PLAN4D: Generative Plannable 4D Worlds

Hangwei Zhang[1,2] **Rujing Dang**[2] **Ruimin Lin**[2]
Jiahong Wu[2,†] **Xiangxiang Chu**[2] **Chuanxia Zheng**[1,†]

[1] PVG, Nanyang Technological University [2] AMAP, Alibaba Group

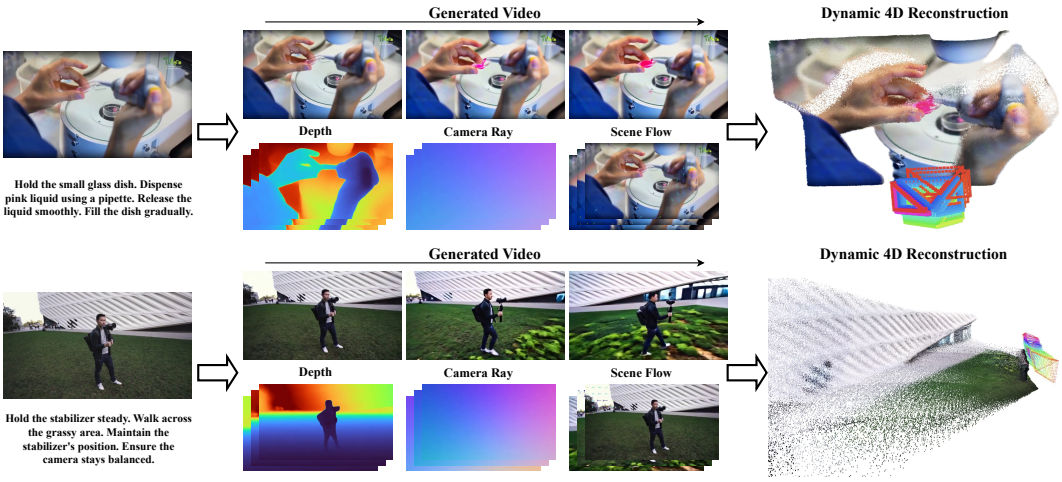

Figure 1: **Exampled future 4D world generated by PLAN4D.** Given an observed scene context and a text-specified action, PLAN4D generates a future video together with synchronized depth, camera rays, and scene flow (subset shown), which are further lifted into a dynamic 4D scene. Top: fine-grained hand-object interaction. Bottom: camera motion with scene-level geometry changes.

Abstract

Most world models for future prediction prioritize the aesthetic quality of generated videos, but offer only limited geometric consistency over time, due to the lack of explicit 4D modeling. In this paper, we introduce PLAN4D, a possible next milestone towards building a *generative plannable 4D world model* that predicts future RGB videos together with dense 4D geometry, including depth, cameras, point clouds, and scene flow, from an observed scene context and text-specified actions. Since training data with paired actions and dense 4D annotations is inherently scarce, PLAN4D leverages the complementary priors of foundational video generators and 3D reconstructors. Specifically, we propose a model-stitching architecture that couples Depth Anything 3 with LongCat-Video and augments it with an auxiliary DPT head for scene-flow prediction. The resulting model allows us to plan and generate complex, long-horizon, and fine-grained actions in dynamic 3D scenes, while substantially improving both visual fidelity and geometric accuracy of the generated future videos. Extensive experiments on both synthetic and real-world datasets demonstrate that PLAN4D achieves state-of-the-art performance in generative 4D modeling of environmental reactions. Project Page: `https://fogradio.github.io/plan4d-project`.

[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Given an observed scene context and a set of text-specified actions, we consider the task of *generative plannable 4D worlds*, a new paradigm for predicting reliable futures for the physical world that jointly produces visually plausible RGB frames and physically grounded 4D geometry –including depth, camera dynamics, and scene flow –within a single feed-forward model. Prior *observation-prediction models* formulate future prediction largely as RGB synthesis [25, 86], often on a per-frame basis, notably UniSim [106], Genie 1–3 [13, 76, 8], NWM [10], and V-JEPA 2 [4]. By contrast, with our actionable 4D future world model, humans and AI agents can plan and execute complex, long-horizon, and fine-grained actions in the explicit 3D environment, such as picking up a cup, opening a bottle, or navigating through a cluttered room. These dynamic 3D quantities are essential for accurate measurement, reliable simulation, and effective execution of downstream tasks [54, 24, 59, 41, 33].

A major bottleneck of this task is the lack of suitable datasets, namely action-conditioned video sequences paired with rich text and *comprehensive* 3D annotations, including camera trajectories, accurate dense depth and motion. For example, large-scale video-caption datasets [7, 98, 18] may still lack detailed captions for physical fine-grained actions; the latest Action-100M [15] dataset provides detailed action annotations but without 3D annotations, while most dynamic 3D datasets [31, 30, 51] are relatively small in scale or lack natural actions.

Correspondingly, powerful baselines have emerged in parallel on these separate data sources, including video generators [35, 12, 11, 81, 34, 53, 108, 90, 66, 87], action-conditioned world models [106, 13, 76, 8, 10, 4, 48, 58], and 3D reconstructors [97, 57, 96, 92, 52, 99, 64, 114]. Despite being trained to build the same physical world, these models still operate as isolated systems for particular tasks. Standard pixel-level generators excel at synthesizing visually realistic RGB frames [102], but typically lack physical consistency [101, 61]; and 3D models recover explicit geometry well [94] but are largely reconstruction-centric and static. Even advanced 4D frameworks [116, 47, 82, 121, 50, 73, 72, 55] only infer motion from *observed* frames and *cannot generate action-conditioned future worlds*. As a result, none of these families alone provides a *generative plannable 4D world*, in which appearance, geometry, camera and action-conditioned future dynamics evolve consistently over time.

What if, given these different dataset sources, we stitch them together to model the same underlying 4D world? To achieve such a *generative plannable 4D world*, we propose **PLAN4D**, a unified framework for *planning* and *generating* dense 4D worlds from observed context and action descriptions. Our *core insight* is to decouple the acquisition of strong geometric priors from the generation of future dynamics, then reconnect them through a stitched latent world model that jointly plans RGB appearance and dense 4D geometry, unlike recent methods [85, 117, 22, 71, 27] that attempt to learn a 4D world model's geometry from scratch. Such a paradigm reduces the task's learning difficulty and data requirements, while allowing us to leverage the powerful priors of pretrained video generators and 3D reconstructors to achieve complex, long-horizon and fine-grained dynamics.

In practice, this paradigm requires solving several coupled challenges. i) the model must extend a frozen 3D foundation model from static scenes to dynamic 4D prediction while preserving its geometric consistency. ii) it must align geometry tokens with the latent space of a video generator without destroying the pretrained generative prior. iii) it must learn from real, unlabeled action videos without dense 4D supervision [112]. iv) it must also ensure consistency across the generated RGB, depth, camera, point clouds, and scene flow. Recent methods, including VIST3A [29], Geometry Forcing [100], and Gen3R [37], couple 3D reconstructors with video generators for static text-to-3D or scene-level 3D generation. We extend this direction to *full actionable 4D world generation*, jointly predicting full 4D geometry and appearance in future video sequences depicting fine-grained actions.

Unlike prior approaches that use geometry mainly as an auxiliary constraint [100, 6, 110, 26, 3], or decode geometry only after RGB generation [24], our PLAN4D turns a video generator into an actionable 4D world model, which *explicitly aligns future appearance and 4D structure inside the shared evolving video latent state.* To preserve strong geometric priors while extending them beyond reconstruction, we augment Depth Anything 3 [64] (DA3) with a lightweight token-only scene-flow head that predicts dense forward 3D motion from intermediate geometry tokens, while keeping its pretrained depth and camera-ray predictions intact. To connect this geometric readout with future generation, we learn a *latent-to-geometry stitching module* that maps video VAE latents into a selected DA3 block and trains it with token distillation and geometric supervision. The resulting stitched

decoder provides a stable, differentiable interface from video latents to full 4D geometry. We then train the video generator against this frozen geometric interface, so unlabeled action videos can supervise not only visual prediction but also physically grounded 4D consistency. At inference time, a single generated latent future yields RGB frames and dense 4D geometry in one feed-forward pass.

Our main contributions can be summarized as follows:

- We take a step towards the new paradigm of *generative plannable 4D worlds*, which jointly generates future RGB appearance and corresponding full 4D geometry, including depth, camera rays, point clouds, and scene flow, from observed context and action descriptions.
- We propose PLAN4D, a unified framework with a scene-flow head and a geometry-video stitching mechanism that connect a frozen 3D foundation model with a pretrained video generator, preserving geometric priors while enabling action-conditioned dynamic 4D future prediction from unlabeled videos.
- We illustrate how PLAN4D can achieve state-of-the-art performance in dense 4D world generation, improving both visual realism and geometric accuracy across scene flow, 3D reconstruction, and action-conditioned video prediction through extensive experiments.

2 Related Works

4D world models. Recent work has begun to model 4D worlds from two complementary directions. Explicit dynamics methods such as ParticleFormer [40] and PointWorld [41] predict particle trajectories or point-level motion, but mainly operate in structured 3D state spaces rather than generating rich future 4D. 4D-native generative models instead learn spatiotemporal world evolution, ranging from dynamic reconstruction, visual planning, and image-to-4D generation in Aether [85] and 4DNeX [21], to RGB-D or geometry-aware scene generation in TesserAct [117] and Robo4DGen [69]. GWM [70], iMoWM [113], and Kinema4D [105] further explore Gaussian, tokenized, and kinematics-driven representations, while FantasyWorld [22], WorldReel [27], TeleWorld [20], HY-World 2.0 [43], and Lyra 2.0 [80] couple video generation with geometric prediction for more unified world modeling. Despite this progress, existing systems typically emphasize either explicit geometric forecasting or large-scale generative simulation, and they rarely integrate the complementary priors of pretrained geometry foundation models into the latent dynamics of video generation.

Aligning 3D Reconstruction and Video Generation. Early uses of generative priors for 3D synthesis mostly follow a cascade, where geometric constraints are applied to generated outputs rather than to internal generative states [77, 28, 95]. Later methods tighten this link through reconstruction-guided diffusion or distillation, as in GenFusion [103], VideoScene [91], and Lyra [5]. Recent work moves the alignment into feature or latent spaces. Geometry Forcing [100] aligns intermediate diffusion features with pretrained geometric representations [111], Gen3R [37] aligns geometry and appearance latents for joint generation, and GLD [46] repurposes geometric features as multi-view diffusion latents. From the model-stitching view [56, 9], VIST3A [29] connects a pretrained video VAE with a 3D network at linearly matchable layers. Our method builds on this direction, but stitches a video diffusion model to a 3D foundation model so that RGB synthesis and dense 4D geometry can be optimized and decoded from a shared latent rollout.

3 Plan4D

Given an observed context $\mathcal{O} = \{\boldsymbol{I}_i\}_{i=1}^{M}$ and the planned action space $\mathcal{A}$, our goal is to learn a generative plannable 4D world model $f_{\boldsymbol{\theta}}$ that is conditioned on $\mathcal{O}$ and $\mathcal{A}$ and predicts a set of future 4D representations: $f_{\boldsymbol{\theta}} : \{\mathcal{O}, \mathcal{A}\} \mapsto \{\boldsymbol{I}_t, \boldsymbol{D}_t, \boldsymbol{r}_t, \boldsymbol{V}_t\}_{t=1}^{T}$, where $\mathcal{A}$ is a sequence of textual action descriptions $\boldsymbol{p}$. The model $f_{\boldsymbol{\theta}}$ learns a video latent trajectory that is conditioned on the observed context and the action description, and reads out the future RGB video frames $\boldsymbol{I}_t \in \mathbb{R}^{H\times W\times 3}$; and its corresponding depth maps $\boldsymbol{D}_t \in \mathbb{R}^{H\times W}$, camera ray maps $\boldsymbol{r}_t \in \mathbb{R}^{H\times W\times 6}$, and full 4D scene flow $\boldsymbol{V}_t \in \mathbb{R}^{H\times W\times 3}$. The ray maps $\boldsymbol{r}_t$ encode the camera origin and the ray directions for each pixel, which can be used to reconstruct the 3D point cloud of the scene. The scene flow $\boldsymbol{V}_t$ encodes the 3D motion of each pixel from time t to $t+1$[1].

[1] Since we only define the forward flow, we then simplify $\boldsymbol{V}_{t\to t+1}$ as $\boldsymbol{V}_t$.

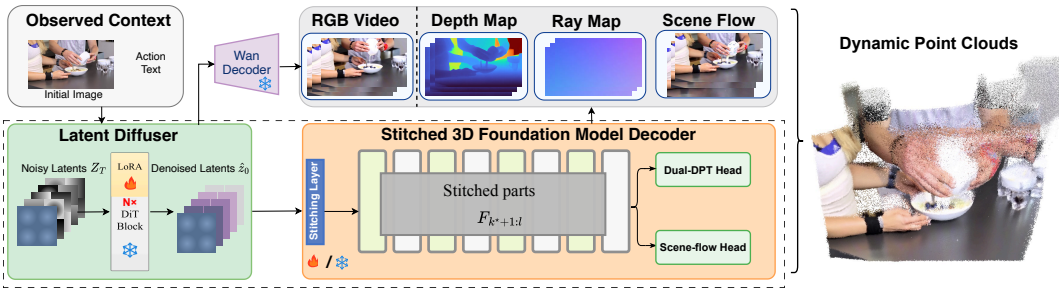

Figure 2: **Overview of the PLAN4D pipeline.** A video generator produces a denoised latent trajectory, while a stitched DA3 decoder reads the same latent state to recover synchronized depth, camera rays, camera parameters, point clouds, and scene flow. During joint training, the stitched decoder is frozen and acts as a differentiable geometry critic for the video generator; at inference time, RGB video and dense 4D structure are decoded from the shared latent trajectory.

We parametrize $f_{\boldsymbol{\theta}}$ with a model-stitching architecture that couples a video generator, LongCat-Video [87] (which uses the Wan 2.1 VAE [90]), with a foundational 3D reconstructor, Depth Anything 3 [64] (DA3). The *key insight* is that, by building on the pretrained video generator and 3D reconstructor, our model can leverage the complementary priors of these two foundational models to achieve both high visual fidelity and strong geometric consistency in the generated future videos, unlike training it from scratch [85, 117, 22, 27], which is extremely challenging and data-hungry. To improve the motion modeling capability of DA3, we first augment it with an auxiliary DPT-style scene flow head in section 3.1. In section 3.2, we then stitch the video generator and the augmented DA3 together through a learnable latent-to-geometry stitching module, followed by a distillation and geometric supervision process to calibrate the stitched model in section 3.3. Finally, we train the joint model in section 3.4.

3.1 Augmenting DA3 for Dynamic Scene Flow Prediction

The physical world is *not* a collection of independent per-frame geometries. To model the dynamics and correspondence of the 4D world, we augment DA3 [64] with a dense *scene flow* head that predicts the 3D motion of each pixel. Following the original DPT-style decoder in DA3, the new head ingests the four intermediate token maps of the frozen encoder, and outputs a dense three-channel motion field via a lightweight prediction layer. Note that, the original encoder, as well as the depth and camera pose branches, remains unchanged; we simply add a dedicated motion branch. Our analysis in Section A shows that frozen DA3 tokens already encode cross-frame motion cues, which supports using this branch as a lightweight readout rather than a full motion model. In particular, for each frame t, the new head predicts a scene flow on the pixel lattice of that frame. Each valid pixel stores the forward 3D displacement from time t to $t+1$. The raw prediction is produced in the local camera basis of frame t and is mapped to a common reference frame with a lightweight adapter,

$$\boldsymbol{V}_t^{(0)}(u,v) = \mathbf{R}_{t\to 0}\, \boldsymbol{V}_t^{(\mathrm{cam})}(u,v), \tag{1}$$

where $\mathbf{R}_{t\to 0}$ denotes the rotation from the camera basis of frame t to the first-frame camera basis. We apply only rotation because scene flow is a vector quantity rather than a 3D point. This representation keeps spatial indexing local to each frame while expressing all motion vectors in a temporally stable basis. As a result, the predicted dynamics remain consistent under camera motion and align naturally with the depth, ray, and point-based geometry produced by DA3. Despite being a seemingly minor change, the new head has a substantial effect in practice: it consistently improves reconstruction, camera, and depth estimation quality on various dynamic scenes compared with stitched variants without motion modeling, as shown in our experiments.

3.2 Latent-to-Geometry Model Stitching

Following recent works in model stitching [75, 29, 37], we design a simple learnable module to stitch the video generator and the augmented DA3 together, allowing the video latent to be directly decoded into full 4D geometry, including depth, camera poses, point clouds, and scene flow. Let $\boldsymbol{I}^{1:T}$ be a video clip and let $\boldsymbol{z}_0^{1:T} = \mathcal{E}_{\mathrm{vae}}(\boldsymbol{I}^{1:T})$ be its Wan 2.1 VAE latent [90]. We first upsample

the latent to the spatial and temporal resolution required by DA3: $\tilde{\boldsymbol{z}}_0^{1:T} = \mathrm{Up}_{(s_t,s_h,s_w)}(\boldsymbol{z}_0^{1:T}) \in \mathbb{R}^{(s_t t)\times(s_h h)\times(s_w w)\times C}$. A learnable 3D stitching module $\mathcal{S}_{\boldsymbol{\theta}}$ then maps the upsampled latent tensor into the channel dimension of a DA3 encoder block, $\mathbf{h}^{(k)} = \mathcal{S}_{\boldsymbol{\theta}}\big(\tilde{\boldsymbol{z}}_0^{1:T}\big)$. In our implementation $\mathcal{S}_{\boldsymbol{\theta}}$ is a compact spatiotemporal projection, instantiated as a 3D convolution. Its temporal kernel aggregates neighboring latent frames, and its spatial stride matches the VAE grid to the DA3 patch grid.

