One4D: Unified 4D Generation and Reconstruction via Decoupled LoRA Control

Zhenxing Mi[1], Yuxin Wang[1], and Dan Xu[1]✉

The Hong Kong University of Science and Technology, Hong Kong, China
{zmiaa,ywangom}@connect.ust.hk, danxu@cse.ust.hk

Abstract. We present One4D, a unified framework for 4D generation and reconstruction that produces dynamic 4D content as synchronized RGB frames and pointmaps. By consistently handling varying sparsities of conditioning frames through a Unified Masked Conditioning (UMC) mechanism, One4D can seamlessly transition between 4D generation from a single image, 4D reconstruction from a full video, and mixed generation and reconstruction from sparse frames. Our framework adapts a powerful video generation model for joint RGB and pointmap generation, with carefully designed network architectures. The commonly used diffusion finetuning strategies for depthmap or pointmap reconstruction often fail on joint RGB and pointmap generation, quickly degrading the base video model. To address this challenge, we introduce Decoupled LoRA Control (DLC), which employs two modality-specific LoRA adapters to form decoupled computation branches for RGB frames and pointmaps, connected by lightweight, zero-initialized control links that gradually learn mutual pixel-level consistency. Trained on a mixture of synthetic and real 4D datasets under modest computational budgets, One4D produces high-quality RGB frames and accurate pointmaps across both generation and reconstruction tasks. This work represents a step toward general, high-quality geometry-based 4D world modeling using video diffusion models. Project page: https://mizhenxing.github.io/One4D.

Keywords: 4D generation and reconstruction · Video diffusion model

1 Introduction

Simulating the dynamics of the physical world has progressed rapidly with video diffusion and flow-matching models [4,9,18,21,28,41,55,67]. Recent open systems such as Wan [41], HunyuanVideo [18], and Cosmos [27] demonstrate remarkable visual fidelity and strong understanding of real-world dynamics. However, these foundation models operate purely in RGB space while lacking explicit geometry. Augmenting them with accurate geometry generation is a key step toward downstream world-simulation tasks such as spatial reasoning [53].

Concurrently, scalable 3D and 4D foundation models have advanced rapidly. Dust3R [45] introduces a pointmap representation encoding both geometry and

✉ Corresponding author.

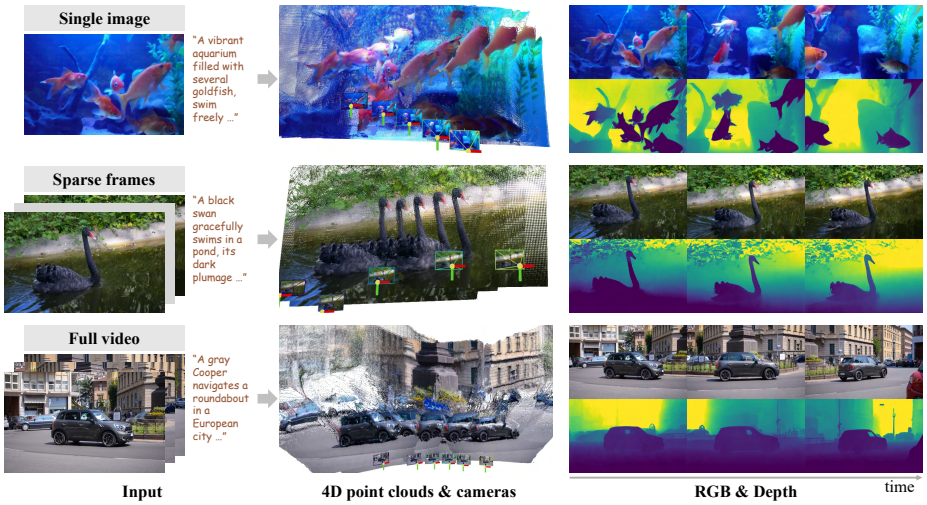

Fig. 1: One4D supports single-image-to-4D generation, sparse-frame-to-4D generation, and full-video reconstruction in a single model. It outputs synchronized RGB frames and pointmaps, visualized as 4D point clouds with cameras, and RGB-depth sequences.

camera information, enabling efficient feedforward 3D reconstruction. Monst3R [59] and VGGT [43], etc., show that pointmaps are effective for static 3D and dynamic 4D reconstruction. Meanwhile, several works extend video or multi-view diffusion models with 6D video representations (RGB+XYZ) for 4D reconstruction [15], 3D world generation [62], 3D generation and reconstruction [38], and 4D generation [6]. However, these methods typically specialize in either reconstruction or generation, or are restricted to static 3D scenes.

In this paper, we take a step further and propose One4D, a unified 4D generation and reconstruction framework that significantly enhances the fidelity of joint RGB-geometry modeling through innovative network designs.

We represent each 4D scene as RGB frames and pointmaps due to their flexibility and scalability, where pointmaps are 3-channel 2D (XYZ) videos analogous to RGB videos. Prior work [6, 15, 38, 62] shows that modern video VAEs can effectively encode and decode pointmaps. A central challenge in joint RGB-geometry modeling is to fully exploit the strong priors of a pretrained video diffusion model to generate distinct modalities. Existing diffusion-based geometry methods [15,16,62] typically couple RGB and geometry through channel-wise concatenation, which in our setting causes severe cross-modal interference and rapid degradation under low-resource fine-tuning.

To address this, we introduce the Decoupled LoRA Control (DLC), an efficient adaptation design guided by three goals: (i) preserve the base model's strong video priors under low-resource finetuning; (ii) decouple RGB and geometry generation to reduce mutual interference; (iii) enable sufficient cross-modal communication for pixel-level consistency. Concretely, DLC attaches two decou-

pled and modality-specific LoRA adapters to the video backbone, one for RGB and the other for geometry. The adapters share the frozen base parameters but *not* the base forward computation, forming two **fully decoupled computation branches**. The RGB branch maintains pretrained video quality, while the geometry branch adapts to the geometry video distribution. To synchronize the two modalities, we add zero-initialized control links [61] between a few corresponding layers across branches. These links gradually learn to transmit information for precise pixel-wise alignment. In practice, DLC prevents mode collapse and yields accurate geometry without sacrificing RGB fidelity.

To unify 4D generation and reconstruction tasks, we further propose Unified Masked Conditioning (UMC). UMC packs different condition types into a single conditioning video that differs in the sparsity of observed frames, filling unobserved frames with zeros. A single-image (with text) corresponds to pure generation. A set of sparse frames corresponds to mixed generation and reconstruction, and a full video corresponds to pure reconstruction. The conditioning video is fed to the diffusion backbone to guide the network to generate missing RGB frames and full pointmaps. With UMC, One4D transitions seamlessly between generation and reconstruction without any architectural changes.

