time. When this property is violated—even if the video appears photorealistic—action extraction becomes unreliable because the underlying 3D structure no longer reflects physically realizable motion [2, 56]. We enforce correspondence consistency during training via **Geometry-Enhanced Velocity Alignment** (Sec. 3.2) shown in Fig. 2, and convert correspondence-consistent rollouts into actions via an **Inverse Dynamic System** (Sec. 3.3).

3.2 Geometry-Enhanced Velocity Alignment

What Governs Inter-Frame Correspondence We begin by formalizing what determines whether two pixels in adjacent frames correspond to the same physical point. Let $\mathbf{X}_t \in \mathbb{R}^3$ be a scene point observed at pixel $\mathbf{p}_t$ in frame t. Under relative camera motion $(\mathbf{R}_{t\to t+1}, \mathbf{T}_{t\to t+1})$ and scene flow $\Delta\mathbf{X}_t$, its projection $\mathbf{p}_{t+1}$ in frame $t{+}1$ is:

$$\mathbf{p}_{t+1} \sim \mathbf{K}\left[\mathbf{R}_{t\to t+1}\,\mathbf{D}(\mathbf{p}_t)\,\mathbf{K}^{-1}\mathbf{p}_t + \mathbf{T}_{t\to t+1} + \Delta\mathbf{X}_t\right]. \tag{1}$$

Pixel in next frame · Intrinsic matrix · Scene flow / object motion · Camera translation · Camera rotation · Depth at $\mathbf{p}_t$ · Pixel in current frame · Inverse intrinsic matrix

This leads to two immediate insights. First, pixel-level reconstruction losses cannot enforce correspondences: the mapping from scene geometry to pixel values is

many-to-one, so different configurations of $(\mathbf{D}, \mathbf{R}, \mathbf{T}, \Delta\mathbf{X})$ can produce visually indistinguishable frames. A pixel loss can reach zero while the underlying correspondences are entirely wrong. Second, a model whose internal representations correctly encode $(\mathbf{D}, \mathbf{R}, \mathbf{T}, \Delta\mathbf{X})$ *necessarily* produces correct correspondences, since Eq. 1 leaves no remaining degree of freedom.

Geometry foundation models as correspondence encoders. Models such as PAGE-4D [72], Depth Anything V3 [33],VGGT [49],and DUSt3R-family models [51, 52, 67] employ cross-frame transformers supervised on two tasks: dense depth estimation and camera pose regression. These are precisely the dominant factors in Eq. 1: given per-frame depth $\mathbf{D}$ and relative camera pose $(\mathbf{R}, \mathbf{T})$, the correspondence of any static scene point is fully determined. For dynamic points, the model must additionally capture how depth changes across frames—which implicitly encodes object motion $\Delta\mathbf{X}$, since a point whose depth evolves inconsistently with the estimated camera motion must be moving independently. The learned feature representations of these models therefore encode the complete correspondence structure of Eq. 1, despite being supervised only on depth and pose. Supervising the video backbone to predict these representations implicitly encourages correspondence consistency, without requiring an explicit correspondence loss. GEM-4D shares REPA's [65] philosophy of leveraging foundation-model supervision to improve generative representations. Unlike REPA, which explicitly aligns the diffusion representation with foundation-model features, GEM-4D uses geometry foundation models as auxiliary supervision to encourage a shared video representation that jointly supports RGB denoising and 3D geometric reasoning.

Flow Matching for Latent Video Generation We adopt flow matching [34] for latent video generation. Let $\mathbf{z}_0$ be the VAE-encoded video latent [29] and $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be noise. A velocity field $\mathbf{v}_\theta^{\text{vid}}$ transports $\mathbf{z}_1$ to $\mathbf{z}_0$ via the ODE $d\mathbf{z}_t/dt = \mathbf{v}_\theta^{\text{vid}}(\mathbf{z}_t, t, c)$, trained by regressing against an analytically derived target velocity $\mathbf{v}^*$:

$$\mathcal{L}_{\text{FM}}^{\text{vid}} = \mathbb{E}_{\mathbf{z}_0, \mathbf{z}_1, t}\left[\|\mathbf{v}_\theta^{\text{vid}}(\mathbf{z}_t, t, c) - \mathbf{v}^*(\mathbf{z}_t, t)\|_2^2 \right]. \tag{2}$$

We parameterize the velocity through a Video DiT [39] with backbone E_θ^{vid} and output head U_θ^{vid}:

$$\mathbf{m}_t = E_\theta^{\text{vid}}(\mathbf{z}_t, t, c), \qquad \mathbf{v}_\theta^{\text{vid}} = U_\theta^{\text{vid}}(\mathbf{m}_t), \tag{3}$$

where $\mathbf{m}_t$ denotes the intermediate features extracted at a mid-level layer. Under standard flow matching, $\mathbf{m}_t$ is optimized only for appearance transport. The next section describes how we shape $\mathbf{m}_t$ to encode correspondence structure.