The injection layer is not chosen heuristically. For a set of paired clips, we extract Wan 2.1 VAE latents and frozen DA3 candidate features from several early encoder blocks. For each candidate block k, we solve a regularized linear fitting problem,

$$(k^\star, \boldsymbol{\theta}_0) = \arg\min_{k,\boldsymbol{\theta}} \sum_i \left\| \mathcal{S}_{\boldsymbol{\theta}}\big(\tilde{\boldsymbol{z}}_0^{1:T}\big) - \mathbf{g}_i^{(k)} \right\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{2}$$

where $\mathbf{g}_i^{(k)}$ is the frozen DA3 feature grid at block k for clip i. This procedure selects the block whose representation is most linearly reachable from the video VAE latent [29, 3]. It also provides a stable initialization $\boldsymbol{\theta}_0$ for the stitching module. The selected block acts as the interface between the generative video prior and the geometric reconstruction prior.

After stitching, the original RGB patch embedding and the encoder blocks before $k^\star$ in DA3 are bypassed. The projected latent tokens are combined with DA3 special tokens and positional encodings, then processed by the remaining DA3 encoder blocks and the native dense heads, $\hat{\mathcal{G}} = \mathcal{H}^{\mathrm{DA3}}_{k^\star:L}\left(\mathbf{h}^{(k^\star)}\right)$. The output $\hat{\mathcal{G}}$ contains depth, camera rays, confidence maps, and scene flow. This design is important because the stitched decoder reads the denoised video latent directly. It does *not* need to first decode the latent into RGB and then run a separate geometry model. Visual RGB generation and 3D reconstruction are instead two synchronized readouts of the same latent video state.

3.3 Stitched Decoder Finetuning

Here, we describe how to finetune the stitched decoder to adapt to the new latent input distribution and obtain accurate geometry predictions. Unlike VIST3A [29] which only distills RGB features, we employ a hybrid training strategy that combines teacher-student distillation with ground-truth (GT) supervision. The teacher-student distillation follows the same principle as VIST3A, where the teacher is the original frozen DA3 model augmented with the new scene flow head. This teacher model takes the ground-truth RGB video clip as input and produces all types of 4D geometric outputs as supervision signals. The student is the stitched decoder, which takes the Wan VAE latent of the same clip as input and produces the same set of geometric outputs. For the GT supervision, we utilize the 3D annotations provided in the dataset as explicit regression targets.

Following the formulation in Lin et al. [64], our training objective integrates dense geometric matching with spatial and temporal regularization constraints,

$$\mathcal{L}_{\text{geo}} = \lambda_D \mathcal{L}_{\text{depth}} + \lambda_R \mathcal{L}_{\text{ray}} + \lambda_C \mathcal{L}_{\text{camera}} + \lambda_M \mathcal{L}_{\text{motion}} + \lambda_T \mathcal{L}_{\text{temporal}}. \tag{3}$$

We also apply distillation to the depth and ray confidence maps using small weighting factors. The training details are discussed in Section B in detail. We keep the video diffusion transformer strictly frozen throughout this stage. This entire training phase exclusively focuses on transforming the DA3 decoder into a robust geometric readout mechanism for the Wan latent space.

3.4 Joint Video and 4D Geometry Training

Once the stitched decoder is finetuned, we use it as a differentiable geometry critic for the video diffusion model. The video model is trained with the standard latent flow-matching objective [115]. For each clip, we sample a scalar diffusion time $\tau \in [0, 1]$ and use the corresponding scheduler noise scale $\sigma_\tau \in \mathbb{R}$. Both values are shared by all latent frames in the clip and broadcast over space and channels. Omitting the batch dimension, let $\boldsymbol{z}_\tau^{1:T} \in \mathbb{R}^{T_\ell \times h \times w \times C_z}$ denote the noisy Wan latent trajectory at time τ, where T_ℓ is the latent-frame length, (h, w) is the VAE spatial grid, and $C_z = 16$ for Wan. The conditioning tuple $c = (\boldsymbol{e_p}, \boldsymbol{z}_{\text{ctx}}, \boldsymbol{m}_{\text{ctx}})$ contains the text-action embeddings $\boldsymbol{e_p} \in \mathbb{R}^{L_p \times d_c}$ and context-frame latents $\boldsymbol{z}_{\text{ctx}} \in \mathbb{R}^{T_c \times h \times w \times C_z}$ with a binary prefix mask $\boldsymbol{m}_{\text{ctx}} \in \{0, 1\}^{T_\ell \times h \times w \times 1}$. Given $\boldsymbol{z}_\tau^{1:T}$, τ, and c, the diffusion transformer predicts a velocity field $\boldsymbol{v}_\phi(\boldsymbol{z}_\tau^{1:T}, \tau, c) \in \mathbb{R}^{T_\ell \times h \times w \times C_z}$. We convert the velocity prediction into a clean-latent estimate

$$\hat{\boldsymbol{z}}_0^{1:T} = \boldsymbol{z}_\tau^{1:T} - \sigma_\tau \, \boldsymbol{v}_\phi(\boldsymbol{z}_\tau^{1:T}, \tau, c), \tag{4}$$

Table 1: **Scene flow evaluation** on MOVi-F [31], Dynamic Replica [49], and PointOdyssey [118] using 81-frame clips. We report training data in millions of frames. EPE and $\text{APD}_{0.05}$ are shown without percentage symbols for readability. The best, second-best, and third-best results within each dataset are highlighted in red, orange, and yellow, respectively.

Method	Train frames / M ↓	MOVi-F [31]		Dynamic Replica [49]		PointOdyssey [118]	
		EPE ↓	$\text{APD}_{0.05}$ ↑	EPE ↓	$\text{APD}_{0.05}$ ↑	EPE ↓	$\text{APD}_{0.05}$ ↑
TraceAnything [68]	1.68	13.27	91.69	0.74	99.23	0.95	95.34
Any4D [50]	1.98	27.97	47.75	3.06	85.85	24.95	37.17
MotionCrafter [121]	1.67	13.48	90.86	0.82	99.25	1.14	94.50
V-DPM [82]	1.74	15.89	81.30	0.75	99.23	**0.92**	**95.54**
Zero-MSF [63] + GT	1.36	13.33	92.47	**0.45**	**99.28**	1.04	95.41
Ours (DA3)	**0.97**	13.94	92.47	0.76	99.23	0.93	95.45
Ours (Stitched)	**0.97**	**13.10**	**93.32**	0.61	99.27	1.22	93.58

where $\hat{z}_0^{1:T}$ has the same tensor shape as $z_\tau^{1:T}$ and is passed directly into the frozen stitched decoder. The joint objective is

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{geo}}\, \mathcal{L}_{\text{geo}}\left(\mathcal{D}_{\text{stitch}}(\hat{z}_0)\right), \tag{5}$$

where $\mathcal{L}_{\text{FM}}$ denotes the latent flow-matching loss and $\mathcal{D}_{\text{stitch}}$ represents the frozen stitched decoder. The geometric penalty $\mathcal{L}_{\text{geo}}$ is strictly identical to the objective formulated in Equation (3). During this joint optimization phase, we exclusively utilize the teacher-student distillation mechanism for action-conditioned videos. As introduced previously, an independently loaded raw DA3 model serves as the teacher network. It processes the ground-truth RGB clip to provide the reference targets for depth, ray maps, and scene flow. For image-to-video generation, the first frame is encoded as the conditioning prefix [87]. The diffusion loss can mask the conditioned prefix, while the geometry loss can be applied to the generated suffix. As a result, Plan4D learns a single latent trajectory that must satisfy both appearance realism and dense 4D consistency. At inference time, the final latent sequence is decoded by the Wan VAE into RGB video and by the stitched DA3 decoder into depth, camera rays, and scene flow.

4 Experiments

4.1 Experimental Settings.

Implementation. PLAN4D is trained in three stages on diverse datasets due to the lack of large-scale data with paired action and dense 4D annotations. **(1)** We train the designed scene-flow head on top of frozen DA3 using PointOdyssey [118], Dynamic Replica [49], Virtual KITTI 2 [14], BlinkVision [62], and OmniWorld-Game [119] for 20 epochs on 16 H100 GPUs. **(2)** We fit the latent-to-geometry stitching module on a small subset of the above datasets to initialize the interface from Wan latents to DA3 tokens. Then we finetune the stitched decoder with the scene-flow-head data mixture augmented by ScanNet++ [109] and MVS-Synth [39]. This stage uses 64 H20 GPUs and runs one hybrid epoch over the full data mixture, where the same mixed sampling schedule is traversed once with ground-truth geometric supervision and once with teacher-student distillation. When a ground-truth-supervised sample does not provide scene-flow annotations, we set its scene-flow loss to zero. **(3)** We finally train the DiT on a subset of Action100M [15] for one epoch using 64 H20 GPUs. All stages use a per-GPU batch size of 1. Additional dataset details, optimization details, losses, and supervision definitions are provided in Section B.

Evaluation. We evaluate PLAN4D along three axes that correspond to its intended use as a generative 4D world model. **(1)** we evaluate dynamic 4D geometry through scene flow on MOVi-F [31], Dynamic Replica, and PointOdyssey using 81-frame clips, and report endpoint error together with $APD_{0.05}$ against reconstruction and motion baselines. **(2)** we test whether the stitched decoder preserves DA3 geometry after replacing the RGB encoder with a Wan-latent input pathway. This evaluation covers reconstruction, camera, and depth benchmarks, including ETH3D [79], ScanNet++, Dynamic Replica, Dynamic-RE10K [19], MOVi-F, and RobbySimVal [83], with comparisons to RGB-based 3D foundation models and latent-to-geometry baselines. Scene-flow metrics are computed

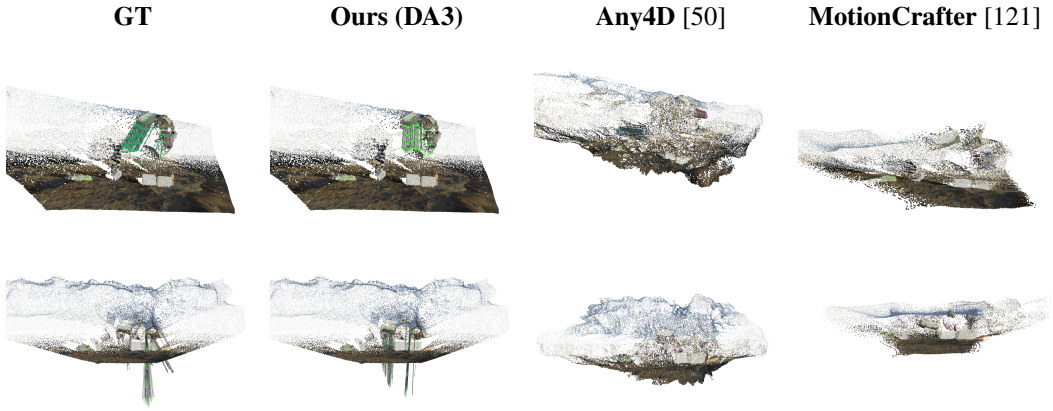

Figure 3: **Qualitative results for scene flow and dynamic reconstruction.** Columns from left to right compare the ground truth, results from our DA3-based prediction, Any4D, and MotionCrafter. Our result better preserves object-level motion and dynamic 3D structure while maintaining consistency with the reference geometry.

after rescaling predictions to the target point-map scale. Depth uses per-clip scale-shift alignment, while camera and reconstruction follow the official DA3 bench protocol, with first-camera pose alignment for relative-pose metrics and Umeyama scale alignment for reconstruction. **(3)** finally, we evaluate full action-conditioned 4D world generation on RLBench [45] (210 samples), HOIGen-1M [67] (200 samples), Dynamic Replica (480 samples), and PointOdyssey (375 samples). To further assess in-the-wild generalization, we additionally sample 100 five-second clips from Epic-Kitchens-100 [23], which contains everyday human-object interactions in kitchen environments, and evaluate all compared methods under the same joint-generation protocol. These benchmarks test whether a single generated rollout can jointly preserve visual quality [65, 120], action-text alignment [16], and temporally coherent 4D structure. We report video quality and instruction following with VBench-style video metrics and video-caption similarity [42]. For depth, reconstruction, and camera estimation, we follow the same protocol as the stitched-decoder evaluation. We compare with recent 4D world-generation and geometry-generation baselines, including VIST3A [29], TesserAct [117], and Gen3R [37], on the modalities each method supports.

4.2 Main Results

Scene Flow Estimation. We first test whether a frozen DA3 geometry prior can support dynamic 4D motion prediction beyond static reconstruction. This is a stringent setting because scene flow requires temporally coherent geometry rather than accurate per-frame depth alone. Table 1 shows that both scene-flow variants remain competitive on all three benchmarks while using the smallest training set among the compared methods. The RGB-based DA3 variant is particularly strong on the harder real dynamic benchmarks, suggesting that frozen DA3 tokens already support reliable dense 4D motion readout. The stitched variant performs best on the controlled synthetic benchmark and remains near the top on Dynamic Replica, which matters because it is the decoder interface used during joint video training. The qualitative results in Figure 3 show cleaner boundaries, more stable articulated motion, and better global 3D structure than the baselines. Overall, these results validate the first step of PLAN4D: DA3 can be extended from geometry reconstruction to strong dynamic 4D readout, and this capability largely survives the latent-to-geometry stitching step.

Stitched Geometry Decoder Preserves DA3 Behavior. We evaluate whether replacing DA3's RGB input with Wan video latents preserves its depth, camera, and reconstruction behavior on held-out benchmarks. We compare against the latest state-of-the-art RGB-based DA3 and VGGT, and the aligned VIST3A-Stitched and Gen3R-VAE decoders. Table 2 shows that our decoder remains closer to the RGB-based models than prior aligned decoders across all three geometry families. It slightly exceeds DA3 on Dynamic-RE10K camera estimation and remains competitive with VGGT on Dynamic Replica. Depth changes little on MOVi-F and declines moderately on RobbySimVal. Reconstruction shifts are larger, but our decoder remains closer to DA3 and VGGT in most settings

Table 2: **Stitched Geometry Decoder Evaluation.** 3D foundation models such as VGGT [92] and DA3 [64] calculate 3D metrics using raw RGB inputs. The remaining three approaches decode geometry from intermediate feature spaces. Gen3R-VAE [37] accomplishes this by projecting and reconstructing VGGT tokens. VIST3A-Stitched [29] and our method directly process Wan video latents to decode scene geometry.

Dataset	Target	Metric	**VGGT** [92]	**DA3** [64]	**VIST3A-Stitched** [29]	**Gen3R-VAE** [37]	**Ours (Stitched)**
ETH3D [79]	Reconstruction	Acc. ↓	0.2802	**0.2783**	0.9184	1.5954	0.2898
ScanNet++ [109]	Reconstruction	Comp. ↓	0.1115	**0.1015**	0.1425	0.2235	0.1425
Dynamic Replica [49]	Camera	RRA@5° ↑	**0.9403**	0.8971	0.9370	0.9383	0.8987
Dynamic Replica [49]	Reconstruction	Acc. ↓	0.0733	**0.0700**	0.3936	0.3402	0.0967
Dynamic Replica [49]	Reconstruction	Comp. ↓	0.0679	0.0834	0.4154	0.4308	**0.0658**
Dynamic Replica [49]	Reconstruction	F-score ↑	0.6298	**0.6843**	0.0740	0.0687	0.6125
Dynamic-RE10K [19]	Camera	AUC_{30} ↑	0.5149	0.5333	0.5186	0.4834	**0.5350**
Dynamic-RE10K [19]	Camera	AUC_{15} ↑	0.3854	0.4105	0.3956	0.3268	**0.4124**
MOVi-F [32]	Depth	δ_1 ↑	**0.9830**	0.8932	0.8455	0.9720	0.8943
MOVi-F [32]	Depth	AbsRel ↓	**0.0374**	0.0877	0.1257	0.0573	0.0931
RobbySimVal [83]	Depth	δ_1 ↑	0.9070	**0.9273**	0.8592	0.9042	0.9055
RobbySimVal [83]	Depth	AbsRel ↓	0.1032	**0.0808**	0.1465	0.1121	0.1085

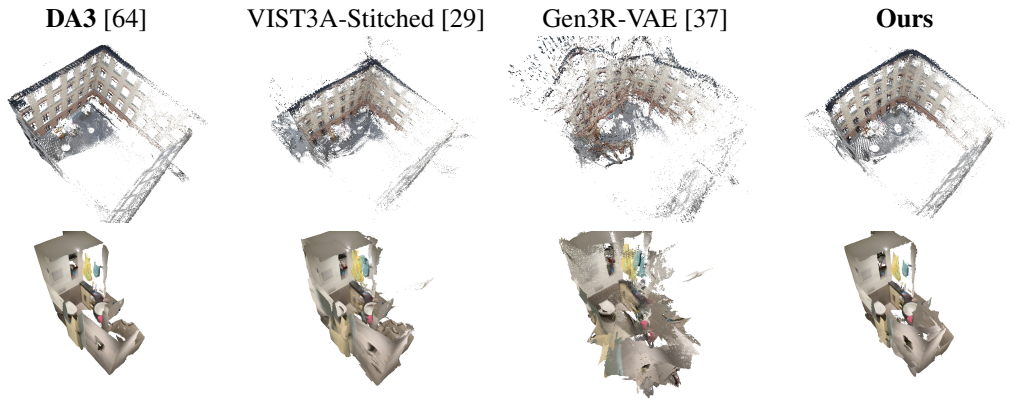

Figure 4: **Qualitative comparison on ETH3D (top) and ScanNet++ (bottom).** Our stitched decoder preserves DA3-like surface layout and scene structure more faithfully than the other aligned decoders.

and achieves the best completeness on Dynamic Replica. Figure 4 shows that our decoder preserves DA3-like surface layout and global scene organization, while competing aligned models produce less faithful structure. This preserved geometry allows the frozen decoder to provide stable supervision for the video generator directly from latent rollouts.