We train One4D on a curated mixture of synthetic and real 4D datasets, combining accurate synthetic geometry and in-the-wild appearance. The resulting model achieves generalizable, high-quality 4D generation and reconstruction in a single framework. Our main contributions are:

- We introduce One4D, a unified 4D framework that bridges 4D generation and 4D reconstruction within a single video diffusion model, achieving strong performance across diverse dynamic scenarios.
- We propose Decoupled LoRA Control (DLC), which uses modality-specific LoRA branches and zero-initialized control links to preserve video priors while enabling accurate, consistent joint RGB-geometry generation.
- We design Unified Masked Conditioning (UMC), a single conditioning interface that supports various 4D generation and reconstruction tasks.

2 Related Work

Video generation models. Video generation models [10,18,21,27,28,41,55,67] have advanced rapidly with diffusion [11,30,33,35,36] and flow matching [7,22]. Modern systems combine 3D causal VAEs [4,17,48,57] with DiTs [30] to model spatio-temporal dynamics. Recent open-weight models such as Wan [41], HunyuanVideo [18], and Cosmos [27] demonstrate strong world-modeling abilities, suggesting possibilities for geometry generation. Image-to-video systems such as Wan [41] and LongCat-Video [26] further unify image-to-video, interpolation, and continuation through frame inpainting. One4D follows this unified-generation spirit with specific designs for joint RGB and geometry modeling.

Scalable geometry modeling. DUSt3R [45] introduces paired pointmaps that encode geometry and camera information in 2D maps, enabling scalable 3D reconstruction with transformers. Subsequent works [15, 39, 43, 44, 52, 59] extend

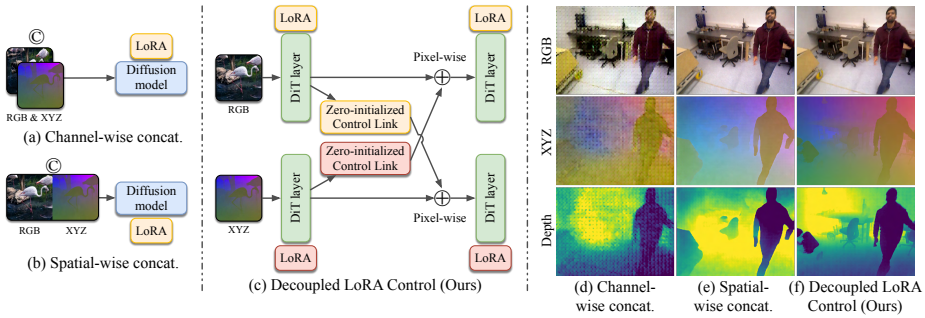

Fig. 2: Architecture and results comparison for joint RGB and geometry modeling. (a) Channel-wise and (b) spatial-wise concatenation feed RGB and XYZ into a single diffusion model with a shared LoRA branch. (c) Our Decoupled LoRA Control (DLC) employs two modality-specific LoRA branches with zero-initialized control links, achieving decoupled yet controlled RGB–XYZ joint generation. (f) Our Decoupled LoRA Control produces cleaner RGB and sharper, more consistent XYZ and depth than (e) spatial-wise concatenation, while channel-wise (d) concatenation severely degrades both appearance and geometry. Ⓒ denotes concatenation and ⊕ denotes pixel-wise addition.

this representation to multiview and dynamic reconstruction. Our method leverages these advances by adopting pointmaps as the geometry representation for joint RGB–geometry generation within a unified video model.

3D and 4D generation. Leveraging strong 2D diffusion models [3, 33] for 3D/4D advances rapidly. Several methods repurpose diffusion models for reconstructing depth and normal maps [16, 24], conditioning noisy geometry with clean RGB latents. JointNet [58] extends a text-to-image model to image-level RGB-depth/normal generation with a dense branch coupled to RGB, while JointDiT [20] uses a parallel depth branch with joint attention. In contrast, One4D targets video-level RGB+XYZ pointmaps for unified generation and reconstruction, where XYZ provides a shared 4D frame for camera recovery. Unlike ControlNet's one-way conditioning [61] or JointNet's dense coupling, DLC uses sparse bidirectional links between RGB and XYZ branches to avoid dense cross-modal interference. Other works use diffusion models as multiview generators conditioned on cameras, such as CAT3D [8], Bolt3D [38], and Stable virtual camera [68], and extend this to multi-view videos for 4D generation [49, 51]. Some other works focus on 3D object generation from images or text [32, 50, 60, 65].

Several methods model geometry and camera poses jointly via pointmaps, raymaps, or depth maps [6, 15, 25, 38, 62]. WVD [62] generates 6D (RGB+XYZ) static scenes from a single image using channel-wise concatenation of RGB and XYZ. 4DNeX [6] generates dynamic scenes via spatial-wise concatenation. Our method adopts the RGB+XYZ representation, but differs in two key aspects. (1) We unify 4D generation and reconstruction within a single model. (2) We introduce Decoupled LoRA Control (DLC), which achieves substantially higher quality and geometry accuracy than channel-wise or spatial-wise concatenation.

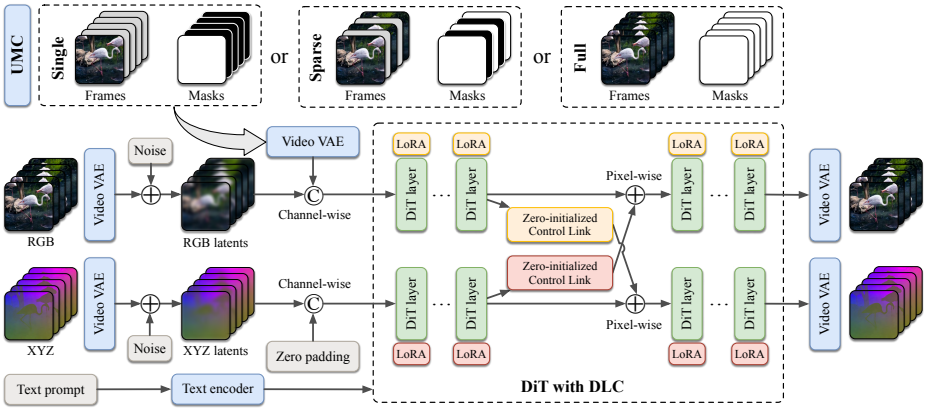

Fig. 3: Overview of the One4D framework. Unified Masked Conditioning (UMC) packs single-image, sparse-frame, and full-video inputs into a masked conditioning video. RGB and XYZ videos are encoded into latent spaces via video VAEs, and the conditioning latents are concatenated only with noisy RGB latents. These RGB and XYZ latents are then processed by a DiT backbone with Decoupled LoRA Control (DLC). DLC employs modality-specific LoRA branches to decouple computation, and zero-initialized cross-modal control links to learn pixel-wise consistency. The denoised RGB and XYZ latents are finally decoded into RGB frames and pointmaps.