Correspondence Distillation via Geometry Flow Given a video sequence $\{\mathbf{I}_t\}_{t=0}^T$, a frozen geometry model G extracts a dense geometric representation:

$$\mathbf{g}_0 = G\big(\{\mathbf{I}_t\}_{t=0}^T\big) \in \mathbb{R}^{T \times \frac{H}{P} \times \frac{W}{P} \times C}. \tag{4}$$

As established in Sec. 3.2, $\mathbf{g}_0$ encodes the factors $(\mathbf{D}, \mathbf{R}, \mathbf{T}, \Delta\mathbf{X})$ in Eq. 1—it is a dense correspondence representation of the input video. G remains frozen throughout; it serves as a correspondence teacher whose knowledge we distill into the video backbone. To perform this distillation, we introduce a parallel flow-matching process over the geometry representation space. A Geometry DiT $\mathbf{v}_\psi^{\text{geo}}$, *conditioned on the video backbone's intermediate features* $\mathbf{m}_t$, predicts the velocity field of the geometry latent:

$$\mathcal{L}_{\text{FM}}^{\text{geo}} = \mathbb{E}_{\mathbf{g}_0, \mathbf{g}_1, t}\left[\|\mathcal{M}(\mathbf{v}_\psi^{\text{geo}}(\mathbf{g}_t, t, \mathbf{m}_t)) - \mathbf{v}^*(\mathbf{g}_t, t)\|_2^2 \right], \tag{5}$$

where $\mathbf{g}_t$ interpolates between $\mathbf{g}_0$ and noise $\mathbf{g}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\mathcal{M}$ denotes mask mechanism applied to noised geometry feature. The critical design choice is that the Geometry DiT receives $\mathbf{m}_t$ as its *only* scene-level conditioning signal. It has no direct access to pixels, camera parameters, or depth maps—all scene information must arrive through $\mathbf{m}_t$. Minimizing $\mathcal{L}_{\text{FM}}^{\text{geo}}$ therefore requires $\mathbf{m}_t$ to contain sufficient information about the geometric factors $(\mathbf{D}, \mathbf{R}, \mathbf{T}, \Delta\mathbf{X})$ to predict how the geometry representation evolves over time. Since these factors determine correspondences (Eq. 1), the geometry loss could serve as a correspondence loss imposed on the video backbone's internal representations.

Joint Objective and Geometric Grounding The training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{FM}}^{\text{vid}} + \alpha\, \mathcal{L}_{\text{FM}}^{\text{geo}}, \tag{6}$$

where α balances the two terms. Because both losses share the intermediate representation $\mathbf{m}_t$, the gradient with respect to the video backbone parameters θ decomposes as:

$$\underbrace{\nabla_\theta \mathcal{L}}_{\text{Total gradient}} = \underbrace{\nabla_\theta \mathcal{L}_{\text{FM}}^{\text{vid}}}_{\text{Appearance supervision}} + \underbrace{\alpha}_{\text{Balancing weight}} \cdot \underbrace{\frac{\partial \mathcal{L}_{\text{FM}}^{\text{geo}}}{\partial \mathbf{m}_t} \cdot \frac{\partial \mathbf{m}_t}{\partial \theta}}_{\text{Geometry-induced gradient}} \tag{7}$$

Because both losses share the intermediate representation $\mathbf{m}_t$, the gradient decomposes into an appearance term and a geometry-induced term. The latter propagates correspondence structure into the video backbone through $\mathbf{m}_t$. The first term drives $\mathbf{m}_t$ to encode how pixels move over time—appearance transport. The second, which is non-zero by construction since $\mathcal{L}_{\text{FM}}^{\text{geo}}$ depends on $\mathbf{m}_t$ (Eq. 3), drives $\mathbf{m}_t$ to additionally encode *why* they move that way: the depth, camera pose, and scene flow that determine correspondences (Eq. 1). At convergence, $\mathbf{m}_t$ must satisfy both objectives, eliminating representations that explain appearance but violate geometric consistency.

3.3 Adaptive Inverse Dynamic System

Given generated frames $\{\mathbf{I}_t\}_{t=0}^{N}$, we make use of **Adaptive Inverse Dynamic System (AIDS)** to convert it into executable 6-DoF action trajectories

Fig. 3: Adaptive Inverse Dynamic System. Given a generated video as input, this system extracts a robot policy through the four steps illustrated in the figure.

$\{\mathbf{a}_t\}_{t=0}^{N-1}$, as illustrated in Fig. 3. AIDS introduces two mechanisms that together make action extraction robust to these artifacts in generated results without any task-specific training: (i) a *dual-criterion confidence-gated tracker* that separates gradual drift from catastrophic collapse and invokes heavyweight VLM re-grounding only when warranted, and (ii) a *geometry–kinematics pose fallback* that decouples translation (well-observed from depth) from rotation (temporally smoothed in $SE(3)$) whenever learned pose estimation becomes unreliable.