Joint 4D World Generation Table 3 reports a compact subset of the full joint 4D generation benchmark in Table 8. The subset covers HoiGen, Epic-Kitchens-100, Dynamic Replica, and PointOdyssey across video quality, depth and reconstruction, and camera estimation. On the 100 five-second Epic-Kitchens-100 clips, PLAN4D leads on 10 of the 13 metrics in the full comparison. On video quality, PLAN4D achieves the best subject consistency on all four datasets and the best caption similarity on Epic-Kitchens-100 and PointOdyssey. VIST3A has the highest caption similarity on HoiGen and Dynamic Replica. On Epic-Kitchens-100, PLAN4D also achieves the best background consistency and image quality, while TesserAct has the best motion smoothness and VIST3A the highest aesthetic quality. These results show that the shared latent rollout supports strong visual consistency and action-text alignment across diverse scenes.

For depth and reconstruction, PLAN4D gives the lowest AbsRel on all four datasets in the subset. It also obtains the best Chamfer-L1 on Epic-Kitchens-100, Dynamic Replica, and PointOdyssey, while Gen3R has the lowest Chamfer-L1 on HoiGen. On Epic-Kitchens-100, PLAN4D leads on all three depth metrics. The HoiGen exception separates local reconstruction quality from the full world-generation objective. Gen3R can produce a strong point-cloud distance on one hand-object benchmark, but its video quality and camera stability are much weaker in the same setting.

Table 3: **Joint 4D generation results.** Limited by the space, we only report a subset here; the full results are reported in Table 8. The subset covers video quality, depth and reconstruction, and camera estimation on four benchmarks.

Dataset	Model	Video Quality		Depth & Reconstruction		Camera	
		SC ↑	Cap.Sim ↑	AbsRel ↓	Chamfer-L1 ↓	RPE-R ↓	RRA@5 ↑
HoiGen	**Ours**	**0.9035**	0.6854	**0.1125**	0.3325	**0.1240**	**0.9809**
	TesserAct	0.8661	0.6401	1.850	0.3534	0.1429	0.9256
	Gen3R	0.7765	0.6803	0.2881	**0.2286**	5.060	0.2689
	VIST3A	0.6721	**0.7851**	0.6301	0.7633	13.09	0.0185
Epic-Kitchens-100	**Ours**	**0.9506**	**0.4010**	**0.0766**	**0.2321**	0.1568	**1.0000**
	TesserAct	0.8923	0.3343	0.4201	1.3321	**0.0946**	0.9736
	Gen3R	0.7813	0.3305	0.3641	0.3390	2.3992	0.3510
	VIST3A	0.5499	0.3710	0.1326	6.4722	34.7905	0.0184
Dynamic Replica	**Ours**	**0.9356**	0.5639	**0.0696**	**0.2579**	**0.1550**	**1.000**
	TesserAct	0.8490	0.4419	1.289	0.4067	0.7609	0.6772
	Gen3R	0.6833	0.4620	0.2018	1.034	22.47	0.1393
	VIST3A	0.6013	**0.5744**	0.4084	1.474	16.31	0.1230
Point Odyssey	**Ours**	**0.9233**	**0.5579**	**0.0822**	**0.3013**	**0.1985**	**0.9864**
	TesserAct	0.9019	0.5094	0.6347	0.3904	0.8804	0.6446
	Gen3R	0.7486	0.4930	0.4227	0.6703	9.409	0.2188
	VIST3A	0.6270	0.5465	0.4371	1.347	6.953	0.2962

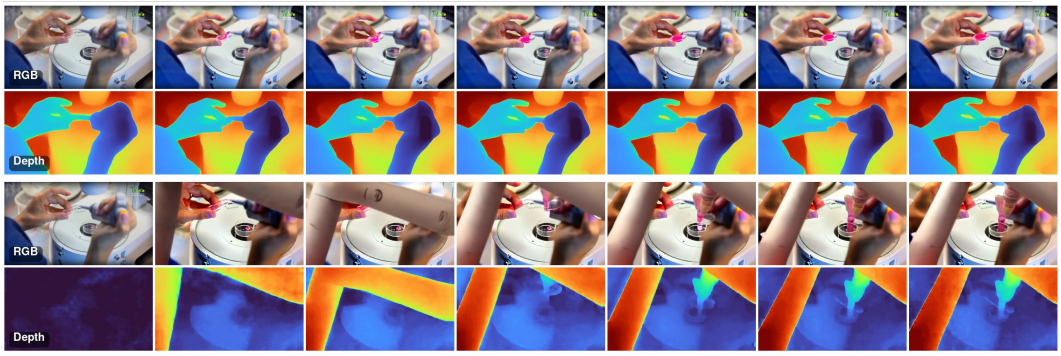

Figure 5: **Qualitative comparison between PLAN4D and TesserAct [117] for action-conditioned RGB-D world generation.** The input consists of the first RGB frame and the action text: *Hold the small glass dish. Dispense pink liquid using a pipette. Release the liquid smoothly. Fill the dish gradually.*. The first two rows show our results, and the lower rows show TesserAct.

Camera metrics show the largest gap in global 4D consistency. PLAN4D obtains the best RRA@5 on all four datasets in the subset and the best RPE-R on HoiGen, Dynamic Replica, and PointOdyssey. On Epic-Kitchens-100, it leads on both relative-rotation accuracy thresholds. TesserAct has the lowest RPE-R on Epic-Kitchens-100, while PLAN4D maintains strong performance across datasets and metric families. The Epic-Kitchens-100 results complement HoiGen with quantitative evidence across two real-world human-object interaction benchmarks, supporting the applicability of PLAN4D to everyday interactions beyond the original evaluation setting. The qualitative comparisons in Figure 5

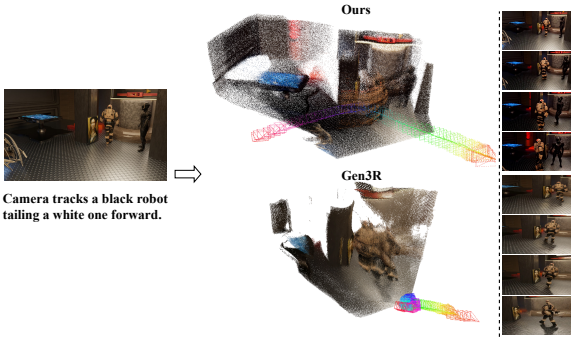

Figure 6: **Qualitative comparison between PLAN4D and Gen3R [37] for dynamic 4D world generation.** The middle panels show the global dynamic point-cloud reconstruction, and the right panels show the corresponding RGB frames.

Table 4: **Ablation of the joint-training objective.** We ablate RGB-only, geometry-only, and full objectives on HoiGen and Dynamic Replica under the same action-conditioned 4D generation protocol. SC and BC denote VBench subject and background consistency. Arrows indicate the preferred direction, and the best and second-best results within each dataset are highlighted.

Dataset	Model	Video Quality		Depth		Camera
		SC ↑	BC ↑	AbsRel ↓	δ_1 ↑	RPE-R ↓
HoiGen	Only-RGB-Loss	0.8889	0.9272	0.1559	0.7769	0.4010
	Only-Geo-Loss	0.6198	0.8184	**0.0745**	**0.9391**	0.3232
	Ours (Full)	**0.9035**	**0.9334**	0.1125	0.8540	**0.1240**
Dynamic Replica	Only-RGB-Loss	0.8998	0.9321	0.0874	0.9169	0.2615
	Only-Geo-Loss	0.6009	0.8459	0.1733	0.7077	0.7698
	Ours (Full)	**0.9356**	**0.9496**	**0.0696**	**0.9306**	**0.1550**

and Figure 6 show the same pattern visually: the generated RGB frames, RGB-D structure, and global dynamic point clouds remain better aligned under complex action-conditioned rollouts. These results support the core design choice of supervising a pretrained video generator with a frozen geometric readout rather than learning dense 4D structure from scratch.

4.3 Ablation Studies

We conduct an ablation study to understand the contribution of each component in our joint-training objective. The results are summarized in Table 4, with results on HoiGen and Dynamic Replica using the same action-conditioned generation protocol. The RGB-only variant retains strong subject and background consistency, but its depth and camera estimates are less accurate than those of the full model on both datasets. The geometry-only variant improves depth on HoiGen, but sacrifices visual consistency and performs worse across all reported metrics on Dynamic Replica. The full objective gives the best subject and background consistency and camera estimation on both datasets, and also leads on depth on Dynamic Replica. These results support combining flow matching with geometric supervision to preserve action-conditioned video quality while improving the depth and camera structure decoded from the same future latent state. Additional settings and analysis appear in Section C.1.

5 Conclusion

We introduce the problem of *generative plannable 4D worlds*, where a model must generate future scenes with rich 4D attributes, including RGB appearance, depth, camera poses, point clouds, and scene flow, from observed context and action descriptions. To this end, we propose PLAN4D, a unified framework that decouples the acquisition of strong geometric priors from the generation of future dynamics, then reconnects them through a stitched latent world model. By augmenting frozen DA3 with a token-only scene-flow head and learning a latent-to-geometry stitching interface from video VAE latents to geometry tokens, PLAN4D turns a pretrained video generator into an actionable 4D world model whose RGB, depth, camera rays, point clouds, and scene flow are read out from the same evolving latent state in a single feed-forward pass. Experiments show that this design preserves strong geometric behavior while improving dense 4D world generation across scene flow, reconstruction, and action-conditioned future prediction, without requiring manual 4D annotations for action videos. We hope this perspective helps move future world modeling beyond video-only prediction toward unified, measurable, and plannable 4D world models.

Acknowledgement

This research was done while Hangwei Zhang was an intern at AMAP, Alibaba Group. This research is supported by the NTU SUG-NAP and National Research Foundation, Singapore, under its NRF Fellowship Award NRF-NRFF17-2025-0009, and the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-PhD-2026-08-079).

References

[1] Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

[2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[3] Zhaochong An, Orest Kupyn, Théo Uscidda, Andrea Colaco, Karan Ahuja, Serge Belongie, Mar Gonzalez-Franco, and Marta Tintore Gazulla. Vggrpo: Towards world-consistent video generation with 4d latent reward. *arXiv preprint arXiv:2603.26599*, 2026.

[4] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[5] Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B Lindell, Zan Gojcic, Sanja Fidler, et al. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. *arXiv preprint arXiv:2509.19296*, 2025.

[6] Yunpeng Bai, Shaoheng Fang, Chaohui Yu, Fan Wang, and Qixing Huang. Geovideo: Introducing geometric regularization into video generation model. *arXiv preprint arXiv:2512.03453*, 2025.

[7] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021.

[8] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

[9] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236, 2021.

[10] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *CVPR*, pages 15791–15801, 2025.

[11] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[12] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023.

[13] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Int. Conf. Mach. Learn.*, 2024.

[14] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

[15] Delong Chen, Tejaswi Kasarla, Yejin Bang, Mustafa Shukor, Willy Chung, Jade Yu, Allen Bolourchi, Theo Moutakanni, and Pascale Fung. Action100m: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026.

[16] Rui Chen, Lei Sun, Jing Tang, Geng Li, and Xiangxiang Chu. Finger: Content aware fine-grained evaluation with reasoning for ai-generated videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3517–3526, 2025.

[17] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *CVPR*, 2025.

[18] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, pages 13320–13331, 2024.

[19] Xuweiyi Chen, Wentao Zhou, and Zezhou Cheng. Wildrayzer: Self-supervised large view synthesis in dynamic environments. *arXiv preprint arXiv:2601.10716*, 2026.

[20] Yabo Chen, Yuanzhi Liang, Jiepeng Wang, Tingxi Chen, Junfei Cheng, Zixiao Gu, Yuyang Huang, Zicheng Jiang, Wei Li, Tian Li, et al. Teleworld: Towards dynamic multimodal synthesis with a 4d world model. *arXiv preprint arXiv:2601.00051*, 2025.

[21] 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.

[22] Yixiang Dai, Fan Jiang, Chiyu Wang, Mu Xu, and Yonggang Qi. Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. In *ICLR*, 2026. URL `https://openreview.net/forum?id=3q9vHEqsNx`.

[23] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020.

[24] Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. In *IEEE Int. Conf. Robot. Autom.*, 2026.

[25] DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, Ruimin Lin, Qingfeng Shi, Bingze Song, Lei Sun, Jing Tang, Ruitian Tian, Jun Wang, Jiahong Wu, Pengfei Zhang, Shen Zhang, and Jiashu Zhu. DreamX-World 1.0: A general-purpose interactive world model. *arXiv preprint arXiv:2606.16993*, 2026.

[26] Hongyang Du, Junjie Ye, Xiaoyan Cong, Runhao Li, Jingcheng Ni, Aman Agarwal, Zeqi Zhou, Zekun Li, Randall Balestriero, and Yue Wang. Videogpa: Distilling geometry priors for 3d-consistent video generation. *arXiv preprint arXiv:2601.23286*, 2026.

[27] Shaoheng Fang, Hanwen Jiang, Yunpeng Bai, Niloy J Mitra, and Qixing Huang. Worldreel: 4d video generation with consistent geometry and motion modeling. *arXiv preprint arXiv:2512.07821*, 2025.

[28] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *NeurIPS*, 2024.

[29] Hyojun Go, Dominik Narnhofer, Goutam Bhat, Prune Truong, Federico Tombari, and Konrad Schindler. Text-to-3d by stitching a multi-view reconstruction network to a video generator. In *ICLR*, 2026. URL `https://openreview.net/forum?id=kI27Niy4xY`.

[30] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, pages 19383–19400, 2024.

[31] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. In *CVPR*, 2022.

[32] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022.

[33] Qiwen Gu, Jifan Li, Bingjie Gao, Rui Chen, Jing Tang, Xiangxiang Chu, and Junqiao Zhao. The right future for action: Learning action-relevant predictive states in world action models. *arXiv preprint arXiv:2609.23369*, 2026.

[34] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. URL `https://openreview.net/forum?id=Fx2SbBgcte`.

[35] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 35:8633–8646, 2022.

[36] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[37] Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3r: 3d scene generation meets feed-forward reconstruction. In *CVPR*, 2026.

[38] Nan Huang, Pengcheng Yu, Weijia Zeng, James M Rehg, Angjoo Kanazawa, Haiwen Feng, and Qianqian Wang. Self-improving 4d perception via self-distillation. *arXiv preprint arXiv:2604.08532*, 2026.

[39] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] 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. *arXiv preprint arXiv:2506.23126*, 2025.

[41] 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.

[42] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[43] Team HY-World, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026.

[44] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020.

[45] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2): 3019–3026, 2020.

[46] Wooseok Jang, Seonghu Jeon, Jisang Han, Jinhyeok Choi, Minkyung Kwon, Seungryong Kim, Saining Xie, and Sainan Liu. Repurposing geometric foundation models for multi-view diffusion. *arXiv preprint arXiv:2603.22275*, 2026.

[47] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. In *ICCV*, 2025.

[48] Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, et al. World and human action models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025.

[49] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023.

[50] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

[51] Marius Kästingschäfer, Théo Gieruc, Sebastian Bernhard, Dylan Campbell, Eldar Insafutdinov, Eyvaz Najafli, and Thomas Brox. Seed4d: A synthetic ego-exo dynamic 4d data generator, driving dataset and benchmark. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 7752–7764. IEEE, 2025.

[52] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. In *Int. Conf. 3D Vision*. IEEE, 2026.

[53] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[54] Akshay Krishnan, Xinchen Yan, Vincent Casser, and Abhijit Kundu. Orchid: Image latent diffusion for joint appearance and geometry generation. In *ICCV*, pages 28217–28227, 2025.

[55] Yushi Lan, Zeren Jiang, Kelvin Li, Xingang Pan, Chuanxia Zheng, and Andrea Vedaldi. Track4d: Representing dense 3d tracking for video diffusion models. In *NeurIPS*, 2026.

[56] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[57] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, pages 71–91. Springer, 2024.

[58] Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *ICCV*, pages 25690–25699, 2025.

[59] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *RSS*, 2025.

[60] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in neural information processing systems*, 34:7838–7851, 2021.

[61] Yanxun Li, Hao Wen, Bingze Song, Jiashu Zhu, Aiming Hao, Chubin Chen, Jintao Chen, Jiahong Wu, Xiangxiang Chu, and Miao Wang. Learning explicit physical parameter control and benchmarking for video generation. *arXiv preprint arXiv:2607.18924*, 2026.

[62] Yijin Li, Yichen Shen, Zhaoyang Huang, Shuo Chen, Weikang Bian, Xiaoyu Shi, Fu-Yun Wang, Keqiang Sun, Hujun Bao, Zhaopeng Cui, et al. Blinkvision: A benchmark for optical flow, scene flow and point tracking estimation using rgb frames and events. In *European conference on computer vision*, pages 19–36. Springer, 2024.

[63] Yiqing Liang, Abhishek Badki, Hang Su, James Tompkin, and Orazio Gallo. Zero-shot monocular scene flow estimation in the wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21031–21044, 2025.

[64] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

[65] Xinran Ling, Chen Zhu, Meiqi Wu, Hangyu Li, Xiaokun Feng, Cundian Yang, Aiming Hao, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. Vmbench: A benchmark for perception-aligned video motion generation. *arXiv preprint arXiv:2503.10076*, 2025.

[66] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. *arXiv preprint arXiv:2511.04675*, 2025.

[67] Kun Liu, Qi Liu, Xinchen Liu, Jie Li, Yongdong Zhang, Jiebo Luo, Xiaodong He, and Wu Liu. Hoigen-1m: A large-scale dataset for human-object interaction video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24001–24010, 2025.

[68] Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802*, 2025.

[69] Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation. *arXiv preprint arXiv:2507.01099*, 2025.

[70] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9263–9274, 2025.

[71] Haoran Lu, Shang Wu, Jianshu Zhang, Maojiang Su, Guo Ye, Chenwei Xu, Lie Lu, Pranav Maneriker, Fan Du, Manling Li, et al. Phys4d: Fine-grained physics-consistent 4d modeling from video diffusion. *arXiv preprint arXiv:2603.03485*, 2026.