3 Method

3.1 Overview

One4D extends a flow-matching video generation model [7,22,41] for joint RGB and geometry generation. As shown in Figure 3, given input images and a text prompt, the model generates synchronized RGB frames $\mathbf{X}_{\text{rgb}} \in \mathbb{R}^{3\times F\times H\times W}$ and pointmaps $\mathbf{X}_{\text{xyz}} \in \mathbb{R}^{3\times F\times H\times W}$, where each pointmap pixel stores the 3D coordinates of the corresponding RGB pixel. F denotes the number of frames and $H\times W$ the spatial resolution. At each training step, RGB and pointmaps are encoded by a video VAE into latents $\mathbf{z}_{\text{rgb}}, \mathbf{z}_{\text{xyz}} \in \mathbb{R}^{c\times f\times h\times w}$. We then sample a timestep $t \in [0,1]$ and Gaussian noise $\epsilon^t_{\text{rgb}}, \epsilon^t_{\text{xyz}} \sim \mathcal{N}(0, I)$, , and construct noisy latents using the Rectified Flow formulation [7,22]:

$$\mathbf{z}^t_{\text{rgb}} = t\mathbf{z}_{\text{rgb}} + (1-t)\epsilon^t_{\text{rgb}}; \qquad \mathbf{z}^t_{\text{xyz}} = t\mathbf{z}_{\text{xyz}} + (1-t)\epsilon^t_{\text{xyz}}. \tag{1}$$

The noisy latents and conditioning inputs are fed into DiTs [30] to predict velocities supervised by:

$$v^t_{\text{rgb}} = \mathbf{z}_{\text{rgb}} - \epsilon_{\text{rgb}}; \qquad v^t_{\text{xyz}} = \mathbf{z}_{\text{xyz}} - \epsilon_{\text{xyz}}. \tag{2}$$

and we train with a mean-squared error loss between predicted and ground-truth velocities for both modalities.

Based on this structure, we introduce *Decoupled LoRA Control* (DLC) for stable, modality-specific adaptation with pixel-level consistency between RGB

and geometry, and *Unified Masked Conditioning* (UMC) to handle 4D generation and reconstruction in a single model. After generation, camera poses and depthmaps are recovered from the pointmaps via a global optimization [15, 45].

3.2 Decoupled LoRA Control

Previous diffusion-based **reconstruction** methods, such as Marigold [16] and Geo4D [15], generate depthmaps or pointmaps conditioned on *clean* RGB inputs. They typically concatenate clean RGB latents with noisy geometry latents channel-wise, which works when RGB is fixed. However, in the **joint generation** setting, both RGB and geometry latents are noisy, and such direct coupling leads to severe quality degradation. WVD [62] still adopts channel-wise concatenation for joint RGB and XYZ generation, but is trained only on static scenes and requires extreme compute (over 1M steps on 64 A100s). As shown in Figure 2, under moderate compute, we observe that both channel-wise and spatial-wise concatenation [6] induce premature and excessive interaction between modalities, degrading RGB quality or preventing high-quality geometry learning.

To address these challenges, we propose Decoupled LoRA Control (DLC), which decouples RGB and XYZ computation to minimize cross-modal interference, while enabling pixel-wise communication in a gradually learnable manner.
Decoupled computation. DLC enforces **decoupling**. We maintain two branches for RGB and XYZ tokens, each equipped with a LoRA adapter, so the model can adapt to each modality without mutual degradation. For a submodule with LoRA in DiTs, the modality-specific computation can be written as:

$$\mathbf{z}'_{\text{rgb}} = \text{DiTSubmoduleWithRGBLoRA}(\mathbf{z}_{\text{rgb}}), \tag{3}$$

$$\mathbf{z}'_{\text{xyz}} = \text{DiTSubmoduleWithXYZLoRA}(\mathbf{z}_{\text{xyz}}) \tag{4}$$

Implementing the decoupled computation by simply duplicating all DiT parameters and attaching distinct LoRAs to each copy is infeasible for large-scale base models (14B parameters in our case). Instead, we share the frozen base parameters between modalities and add two separate LoRA adapters on each DiT submodule, while keeping the forward computation disjoint.

$$\mathbf{z}'_{\text{rgb}} = \text{DiTSubmodule}(\mathbf{z}_{\text{rgb}}) + \text{RGBLoRA}(\mathbf{z}_{\text{rgb}}), \tag{5}$$

$$\mathbf{z}'_{\text{xyz}} = \text{DiTSubmodule}(\mathbf{z}_{\text{xyz}}) + \text{XYZLoRA}(\mathbf{z}_{\text{xyz}}) \tag{6}$$

Each DiT submodule is evaluated once per modality, reusing weights but keeping computations decoupled. This drastically reduces memory usage compared to duplicating parameters, making finetuning feasible for large models. The decoupled design of DLC preserves pretrained video priors in RGB branch while allowing XYZ branch to adapt to pointmap generation, achieving high fidelity in both modalities without cross-modal degradation.
Control links. DLC's second principle is **control**. For 4D generation, RGB and geometry outputs must be pixel-wise consistent. Channel-wise concatenation enforces strong coupling but harms video quality, while spatial-wise concatenation

relies on non-pixel-aligned attention interactions that are relatively weak, as shown in Figure 2. To obtain strong yet controllable cross-modal consistency, we introduce lightweight *control links* to connect the two branches.

Let the DiT backbone have N layers, and let $\{L_{i_1}, \ldots, L_{i_m}\}$, with $1 \le i_1 < \cdots < i_m \le N$ denote m layers where we insert control links. At a linked layer l, features of one modality are updated by features from the other modality:

$$\begin{aligned} \hat{\mathbf{z}}^{(l)}_{\text{rgb}} &= \mathbf{z}^{(l)}_{\text{rgb}} + \text{ZCL}_{\text{rgb}\leftarrow\text{xyz}}\big(\mathbf{z}^{(l)}_{\text{xyz}}\big), \\ \hat{\mathbf{z}}^{(l)}_{\text{xyz}} &= \mathbf{z}^{(l)}_{\text{xyz}} + \text{ZCL}_{\text{xyz}\leftarrow\text{rgb}}\big(\mathbf{z}^{(l)}_{\text{rgb}}\big), \end{aligned} \tag{7}$$

where $\text{ZCL}_{\text{rgb}\leftarrow\text{xyz}}$ and $\text{ZCL}_{\text{xyz}\leftarrow\text{rgb}}$ are zero-initialized linear control links. The updated features $\hat{\mathbf{z}}^{(l)}_{\text{rgb}}$ are $\hat{\mathbf{z}}^{(l)}_{\text{xyz}}$ then passed to the next DiT layer. Zero initialization [61] keeps two branches fully independent at the start of training, preserving the pretrained video priors and the links gradually learn pixel-level alignment between RGB and geometry. In practice, we link a small subset of layers.