3D scene grounding. We first localize the target object and end-effector (EE) in 3D. Given the instruction, the depth map, and camera intrinsics (estimated by the geometry foundation model), Qwen3.5-VL [57] and SAM 2 [30, 42] generate segmentation masks for the target object and the end effector (EE), which are then used to extract their corresponding point clouds. We then align the EE CAD to the EE point cloud with FoundationPose [55] to recover the initial EE translation and rotation $\left(\mathbf{R}_{\text{ee}}^0, \mathbf{T}_{\text{ee}}^0\right) \in SE(3)$.

Dual-Criterion Confidence-Gated Tracker. With initial EE pose, we propagate dense keypoints sampled from the mask of end effector $\mathcal{M}_{\text{ee}}$ through the rollout using CoTracker3 [25, 26]. Let $\mathcal{V}_{t_0} \subseteq \mathcal{M}_{\text{ee}}$ be the anchor keypoint set at initial frame, and $\mathcal{V}_t \subseteq \mathcal{V}_{t_0}$ the subset still reliably tracked at frame t. We monitor following two metrics:

$$s_t = \frac{|\mathcal{V}_t|}{|\mathcal{V}_{t_0}|} \in [0,1], \qquad \Delta s_t = s_t - s_{t-1}, \tag{8}$$

where s_t denotes the anchor retention ratio ($s_t = 1$ means all anchors remain reliably tracked, $s_t \to 0$ means tracker failure) and Δs_t denotes the frame-to-frame change (negative value signals a loss of correspondence). These two signals

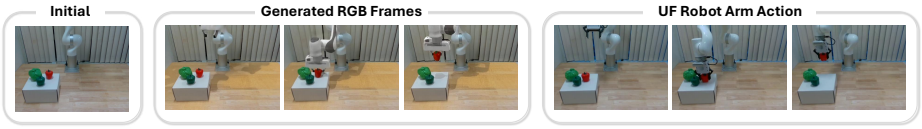

Fig. 4: Generated Frames to Arm Action. Starting from an initial observation, GEM-4D predicts future frames, which are then converted into executable UF arm actions. The generated frames are displayed with higher brightness to distinguish them from the real frames.

separate two qualitatively different failure modes. A *gradual drift* manifests as s_t decaying smoothly below the retention threshold $\tau \in (0, 1)$; an *abrupt collapse*, typically caused by frame-level generative artifacts, manifests as a sharp drop $\Delta s_t < -\delta$, where $\delta > 0$ is the drop threshold. We handle these two failure modes with different interventions:

$$\hat{\mathcal{M}}_{\text{ee}}^{t} = \begin{cases} \text{re-anchor tracker at } t, & \text{if } s_t < \tau, \\ \text{Qwen3.5-VL}(I_t,\, c), & \text{if } \Delta s_t < -\delta, \\ \mathcal{M}_{\text{ee}}^{t}, & \text{otherwise.} \end{cases} \tag{9}$$

The first branch corresponds to a *gradual drift* (resampling fresh keypoints from the most recent reliable mask), and the second to an *abrupt collapse* (re-grounding the EE mask itself via vision-language semantics).

Geometry–kinematics pose fallback. Given the mask of end effector $\mathcal{M}_{\text{ee}}^{t}$, the frame $\mathbf{I}_t$, the depth $\mathbf{D}_t$, and the EE CAD model, FoundationPose [55] predicts the pose of EE $(\mathbf{R}_{\text{ee}}^{t}, \mathbf{T}_{\text{ee}}^{t})$ together with a confidence $\kappa_t \in [0, 1]$ for each frame.

$$(\mathbf{R}_{\text{ee}}^{t}, \mathbf{T}_{\text{ee}}^{t}, \kappa_t) = \text{FoundationPose}\big(\mathbf{I}_t, \mathbf{D}_t, \mathcal{M}_{\text{ee}}^{t}, \text{CAD}\big), \tag{10}$$

When $\kappa_t < \kappa^*$, where $\kappa^* \in [0, 1]$ is a FoundationPose acceptance confidence threshold, we apply a temporal consistency check and reject the FoundationPose estimate if either the translation jump or the rotation jump exceeds its corresponding threshold:

$$\|\mathbf{T}_{\text{ee}}^{t} - \mathbf{T}_{\text{ee}}^{t-1}\|_2 > \epsilon_t, \quad d_{\text{geo}}(\mathbf{R}_{\text{ee}}^{t}, \mathbf{R}_{\text{ee}}^{t-1}) > \epsilon_R. \tag{11}$$

Where $d_{\text{geo}}(\cdot, \cdot)$ denotes the geodesic distance on $SO(3)$. For rejected frames, we recover the EE translation by back-projecting valid depth pixels within the EE mask and taking their 3D centroid, while the missing rotation is estimated by spherical linear interpolation [43] between the nearest accepted poses in time. Finally, we'll have the recovered EE trajectory $\{(\mathbf{R}_{\text{ee}}^{t}, \mathbf{T}_{\text{ee}}^{t})\}_{t=0}^{N}$.