[72] Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4world: Feedforward world-centric dense 3d tracking of all pixels. *arXiv preprint arXiv:2603.02573*, 2026.

[73] Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

[74] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[75] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16102–16112, 2023.

[76] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia

Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. URL https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

[77] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[78] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[79] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017.

[80] Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

[81] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023. URL https://openreview.net/forum?id=nJfylDvgzlq.

[82] Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. In *CVPR*, 2026.

[83] Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, et al. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

[84] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in neural information processing systems*, 36:1363–1389, 2023.

[85] 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*, October 2025.

[86] DreamX Team, Rui Chen, Xiangxiang Chu, Geng Li, Jifan Li, Qingfeng Shi, Datao Tang, Jing Tang, Jun Wang, and Pengfei Zhang. Dreamx-phi 1.0: Action-conditioned video world model for robotic manipulation. *arXiv preprint arXiv:2608.13489*, 2026.

[87] Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

[88] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, pages 8375–8384, 2021.

[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[90] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[91] Hanyang Wang, Fangfu Liu, Jiawei Chi, and Yueqi Duan. Videoscene: Distilling video diffusion model to generate 3d scenes in one step. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16475–16485. IEEE, 2025.

[92] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025.

[93] Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω. *arXiv preprint arXiv:2605.15195*, 2026.

[94] JiYuan Wang, Chunyu Lin, Lei Sun, Rongying Liu, Lang Nie, Mingxing Li, Kang Liao, Xiangxiang Chu, and Yao Zhao. From editor to dense geometry estimator. *arXiv preprint arXiv:2509.04338*, 2025.

[95] Jiyuan Wang, Chunyu Lin, Lei Sun, Zhi Cao, Yuyang Yin, Lang Nie, Zhenlong Yuan, Xiangxiang Chu, Yunchao Wei, Kang Liao, et al. Edit in 2d, verify in 3d: Reinforcement learning for multi-view consistent scene editing. In *European Conference on Computer Vision*, pages 260–278. Springer, 2026.

[96] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, June 2025.

[97] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024.

[98] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023.

[99] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Scalable permutation-equivariant visual geometry learning. In *ICLR*, 2026.

[100] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *ICLR*, 2026. URL `https://openreview.net/forum?id=ULXYZCms41`.

[101] Meiqi Wu, Zhixin Cai, Fufangchen Zhao, Xiaokun Feng, Rujing Dang, Bingze Song, Ruitian Tian, Jiashu Zhu, Jiachen Lei, Hao Dou, Jing Tang, Lei Sun, Jiahong Wu, Xiangxiang Chu, Zeming Liu, and Kaiqi Huang. Omni-WorldBench: Towards a comprehensive interaction-centric evaluation for world models. *arXiv preprint arXiv:2603.22212*, 2026.

[102] Meiqi Wu, Bingze Song, Ruimin Lin, Chen Zhu, Xiaokun Feng, Jiahong Wu, Xiangxiang Chu, and Kaiqi Huang. Latent temporal discrepancy as motion prior: A loss-weighting strategy for dynamic fidelity in t2v. *arXiv preprint arXiv:2601.20504*, 2026.

[103] Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. Genfusion: Closing the loop between reconstruction and generation via videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6078–6088, 2025.

[104] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[105] Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4d: Kinematic 4d world modeling for spatiotemporal embodied simulation. *arXiv preprint arXiv:2603.16669*, 2026.

[106] 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*, 2024. URL `https://openreview.net/forum?id=sFyTZEqmUY`.

[107] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[108] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

[109] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[110] Tengjiao Yin, Jinglei Shi, Heng Guo, and Xi Wang. Vigor: Video geometry-oriented reward for temporal generative alignment. *arXiv preprint arXiv:2603.16271*, 2026.

[111] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[112] Chengbo Yuan, Geng Chen, Li Yi, and Yang Gao. Self-supervised monocular 4d scene reconstruction for egocentric videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8863–8874, 2025.

[113] 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*, 2025.

[114] Hangwei Zhang, Armando Fortes, Tianyi Wei, and Xingang Pan. Boosting monocular metric depth estimation via bokeh rendering. In *International Conference on Machine Learning*, pages 155450–155481. PMLR, 2026.

[115] Hangwei Zhang, Nathan Rouillé, and Chuanxia Zheng. Corrflow: Self-correcting flow matching. *arXiv preprint arXiv:2610.24750*, 2026.

[116] 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. In *ICLR*, 2025.

[117] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

[118] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, pages 19855–19865, 2023.

[119] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025.

[120] Chen Zhu, Jiashu Zhu, Yanxun Li, Meiqi Wu, Bingze Song, Chubin Chen, Jiahong Wu, Xiangxiang Chu, and Yangang Wang. Artifact-aware evaluation for high-quality video generation. *arXiv preprint arXiv:2601.20297*, 2026.

[121] Ruijie Zhu, Jiahao Lu, Wenbo Hu, Xiaoguang Han, Jianfei Cai, Ying Shan, and Chuanxia Zheng. Motioncrafter: Dense geometry and motion reconstruction with a 4d vae. In *CVPR*, 2026.

A Frozen DA3 Tokens Already Encode Cross-Frame Motion Cues

The lightweight DPT scene flow head above is only useful if the frozen DA3 encoder tokens already preserve motion-relevant structure. The central question of this section is therefore simple: before any scene-flow supervision reaches the decoder, do frozen DA3 tokens already know which location in frame $t+1$ corresponds to a location in frame t? If the answer is yes, then a shallow head can decode dense motion by reading out a representation that already contains cross-frame geometry and correspondence cues. If the answer is no, then good scene-flow predictions would have to come from the capacity of the head alone.

To make this question concrete, let $z_t(p)$ denote the frozen DA3 token at spatial cell p in frame t. Point Odyssey [118] provides ground-truth tracks, which let us identify the corresponding target cell p^+ in frame $t+1$ for each valid source cell p. We then probe the token space without using the scene flow head. Our analysis uses the Point Odyssey test split with 128 clips from 5 scenes. All values in Table 5 are weighted over clips by the number of valid token pairs or valid flow tokens. Figure 7 aggregates all 6,132,636 positive pairs, 6,132,636 negative pairs, and 2,160,723 token-difference vectors.

We use four complementary probes, each answering a different question about the frozen representation. The first probe is **correspondence similarity** [84]. It asks whether a token is more similar to its true match than to an unrelated token from the next frame:

$$s_{\mathrm{pos}} = \cos\big(z_t(p), z_{t+1}(p^+)\big), \qquad s_{\mathrm{neg}} = \cos\big(z_t(p), z_{t+1}(q^-)\big), \tag{6}$$

where q^- is a random negative sampled from frame $t+1$. This is the cleanest test of whether correspondence information is present [44]. If frozen DA3 tokens contain motion-relevant structure, the s_{pos} distribution should shift to the right of the s_{neg} distribution. The margin $\Delta s = s_{\mathrm{pos}} - s_{\mathrm{neg}}$ and the probability $\Pr(s_{\mathrm{pos}} > s_{\mathrm{neg}})$ summarize how reliably the two distributions separate.

The second probe is **token retrieval** [74]. Instead of comparing one positive and one negative, we search over all target tokens in frame $t+1$:

$$\hat{p} = \arg\max_{q \in \mathcal{P}_{t+1}} \cos\big(z_t(p), z_{t+1}(q)\big). \tag{7}$$

This is a stronger operational test. It asks whether nearest-neighbor search in frozen token space can recover the correct destination cell. We report *Top-1 exact*, which measures whether the retrieved cell is exactly the ground-truth cell, and *PCK@1* and *PCK@3*, which measure whether retrieval falls within 1 or 3 token cells of the ground-truth location [107]. High retrieval accuracy means that the token space is not only separable on average, but also organized enough to support direct correspondence lookup.

The third probe is a **token-flow readout**. After retrieval returns $\hat{p}$, we interpret it as a token-level motion estimate $\hat{d} = \hat{p} - p$ and compare it with the ground-truth token displacement. This probe is more demanding than similarity or retrieval because it asks a single nearest-neighbor lookup to act as a motion predictor. The resulting endpoint error and direction cosine should be interpreted as evidence for how directly motion can be read out from the frozen tokens without a learned decoder [104].

The fourth probe is **PCA on token differences**. For each matched pair, we form

$$\Delta z_t(p) = z_{t+1}(p^+) - z_t(p). \tag{8}$$

We then perform PCA on these token differences [78]. This analysis does *not* test whether motion is present at all. The earlier probes already answer that question more directly. Instead, PCA asks a more specific question: is the motion signal concentrated in a small linear subspace, or is it distributed across many token dimensions? If the first few principal components explain most of the variance, then motion could be close to linearly disentangled [2]. If the variance remains spread out, then motion is still present, but encoded in a more distributed form. This distinction matters for interpreting the role of the DPT head. A low-dimensional linear code would be easy to read out with almost any linear probe, while a distributed code benefits from structured multiscale decoding.

Before discussing the numbers, it is useful to clarify how they should be interpreted. Similarity and retrieval are the strongest probes in this section because they make the fewest modeling assumptions. They only ask whether true cross-frame matches are already nearby in frozen token space. Token flow is a harder probe because it asks nearest-neighbor retrieval to behave like a motion estimator.

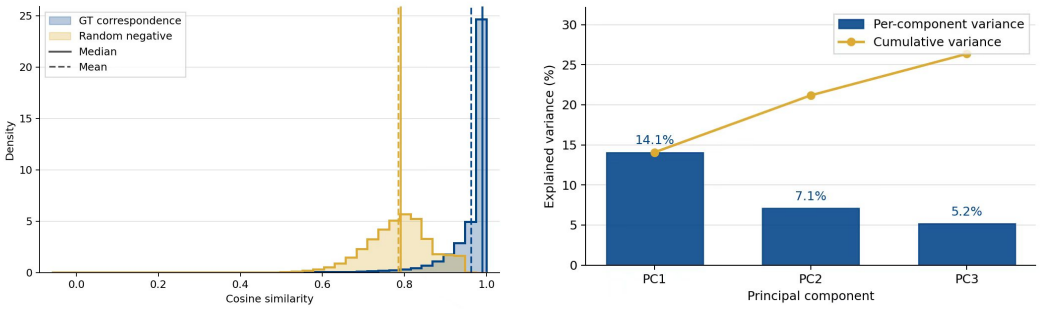

Figure 7: Aggregated token-space evidence on the Point Odyssey test split. Left: cosine similarity distributions computed from frozen DA3 tokens. Right: explained variance of the top principal components of token differences $\Delta z_t = z_{t+1} - z_t$, aggregated over all test clips.

Table 5: Dataset-level evidence from frozen DA3 tokens on the Point Odyssey test split. Each block should be read differently. Similarity measures token-level separability between true matches and negatives. Retrieval measures whether nearest-neighbor search recovers the correct target cell. Token flow measures how well that retrieval can be turned into a direct motion estimate. PCA measures whether motion is concentrated in a small linear subspace.

Category	Metric	Value
Similarity	Mean GT similarity s_{pos}	0.962
Similarity	Mean negative similarity s_{neg}	0.784
Similarity	Mean gap Δs	0.178
Similarity	$\Pr(s_{\text{pos}} > s_{\text{neg}})$	96.98%
Retrieval	Top-1 exact	69.65%
Retrieval	PCK@1 token	86.01%
Retrieval	PCK@3 token	95.66%
Retrieval	Mean token error (tokens)	0.57
Token flow	EPE (tokens)	0.62
Token flow	Direction cosine	0.331
Token flow	PCK@1 token	83.35%
PCA	PC1 explained variance	14.1%
PCA	PC2 explained variance	7.1%
PCA	PC3 explained variance	5.2%
PCA	Top-3 cumulative variance	26.4%

PCA is the most indirect probe because a failure to find a low-dimensional linear motion axis does not mean that motion information is absent. It only means that the information is not cleanly isolated in the top principal components.

Figure 7 and Table 5 provide direct evidence that frozen DA3 tokens preserve cross-frame correspondence structure. The similarity distributions are cleanly separated, with ground-truth matches consistently closer in token space than random negatives. Nearest-neighbor retrieval leads to the same conclusion under a stronger all-candidates test. These two probes show that, even without any learned motion decoder, the frozen representation already places true future matches near the source token and usually retrieves a target very close to the correct location.

The token-flow probe refines this picture. A simple nearest-neighbor displacement readout already recovers meaningful motion, which indicates that the frozen tokens encode where points are likely to move. At the same time, this probe is weaker than similarity and retrieval, which suggests that the motion signal is not exposed as a trivial one-step velocity code. This is precisely where a structured decoder becomes useful. Our DPT head can pool multiscale context and impose spatial coherence to transform this latent correspondence signal into dense scene flow.

The PCA results should be read as auxiliary evidence. The dominant components explain only a limited portion of the token-difference variance, and their alignment with flow magnitude remains weak. This does not argue against motion information in the frozen tokens. Instead, PCA indicates that motion is encoded in a distributed high-dimensional form rather than concentrated in a single linear direction.

Table 6: Stage-wise training datasets for the 81-frame PLAN4D pipeline. Counts are the effective rows used in each stage. Stage 1 keeps only clips with `scene_flow_available=true`, while Stage 2 uses the same GT-capable manifests to build a GT branch and a distillation branch, so one hybrid epoch traverses the full mixture once under GT supervision and once under teacher-student supervision.

Stage	Dataset	Clips	Primary Supervision
1	PointOdyssey train	2,619	GT scene flow
1	Dynamic Replica train	3,864	GT scene flow
1	BlinkVision train	223	GT scene flow
1	Virtual KITTI 2 train	580	GT scene flow
1	OmniWorld train	4,670	GT scene flow
2	PointOdyssey	3,040	GT geometry + flow
	BlinkVision	241	GT geometry + flow
	Dynamic Replica	3,864	GT geometry + flow
	Virtual KITTI 2	580	GT geometry + flow
	OmniWorld-Game	5,796	GT geometry + flow
	MVS-Synth	240	GT geometry only
	ScanNet++	108,464	GT geometry only
3	Action100M	249,422	Distill only

These results support a precise interpretation of our method. Frozen DA3 encoder tokens already contain scene-flow-relevant information, especially in the form of cross-frame correspondence structure. The strongest evidence comes from the probes that require almost no decoder assumptions, namely similarity separation and nearest-neighbor retrieval. The weaker token-flow and PCA probes refine this picture. They suggest that the motion signal is real, but distributed and not trivially linearized. Our DPT head succeeds because it decodes and regularizes this latent signal into dense metric predictions. In other words, the scene flow branch acts as an informed readout of the pretrained geometry representation, rather than a separate module that has to create motion information from scratch.

B Training Details

B.1 Dataset and Optimization Summary

All dataset statistics in this paper correspond to the 81-frame manifests. Every manifest row therefore satisfies the temporal constraint $T = 4k + 1$ with $T = 81$. Table 6 summarizes the stage-wise training data. Stage 1 trains only the DA3 scene-flow readout. Stage 2 finetunes the stitched decoder with hybrid ground-truth and teacher-student supervision. Stage 3 optimizes the action-conditioned video model, where Action100M contributes text-conditioned video clips without paired 4D labels. Stage 1 mixes 11,956 training clips from five dynamic datasets. The sampler uses dataset-wise mixture weights of 1.0 together with a size-aware mixing exponent $\alpha = 0.5$, which interpolates between uniform-over-datasets and proportional-to-size sampling. The stage-1 manifest directory also contains held-out or diagnostic subsets such as PointOdyssey test, PointOdyssey sample, and Dynamic Replica valid clips for validation and smoke tests.

Stage 2 contains 122,225 GT-capable training clips. Among them, 13,521 clips expose scene-flow labels, while the remaining 108,704 clips provide only depth, ray, camera, and point-cloud supervision. In hybrid decoder training, we build separate GT and distill dataloaders from the same manifest list and randomly shuffle the union of GT and distill steps once per epoch. This realizes one GT pass and one distillation pass over the full mixture in a single hybrid epoch.

The stage-3 Action100M subset is constructed in two steps. We first start from the filtered 81-frame TI2V manifest `action100m_train_ti2v_81f.filtered.jsonl`, whose rows already satisfy `task_type=ti2v`, `cond_frames=1`, and `num_frames=81`. We then score the corresponding clip-level entries from `action100m_clip_index_81f.filtered.jsonl` and keep only clips that pass a high-precision action-quality filter. The clip-level decision uses fields such as

`prompt_action`, `level`, and `node_duration_sec`. We drop clips with node duration shorter than 1.6 seconds, hierarchy depth above 10, missing action text, action text identical to the generic description, action text shorter than four words, prompt patterns dominated by on-screen text or audio-style narration, uncertain descriptions, or generic action stubs. The retained manifest `action100m_train_ti2v_81f.action_hq.keep.jsonl` contains 249,422 clips, which is 28.6% of the 871,785 filtered TI2V rows. Because Action100M does not provide paired 4D geometry, these clips are used only in the distillation branch during stage 3.

Per-dataset roles. PointOdyssey provides articulated outdoor motion with sparse track supervision that we rasterize into dense forward scene flow, which makes it particularly useful for supervising correspondence-aware motion. Dynamic Replica contributes denser dynamic stereo geometry with strong camera and scene-flow signals. BlinkVision, Virtual KITTI 2, and OmniWorld broaden the synthetic motion distribution across indoor scenes, driving scenes, and open-world gameplay trajectories. ScanNet++ and MVS-Synth do not provide scene flow in our 81-frame setup, but they substantially enrich multiview depth, camera, and point-cloud supervision during stitched-decoder finetuning. Action100M serves a different purpose from the geometry datasets. It injects large-scale action-conditioned video semantics into the joint model while relying on the frozen DA3 teacher for geometric targets.