Compared to concatenation-based coupling, DLC enables a sparse, learned pathway for cross-modal communication. As shown in Figure 2, it achieves stronger pixel-wise consistency while maintaining high RGB fidelity and geometric accuracy. It also avoids the token-doubling issue of spatial-wise concatenation within a single attention operation, thereby reducing memory and compute cost.

3.3 Unified Masked Conditioning

We introduce Unified Masked Conditioning to express single-image, sparse-frame, and full-video conditioning within a single interface. Prior video generation models such as Wan [41] and LongCat-Video [26] unify different video generation tasks via frame inpainting. We extend this idea to unified 4D generation and reconstruction, where RGB and geometry are modeled jointly.

Condition construction. We assemble the available image conditions into a conditioning video $\mathbf{X}_c \in \mathbb{R}^{C\times F\times H\times W}$, matching the RGB shape, and fill unobserved frames with zeros. $\mathbf{X}_c$ is encoded by the video VAE into $\mathbf{z}_{\text{c}} \in \mathbb{R}^{c\times f\times h\times w}$, which is concatenated channel-wise with the RGB latents $\mathbf{z}_{\text{rgb}}$. We also build a binary mask $\mathbf{M}_c \in \mathbb{R}^{1\times F\times H\times W}$ [41] indicating observed vs. unobserved frames, reshape it to latent resolution $\mathbf{M}_c \in \mathbb{R}^{c_m\times f\times h\times w}$, and concatenate it with RGB latents. The DiT input is:

$$\mathbf{z}_{\text{input}} = \text{Concat}(\mathbf{z}_{\text{rgb}}, \mathbf{z}_{\text{c}}, \mathbf{M}_c) \tag{8}$$

Given $\mathbf{z}_{\text{c}}$ and $\mathbf{M}_c$, the model will generate missing RGB frames and the full set of pointmaps. This unified construction handles single-image, sparse-frame, and full-video conditioning without changing the architecture.

Controlling geometry. Since all XYZ frames are always generated, we do not feed $\mathbf{z}_{\text{c}}$ or $\mathbf{M}_c$ directly to the geometry branch, avoiding conditioning artifacts in geometry. Instead, conditioning signals reach the geometry branch through DLC control links from the RGB branch, allowing the geometry branch to focus on accurate 3D structure.

With the above designs, UMC makes One4D seamlessly switch between 4D generation and reconstruction tasks within one unified model.

3.4 Post-Optimization

After 4D generation, pointmaps are already in a consistent coordinate system and encode full geometry. We apply a simple global optimization to compute camera parameters and depth maps from the generated pointmaps, following MonST3R [59] and Geo4D [15]. Given the N generated point maps $\{\hat{\mathbf{X}}^i\}_{i=1}^N$, each $\hat{\mathbf{X}}^i \in \mathbb{R}^{H\times W\times 3}$, we recover $\{\mathbf{K}^i, \mathbf{R}^i, \mathbf{o}^i, \mathbf{D}^i\}_{i=1}^N$, where $\mathbf{K}^i$ is the intrinsic matrix, $\mathbf{R}^i$ is the world-to-camera rotation matrix, $\mathbf{o}^i$ is the camera center in the global frame, and $\mathbf{D}^i$ is the depth map of frame i.

For each frame i and pixel (u, v), the corresponding 3D point in the global reference frame is parameterized as:

$$\mathbf{X}^i_{uv} = \mathbf{R}^{i\top}\left(D^i_{uv}\,\mathbf{K}^{i-1}(u, v, 1)^\top\right) + \mathbf{o}^i, \tag{9}$$

where D^i_{uv} is the depth value at pixel (u, v) in frame i.

The predicted point maps $\hat{\mathbf{X}}^i$ is treated as the observations of $\mathbf{X}^i$ and they are aligned by the loss: $\mathcal{L}_{\mathrm{p}} = \sum_{i=1}^N \sum_{u,v} \left\|\mathbf{X}^i_{uv} - \hat{\mathbf{X}}^i_{uv}\right\|_1$. We minimize $\mathcal{L}_{\mathrm{p}}$ with respect to $\{\mathbf{K}^i, \mathbf{R}^i, \mathbf{o}^i, \mathbf{D}^i\}_{i=1}^N$, yielding globally consistent camera parameters and depth maps. We further regularize the camera trajectory by a temporally smooth loss [15, 59]:

$$\mathcal{L}_{\mathrm{s}}(\mathbf{R}, \mathbf{o}) = \sum_{i=1}^{N-1}\left(\left\|\mathbf{R}^{i\top}\mathbf{R}^{i+1} - \mathbf{I}\right\|_{\mathrm{f}} + \left\|\mathbf{o}^{i+1} - \mathbf{o}^i\right\|_2\right). \tag{10}$$

The final post-optimization objective is a weighted combination of the point-map alignment and trajectory smoothness losses: $\mathcal{L}_{\mathrm{all}} = \alpha_1\mathcal{L}_{\mathrm{p}} + \alpha_2\mathcal{L}_{\mathrm{s}}$.

On one H800 GPU, sampling 81 frames at 352×624 with 50 flow-matching steps takes ∼12min and 57.7GB peak VRAM. Since post-optimization does not solve geometry from scratch, its 500 iterations take only ∼30s.