Grasp insertion and action synthesis. From the recovered EE trajectory, we first select the pose closest to the target object as a reference pose, denoted by $(\mathbf{R}_{\text{ref}}, \mathbf{T}_{\text{ref}})$. GraspGen [38] then predicts a set of grasp candidates $\{(\mathbf{R}_{\text{grasp}}^{i}, \mathbf{T}_{\text{grasp}}^{i})\}_{i=0}^{M}$ on the target object point cloud. We rank these candidates

by their weighted pose deviation from reference pose, combining translation and rotation errors, and select the best-scoring candidate as the final grasp pose:

$$\mathbf{T}^{*}_{\text{grasp}} = \arg\min_{\mathbf{T}^{(i)}_{\text{grasp}}} \Big(\lambda_t \, \|\mathbf{t}^{(i)}_{\text{grasp}} - \mathbf{t}_{\text{ref}}\|_2 + \lambda_R \, d_{\text{geo}}\big(\mathbf{R}^{(i)}_{\text{grasp}}, \mathbf{R}_{\text{ref}}\big) \Big), \tag{12}$$

where λ_t and λ_R balance translation and rotation consistency, respectively. We then insert $\mathbf{T}^{*}_{\text{grasp}}$ into the recovered EE trajectory as an intermediate target and smooth the resulting motion via interpolation. Inverse kinematics then converts the smoothed trajectory into an executable action sequence $\{\mathbf{a}_t\}_{t=1}^{N}$ for controlling the robot arm, as illustrated in Fig. 4.

4 Experiments

Overview. Our experiments are structured around two research questions. First, does geometry distillation improve 4D scene prediction in terms of appearance, depth, and correspondence consistency? Second, can the resulting geometry-consistent rollouts improve downstream robot manipulation when converted into executable actions? We therefore organize the evaluation into two parts: Sec. 4.1 studies 4D scene reconstruction quality. Sec. 4.2 evaluates whether the generated rollouts support more reliable robot action planning and execution. Sec. 4.3 examines the effectiveness of the proposed geometry flow branch.

Datasets. We train our model on tasks simulated in ManiSkill3 [44] and RLBench [24], as well as on the Bridge [46] and RT-1 [6] datasets. To evaluate both video generation quality and robotic modeling capability, we conduct experiments in real-world and synthetic domains. In the real-world setting, we use 400 unseen samples from the Droid dataset [27], where depth is estimated using Depth Anything V3 [33] and the point tracking is estimated using CoTracker3 [25]. In the synthetic setting, we evaluate on 780 unseen samples from RLBench [24], where ground-truth depth is directly available and the point tracking is estimated using CoTracker3 [25].

Baselines. Our method is compared against the following: (1) **CogVideoX** [64], a large-scale image-to-video generation model based on a diffusion transformer architecture, pretrained on approximately 35M video clips. We fine-tune it on our training set. (2) **WAN 2.2-14B** [47], a large-scale image-to-video generation model built upon a spatio-temporal variational autoencoder. We fine-tune it on our training set. (3) **TesserAct** [69], a CogVideoX-based 4D embodied world model that further fine-tunes CogVideoX on robotics datasets while jointly modeling depth and surface normals. (4) **Geometry Forcing** [58], a representation-alignment-based method that incorporates geometric prior knowledge into the generative model through feature alignment. We do not compare directly against RoboTransfer [35] (multi-view RGB with depth/normal conditioning), 3DFlowAction [70] (predicts 3D flow rather than RGB), or Liu et al. [37] (stereo RGB-D with cross-view pointmap supervision), as their input modalities and output spaces differ from our single-view RGB-only setting.

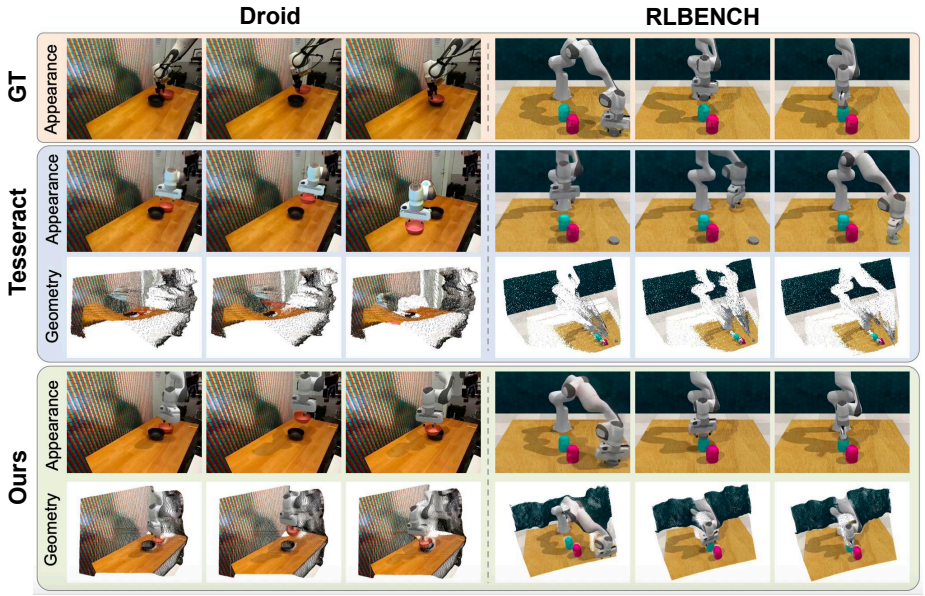

Fig. 5: Qualitative 4D scene generation results on Droid (real) and RLBench (synthetic). Top: ground truth. Middle: TesserAct generates plausible RGB but produces inconsistent depth — note the warping on the manipulator and the noisy depth strip on Droid. Bottom: GEM-4D preserves geometric structure across frames, yielding cleaner depth and tighter object boundaries.