Evaluation datasets. We report three groups of held-out evaluations. For dynamic 4D motion, we evaluate on MOVi-F, Dynamic Replica, and PointOdyssey using 81-frame clips. MOVi-F tests controlled synthetic object motion, Dynamic Replica tests realistic dynamic stereo scenes, and PointOdyssey emphasizes articulated motion with long temporal support. For stitched-decoder geometry preservation, we evaluate on ETH3D, ScanNet++, Dynamic Replica, Dynamic-RE10K, MOVi-F, and RobbySimVal. These datasets cover static reconstruction, indoor multiview geometry, dynamic camera estimation, and robot-centric depth prediction. For action-conditioned world generation, we evaluate on RLBench with 210 samples, HOIGen-1M with 200 samples, Dynamic Replica with 480 samples, and PointOdyssey with 375 samples, which together span manipulation, hand-object interaction, and general dynamic-scene futures. We additionally evaluate 100 five-second Epic-Kitchens-100 clips to assess in-the-wild generalization to everyday human-object interactions in kitchen environments. This evaluation complements HoiGen with a second real-world human-object interaction benchmark.

Stage-3 comparison scope. We exclude several recent world-generation systems from the stage-3 comparison because they assume additional control inputs beyond the setting targeted in this paper. PointWorld requires depth input [41]. FantasyWorld, HY-World 2.0, Lyra 2.0, Geometry Forcing, and TeleWorld rely on trajectory-level control signals [22, 43, 80, 100, 20]. Kinema4D further requires robot-action kinematic representations [105]. These methods therefore do not match the task of predicting future RGB video together with dense 4D geometry from only an observed scene context and a text-specified action, so we do not include them in the stage-3 comparison.

Evaluation protocol and metric definitions. We follow the same three-way evaluation split as in the main paper and expand the scoring protocol here. **Scene flow.** We score only valid moving pixels indicated by the deformation mask. Predictions are first rescaled to the target point-map scale of each clip. When a model predicts relative flow, we restore the target scale with the clip-level scene-flow scale before metric computation. If a benchmark provides native-resolution labels, we resize predictions to that evaluation resolution before scoring. We report

$$\mathrm{EPE} = \frac{1}{|\mathcal{V}_{\mathrm{sf}}|} \sum_{(t,p)\in\mathcal{V}_{\mathrm{sf}}} \left\|\hat{\mathbf{m}}_{t,p} - \mathbf{m}_{t,p}\right\|_2, \qquad APD_{0.05} = \frac{1}{|\mathcal{V}_{\mathrm{sf}}|} \sum_{(t,p)\in\mathcal{V}_{\mathrm{sf}}} \mathbb{1}\left[\left\|\hat{\mathbf{m}}_{t,p} - \mathbf{m}_{t,p}\right\|_2 < 0.05\right], \tag{9}$$

where $\mathcal{V}_{\mathrm{sf}}$ is the set of valid motion pixels, $\hat{\mathbf{m}}_{t,p}$ is the predicted 3D displacement from frame t to frame $t+1$, and $\mathbf{m}_{t,p}$ is the ground-truth displacement. Lower EPE is better, while higher $APD_{0.05}$ is better.

Depth. Depth metrics use per-clip scale-shift alignment. For each clip, we solve one scale a and one shift b jointly over all valid depth pixels and evaluate $a\hat{d} + b$ against ground-truth depth:

$$(a^\star, b^\star) = \arg\min_{a,b} \sum_{(t,p)\in\mathcal{V}_d} \left(a\hat{d}_{t,p} + b - d_{t,p}\right)^2. \tag{10}$$

We report

$$\mathrm{AbsRel} = \frac{1}{|\mathcal{V}_d|} \sum_{(t,p)\in\mathcal{V}_d} \frac{\left| a^\star \hat{d}_{t,p} + b^\star - d_{t,p} \right|}{d_{t,p}} \tag{11}$$

$$\delta_1 = \frac{1}{|\mathcal{V}_d|} \sum_{(t,p)\in\mathcal{V}_d} \mathbb{1}\left[\max\left(\frac{a^\star \hat{d}_{t,p} + b^\star}{d_{t,p}}, \frac{d_{t,p}}{a^\star \hat{d}_{t,p} + b^\star} \right) < 1.25 \right]. \tag{12}$$

Lower AbsRel is better, while higher δ_1 is better.

Camera. Camera metrics follow the official DA3 bench. Predicted and ground-truth camera trajectories are first converted to the first-camera gauge. We then evaluate all frame pairs (i, j) through their relative transforms. Let e^r_{ij} denote the relative rotation error in degrees and let e^t_{ij} denote the angle between the predicted and ground-truth relative translation directions. On Dynamic Replica we report

$$RRA@5^\circ = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \mathbb{1}\left[e^r_{ij} < 5^\circ \right], \tag{13}$$

which measures relative rotation accuracy over all frame pairs. On Dynamic-RE10K we report AUC_{15} and AUC_{30}. These metrics are computed from the per-pair error

$$m_{ij} = \max\left(e^r_{ij}, e^t_{ij} \right), \tag{14}$$

and summarize the cumulative accuracy curve of m_{ij} up to the corresponding angular threshold. Higher camera metrics are better.

3D reconstruction. Reconstruction also follows the official DA3 bench. We estimate a global Umeyama alignment between predicted and ground-truth camera trajectories and use the recovered scale to place predicted depth in metric units before TSDF fusion. When predicted poses are used for reconstruction, the same fit also aligns the fused camera trajectory. The fused prediction is then compared with the benchmark ground-truth geometry using the dataset-specific threshold and downsampling settings from the DA3 bench. ETH3D and ScanNet++ evaluate against points sampled from the ground-truth mesh, and ScanNet++ additionally crops the predicted point cloud to the ground-truth bounding box with a small margin before scoring. For predicted points P and ground-truth points G, we report

$$\mathrm{Acc.} = \frac{1}{|P|} \sum_{x\in P} \min_{y\in G} \|x - y\|_2, \qquad \mathrm{Comp.} = \frac{1}{|G|} \sum_{y\in G} \min_{x\in P} \|x - y\|_2. \tag{15}$$

Precision and recall are the fractions of predicted and ground-truth points whose nearest-neighbor distance is below the dataset threshold ϵ, and the F-score is their harmonic mean:

$$\mathrm{Prec.} = \frac{1}{|P|} \sum_{x\in P} \mathbb{1}\left[\min_{y\in G} \|x - y\|_2 < \epsilon \right], \tag{16}$$

$$\mathrm{Rec.} = \frac{1}{|G|} \sum_{y\in G} \mathbb{1}\left[\min_{x\in P} \|x - y\|_2 < \epsilon \right], \tag{17}$$

$$\text{F-score} = \frac{2\,\mathrm{Prec.}\,\mathrm{Rec.}}{\mathrm{Prec.} + \mathrm{Rec.}}. \tag{18}$$

Lower Acc. and Comp. are better, while higher F-score is better.

Joint world generation. For RGB video quality, higher video-caption similarity indicates stronger alignment between the generated video and the conditioning action text. RLBench and HOIGen-1M use their native action annotations, while Dynamic Replica and PointOdyssey do not provide action-level text in our stage-3 evaluation split, so we generate per-clip action captions with Qwen-VL-30B-Thinking and use them as the conditioning text for all compared methods. When a baseline exposes a compatible geometry output, we score its generated 4D structure with the same geometry protocol described above.

Epic-Kitchens-100 evaluation setting. We sample 100 clips, each five seconds long, and follow the same action-conditioned joint 4D generation protocol as the original comparison. Each method receives the initial RGB frame and an action description. We compare PLAN4D, TesserAct, Gen3R, and VIST3A using the same metric definitions and alignment procedures. The full evaluation reports six video metrics, namely subject consistency (SC), background consistency (BC), aesthetic quality (AQ), image quality (IQ), motion smoothness (MS), and video-caption similarity (Cap.Sim). Geometry is evaluated with depth AbsRel, δ_1, δ_2, and Chamfer-L1, together with camera RPE-R, RRA@10, and RRA@5. Higher scores are better except for AbsRel, Chamfer-L1, and RPE-R. The complete results appear in Table 8, and Table 3 retains six representative metrics for comparison with the other benchmarks.

Stage-wise optimization details. For stage 1, the 81-frame manifest configuration uses AdamW with learning rate 10^{-5}, weight decay 10^{-2}, $\beta = (0.9, 0.999)$, 100 warmup steps, gradient accumulation 2, and bf16 mixed precision. Inputs are resized to 504×504 with upper-bound resizing, the clip overlap is 1 frame, and the scene-flow loss adds a zero-last-frame penalty with weight 0.1. The main paper reports a 20-epoch run on 16 H100 GPUs with per-GPU batch size 1.

For stage 2, the stitched-decoder hybrid launchers use AdamW with learning rate 10^{-4}, weight decay 10^{-4}, bf16 mixed precision, 504×504 inputs, and per-GPU batch size 1. The representative 81-frame launchers use a 50-step warmup and train the stitching layer together with LoRA adapters on the stitched DA3 blocks by default, while still supporting full finetuning of the stitched parts. The main paper reports one hybrid epoch on 64 H20 GPUs.

For stage 3, the stitched joint LoRA launcher uses AdamW with learning rate 10^{-4}, cosine decay, 1000 warmup steps, LoRA rank 32 on the DiT, gradient checkpointing, 81 frames at 532×532 resolution, condition dropout 0.1, and geometry-loss warmup from 0.001 to 1.0 over the first 500 steps. We apply `mask_condition_loss` on conditioned prefixes and `geo_supervise_generated_only` on the generated suffix. The main paper reports one epoch on 64 H20 GPUs with per-GPU batch size 1.

B.2 Scene Flow Head Training

We train the scene-flow branch as a readout on top of a fixed DA3 representation. The DA3 backbone, depth head, ray head, and camera modules are loaded from the pretrained checkpoint and kept frozen. We inject the DPT-style scene-flow head and its coordinate adapter from the scene-flow preset, load the original DA3 weights with non-strict matching, and optimize only the parameters of the new head. During training, each clip is passed through the frozen DA3 encoder to extract the four multiscale token maps. The head decodes these tokens into a dense three-channel motion field, and the adapter rotates the local camera-frame prediction into the first-frame camera basis.

The training target is a forward displacement field from frame t to frame $t+1$. All datasets expose the same fields: RGB frames, camera intrinsics, world-to-camera extrinsics, a scene-flow tensor, a per-clip scale, and a binary deformation mask. We use relative scene flow by default. Metric displacements are expressed in the first-frame camera basis and normalized by a robust clip-level scale, which makes the head less sensitive to the absolute size of a scene.

Let $\hat{\mathbf{m}}_{t,p}$ be the predicted motion at pixel p in frame t, let $\mathbf{m}_{t,p}$ be the target, and let $a_{t,p} \in \{0, 1\}$ be the deformation mask. The scene-flow loss [60, 88] is

$$\mathcal{L}_{\text{sf}} = \frac{\sum_{t,p} a_{t,p} \left\| \hat{\mathbf{m}}_{t,p} - \mathbf{m}_{t,p} \right\|_1}{\max\left(1, \sum_{t,p} a_{t,p}\right)} + \lambda_{\text{last}} \frac{1}{HW} \sum_{p} \left\| \hat{\mathbf{m}}_{T,p} \right\|_2 . \tag{19}$$

The first term is evaluated only at valid motion pixels. It never falls back to full-image supervision, which is important for datasets where motion labels come from sparse tracks. The second term softly enforces the last frame to be zero because there is no target frame $T+1$. We use masked L1 loss in the main configuration.

B.3 Latent-to-Geometry Model Stitching Training

The stitching stage determines where the video latent space should enter DA3 and initializes the projection into that token space. For each training clip, we encode the RGB video with the frozen Wan

VAE and extract candidate DA3 token maps from encoder blocks. The Wan latent has 16 channels and is temporally compressed. Before fitting the stitch, we upsample it back to the video frame rate with $T = 4(T_{\text{vae}} - 1) + 1$ and resize it to the spatial resolution required by the DA3 patch grid.

For each candidate block k, we fit a 3D convolutional stitch $\mathcal{S}_\theta^{(k)}$ from the upsampled Wan latent to the frozen DA3 token grid. The fit is a streaming ridge regression problem shown in Equation (2).

The streaming solver accumulates normal-equation statistics over cached tensor batches, which avoids storing a large design matrix. We rank candidate blocks by the reconstruction mean squared error and choose the block with the lowest error. In our DA3 setting, the stitch maps the 16-channel Wan latent to the 1536-channel DA3 token width through a compact spatiotemporal convolution. This procedure gives both the injection layer and the initial weights for the later trainable stitched decoder.

B.4 Stitched Decoder Finetuning

The linear stitch is only an initialization. We next train a stitched decoder that accepts Wan latents but preserves the geometry behavior of DA3. The Wan VAE is frozen. The pretrained DA3 teacher is frozen. The trainable student contains the stitching module and stitched DA3 parts. The video diffusion transformer is not trained in this stage.

We use two complementary sources of supervision. In the distillation branch, the teacher consumes the ground-truth RGB clip, while the student consumes the Wan latent of the same clip. The teacher and student outputs are aligned in time and space before computing losses. The distillation objective is

$$\begin{aligned}\mathcal{L}_{\text{distill}} =& \lambda_D \left\| \hat{\mathbf{D}} - \mathbf{D}^{\mathrm{T}} \right\|_1 + \lambda_{\nabla D} \mathcal{L}_{\nabla D} \left(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{T}} \right) + \lambda_R \left\| \hat{\mathbf{R}} - \mathbf{R}^{\mathrm{T}} \right\|_1 \\ &+ \lambda_{cD} \left\| \hat{\mathbf{C}}^D - \mathbf{C}^{D,\mathrm{T}} \right\|_1 + \lambda_{cR} \left\| \hat{\mathbf{C}}^R - \mathbf{C}^{R,\mathrm{T}} \right\|_1 + \lambda_C \mathcal{L}_C \\ &+ \lambda_M \left\| \hat{\mathbf{M}} - \mathbf{M}^{\mathrm{T}} \right\|_1 + \lambda_T \mathcal{L}_T.\end{aligned} \tag{20}$$

Here $\mathbf{D}$ denotes depth, $\mathbf{R}$ denotes the six-channel ray map, $\mathbf{C}^D$ and $\mathbf{C}^R$ denote confidence maps, and $\mathbf{M}$ denotes scene flow. The camera term $\mathcal{L}_C$ compares decoded intrinsics and relative camera motion. The temporal term $\mathcal{L}_T$ matches depth differences between adjacent frames, which discourages frame-wise geometry flicker.

In the ground-truth branch, the student is supervised against dataset depth, ray, camera, valid-mask, and optional scene-flow annotations. The objective is

$$\begin{aligned}\mathcal{L}_{\text{GT}} =& \lambda_D \mathcal{L}_D + \lambda_{\nabla D} \mathcal{L}_{\nabla D} + \lambda_R \mathcal{L}_R + \lambda_P \mathcal{L}_P \\ &+ \lambda_C \mathcal{L}_C + \lambda_M \mathcal{L}_M + \lambda_T \mathcal{L}_T.\end{aligned} \tag{21}$$

The depth loss follows DA3's confidence-aware form. For sample b, let γ_b be the common scene scale computed from the valid target point map, and let Ω_b be the valid pixel set over all frames. The depth prediction is first aligned to the target camera scale, producing $\tilde{D}$. We then use

$$\mathcal{L}_D = \frac{1}{B} \sum_b \frac{1}{\gamma_b} \frac{1}{|\Omega_b|} \sum_{(t,p) \in \Omega_b} \left[\hat{c}^D_{b,t,p} \left| \tilde{D}_{b,t,p} - D^\star_{b,t,p} \right| - \beta_D \log \hat{c}^D_{b,t,p} \right]_+, \tag{22}$$

where $\hat{c}^D \geq 1$ is the predicted depth confidence and $[\cdot]_+$ clamps the per-pixel contribution to be non-negative. The confidence term allows the model to express uncertainty, while the logarithmic regularizer prevents the trivial low-confidence solution.

The depth-gradient loss supervises local geometry changes rather than absolute depth alone,

$$\mathcal{L}_{\nabla D} = \frac{1}{B} \sum_b \frac{1}{\gamma_b} \frac{1}{|\Omega_b^x| + |\Omega_b^y|} \left(\sum_{(t,p) \in \Omega_b^x} \left| \nabla_x \tilde{D}_{b,t,p} - \nabla_x D^\star_{b,t,p} \right| + \sum_{(t,p) \in \Omega_b^y} \left| \nabla_y \tilde{D}_{b,t,p} - \nabla_y D^\star_{b,t,p} \right| \right), \tag{23}$$

where Ω_b^x and Ω_b^y contain neighboring valid pixels along the horizontal and vertical directions. This term preserves depth discontinuities and discourages over-smoothed surfaces.

Each ray map is represented as a direction-origin pair $\mathbf{R}_{t,p} = (\mathbf{d}_{t,p}, \mathbf{o}_{t,p})$, where $\mathbf{d}_{t,p} \in \mathbb{R}^3$ is a unit ray direction and $\mathbf{o}_{t,p} \in \mathbb{R}^3$ is the camera-origin field. Before supervision, the predicted ray map is

aligned to the official DA3 camera-pose space. The ray loss is

$$\mathcal{L}_R = \frac{1}{B}\sum_b \left[\frac{1}{|\Omega_b|} \sum_{(t,p)\in\Omega_b} \left\| \tilde{\mathbf{d}}_{b,t,p} - \mathbf{d}^\star_{b,t,p} \right\|_1 + \lambda_o \frac{1}{|\Omega_b|} \sum_{(t,p)\in\Omega_b} \left\| \tilde{\mathbf{o}}_{b,t,p} - \mathbf{o}^\star_{b,t,p} \right\|_1 \right] \tag{24}$$

The direction term controls viewing geometry, while the origin term keeps the predicted camera-origin field at the correct scale.