4 Experiments

4.1 Implementation Details

Datasets. We construct 4D training datasets from both synthetic and real videos. We first collect dynamic synthetic 4D datasets OmniWorld-Game [69], BEDLAM [2], PointOdyssey [66], TarTanAir [46], which provide accurate geometry data. To increase dataset scale and diversity, we further annotate real-world videos in SpatialVID [42] with pseudo geometry using Geo4D [15], covering diverse in-the-wild dynamic scenes. Long videos are clipped into segments of about 81 frames and captioned with Gemini-2.0-Flash [31]. In total, we obtain about 17k synthetic and 17k real-world clips, with roughly 2M frames. Given camera

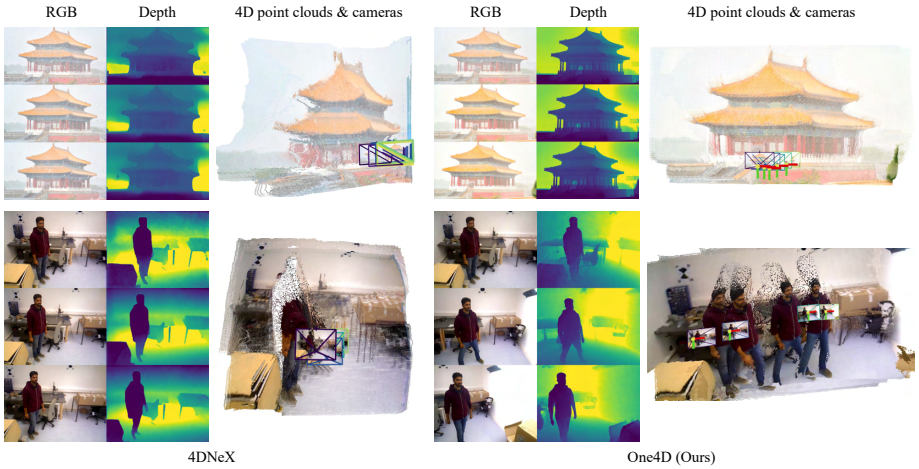

Fig. 4: Single-image-to-4D generation comparison between 4DNeX [6] and our One4D. Compared to 4DNeX, One4D produces more dynamic and realistic videos, sharper and cleaner depth, and more complete, coherent 4D point clouds with cameras.

parameters, depth maps are lifted to 3D pointmaps using the first frame as the global coordinate frame, and pointmaps are normalized to $[-1, 1]$.

Training. One4D is built on Wan2.1-Fun-V1.1-14B-InP [1], a community fine-tuned version of Wan2.1-I2V-14B [40] for video inpainting. We apply LoRA [12] with rank 64 to all linear layers in the DiT for both RGB and XYZ branches, with 685M parameters. We add DLC control links to five DiT layers, introducing 250.7M parameters. Overall, our model has 935.7M trainable parameters. Training is done on 8 NVIDIA H800 GPUs with batch size 1 per GPU and gradient accumulation of 4, for 5500 steps at a learning rate of 1×10^{-4}. We randomly switch among tasks by varying the number of masked frames. The task sampling ratios are 0.35 for single-image, 0.30 for sparse-frame, and 0.35 for full-video input. The maximum number of training frames is 81, at a resolution of 352×624.

Inference. At inference, we use 50 flow-matching steps with a classifier-free guidance scale of 6.0. The model jointly generates RGB frames and corresponding pointmaps (XYZ). From the pointmaps, we derive depth maps and estimate camera trajectories via post-optimization. For visualization, pointmaps (XYZ) are interpreted as RGB images. Depth maps are mapped to 3-channel images. We use Viser [56] to visualize 4D point clouds together with their camera trajectories, typically subsampling frames with a temporal stride to better show motion.

4.2 4D Generation

We compare One4D with Free4D [23], 4DNeX [6], and GenXD [64]. GenXD serves as an additional image-conditioned baseline with camera paths. The methods are compared by VBench [14], user studies, and qualitative comparisons. The

Fig. 5: Additional single-image-to-4D generation results from One4D. It can generate coherent 4D geometry for various types of scenes.

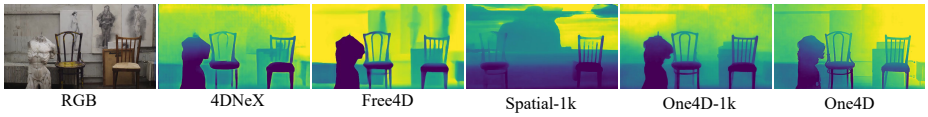

Fig. 6: Generation qualitative comparison of different methods. 4DNeX [6] and Free4D [23] cannot produce accurate and sharp geometry. Spatial-wise concatenation struggles with pixel-wise alignment between RGB and XYZ. Our method can produce accurate, sharp, and aligned XYZ.

user studies use held-out anonymous pairs with randomized order and the same image/prompt. They were conducted separately, using separate participant pools of 11, 6, and 11 for 4DNeX, Free4D, and GenXD, respectively, with 10 cases each.

As shown in Table 1, One4D is preferred over Free4D, 4DNeX, and GenXD across image-to-video consistency, motion dynamics, aesthetics, and geometry-related criteria such as depth and 4D coherence. VBench scores in Table 2 further show that One4D produces more natural motion while maintaining comparable image-to-video consistency. These results support the effectiveness of our decoupled design, which learns accurate geometry without sacrificing video quality.

Figure 4 qualitatively compares RGB frames, 4D point clouds, and depth maps generated by the 4DNex and our One4D. One4D generates finer geometric details, more accurate depth, and richer motion, whereas 4DNeX's spatial-wise concatenation struggles to produce fine-grained geometry, as also illustrated in Figure 2 and Figure 6. After lifting to 4D, One4D yields coherent scenes with stable backgrounds and naturally moving foreground objects, while 4DNeX often shows limited dynamics. Additional single-image-to-4D results in Figure 5 demonstrate that One4D produces high-quality geometry and realistic 4D structure across diverse indoor, outdoor, and static, dynamic scenarios.

These high-quality RGB videos and fine-grained, pixel-wise aligned geometry validate the design of Decoupled LoRA Control (DLC), which preserves the strong generative priors of the base model while learning accurate geometry and maintaining pixel-wise consistency.

4.3 4D Reconstruction

We evaluate 4D reconstruction in both full-video and sparse-frame settings on benchmarks to assess our unified generation–reconstruction framework. Sintel [5]

Table 1: User study comparing 4DNeX [6], Free4D [23], and GenXD [64] to One4D for 4D generation. Percentages indicate user preference. Our model shows a clear overall advantage, especially on geometry-related criteria.

Method	Consistency ↑	Dynamic ↑	Aesthetic ↑	Depthmap ↑	4D ↑
4DNeX [6]	21.0%	16.7%	17.7%	11.7%	10.0%
One4D (Ours)	**78.9%**	**83.3%**	**82.3%**	**88.3%**	**90.0%**
Free4D [23]	16.7%	5.0%	5.0%	6.7%	3.3%
One4D (Ours)	**83.3%**	**95.0%**	**95.0%**	**93.3%**	**96.7%**
GenXD [64]	5.5%	4.5%	4.5%	2.7%	6.4%
One4D (Ours)	**94.5%**	**95.5%**	**95.5%**	**97.3%**	**93.6%**

Table 2: VBench [14] for video quality. One4D significantly improves motion dynamics and aesthetic quality, while maintaining comparable image-to-video (I2V) consistency.