4.1 4D Scene Prediction

Metrics. We evaluate video generation quality using FVD [45], SSIM [54], and PSNR [73]; depth estimation accuracy using AbsRel, δ_1, and δ_2 [16]. To assess 3D reconstruction quality, we compute the L1 Chamfer Distance [17, 71] between the predicted and ground-truth point clouds. The correspondence quality is measured by using δ^{vis}_{avg} (fraction of visible points tracked within 1, 2, 4, 8 and 16 pixels, averaged over thresholds) [13]. Each sample is generated 20 times, and we report the averaged results.

Results and analysis. Table 1 reports quantitative comparisons on both real and synthetic datasets. Qualitative results are shown in Fig 5. Our method consistently achieves the best overall performance across most RGB appearance metrics and depth reconstruction metrics. In particular, GEM-4D achieves the lowest FVD and the highest SSIM/PSNR on the real-world dataset, indicating that it generates videos with higher visual quality and temporal consistency than the baseline models. More importantly, when evaluating geometric quality using DA3 [33] for depth estimation, our method significantly outperforms existing baselines. On the real dataset, GEM-4D achieves the lowest AbsRel and the highest δ_1 and δ_2, demonstrating that the generated frames preserve more accurate scene geometry across time. A similar trend is observed on the synthetic dataset, where our method produces the most accurate depth reconstruction.

Table 1: Quantitative comparison on 4D scene generation. We evaluate RGB reconstruction, depth reconstruction and points correspondence quality across both real and synthetic domains. The best results are highlighted in **bold**, and the second-best results are underlined. **D** denotes dataset; **R** denote realistic dataset and **S** denote simulation dataset. Std. and dev. across 20 generations is around $1\% - 2\%$.

D	Method	RGB			Depth			Points	Tracking
		FVD ↓	SSIM ↑	PSNR ↑	AbsRel ↓	δ_1 ↑	δ_2 ↑	Chamfer ↓	δ^{vis}_{avg} ↑
R	CogVideoX [64]	35.56	75.91	20.18	22.33	68.32	83.17	0.2670	66.22
	Wan 2.2-14B [47]	33.43	76.24	20.70	21.39	71.18	84.35	0.2349	68.18
	TesserAct [69]	33.28	75.66	20.08	22.07	66.80	82.60	0.2630	67.14
	Geometry-Forcing [58]	33.17	76.12	20.53	21.96	69.74	83.83	0.2443	67.97
	GEM-4D	**31.82**	**82.05**	**21.11**	**20.13**	**78.19**	**88.21**	**0.2001**	**71.23**
S	CogVideoX [64]	40.21	75.51	20.03	15.41	70.99	92.90	0.2913	58.32
	Wan 2.2-14B [47]	49.20	73.01	19.87	17.81	67.07	90.16	0.1762	61.99
	TesserAct [69]	41.97	76.72	19.71	16.02	69.26	93.03	0.1813	61.15
	Geometry-Forcing [58]	34.06	77.92	19.48	15.34	68.96	92.80	0.1488	60.84
	GEM-4D	**27.94**	**80.27**	**23.36**	**14.11**	**74.13**	**95.01**	**0.0702**	**68.18**

Table 2: Quantitative comparison on task success rates. Comparison of task success rates across Droid and RLBench tasks. For the real-world Droid dataset, success rates are measured through a human study. For the simulated RLBench dataset, success rates are measured by executing the generated trajectories in simulation.

	Droid (Real)			RLBench (Simulator)						
	AUTOLab	CLVR	RAIL	Lift Numbered Block	Put Rubbish InBin	Reach Target	Lamp On	Pick Up Cup	Slide Block ToTarget	Solve Puzzle
CogVideoX [64]	49	64	39	-	-	-	-	-	-	-
Tesseract [69]	58	65	59	21	0	2	36	49	18	33
Ours	**75**	**83**	**87**	**78**	**75**	**82**	**67**	**81**	**80**	**63**

These results suggest that the proposed geometry-aware training effectively improves both appearance generation and geometric consistency. By incorporating geometry dynamics during training, the model learns to generate videos that not only look realistic but also maintain coherent 3D structure, which further leads to improved downstream reconstruction quality [22] when processed by DA3.

4.2 Embodied Action Planning

Quantitative Experiment Dataset. We evaluate our model on 7 challenging manipulation tasks from RLBench [24] and unseen realistic dataset from Droid [27]. These tasks are selected for their requirements of precise grasping and complex spatial reasoning. **Metric.** To ensure a fair evaluation, we adopt two complementary evaluation protocols. For realistic dataset, we conduct a human study to assess the quality of generated videos, focusing on task success, object and robot-arm deformation, and instruction following. (Averaged over 15 participants; more generated results are provided at the link in the abstract.)