The point-cloud loss turns depth and ray geometry into explicit 3D points. In ray-map form, each valid pixel produces

$$\mathbf{P}_{b,t,p} = \mathbf{o}_{b,t,p} + D_{b,t,p}\mathbf{d}_{b,t,p}. \tag{25}$$

Let $\hat{\mathcal{P}}_b$ and $\mathcal{P}^\star_b$ be the resulting valid point sets sampled from all frames. We use a symmetric dense point-set distance,

$$\mathcal{L}_P = \frac{1}{B}\sum_b \frac{1}{2\gamma_b} \left[\frac{1}{|\hat{\mathcal{P}}_b|} \sum_{\hat{\mathbf{p}}\in\hat{\mathcal{P}}_b} \min_{\mathbf{q}\in\mathcal{P}^\star_b} \|\hat{\mathbf{p}} - \mathbf{q}\|_2 + \frac{1}{|\mathcal{P}^\star_b|} \sum_{\mathbf{q}\in\mathcal{P}^\star_b} \min_{\hat{\mathbf{p}}\in\hat{\mathcal{P}}_b} \|\hat{\mathbf{p}} - \mathbf{q}\|_2 \right]. \tag{26}$$

This term is a training-time surrogate for reconstruction quality. It penalizes global 3D misalignment that can be hidden if depth and ray maps are considered independently.

The camera loss is computed after converting both predicted and target poses to a first-camera-relative gauge. Let $\bar{\mathbf{E}}_{b,t}$ be the aligned world-to-camera extrinsic, let $\bar{\mathbf{K}}_{b,t}$ be the image-size-normalized intrinsic matrix, and let $\mathbf{u}_{b,t}$ be the unit translation direction. We use

$$\mathcal{L}_C = \frac{1}{B}\sum_b \frac{1}{|\mathcal{T}_b|} \sum_{t\in\mathcal{T}_b} \left(\left\| \hat{\bar{\mathbf{E}}}^R_{b,t} - \bar{\mathbf{E}}^{\star,R}_{b,t} \right\|_1 + 1 - \left\langle \hat{\mathbf{u}}_{b,t}, \mathbf{u}^\star_{b,t} \right\rangle + \left\| \hat{\bar{\mathbf{K}}}_{b,t} - \bar{\mathbf{K}}^\star_{b,t} \right\|_1 \right), \tag{27}$$

where $\mathcal{T}_b$ denotes frames with valid camera estimates and $\bar{\mathbf{E}}^R$ denotes the rotation block.

The temporal depth term matches depth changes across adjacent valid frames [17],

$$\mathcal{L}_T = \frac{1}{B}\sum_b \frac{1}{\gamma_b} \frac{1}{|\Psi_b|} \sum_{(t,p)\in\Psi_b} \left| \left(\tilde{D}_{b,t+1,p} - \tilde{D}_{b,t,p} \right) - \left(D^\star_{b,t+1,p} - D^\star_{b,t,p} \right) \right|, \tag{28}$$

where Ψ_b contains adjacent frame-pixel pairs that are valid in both frames and pass the depth-change filter. This term prevents the stitched decoder from producing temporally plausible RGB-aligned depth in each frame but inconsistent depth evolution across the clip.

The motion term $\mathcal{L}_M$ uses the same masked scene-flow loss as Section B.2. For GT datasets without scene-flow annotations, including ScanNet++ and MVS-Synth in our 81-frame stage-2 manifests, we set $\mathcal{L}_M = 0$ and keep the remaining geometry losses active.

B.5 Joint Video and 4D Geometry Training

After the stitched decoder is finetuned, we freeze it and use it as a differentiable geometry critic for LongCat-Video. The only trainable video parameters in our main joint run are LoRA parameters on the DiT. The training batch first follows standard latent flow matching. Given a clean latent $\mathbf{z}_0$, noise $\boldsymbol{\epsilon}$, and sampled noise level τ, the scheduler constructs a noisy latent $\mathbf{z}_\tau$ and a target velocity $\mathbf{v}^\star_\tau$. The diffusion loss is a masked weighted mean squared error,

$$\mathcal{L}_{\mathrm{FM}} = w(\tau) \frac{\sum_j b_j \left\| \mathbf{v}_{\phi,j} - \mathbf{v}^\star_{\tau,j} \right\|_2^2}{\max\left(10^{-8}, \sum_j b_j\right)}, \tag{29}$$

where b_j is the latent loss mask. For image-to-video training, the first frame is used as the conditioning prefix. For continuation training, the observed prefix is encoded as LongCat condition latents. When condition masking is enabled, the prefix receives zero diffusion loss.

The predicted velocity also gives a denoised latent estimate,

$$\hat{\mathbf{z}}_0 = \mathbf{z}_\tau - \sigma_\tau \mathbf{v}_\phi\left(\mathbf{z}_\tau, \tau, c\right). \tag{30}$$

We pass $\hat{\mathbf{z}}_0$ directly into the frozen stitched decoder and compute a geometry loss on the decoded 4D outputs. The joint objective is shown in Equation (5).

For action-conditioned joint training, the teacher-student distillation branch is always active and supplies the only supervision available for Action100M clips. A raw DA3 teacher is loaded independently from the stitched decoder and is kept frozen. The raw teacher processes the ground-truth RGB video and supplies depth, ray, confidence, camera, and scene-flow targets. The frozen stitched decoder processes the model-predicted latent $\hat{\mathbf{z}}_0$. The same distillation loss used in decoder finetuning is then applied to the two geometry outputs.

In the manifest-based joint setup, GT-capable geometry datasets can be used with `distill`, `gt`, or `hybrid` supervision modes. Action100M rows are wrapped as distill-only manifest entries with prompt metadata and no GT geometry. For the subset used in the main paper, every Action100M row is an 81-frame TI2V clip with `cond_frames=1`. This design lets the video model learn long-horizon action semantics from large-scale web video while still receiving metric 4D regularization from the stitched geometry branch.

B.6 Model Information

Our system combines a large pretrained video generator with a frozen geometric readout. The dominant parameter budget comes from the LongCat-Video diffusion transformer, while the Wan VAE provides the latent video interface used by both the RGB and geometry branches. Based on the local pretrained weights used in our pipeline, the LongCat-Video DiT contains 13,581.07M parameters, and the Wan VAE contains 126.89M parameters. These two components are both frozen during stitched-decoder finetuning, and only the DiT is updated during the final joint training stage, where our main setting optimizes LoRA parameters on top of the pretrained backbone.

The geometry branch is implemented as a stitched DA3 decoder that reads the same Wan latent trajectory and recovers depth, camera rays, camera parameters, and scene flow. This branch reuses the pretrained DA3 backbone and dense geometry heads together with our added scene-flow head and the latent-to-token stitching module. We do not count the raw DA3 teacher separately because it is only used to provide supervision during training and is not part of the inference-time Plan4D pipeline. In this sense, the deployed system consists of three principal model components: a 13.58B video diffusion transformer, a 126.89M video VAE, and a stitched geometric decoder built on top of DA3. The overall parameter budget is therefore dominated by the video backbone rather than the geometric readout.

C Other Ablation Studies

C.1 Ablation of the Joint Training Objective

We ablate the final joint-training objective and evaluate each variant on HoiGen and Dynamic Replica under the action-conditioned world-generation protocol in Section 4.1. Each test example provides an initial RGB frame and an action description. HoiGen uses its native action annotations, while Dynamic Replica uses the per-clip action captions from the main evaluation. Each variant produces RGB video and dense geometry from its predicted latent trajectory using the frozen stitched decoder described in Section B.5.

The variants isolate the terms of Equation (5). **Only-RGB-Loss** optimizes the latent flow-matching loss $\mathcal{L}_{\mathrm{FM}}$ without the geometry term. **Only-Geo-Loss** optimizes the weighted geometric loss $\lambda_{\mathrm{geo}}\mathcal{L}_{\mathrm{geo}}$ without the flow-matching term. **Ours (Full)** combines both losses to train the video generator while keeping the stitched geometry decoder frozen. When geometric supervision is used, its targets come from the frozen DA3 teacher described in Section B.5.

For both datasets, we report subject consistency, background consistency, depth AbsRel, depth δ_1, and camera rotation RPE_R in Table 4. Higher consistency scores and δ_1 are better, while lower AbsRel and RPE_R are better. Depth uses per-clip scale-shift alignment, and camera evaluation uses the same first-camera alignment and relative-rotation protocol as the main joint-generation comparison.

On Dynamic Replica, the full objective achieves the best result on all five metrics. Compared with RGB-only training, subject and background consistency increase from 0.8998 and 0.9321 to 0.9356 and 0.9496. Depth AbsRel decreases from 0.0874 to 0.0696, δ_1 increases from 0.9169 to

0.9306, and camera RPE_R decreases from 0.2615 to 0.1550. Geometry-only training performs worse on all reported metrics than both RGB-only training and the full objective, indicating that geometric supervision alone does not preserve visual consistency or ensure accurate geometry on this benchmark.

The HoiGen results reveal a different depth tradeoff. Geometry-only training attains the best depth scores, but weakens subject and background consistency and has a higher camera rotation error than the full objective. Across both datasets, the full model gives the best subject and background consistency and camera estimates. These results support the complementary roles of the two losses: flow matching preserves action-conditioned visual quality, while geometric supervision regularizes the shared future latent state toward consistent depth and camera structure.

C.2 Ablation for Stitching Adapter Architecture

The stitching adapter has a narrow but important role in PLAN4D. It maps the upsampled Wan VAE latent into the DA3 token space at a candidate encoder block, after which the remaining DA3 encoder and dense heads perform the geometric decoding. We therefore ask whether this interface needs a high-capacity nonlinear module, or whether the compact linear 3D convolution used in the main method is sufficient.

We compare three adapter architectures under the same cached OmniWorld-Game [119] feature sweep. Let $\mathbf{Z} \in \mathbb{R}^{B\times 16\times T\times H\times W}$ denote the temporally upsampled Wan latent, and let $\mathbf{Y} \in \mathbb{R}^{B\times 1536\times T\times H'\times W'}$ denote the target DA3 token features. The adapter learns a mapping

$$\mathcal{A}_\theta : \mathbf{Z} \mapsto \hat{\mathbf{Y}}, \qquad \hat{\mathbf{Y}} \in \mathbb{R}^{B\times 1536\times T\times H'\times W'}. \tag{31}$$

We study three designs with increasing modeling capacity. The linear adapter provides a purely local projection, the residual bottleneck augments this projection with a nonlinear correction, and the axial-attention adapter further introduces long-range mixing along the temporal and spatial axes.

Linear convolutional adapter. The first variant uses a single 3D convolution to map the 16-channel Wan latent directly into the 1536-channel DA3 token space:

$$\hat{\mathbf{Y}} = \mathcal{A}^{\text{lin}}_\theta(\mathbf{Z}) = \text{Conv3D}^{16\to 1536}_{5\times 3\times 3,\ 1\times 2\times 2,\ 2\times 1\times 1}(\mathbf{Z}). \tag{32}$$

Equivalently, after unfolding the local spatiotemporal neighborhoods, this model is a linear regression problem:

$$\text{vec}(\hat{\mathbf{Y}}) = \mathbf{\Phi}(\mathbf{Z})\mathbf{w}, \tag{33}$$

where $\mathbf{\Phi}(\mathbf{Z})$ is the design matrix induced by the convolutional receptive field and $\mathbf{w}$ contains the convolutional weights. We can fit this adapter by streaming ridge regression as in Equation (2). This gives a strong linear baseline with minimal optimization overhead, but its receptive field and representational capacity are limited.

Residual bottleneck adapter. The second variant keeps the same linear convolution as a skip path and adds a nonlinear bottleneck branch. The skip branch is

$$\mathbf{S} = \text{Conv3D}^{16\to 1536}_{5\times 3\times 3,\ 1\times 2\times 2,\ 2\times 1\times 1}(\mathbf{Z}). \tag{34}$$

The residual branch first expands the channel dimension with a pointwise projection, applies normalization and GELU, performs local spatiotemporal mixing, and then projects the features back to the DA3 token dimension:

$$\mathbf{U} = \text{GELU}\left(\text{Norm}\left(\text{Conv3D}^{16\to d}_{1\times 1\times 1}(\mathbf{Z})\right)\right), \tag{35}$$

$$\mathbf{V} = \text{GELU}\left(\text{Norm}\left(\text{Conv3D}^{d\to d}_{5\times 3\times 3,\ 1\times 2\times 2,\ 2\times 1\times 1}(\mathbf{U})\right)\right), \tag{36}$$

$$\mathbf{R} = \text{Conv3D}^{d\to 1536}_{1\times 1\times 1}(\mathbf{V}). \tag{37}$$

The final prediction is the sum of the linear fit and the nonlinear residual correction:

$$\hat{\mathbf{Y}} = \mathcal{A}^{\text{res}}_\theta(\mathbf{Z}) = \mathbf{S} + \mathbf{R}. \tag{38}$$

This design preserves the stability of the linear adapter while allowing the bottleneck branch to model nonlinear deviations from the ridge-regression solution.

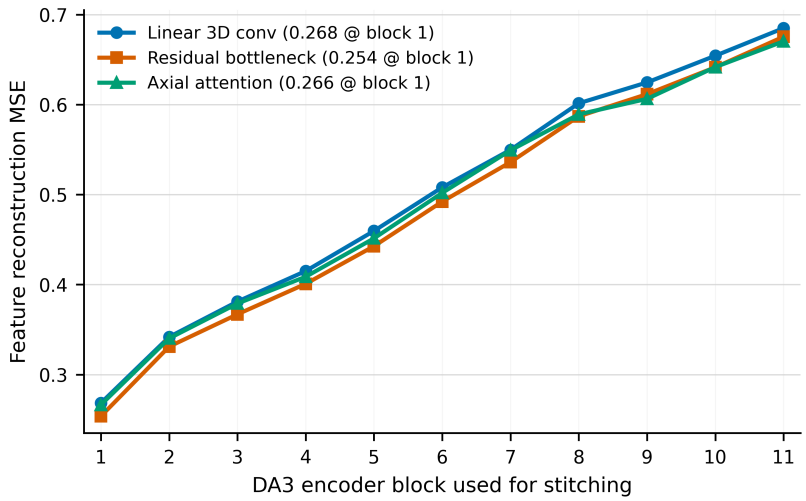

Figure 8: Stitching adapter architecture ablation over DA3 encoder blocks 1–11. Lower MSE means that the adapter reconstructs the frozen DA3 feature grid more accurately from the Wan VAE latent. All three variants prefer the earliest candidate block, and the linear 3D convolution remains close to the heavier residual and attention-based adapters across the sweep.

Axial-attention adapter. The third variant uses a convolutional stem followed by axial self-attention [89]. The stem first lifts the Wan latent into a hidden feature space:

$$\mathbf{H}_0 = \mathrm{GELU}\left(\mathrm{Norm}\left(\mathrm{Conv3D}^{16\to d}_{5\times3\times3,\ 1\times2\times2,\ 2\times1\times1}(\mathbf{Z})\right)\right). \tag{39}$$

We then apply multi-head self-attention independently along the temporal, height, and width axes. For an axis $a \in \{T, H, W\}$, axial attention is written as

$$\mathrm{Attn}_a(\mathbf{H}) = \mathrm{Reshape}_a^{-1}\left[\mathrm{MHA}\left(\mathrm{LN}\left(\mathrm{Reshape}_a(\mathbf{H})\right)\right)\right], \tag{40}$$

where $\mathrm{Reshape}_a$ groups tokens into one-dimensional sequences along axis a while treating the remaining axes as batch dimensions. The hidden state is updated by residual axial attention:

$$\mathbf{H}_T = \mathbf{H}_0 + \mathrm{Attn}_T(\mathbf{H}_0), \tag{41}$$

$$\mathbf{H}_{TH} = \mathbf{H}_T + \mathrm{Attn}_H(\mathbf{H}_T), \tag{42}$$

$$\mathbf{H}_{THW} = \mathbf{H}_{TH} + \mathrm{Attn}_W(\mathbf{H}_{TH}). \tag{43}$$

Finally, a pointwise projection maps the attended representation to the DA3 token dimension:

$$\hat{\mathbf{Y}} = \mathcal{A}^{\mathrm{axial}}_\theta(\mathbf{Z}) = \mathrm{Conv3D}^{d\to1536}_{1\times1\times1}(\mathbf{H}_{THW}). \tag{44}$$

This architecture gives the adapter access to long-range temporal and spatial context while avoiding the quadratic cost of full 3D attention. It is the most expressive variant among the three, but it also has the highest computational and memory cost.

Figure 8 shows the reconstruction MSE for blocks 1–11. All three curves have the same dominant trend. The earliest DA3 block is easiest to reach from the Wan latent, while deeper blocks become progressively harder to reconstruct. This aligns with the insight from Go et al. [29] that early 3D foundation model encoder features serve as a foundational geometric interface. This observation directly motivates our architectural design in Section 3. By introducing the stitch at an early stage, we capture these representations before the network accumulates highly specialized geometric processing.

The architectural comparison is more important than the absolute block ranking. At the best block, the linear 3D convolution obtains an MSE of 0.268. The residual bottleneck obtains 0.254, and axial attention obtains 0.266. Averaged over blocks 1–11, the MSEs are 0.499, 0.485, and 0.491 for the linear convolution, residual bottleneck, and axial attention variants. The nonlinear residual branch

Table 7: Ablation of stitched-decoder finetuning strategies under the same evaluation protocol as Table 2. The table reports representative reconstruction and camera metrics that diagnose metric 3D preservation and dynamic-scene camera stability. In each metric column, the best value is highlighted in red and bold, the second-best value in orange, and the third-best value in yellow.

Category	Variant	ETH3D Acc. ↓	Dynamic-RE10K AUC_{30} ↑
Supervision	Distillation only	1.4276	0.5259
Supervision	Distillation with feature loss	0.5209	0.5330
Supervision	GT supervision only	1.7899	0.3141
Architecture	No scene-flow head	0.8459	0.5139
Adaptation	LoRA adaptation	2.6344	0.5144
Full model	Ours (full)	**0.2898**	**0.5350**

gives the lowest standalone fitting error, but the gain is small relative to the added normalization, activation, and extra convolutional layers. Axial attention is even closer to the linear convolution despite its substantially higher memory and compute cost.