Method	Dynamic ↑	I2V consistency ↑	Aesthetic ↑
4DNeX [6]	25.6%	98.7%	61.9%
One4D (Ours)	**55.7%**	97.8%	**63.8%**

provides synthetic video with accurate ground-truth depthmaps, about 50 frames per video. Bonn [29] contains real dynamic indoor scenes, about 110 frames per video. TUM-dynamics [37] contains dynamic scenes sampled to 30 frames for each video. Our evaluation setting closely follows MonST3R [59] and Geo4D [15].

We compared the depth and camera accuracy derived from our generated pointmaps to reconstruction-only baselines. Marigold [16] and Depth-Anything-V2 [54] are single-image depth methods. NVDS [47], ChronoDepth [34] and DepthCrafter [13] operate on videos. In addition, Robust-CVD [19], Casual-SAM [63], MonST3R [59], CUT3R [44], Geo4D [15] jointly reconstruct video depth maps and camera poses. Although Geo4D [15] is used to annotate our real-world training data, we still report its numbers (Geo4D-Ref) as a reconstruction-only reference. For depth evaluation, we use Sintel [5] and Bonn [29], aligning predicted depths to the ground truth. We report the absolute relative error (Abs Rel) and the percentage of inlier points with $\delta < 1.25$. For camera evaluation, we use Sintel [5] and TUM-dynamics [37], and report Absolute Trajectory Error (ATE), Relative Pose Error in translation (RPE-T), and Relative Pose Error in rotation (RPE-R).

Full-video-to-4D. Table 3 reports full-video depth accuracy. Among pointmap-based methods, One4D clearly outperforms MonST3R [59] and CUT3R [44] on Sintel (70.4% vs 58.5 / 62.0 on $\delta < 1.25$), and remains close to the reconstruction-only Geo4D-ref [15], even though those models are trained purely for reconstruction and One4D is trained once for both 4D generation and reconstruction. This indicates that our Decoupled LoRA Control and Unified Masked Conditioning allow a single generative model to recover highly accurate geometry. Results for

Table 3: Trained as a unified generation–reconstruction model (G&R), One4D outperforms reconstruction-only (R) pointmap-based methods such as MonST3R and CUT3R, and remains reasonably close to the reconstruction-only Geo4D-ref, demonstrating effective geometry reconstruction within a unified architecture.

Method	Task	Sintel [5]		Bonn [29]	
		Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Marigold [16]	R	0.532	51.5	0.091	93.1
Depth-Anything [54]	R	0.367	55.4	0.106	92.1
NVDS [47]	R	0.408	48.3	0.167	76.6
ChronoDepth [34]	R	0.687	48.6	0.100	91.1
DepthCrafter [13]	R	0.270	69.7	0.071	97.2
Robust-CVD [19]	R	0.703	47.8	-	-
CasualSAM [63]	R	0.387	54.7	0.169	73.7
MonST3R [59]	R	0.335	58.5	0.063	96.4
CUT3R [44]	R	0.311	62.0	0.070	96.7
Geo4D-ref [15]	R	0.205	73.5	0.059	97.2
One4D (Ours)	G&R	0.273	70.4	0.092	93.7

CUT3R are obtained using the official Geo4D evaluation scripts. Results of other baselines are taken from MonST3R [59] and Geo4D [15] papers.

Camera accuracy in Table 4 shows that One4D achieves competitive ATE and RPE scores within the same range of Geo4D-ref and other reconstruction baselines, confirming that our pointmaps are sufficiently accurate to support robust camera estimation. Qualitative results in Figure 7 further show that One4D can recover fine geometric structures (e.g., bamboo leaves, ropes) and robustly handles challenging scenes such as dense bamboo forests.

Overall, the strong depth and camera reconstruction results across Sintel, Bonn, and TUM-dynamics highlight that our designs of DLC and UMC enable One4D to effectively reconstruct 4D geometry and camera trajectories, even if we train it as a single unified model for both generation and reconstruction.

Sparse-frame-to-4D. In the sparse-frame-to-4D setting, we keep the first frame and last frame of each video and uniformly sample additional frames in between, with the total number of observed frames controlled by a sparsity ratio. The model must generate all missing RGB frames and the complete pointmaps.

Table 5 reports depth accuracy on Sintel and Bonn with different spatial ratios. Remarkably, with only 50% or 25% of frames, One4D is almost as accurate as in the full-video setting on both datasets. Even at sparsity 0.10, degradation is modest. Moreover, under extreme sparsity (e.g., 0.05 or 0.03), the model still produces reasonable geometry from only 2 or 3 frames, typically just the first and last frames. This is also partly because these boundary frames already capture sufficient background information.

Overall, the sparse-frame evaluation shows that our model can reliably generate unobserved RGB frames and corresponding 4D structure from very sparse

Table 4: Camera trajectory accuracy. The *Task* column distinguishes reconstruction-only (R) methods from our unified generation–reconstruction model (G&R). One4D achieves camera accuracy comparable to strong reconstruction-only baselines, indicating our unified 4D model performs competitive camera estimation.

Method	Task	Sintel [5]			TUM-dynamics [37]		
		ATE↓	RPE-T↓	RPE-R↓	ATE↓	RPE-T↓	RPE-R↓
Robust-CVD [19]	R	0.360	0.154	3.443	0.153	0.026	3.528
CasualSAM [63]	R	0.141	0.035	0.615	0.071	0.010	1.712
MonST3R [59]	R	0.108	0.042	0.732	0.063	0.009	1.217
CUT3R [44]	R	0.208	0.062	0.610	0.046	0.014	0.446
Geo4D-ref [15]	R	0.185	0.063	0.547	0.073	0.020	0.635
One4D (Ours)	G&R	0.213	0.057	0.818	0.129	0.022	1.447

Table 5: Depth accuracy for sparse-frame-to-4D generation on Sintel and Bonn. Using only a fraction of frames (Sparsity), One4D remains competitive and degrades gracefully, indicating strong geometry generation from sparse observations.

Sparsity	Sintel [5]		Bonn [29]	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
0.50	0.314	70.3	0.094	93.5
0.25	0.443	67.7	0.094	93.3
0.10	0.453	64.0	0.099	92.9
0.05	0.641	57.6	0.151	87.2
0.04	-	-	0.191	82.5
0.03	-	-	0.277	71.1
Full Model	**0.273**	**70.4**	**0.092**	**93.7**

observations, complementing the single-image-to-4D generation results and highlighting the strength of our unified generation–reconstruction design.