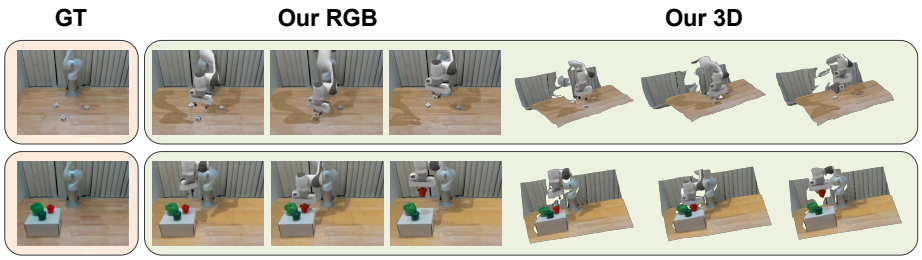

Fig. 6: Real-robot rollouts. From left: ground-truth images, GEM-4D-generated RGB, and the back-projected 3D point cloud. The model produces realistic and geometrically coherent rollouts supporting transfer to UF Arm manipulation.

Table 3: Ablation studies on 4D scene generation. We evaluate RGB reconstruction, depth reconstruction and points correspondence quality on real domain.

Domain	Method	RGB (Appearance)			Depth (Geometry)			P (Geometry)
		FVD ↓	SSIM ↑	PSNR ↑	AbsRel ↓	δ_1 ↑	δ_2 ↑	Chamfer ↓
Real	CogVideoX [64]	35.56	75.91	20.18	22.33	68.32	83.17	0.2670
	Wan 2.2-14B [47]	33.43	76.24	20.70	21.39	71.18	84.35	0.2349
	GEM-4D(Dep)	<u>32.91</u>	<u>78.58</u>	<u>20.75</u>	<u>20.89</u>	<u>74.60</u>	<u>86.67</u>	<u>0.2229</u>
	GEM-4D(VGGT) [49]	33.68	75.89	20.64	21.73	71.03	83.80	0.2370
	GEM-4D	**31.82**	**82.05**	**21.11**	**20.13**	**78.19**	**88.21**	**0.2001**

For simulation dataset, we convert generated videos into executable action trajectories, and measure task success rate by replaying the resulting trajectories in the simulator.

Realistic Dataset. Human evaluation on real-world Droid tasks demonstrates that the benefits of GEM-4D extend well beyond simulation (Tab. 2). GEM-4D consistently achieves substantially higher task success across all benchmarks, improving performance from 58% → 75% on AUTOLab, 65% → 83% on CLVR, and 59% → 87% on RAIL. The consistent gains across diverse real-world environments indicate that geometry-aware supervision yields more robust and transferable world representations for long-horizon manipulation. *Simulation Dataset.* The right part of Tab. 2 further evaluates policy extraction by re-executing predicted trajectories in simulation. Compared with TesserAct and CogVideoX, GEM-4D achieves higher planning success across most tasks. As the inverse dynamics module of TesserAct is not open-source, we use our AIDS to process its generated videos. Because most videos generated by CogVideoX cannot be reliably processed by our AIDS, we do not report task success rates for this baseline.

While TesserAct frequently fails to produce executable trajectories, GEM-4D reaches 63%–82% success on RLBench tasks and consistently improves performance on long-horizon manipulation benchmarks. This shows that geometry-consistent world modeling significantly improves downstream action planning.

Qualitative Experiment To evaluate our model in realistic settings, we conducted several tasks using a UF robot arm in real-world scenarios. The results are shown in Fig. 6. We observe that, even under unseen backgrounds, our method generates realistic visual results and reasonable geometry, demonstrating strong potential for transfer to real robot manipulation.

4.3 Ablation Studies

We conduct a series of ablation studies to examine the contributions of the proposed geometry flow branch and different geometry priors. The results are summarized in Table 3. First, we consider a baseline obtained by directly fine-tuning Generation backbone without any geometry guidance, in order to test whether fine-tuning alone is sufficient to improve 4D generation. Second, we replace the proposed geometry representation with explicit depth supervision, training the model to predict depth velocity rather than geometry features (**GEM-4D (Dep)**). This setting evaluates whether direct depth constraints can provide comparable geometric guidance. Third, we compare different geometry foundation models, including VGGT, to study how the choice of geometry prior influences learning (**GEM-4D (VGGT)**). The results reveal several consistent trends. Directly incorporating VGGT features slightly degrades performance, which may stem from the fact that VGGT is primarily trained for static or quasi-static scenes and is therefore less well matched to the dynamic scene evolution required in robotic manipulation. Using depth as a geometric constraint, by contrast, yields competitive performance. Notably, unlike Tesseract, which conditions on first-frame depth and predicts future depth, our depth ablation receives only noise as input and learns depth through the flow-matching objective alone.