These results support the simpler adapter used in the main system. The stitching layer is not intended to replace the remaining DA3 decoder. Its job is to place the video latent on a compatible DA3 token manifold and provide a stable initialization for the stitched decoder finetuning stage. A linear 3D convolution already captures most of this alignment. More expressive residual or attention-based adapters provide only modest reconstruction gains, so they do not justify their additional cost for this interface.

C.3 Ablation for Stitched Decoder Finetuning Strategy

The stitched decoder is trained for a specific role. It must accept Wan VAE latents as input while retaining the geometric behavior of the original DA3 model. This creates a tension between adaptation and preservation. Teacher-student distillation keeps the student close to the pretrained DA3 function, while ground-truth geometric supervision anchors the predictions to metric depth, camera, point-cloud, and motion annotations when such labels are available. We ablate this finetuning strategy to identify which ingredients are needed for a stable latent-to-geometry decoder.

We follow the same benchmarks and metrics as the main stitched-model evaluation in Table 2. The variants isolate the main design choices in the finetuning stage. **Distillation only** trains the student only from the frozen DA3 teacher outputs. The teacher receives the ground-truth RGB clip, and the student receives the Wan latent of the same clip. **Distillation with feature loss** introduces an auxiliary feature-matching term to regularize the stitched DA3 representation. Specifically, it applies an L_1 penalty between the output features of the stitching layer and the corresponding block features from the teacher model. This constraint forces the projected latent tokens to remain aligned with the teacher's feature manifold prior to dense decoding. **GT supervision only** removes the teacher and trains only from available dataset annotations. **No scene-flow head** keeps the depth, ray, camera, and reconstruction objectives but removes the motion branch during stitched-decoder finetuning. **LoRA adaptation** replaces the full update of the stitched decoder with a parameter-efficient alternative [36] on the adapted DA3 computation path. Specifically, we apply a low-rank update with both the rank and scaling factor α set to 32 and maintain a learning rate of 1×10^{-4}.

The results show that output distillation alone is not enough. It preserves coarse camera behavior relatively well, but its reconstruction quality degrades noticeably. This suggests that matching final teacher predictions can transfer some global geometric structure, yet it does not fully constrain the accurate geometry after the RGB encoder has been replaced by a latent input pathway. Compared to our full model trained with mixed supervision, combining output-space distillation and ground-truth data, incorporating an additional feature loss actually degrades performance. This phenomenon aligns with the insights of Huang et al. [38]. It suggests that rigid representation alignment cannot independently recover all the geometric behaviors required by the stitched decoder. Furthermore, enforcing such strict alignment may impair the flexibility of the learned representations since the underlying model architecture has changed.

The GT-only variant fails for a distinct reason. While direct supervision provides explicit targets, it eliminates the pretrained DA3 teacher signal that contains broad reconstruction priors. Furthermore,

some low quality ground-truth data can be sparse or inaccurate locally, hindering the transfer learning capabilities of the model. This observation aligns with the insights of Lin et al. [64]. Consequently, the network becomes highly sensitive to dataset coverage, annotation quality, and task imbalances. This vulnerability is particularly detrimental to camera estimation, where the teacher provides a strong geometric prior that is exceedingly difficult to learn from supervised labels alone. These results clarify the proper role of ground-truth supervision within our system. It serves as a valuable corrective signal but must complement distillation rather than replace it.

The architectural and adaptation ablations further clarify what the finetuning stage must optimize. Removing the scene-flow head weakens both reported axes, even though the compact table does not directly score scene flow. This indicates that motion supervision contributes useful temporal pressure to the shared decoder representation and benefits the geometry heads beyond the motion output itself. LoRA adaptation is also insufficient for this transfer. It keeps some camera behavior, but reconstruction degrades severely, which suggests that the distribution shift from RGB tokens to Wan latents is too large for a narrow low-rank update.

The full model is strongest on both reported axes. Its advantage is most visible in reconstruction, where combining distillation and ground-truth supervision best preserves 3D structure under the latent input path. The camera gains are more modest because several variants already retain coarse camera cues, but the full objective still gives the most stable result. This ablation supports the hybrid finetuning strategy used in PLAN4D. Distillation preserves the broad DA3 prior, ground-truth geometry corrects metric errors left by teacher-only learning, and scene-flow supervision helps maintain temporal structure in the shared decoder features.

D Limitations

Our method serves as a geometry-aware intermediate world model for predicting future visual and 4D geometric states, rather than an end-to-end embodied agent. Because it lacks closed-loop physical deployment and relies purely on offline benchmarks, the impact of prediction error accumulation on downstream decision-making remains untested. Future work must therefore integrate this predictive representation with explicit action policies to evaluate task success under real-world execution constraints.

E Broader Impact

While our geometry-aware world model advances embodied planning and simulation through structured 4D future predictions, it introduces potential risks, including misuse in surveillance and overreliance in safety-critical applications susceptible to prediction errors and data biases. To ensure responsible deployment, these predictive outputs must be treated as probabilistic forecasts requiring uncertainty-aware downstream validation, augmented with strict task-specific safeguards prior to physical execution, and rigorously evaluated against distribution shifts and embodied failure modes. These measures support more responsible use of geometry-aware world models.

F Licenses and Terms of Use

All external code, models, and datasets utilized in our experiments are properly cited and strictly governed by their respective licenses and usage restrictions.

G LLM Usage Statement

We used a general-purpose large language model as a writing assistant for limited language polishing during manuscript preparation, including improvements to phrasing, readability, and English fluency. We also used Qwen-VL-30B-Thinking to generate per-clip action captions for the Dynamic Replica and PointOdyssey evaluation splits. These captions were used as the conditioning text for all compared methods. The captioning model was used for evaluation annotation preparation and was not a component of the proposed PLAN4D model.

Table 8: Quantitative comparison across five benchmarks. Epic-Kitchens-100 uses 100 five-second clips of in-the-wild kitchen interactions evaluated under the same protocol as the other benchmarks. Arrows indicate the preferred direction. Best results for each dataset and metric are in **bold**.

		Video Quality						Depth & Reconstruction				Camera		
Dataset	Model	SC ↑	BC ↑	AQ ↑	IQ ↑	MS ↑	Cap.Sim ↑	AbsRel ↓	δ_1 ↑	δ_2 ↑	Chamfer-L1 ↓	RPE-R ↓	RRA@10 ↑	RRA@5 ↑
HoiGen	**Ours**	**.9035**	**.9334**	**.4802**	.6102	**.9832**	.6854	**.1125**	**.854**	**.9596**	.3325	**.1240**	**.9903**	**.9809**
	TesserAct	.8661	.9187	.4480	**.6164**	.9802	.6401	1.850	.1839	.3402	.3534	.1429	.9725	.9256
	Gen3R	.7765	.8774	.4391	.4826	.9532	.6803	.2881	.5685	.7789	**.2286**	5.060	.4891	.2689
	VIST3A	.6721	.8346	.4730	.5600	.9383	**.7851**	.6301	.2570	.3761	.7633	13.09	.1298	.0185
Epic-Kitchens-100	**Ours**	**.9506**	**.9555**	.4207	**.6435**	.9867	**.4010**	**.0766**	**.9265**	**.9846**	**.2321**	.1568	**1.0000**	**1.0000**
	TesserAct	.8923	.9223	.4061	.6072	**.9874**	.3343	.4201	.7519	.9207	1.3321	**.0946**	.9983	.9736
	Gen3R	.7813	.8691	.3618	.4706	.9793	.3305	.3641	.4580	.6980	.3390	2.3992	.6104	.3510
	VIST3A	.5499	.8055	**.4556**	.4994	.9768	.3710	.1326	.8204	.9708	6.4722	34.7905	.0275	.0184
Dynamic Replica	**Ours**	**.9356**	**.9496**	.5078	**.6735**	**.9927**	.5639	**.0696**	**.9306**	**.9829**	**.2579**	**.1550**	**1.000**	**1.000**
	TesserAct	.8490	.9242	**.5115**	.6646	.9864	.4419	1.289	.3523	.5674	.4067	.7609	.9501	.6772
	Gen3R	.6833	.8512	.4053	.4906	.9700	.4620	.2018	.6240	.9013	1.034	22.47	.2998	.1393
	VIST3A	.6013	.8163	.4867	.5467	.9604	**.5744**	.4084	.4502	.5430	1.474	16.31	.3098	.1230
Point Odyssey	**Ours**	**.9233**	**.9423**	**.5355**	**.5941**	.9856	**.5579**	**.0822**	**.9215**	**.9720**	**.3013**	**.1985**	**.9965**	**.9864**
	TesserAct	.9019	.9391	.4920	.5351	**.9932**	.5094	.6347	.2953	.5390	.3904	.8804	.8062	.6446
	Gen3R	.7486	.8725	.4389	.4164	.9708	.4930	.4227	.3875	.6961	.6703	9.409	.4066	.2188
	VIST3A	.6270	.8173	.4983	.4373	.9777	.5465	.4371	.4054	.5670	1.347	6.953	.4098	.2962
RLBench	**Ours**	**.8459**	.9017	**.4619**	.4767	**.9897**	.5583	**.0822**	**.9215**	**.9721**	**.3012**	.1985	.9963	.9864
	TesserAct	.8424	**.9235**	.4312	**.5932**	.9842	.1791	5.481	.2003	.3950	.3904	**.0566**	**1.000**	**.9999**
	Gen3R	.6434	.8442	.3732	.4086	.9751	.1769	.2468	.6220	.8306	1.663	11.87	.3712	.1840
	VIST3A	.5608	.8052	.4375	.4913	.9754	**.6071**	.3441	.4380	.5001	.8709	10.701	.2101	.1023

H Experimental Compute Resources

Sec. 4 and Appendix B report GPU types (H100 80GB for Stage 1, H20 96GB for Stages 2 and 3), GPU counts (16 for Stage 1, 64 for Stages 2 and 3), per-GPU batch size (1), and training epochs for all three stages. Stage 1 takes approximately 60 hours on 16 H100 GPUs ($\sim$960 H100 GPU hours); Stage 2 takes approximately 90 hours on 64 H20 GPUs ($\sim$5,760 H20 GPU hours); Stage 3 takes approximately 160 hours on 64 H20 GPUs ($\sim$10,240 H20 GPU hours), totaling approximately 16,960 GPU hours across all three stages. Inference runs on 8 H20 GPUs and takes approximately 15 minutes per clip. The full research project required additional compute beyond the reported training runs. Ablation studies on stitching adapter architectures and stitched-decoder finetuning strategies follow the same stage-wise GPU configuration and duration as the corresponding main experiments, contributing an estimated $\sim$34,000 additional GPU hours in total across all ablation variants.

I More Experimental Results

Full quantitative comparison. Table 8 reports the complete joint 4D generation evaluation across HoiGen, Epic-Kitchens-100, Dynamic Replica, PointOdyssey, and RLBench. Epic-Kitchens-100 extends this evaluation to everyday kitchen interactions under the same protocol, where PLAN4D achieves the best values on 10 of the 13 metrics. We group the metrics into video quality, depth and reconstruction, and camera estimation. The discussion below focuses only on metrics that are reported for the corresponding methods.

Across video-quality metrics, PLAN4D achieves the best subject consistency on all five benchmarks and the best motion smoothness on HoiGen, Dynamic Replica, and RLBench. It leads in caption similarity on Epic-Kitchens-100 and PointOdyssey, while VIST3A leads on HoiGen, Dynamic Replica, and RLBench. On Epic-Kitchens-100, its SC and BC are 0.9506 and 0.9555, compared with 0.8923 and 0.9223 for TesserAct, and its Cap.Sim is 0.4010, compared with 0.3710 for the next-best method, VIST3A. TesserAct remains competitive on individual appearance metrics, such as image quality on HoiGen and RLBench, aesthetic quality on Dynamic Replica, and motion smoothness on Epic-Kitchens-100. VIST3A also leads in aesthetic quality on Epic-Kitchens-100. These results indicate strong visual consistency across benchmarks, while individual appearance and caption-alignment metrics favor different methods.

The depth and reconstruction columns show the value of reading geometry from the same latent future. PLAN4D obtains the best depth metrics on all five datasets, with substantially lower AbsRel and higher threshold accuracy than TesserAct and Gen3R. On Epic-Kitchens-100, it obtains an AbsRel of 0.0766 with $\delta_1 = 0.9265$ and $\delta_2 = 0.9846$. It also gives the best Chamfer-L1 reconstruction on Epic-Kitchens-100, Dynamic Replica, PointOdyssey, and RLBench among methods with reported reconstruction outputs. On Epic-Kitchens-100, its Chamfer-L1 of 0.2321 is lower than Gen3R's 0.3390, TesserAct's 1.3321, and VIST3A's 6.4722. Gen3R is strongest on HoiGen Chamfer-L1,

Table 9: Comparison with video-only generation followed by post hoc geometry reconstruction on HoiGen under the main action-conditioned evaluation protocol. Geometric references are reconstructed by applying DA3 to the ground-truth videos. Arrows indicate the preferred direction, and the best value in each column is in **bold**.

Method	Subj. Cons. ↑	BG Cons. ↑	Depth AbsRel ↓	Depth δ_1 ↑	Camera RPE-R ↓
Cosmos-3-Nano + VGGT-Omega	0.8760	0.9213	0.4547	0.5850	0.2933
LongCat-Video + DA3	0.8864	0.9235	0.2815	0.6875	0.3450
PLAN4D (Ours)	**0.9035**	**0.9334**	**0.1125**	**0.8540**	**0.1240**

but its depth and camera metrics are much worse on the same benchmark. This indicates that a lower point-cloud distance in one setting does not necessarily imply a temporally coherent RGB-D world model. By contrast, PLAN4D keeps depth and reconstruction strong across scene types, from hand-object interaction to dynamic stereo and robot manipulation.

Camera estimation further separates the methods that preserve global 4D structure from those that only recover local geometry. PLAN4D achieves the best reported RPE-R on HoiGen, Dynamic Replica, and PointOdyssey, and matches or approaches perfect relative-rotation accuracy on Dynamic Replica and PointOdyssey. On Epic-Kitchens-100, it achieves RRA@10 and RRA@5 of 1.0000, although TesserAct has a lower RPE-R of 0.0946 compared with our 0.1568. TesserAct also leads on the RLBench camera metrics. Gen3R consistently has much larger rotation errors, which reflects the difficulty of maintaining stable global camera motion when geometry is generated without the same latent-level coupling to the video model. The Epic-Kitchens-100 results provide quantitative evidence of visual, geometric, and camera consistency on in-the-wild interactions and, together with HoiGen, support the generality of PLAN4D across two real-world human-object interaction benchmarks. Overall, the full table supports the same conclusion as the main paper: PLAN4D balances action-conditioned video quality, dense geometry, and camera stability in one forward generation process. These quantitative scores complement the qualitative snapshots in the paper, while full qualitative videos remain useful for inspecting the temporal behavior of generated 4D scenes.

I.1 Video-Only World Models with Post Hoc Geometry Reconstruction

This comparison tests whether a strong video-only world model followed by an independent geometry reconstructor can match joint video and 4D generation. We evaluate two additional baselines on HoiGen, including a direct combination of the two pretrained backbones used by PLAN4D.

Evaluation setting. We use the same 200 HoiGen test examples and action-conditioned generation protocol as the main comparison in Section 4.1 and table 8. Each method receives the initial RGB frame and an action description. For geometric evaluation, we feed the ground-truth video into DA3 and use its reconstructed depth and camera poses as the common reference. The geometric scores therefore measure agreement with DA3-derived pseudo ground truth. We report subject consistency and background consistency for video quality, depth AbsRel and δ_1 for depth accuracy, and camera relative rotation error (RPE-R). Depth uses the same per-clip scale-shift alignment as the main evaluation, and camera evaluation follows the same pose-alignment protocol described in Section B. Higher consistency scores and δ_1 are better, while lower AbsRel and RPE-R are better.

Baseline construction. **Cosmos-3-Nano + VGGT-Omega** uses Cosmos-3-Nano [1] to generate RGB video and applies VGGT-Omega [93] to reconstruct geometry from the generated frames. This baseline tests post hoc geometry recovery from a strong video-only world model. **LongCat-Video + DA3** first generates an RGB video with LongCat-Video and then applies DA3 directly to the generated frames. It combines our two pretrained backbones without joint training and tests whether their independent use is sufficient for consistent 4D generation. In both baselines, geometric reconstruction takes place after RGB video generation.

Results and analysis. Table 9 shows that PLAN4D achieves the best result on all five reported metrics. Compared with LongCat-Video + DA3, subject and background consistency increase from 0.8864 and 0.9235 to 0.9035 and 0.9334. Depth AbsRel decreases from 0.2815 to 0.1125, depth δ_1 increases from 0.6875 to 0.8540, and camera RPE-R decreases from 0.3450 to 0.1240. The

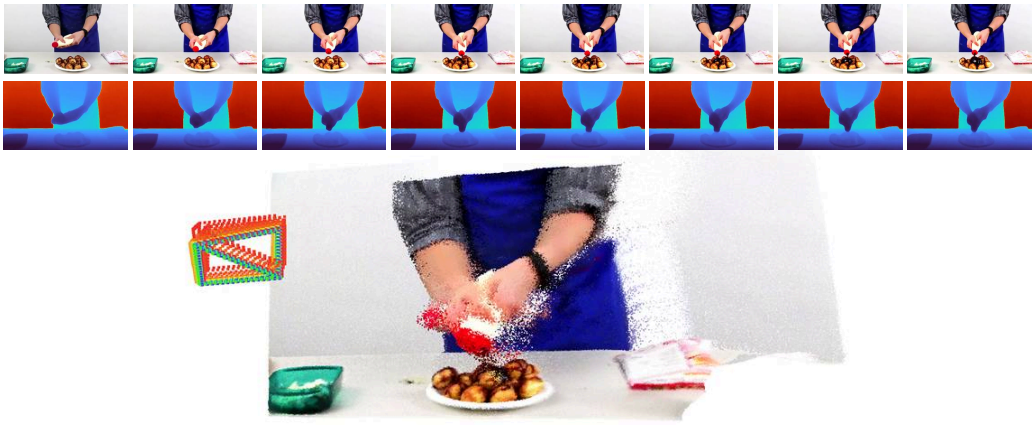

Figure 9: **Qualitative results of PLAN4D.** We provide the model with an initial frame and a descriptive text prompt: *"Removes the red cap from the squeeze bottle. Holds the bottle with both hands. Squeezes the bottle. Drizzles the sauce over the takoyaki in a controlled pattern. Finishes adding the sauce."* Based on these inputs, PLAN4D generates a dense 4D prediction. The figure visualizes the synthesized RGB frames (top row), depth maps (middle row), and 4D global point clouds (bottom row).

comparison with Cosmos-3-Nano + VGGT-Omega shows the same advantage across video, depth, and camera metrics.