4.4 Ablation Study

Dataset. We investigate how the synthetic data can influence the reconstruction accuracy. As shown in Table 7, with only synthetic data trained by 1k steps, the reconstruction metrics improve, especially on the synthetic Sintel benchmark. Since our method targets unified generation and reconstruction, in our main experiment, we still add pseudo-labeled real-world videos to enrich the in-the-wild appearance and motion diversity.

Spatial-wise concatenation. Figure 2 already shows that channel-wise concatenation collapses training. We provide ablation of spatial-wise concatenation trained on our datasets and settings for fair comparison. As shown in Table 8, spatial-wise concatenation (Spatial-1k) yields consistently worse geometry under different sparsity ratios than One4D (One4D-1k). Figure 6 also visualizes RGB-XYZ misalignment of spatial-wise concatenation (Spatial-1k).

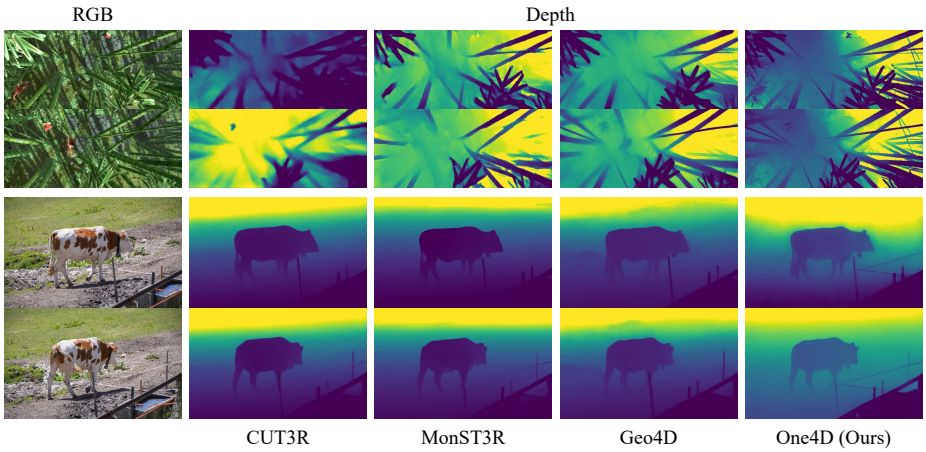

Fig. 7: Qualitative full-video 4D reconstruction comparison with CUT3R [44], MonST3R [59], and Geo4D [15]. The proposed One4D recovers sharper object boundaries and more accurate depth, especially on thin structures and challenging geometry.

Table 6: Ablation on CFG scale and training steps for depth reconstruction. Our model is robust to CFG choice and attains good accuracy with few training steps, improving with longer training.

Setting	Sintel [5]		Bonn [29]	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
CFG=4	0.257	71.5	0.092	94.0
CFG=5	0.259	70.9	0.090	94.1
Step=1000	0.331	65.4	0.114	88.9
Step=3000	0.284	68.1	0.097	91.8
One4D (Ours)	**0.273**	**70.4**	**0.092**	**93.7**

Classifier-free guidance (CFG) scale. We use CFG = 6 as the default setting in all main experiments. As shown in Table 6, using CFG = 4 or CFG = 5 yields very similar depth accuracy, indicating that One4D is robust to the choice of guidance scale. This robustness simplifies deployment in real-world applications.
Training steps. We study the effect of training steps by training models with 1K, 3K, and 5.5K optimization steps. Even with only 1k steps, One4D already achieves reasonable accuracy, and at 3k steps, it is close to the full model. Compared to WVD [62], which requires around 1M steps over two weeks on 64 A100 GPUs with channel-wise concatenation, our decoupled design adapts the base video model with much fewer training steps. This efficiency supports our claim that the proposed architecture preserves and effectively leverages pretrained video priors. As the number of training steps increases, performance generally improves, indicating promising potential for further gains when scaling to larger datasets and more computation.

Table 7: Effect of synthetic data on reconstruction accuracy. Synthetic-only training (1k steps) improves metrics, especially on Sintel.

Method	Sintel [5]		Bonn [29]	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Sythentic-1k	0.281	69.3	0.099	92.0
One4D-1k	0.331	65.4	0.114	88.9

Table 8: Bonn benchmark results trained by 1k steps, comparing DLC and spatial-wise concatenation under different sparsity ratios.

Method	Full		$s = 0.05$		$s = 0.04$	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Spatial-1k	0.121	87.4	0.186	76.8	0.227	73.2
One4D-1k	**0.114**	**88.9**	**0.167**	**83.0**	**0.208**	**76.9**

5 Limitations

One4D builds on pretrained video diffusion priors, so its generation fidelity is related to how well the backbone generalizes to target motions and camera patterns. It may also remain challenging in difficult in-the-wild cases with ambiguous geometry or complex non-rigid motion. Expanding data coverage and incorporating stronger physical priors are promising directions for future work.

6 Conclusion

We presented One4D, a unified 4D framework that bridges 4D generation and 4D reconstruction within a single video diffusion model. We introduced Decoupled LoRA Control for robust joint RGB and geometry modeling. We proposed Unified Masked Conditioning, a simple conditioning scheme that seamlessly handles pure generation, mixed generation–reconstruction, and pure reconstruction without modifying the architecture. Trained on a curated mixture of synthetic and real 4D data, One4D achieves generalizable, high-quality 4D results and takes a step toward geometry-aware world simulation with video foundation models.

Acknowledgements

The research is supported in part by Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321, ITF PRP/046/24FX, Science & Technology Cooperation Program of Shandong under grant No. SDST26EG01, SAIL Research Project, HKUST-Zeekr Coolaborative Research Fund, and Tencent Rhino-Bird Focused Research Program.

References

1. Alibaba PAI Team: Wan2.1-Fun-V1.1-14B-InP. https://huggingface.co/alibaba-pai/Wan2.1-Fun-V1.1-14B-InP (2025), accessed: 24 June 2026
2. Black, M.J., Patel, P., Tesch, J., Yang, J.: Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In: CVPR (2023)
3. Black Forest Labs: Flux. https://github.com/black-forest-labs/flux (2024), gitHub repository. Accessed: 24 June 2026
4. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
5. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: ECCV (2012)
6. Chen, Z., Liu, T., Zhuo, L., Ren, J., Tao, Z., Zhu, H., Hong, F., Pan, L., Liu, Z.: 4dnex: Feed-forward 4d generative modeling made easy. arXiv preprint arXiv:2508.13154 (2025)
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
8. Gao, R., Hołyński, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P., Barron, J.T., Poole, B.: Cat3d: create anything in 3d with multi-view diffusion models. In: NeurIPS (2024)
9. Genmo Team: Mochi 1. https://github.com/genmoai/models (2024), accessed: 24 June 2026
10. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity long video generation. arXiv preprint arXiv:2211.13221 (2022)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
12. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: ICLR (2022)
13. Hu, W., Gao, X., Li, X., Zhao, S., Cun, X., Zhang, Y., Quan, L., Shan, Y.: Depthcrafter: Generating consistent long depth sequences for open-world videos. In: CVPR (2025)
14. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
15. Jiang, Z., Zheng, C., Laina, I., Larlus, D., Vedaldi, A.: Geo4d: Leveraging video generators for geometric 4d scene reconstruction. In: ICCV (2025)
16. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: CVPR (2024)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
18. Kong, W., Tian, Q., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
19. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: CVPR (2021)
20. Kwon, B.K., Dai, Q., Lee, H., Luo, C., Oh, T.H.: Jointdit: Enhancing rgb-depth joint modeling with diffusion transformers. In: ICCV. pp. 25261–25271 (October 2025)

21. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131 (2024)
22. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
23. Liu, T., Huang, Z., Chen, Z., Wang, G., Hu, S., Shen, L., Sun, H., Cao, Z., Li, W., Liu, Z.: Free4d: Tuning-free 4d scene generation with spatial-temporal consistency. In: ICCV. pp. 25571–25582 (October 2025)
24. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. In: CVPR (2024)
25. Lu, Y., Zhang, J., Fang, T., Nahmias, J.D., Tsin, Y., Quan, L., Cao, X., Yao, Y., Li, S.: Matrix3d: Large photogrammetry model all-in-one. In: CVPR (2025)
26. Meituan LongCat Team: Longcat-video technical report (2025), https://arxiv.org/abs/2510.22200, accessed: 24 June 2026
27. NVIDIA, Agarwal, N., Ali, A., Bala, M., Balaji, Y., et al.: Cosmos world foundation model platform for physical ai (2025), https://arxiv.org/abs/2501.03575, accessed: 24 June 2026
28. OpenAI: Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/ (2024), accessed: 24 June 2026
29. Palazzolo, E., Behley, J., Lottes, P., Giguere, P., Stachniss, C.: Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In: IROS (2019)
30. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
31. Pichai, S., Hassabis, D., Kavukcuoglu, K.: Introducing gemini 2.0: our new ai model for the agentic era. Online at https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/ (2024), accessed: 24 June 2026
32. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
34. Shao, J., Yang, Y., Zhou, H., Zhang, Y., Shen, Y., Guizilini, V., Wang, Y., Poggi, M., Liao, Y.: Learning temporally consistent video depth from video diffusion priors. In: CVPR (2025)
35. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
36. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
37. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS (2012)
38. Szymanowicz, S., Zhang, J.Y., Srinivasan, P., Gao, R., Brussee, A., Holynski, A., Martin-Brualla, R., Barron, J.T., Henzler, P.: Bolt3D: Generating 3D Scenes in Seconds. In: ICCV (2025)
39. Tang, Z., Fan, Y., Wang, D., Xu, H., Ranjan, R., Schwing, A., Yan, Z.: Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In: CVPR (2025)
40. Wan-AI Team: Wan2.1-I2V-14B-480P. https://huggingface.co/Wan-AI/Wan2.1-I2V-14B-480P (2025), accessed: 24 June 2026
41. Wan Team, Wang, A., Ai, B., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)

42. Wang, J., Yuan, Y., Zheng, R., Lin, Y., Gao, J., Chen, L.Z., Bao, Y., Zhang, Y., Zeng, C., Zhou, Y., et al.: Spatialvid: A large-scale video dataset with spatial annotations. arXiv preprint arXiv:2509.09676 (2025)
43. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025)
44. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: CVPR (2025)
45. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR (2024)
46. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. In: IROS (2020)
47. Wang, Y., Shi, M., Li, J., Huang, Z., Cao, Z., Zhang, J., Xian, K., Lin, G.: Neural video depth stabilizer. In: ICCV (2023)
48. Wu, P., Zhu, K., Liu, Y., Zhao, L., Zhai, W., Cao, Y., Zha, Z.J.: Improved video vae for latent video diffusion model. In: CVPR (2025)
49. Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J.T., Holynski, A.: Cat4d: Create anything in 4d with multi-view video diffusion models. In: CVPR (2025)
50. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In: NeurIPS (2024)
51. Xie, Y., Yao, C.H., Voleti, V., Jiang, H., Jampani, V.: Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. In: ICLR (2025)
52. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: CVPR (2025)
53. Yang, J., Yang, S., Gupta, A.W., Han, R., Fei-Fei, L., Xie, S.: Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. arXiv preprint arXiv:2412.14171 (2024)
54. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. In: NeurIPS (2024)
55. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
56. Yi, B., Kim, C.M., Kerr, J., Wu, G., Feng, R., Zhang, A., Kulhanek, J., Choi, H., Ma, Y., Tancik, M., et al.: Viser: Imperative, web-based 3d visualization in python. arXiv preprint arXiv:2507.22885 (2025)
57. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., et al.: Language model beats diffusion–tokenizer is key to visual generation. In: ICLR (2024)
58. Zhang, J., Li, S., Lu, Y., Fang, T., McKinnon, D., Tsin, Y., Quan, L., Yao, Y.: Jointnet: Extending text-to-image diffusion for dense distribution modeling. In: ICLR (2024), https://arxiv.org/abs/2310.06347, accessed: 24 June 2026
59. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. In: ICLR (2025)
60. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. TOG pp. 1–20 (2024)
61. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)

62. Zhang, Q., Zhai, S., Ángel Bautista, M., Miao, K., Toshev, A., Susskind, J., Gu, J.: World-consistent video diffusion with explicit 3d modeling. In: CVPR (2025)
63. Zhang, Z., Cole, F., Li, Z., Rubinstein, M., Snavely, N., Freeman, W.T.: Structure and motion from casual videos. In: ECCV (2022)
64. Zhao, Y., Lin, C.C., Lin, K., Yan, Z., Li, L., Yang, Z., Wang, J., Lee, G.H., Wang, L.: Genxd: Generating any 3d and 4d scenes. In: ICLR (2025)
65. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202 (2025)
66. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: CVPR (2023)
67. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)
68. Zhou, J., Gao, H., Voleti, V., Vasishta, A., Yao, C.H., Boss, M., Torr, P., Rupprecht, C., Jampani, V.: Stable virtual camera: Generative view synthesis with diffusion models. arXiv preprint arXiv:2503.14489 (2025)
69. Zhou, Y., Wang, Y., Zhou, J., Chang, W., Guo, H., Li, Z., Ma, K., Li, X., Wang, Y., Zhu, H., et al.: Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. arXiv preprint arXiv:2509.12201 (2025)