5 Conclusion

We presented GEM-4D, a geometry-enhanced video world model for robot manipulation. Unlike prior video world models that often generate visually plausible but geometrically inconsistent futures, GEM-4D incorporates supervision from geometry foundation models to enforce correspondence-consistent scene evolution during training, while retaining a single-stream architecture with no additional inference cost. Our approach improves both appearance generation and geometric fidelity, enabling predicted rollouts to support downstream action extraction. To close the loop from generation to control, we further introduced an adaptive inverse dynamics system that converts generated videos into executable robot trajectories. Experiments across real-world and simulated benchmarks demonstrate consistent gains in video prediction, geometric consistency, and manipulation success. Overall, our results suggest that distilling geometric structure into video world models is a simple and effective step toward more reliable world models for embodied AI.

Bibliography

[1] Bharadhwaj, H., Dwibedi, D., Gupta, A., Tulsiani, S., Doersch, C., Xiao, T., Shah, D., Xia, F., Sadigh, D., Kirmani, S.: Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. arXiv preprint arXiv:2409.16283 (2024)

[2] Bharadhwaj, H., Mottaghi, R., Gupta, A., Tulsiani, S.: Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In: European Conference on Computer Vision (ECCV) (2024)

[3] Björck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: GR00T N1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)

[4] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: pi0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)

[5] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)

[6] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)

[7] Bruce, J., Dennis, M.D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al.: Genie: Generative interactive environments. In: Forty-first International Conference on Machine Learning (2024)

[8] Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024)

[9] Chen, B., Zhang, T., Geng, H., Song, K., Zhang, C., Li, P., Freeman, W.T., Malik, J., Abbeel, P., Tedrake, R., et al.: Large video planner enables generalizable robot control. arXiv preprint arXiv:2512.15840 (2025)

[10] Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Easi3r: Estimating disentangled motion from dust3r without training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9158–9168 (2025)

[11] Chi, X., Ge, K., Liu, J., Zhou, S., Jia, P., He, Z., Liu, Y., Li, T., Han, L., Han, S., et al.: Mind: Learning a dual-system world model for real-time planning and implicit risk analysis. arXiv preprint arXiv:2506.18897 (2025)

[12] Chi, X., Zhang, H., Fan, C.K., Qi, X., Zhang, R., Chen, A., Chan, C.m., Xue, W., Luo, W., Zhang, S., et al.: Eva: An embodied world model for future video anticipation. arXiv preprint arXiv:2410.15461 (2024)

[13] Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: TAP-Vid: A benchmark for tracking any point in a video. In: Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track (2022)
[14] Du, Y., Yang, M., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J.B., et al.: Video language planning. arXiv preprint arXiv:2310.10625 (2023)
[15] Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., Abbeel, P.: Learning universal policies via text-guided video generation. Advances in neural information processing systems **36**, 9156–9172 (2023)
[16] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
[17] Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 605–613 (2017)
[18] Feng, Y., Xiang, C., Mao, X., Tan, H., Zhang, Z., Huang, S., Zheng, K., Liu, H., Su, H., Zhu, J.: Vidarc: Embodied video diffusion model for closed-loop control. arXiv preprint arXiv:2512.17661 (2025)
[19] Fu, X., Wang, X., Liu, X., Bai, J., Xu, R., Wan, P., Zhang, D., Lin, D.: Learning video generation for robotic manipulation with collaborative trajectory control. arXiv preprint arXiv:2506.01943 (2025)
[20] Guo, Y., Shi, L.X., Chen, J., Finn, C.: Ctrl-world: A controllable generative world model for robot manipulation. arXiv preprint arXiv:2510.10125 (2025)
[21] Huang, A., Chen, J., Cheng, J., Song, R., Pan, W., Zhang, W.: Skill-aware diffusion for generalizable robotic manipulation. arXiv preprint arXiv:2601.11266 (2026)
[22] Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. arXiv preprint arXiv:2601.03782 (2026)
[23] Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
[24] James, S., Ma, Z., Arrojo, D.R., Davison, A.J.: Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters **5**(2), 3019–3026 (2020)
[25] Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6013–6022 (2025)
[26] Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European conference on computer vision. pp. 18–35. Springer (2024)
[27] Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., et al.:

Droid: A large-scale in-the-wild robot manipulation dataset. arXiv preprint arXiv:2403.12945 (2024)

[28] Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)

[29] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[30] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)

[31] Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024)

[32] Li, Z., Wu, P., Han, X., Cai, R., Du, Y.: 4d latent world model for robot planning. arXiv preprint arXiv:????.???? (2025)

[33] Lin, H., Chen, S., Liew, J., 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)

[34] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)

[35] Liu, L., Wang, X., Zhao, G., Li, K., Qin, W., Zhu, J., Qiu, J., Zhu, Z., Huang, G., Su, Z.: Robotransfer: Controllable geometry-consistent video diffusion for manipulation policy transfer. arXiv preprint arXiv:2505.23171 (2025)

[36] Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al.: Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177 (2024)

[37] Liu, Z., Li, S., Cousineau, E., Feng, S., Burchfiel, B., Song, S.: Geometry-aware 4d video generation for robot manipulation. arXiv preprint arXiv:2507.01099 (2025)