These results illustrate a limitation of post hoc geometry recovery. A video generator trained only with appearance objectives receives no explicit supervision on the geometry, camera motion, or scene dynamics of its generated future. A downstream reconstructor must infer these quantities from the generated images, so visual inconsistencies can propagate into the recovered geometry. In PLAN4D, appearance, depth, camera geometry, and scene flow supervise a shared latent representation during joint training. The resulting latent future must support both RGB generation and geometric readout, encouraging visual and 4D consistency within the same rollout. Our stitching framework makes this supervision practical by connecting large-scale video data to a pretrained geometry model, while paired dynamic 3D datasets provide additional supervision where available.

I.2 More Qualitative Results

We provide additional qualitative examples to further evaluate the behavior of PLAN4D across diverse action scenarios. Each example is conditioned on a single initial frame and a natural-language description, from which the model predicts an evolving 4D future. We visualize the generated RGB frames, depth maps, and global point clouds from the same prediction, allowing appearance, geometry, and temporal consistency to be inspected jointly. The examples cover object manipulation, human motion, physical interaction, and multi-entity indoor dynamics, showing that PLAN4D can produce action-aligned visual changes while preserving coherent 3D structure over time.

I.3 Diagnostic Study on Image–Text Misalignment

Action-conditioned 4D generation assumes that the input frame and the text prompt describe a compatible scene. In practice, the two conditions may conflict, and it is important to understand which condition PLAN4D follows in that case. No established benchmark currently evaluates image–text conflicts for action-conditioned 4D generation, so we conduct a targeted diagnostic study.

Setting. We select 34 held-out image–text pairs and keep each original input frame fixed. For each frame, we construct two mismatched prompts, which yields 68 rollouts in total. The two prompt types probe different degrees of conflict.

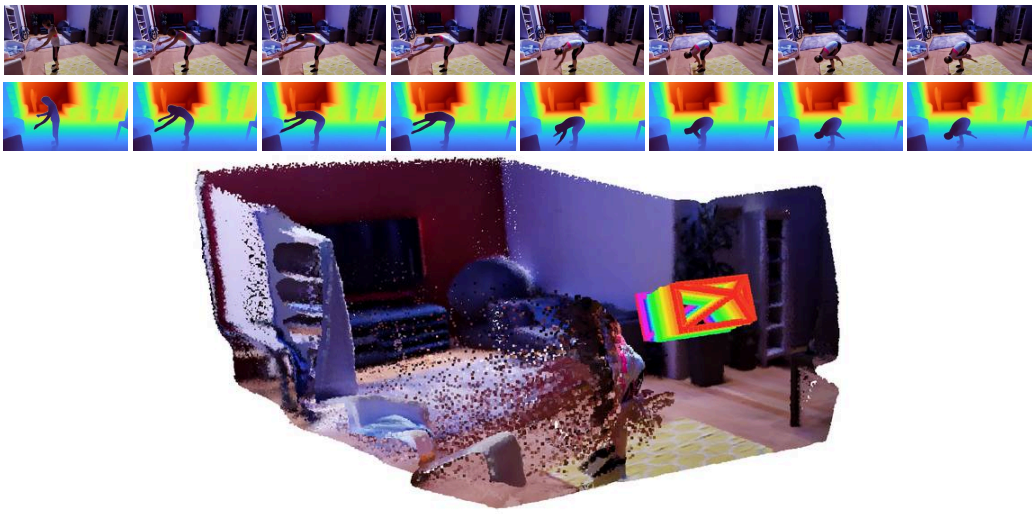

Figure 10: **Qualitative results of PLAN4D.** We provide the model with an initial frame and a descriptive text prompt: *"Person bends forward, arms extended, performing a slow motion movement on the yellow rug."* Based on these inputs, PLAN4D generates a dense 4D prediction. The figure visualizes the synthesized RGB frames (top row), depth maps (middle row), and 4D global point clouds (bottom row).

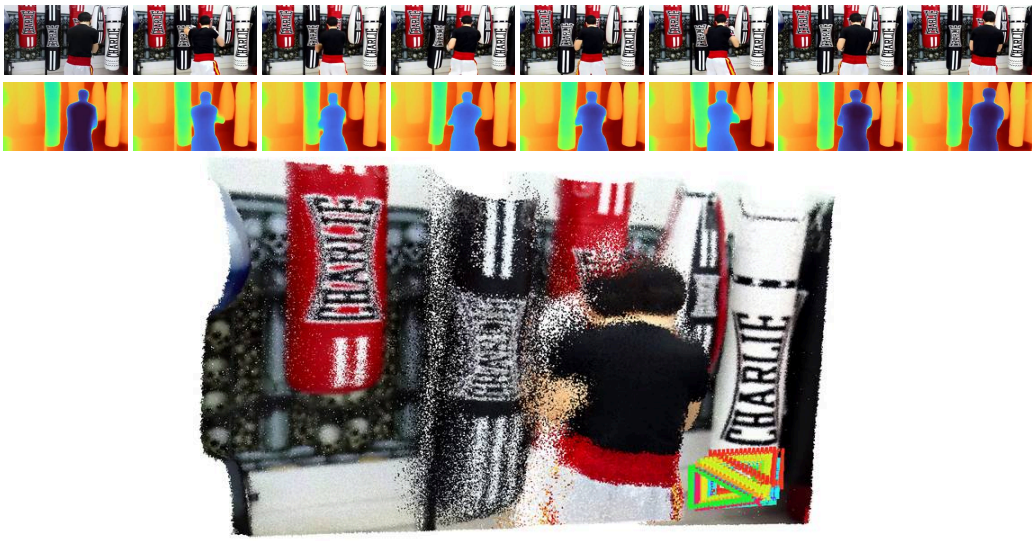

Figure 11: **Qualitative results of PLAN4D.** We provide the model with an initial frame and a descriptive text prompt: *"Delivers a series of punches to the heavy bag. The bag swings backward / recoils away upon impact."* Based on these inputs, PLAN4D generates a dense 4D prediction. The figure visualizes the synthesized RGB frames (top row), depth maps (middle row), and 4D global point clouds (bottom row).

- **Type A, semantically disjoint.** The prompt describes an event that cannot occur in the observed scene. For example, an egocentric kitchen frame showing a hand on dough is paired with *"A group of penguins dives from an iceberg into the cold ocean."*
- **Type B, scene editing.** The prompt requests an implausible transformation of the observed scene, such as *"Transform the dough into a pile of transparent glass marbles and make the dark counter become rippling beach sand."*

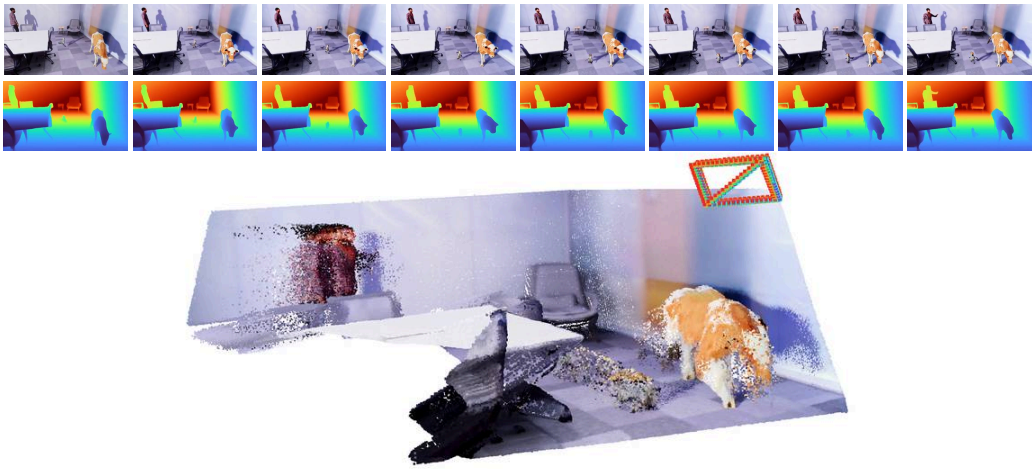

Figure 12: **Qualitative results of PLAN4D.** We provide the model with an initial frame and a descriptive text prompt: *"Man lifts leg, monkey stands and walks, cow shifts position in room."* Based on these inputs, PLAN4D generates a dense 4D prediction. The figure visualizes the synthesized RGB frames (top row), depth maps (middle row), and 4D global point clouds (bottom row).

Table 10: Diagnostic study on image–text misalignment. Each of the 34 held-out input frames is paired with one semantically disjoint prompt (Type A) and one scene-editing prompt (Type B), giving 68 rollouts. Each entry counts the rollouts manually assigned to an outcome. C denotes satisfying the prompt through a scene cut, M denotes satisfying it through an instantaneous morph, and P denotes preserving the original scene and continuing a plausible action. Each row sums to $n = 34$.

Mismatch type	C: cut	M: morph	P: plausible
Type A: semantically disjoint ($n = 34$)	34	0	0
Type B: scene editing ($n = 34$)	0	30	4

Categorization protocol. We manually assign each rollout to one of three outcomes. **C (cut)** denotes a rollout that satisfies the prompt through a scene cut to the described content. **M (morph)** denotes a rollout that satisfies the prompt through an instantaneous transformation of objects or surfaces in the observed scene. **P (plausible)** denotes a rollout that preserves the original scene and continues a plausible action instead of realizing the conflicting prompt.

Results. Table 10 summarizes the outcomes. For Type A, all 34 rollouts begin by preserving the input scene and then transition to the scene described by the prompt, in a manner reminiscent of a cinematic montage cut. For Type B, 30 of the 34 rollouts introduce the requested objects through abrupt transformations, while the remaining 4 preserve the original scene and continue a plausible action.

Qualitative examples and input prompts. Figure 13 illustrates the three outcomes with four selected rollouts. We show the initial image and sample the generated RGB and depth at 0.5, 1, 2, 3, 4, and 5 seconds. The input text prompts are:

- **(a) Scene transition.** *"A violinist performs alone under a spotlight in an empty concert hall."*
- **(b) Object morph.** *"Keep the same man, pose, and restaurant background, but transform the pizza in his hands into a large bouquet of yellow sunflowers and make his black shirt turn bright red."*
- **(c) Scene morph.** *"Keep the same man, pose, lawn, and modern building, but transform the camera stabilizer into an open yellow umbrella and turn the grass beneath him into shallow reflective water."*

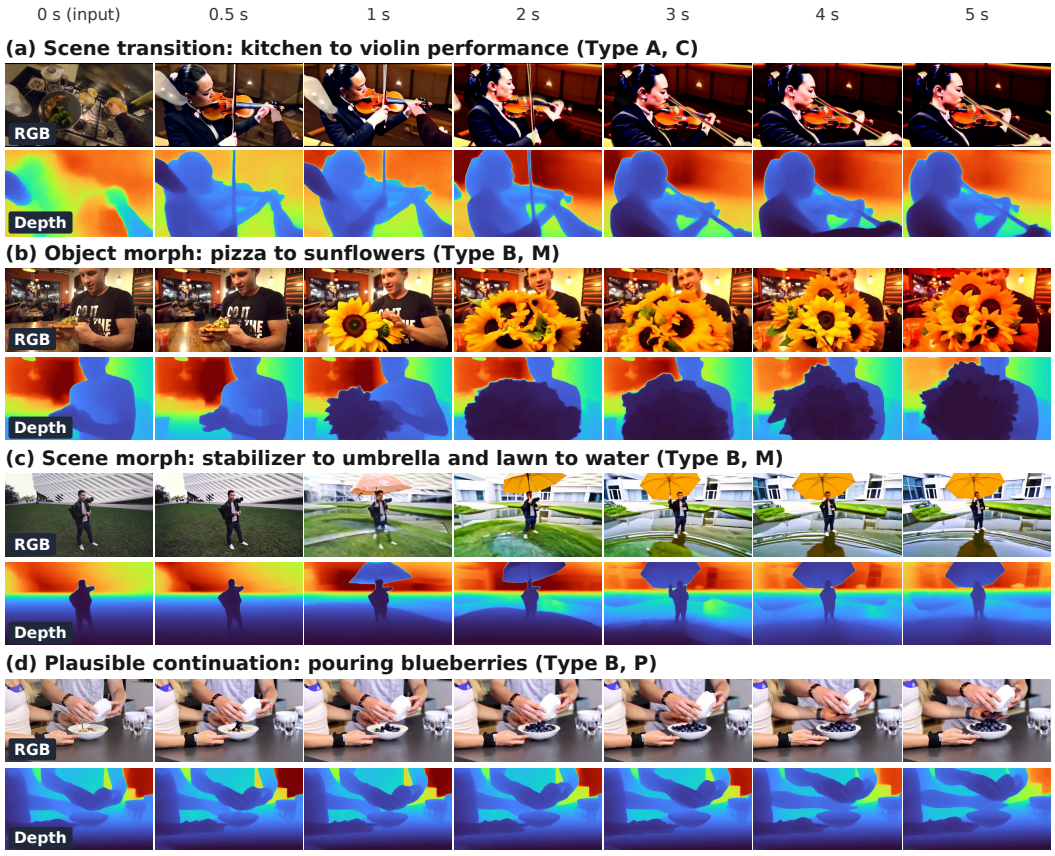

Figure 13: **Qualitative outcomes under image–text misalignment.** Each case shows RGB above the corresponding predicted depth at seven shared time points. The first RGB column is the conditioning image. Case (a) illustrates a scene transition (C) under Type A. Cases (b) and (c) illustrate object and scene transformations (M) under Type B. Case (d) preserves the original scene and continues a plausible action (P) under Type B. The complete input prompts are reported in the text.

- **(d) Plausible continuation.** *"Keep the same people, bowl, and tabletop, but make every blueberry disappear and fill the white bowl with shiny gold coins as the container is tilted above it."*

In (a), the egocentric kitchen view gives way to a close-up of a violinist. In (b), a sunflower bouquet replaces the pizza while the person and restaurant remain recognizable. In (c), an umbrella appears in place of the stabilizer and the lawn develops a reflective water surface. In (d), the model continues pouring blueberries into the bowl without introducing the requested gold coins. The paired depth maps follow the main foreground boundaries in the displayed frames, including the violinist, expanding bouquet, umbrella, and hands around the bowl. This correspondence can coexist with an implausible scene change or incomplete prompt fulfillment, as also illustrated by the unchanged black shirt in (b). The snapshots provide qualitative evidence of agreement between RGB and predicted depth, while physical continuity and complete adherence to the conflicting prompt remain separate properties.

Analysis. The diagnostic results in Table 10 and fig. 13 show that PLAN4D generally prioritizes the text condition when the image and text conflict. When the prompt cannot be naturally realized in the observed scene, the model resolves the mismatch through a scene transition or an instantaneous transformation. This behavior mainly originates from the text-conditioned video prior. The stitched geometry decoder still produces synchronized depth, camera rays, and scene flow for the generated trajectory, although physical continuity may be broken at the transition. Because PLAN4D explicitly predicts geometry such as scene flow and camera rays, these abrupt transitions may also be detectable as geometric discontinuities. Such discontinuities could provide a lightweight infeasibility signal for downstream planning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Sec. 1 state the main contributions and scope clearly, and these claims are supported by the experimental evidence summarized in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The manuscript now discusses the main limitations explicitly in Sec. D, including that the current model does not directly output executable actions and has not yet been validated as a full closed-loop embodied system.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theorem-style theoretical results or formal proofs. Its mathematical content defines the model, losses, and evaluation setup in Sec. 3 and the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 3, Sec. 4, and Sec. B describe the architecture, staged training pipeline, losses, datasets, and evaluation protocols that underpin the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: This paper relies entirely on open-source data, and the code will be made available upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4 specifies the training stages, datasets, evaluation benchmarks, clip lengths, and hardware scale, while Sec. B details the supervision sources and loss formulations needed to interpret the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Experimental results are tested multiple times to ensure stability and reliability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4 reports GPU types, GPU counts, batch size, and training epochs for the three stages.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes. The study is an algorithmic evaluation on existing datasets and pretrained models, it does not involve human subjects, and we are not aware of any aspect of the work that departs from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Sec. E discusses both positive uses for embodied planning and simulation and possible negative impacts such as surveillance, overreliance on incorrect predictions, and misuse in safety-critical settings, together with mitigation considerations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This submission poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The manuscript now summarizes the external code, model, and dataset assets together with their licenses or usage terms in Sec. F, and the main paper cites the original creators of the assets used in training, evaluation, and comparison.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a new method and model, but the submission does not yet release a dataset that would require accompanying asset documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not report crowdsourcing experiments or research with human subjects. All experiments are model training and evaluation on existing datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve human subjects or crowdsourcing, so IRB approval or an equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Sec. G states that a general-purpose LLM was used only for limited language polishing during manuscript preparation and did not affect the method, experiments, or conclusions, so this item remains not applicable under the checklist guidance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.