[38] Murali, A., Sundaralingam, B., Chao, Y.W., Yamada, J., Yuan, W., Carlson, M., Ramos, F., Birchfield, S., Fox, D., Eppner, C.: GraspGen: A diffusion-based framework for 6-DOF grasping with on-generator training. arXiv preprint arXiv:2507.13097 (2025)

[39] Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)

[40] Qi, H., Yin, H., Du, Y., Yang, H.: Strengthening generative robot policies through predictive world modeling. arXiv preprint arXiv:2502.00622 (2025)

[41] 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)

[42] Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
[43] Shoemake, K.: Animating rotation with quaternion curves. In: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). pp. 245–254. ACM (1985)
[44] Tao, S., Xiang, F., Shukla, A., Qin, Y., Hinrichsen, X., Yuan, X., Bao, C., Lin, X., Liu, Y., Chan, T.k., et al.: Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. arXiv preprint arXiv:2410.00425 (2024)
[45] Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Fvd: A new metric for video generation (2019)
[46] Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., Kim, M.J., Du, M., et al.: Bridgedata v2: A dataset for robot learning at scale. In: Conference on Robot Learning. pp. 1723–1736. PMLR (2023)
[47] Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
[48] Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: International Conference on 3D Vision (3DV) (2025)
[49] Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
[50] Wang, J., Chen, M., Zhang, S., Karaev, N., Schönberger, J., Labatut, P., Bojanowski, P., Novotny, D., Vedaldi, A., Rupprecht, C.: Vggt-omega. arXiv preprint arXiv:2605.15195 (2026)
[51] Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025)
[52] Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024)
[53] Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Scalable permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
[54] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
[55] Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17868–17879 (2024)
[56] Wen, C., Lin, X., So, J., Chen, K., Dou, Q., Gao, Y., Abbeel, P.: Any-point trajectory modeling for policy learning. In: Robotics: Science and Systems (RSS) (2024)

[57] Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
[58] Wu, H., Wu, D., He, T., Guo, J., Ye, Y., Duan, Y., Bian, J.: Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. arXiv preprint arXiv:2507.07982 (2025)
[59] Wu, H., Yu, J., Zou, Y., Liu, X.: Multiworld: Scalable multi-agent multi-view video world models. arXiv preprint arXiv:2604.18564 (2026)
[60] Wu, J., Yin, S., Feng, N., He, X., Li, D., Hao, J., Long, M.: ivideogpt: Interactive videogpts are scalable world models. Advances in Neural Information Processing Systems **37**, 68082–68119 (2024)
[61] Xu, K., Tse, T.H.E., Peng, J., Yao, A.: Das3r: Dynamics-aware gaussian splatting for static scene reconstruction. arXiv preprint arXiv:2412.19584 (2024)
[62] Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. arXiv preprint arXiv:2501.13928 (2025)
[63] Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., Abbeel, P.: Learning interactive real-world simulators. arXiv preprint arXiv:2310.06114 (2023)
[64] 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)
[65] Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. In: International Conference on Learning Representations (ICLR) (2025)
[66] Zhang, C., Wu, Z., Lu, G., Tang, Y., Wang, Z.: imowm: Taming interactive multi-modal world model for robotic manipulation. arXiv preprint arXiv:2510.09036 (2025)
[67] Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. In: ICLR (2025)
[68] Zhang, X., Liao, J., Zhang, S., Meng, F., Wan, X., Yan, J., Cheng, Y.: Videorepa: Learning physics for video generation through relational alignment with foundation models. arXiv preprint arXiv:2505.23656 (2025)
[69] Zhen, H., Sun, Q., Zhang, H., Li, J., Zhou, S., Du, Y., Gan, C.: Tesseract: learning 4d embodied world models. arXiv preprint arXiv:2504.20995 (2025)
[70] Zhi, H., Chen, P., Zhou, S., Dong, Y., Wu, Q., Han, L., Tan, M.: 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. arXiv preprint arXiv:2506.06199 (2025)
[71] Zhou, K., Dodds, L., Afzal, S.S., Adib, F.: Rise: Single static radar-based indoor scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 32194–32205 (2026)
[72] Zhou, K., Wang, Y., Chen, G., Beaudouin, G., Zhan, F., Liang, P.P., Wang, M.: Page-4d: Vggt-4d perception via disentangled pose and geometry esti-

mation. In: The Fourteenth International Conference on Learning Representations

[73] Zhou, K., Zhong, J.X., Shin, S., Lu, K., Yang, Y., Markham, A., Trigoni, N.: Dynpoint: Dynamic neural point for view synthesis. Advances in Neural Information Processing Systems **36**, 69532–69545 (2023)

[74] Zhou, S., Du, Y., Chen, J., Li, Y., Yeung, D.Y., Gan, C.: Robodreamer: Learning compositional world models for robot imagination. arXiv preprint arXiv:2404.12377 (2024)

[75] Zhu, H., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Chen, J., Shen, C., Pang, J., He, T.: Aether: Geometric-aware unified world modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8535–8546 (2025)