HunyuanWorld-Voyager: Technical Report

Tencent Hunyuan*

https://3d-models.hunyuan.tencent.com/world/
https://huggingface.co/tencent/HunyuanWorld-Voyager
https://github.com/Tencent-Hunyuan/HunyuanWorld-Voyager

Abstract

While recent HunyuanWorld 1.0 enables explorable 3D world generation, challenges remain with occluded views and limited exploration range. To address these limitations, we introduce ***Voyager*** for consistent world extrapolation, which is a video diffusion framework generating world-consistent 3D point-cloud sequences from a single image with user-defined camera path. Unlike existing approaches, Voyager achieves end-to-end scene generation and reconstruction with inherent consistency across frames, eliminating the need for 3D reconstruction pipelines (*e.g.,* structure-from-motion or multi-view stereo). Our method integrates three key components: 1) **World-Consistent Video Diffusion**: A unified architecture that jointly generates aligned RGB and depth video sequences, conditioned on existing world observation to ensure global coherence 2) **Long-Range World Exploration**: An efficient world cache with point culling and an auto-regressive inference with smooth video sampling for iterative scene extension with context-aware consistency, and 3) **Scalable Data Engine**: A video reconstruction pipeline that automates camera pose estimation and metric depth prediction for arbitrary videos, enabling large-scale, diverse training data curation without manual 3D annotations. Collectively, these designs result in a clear improvement over existing methods in visual quality and geometric accuracy, with versatile applications.

∗ Team contributors are listed in the end of report.

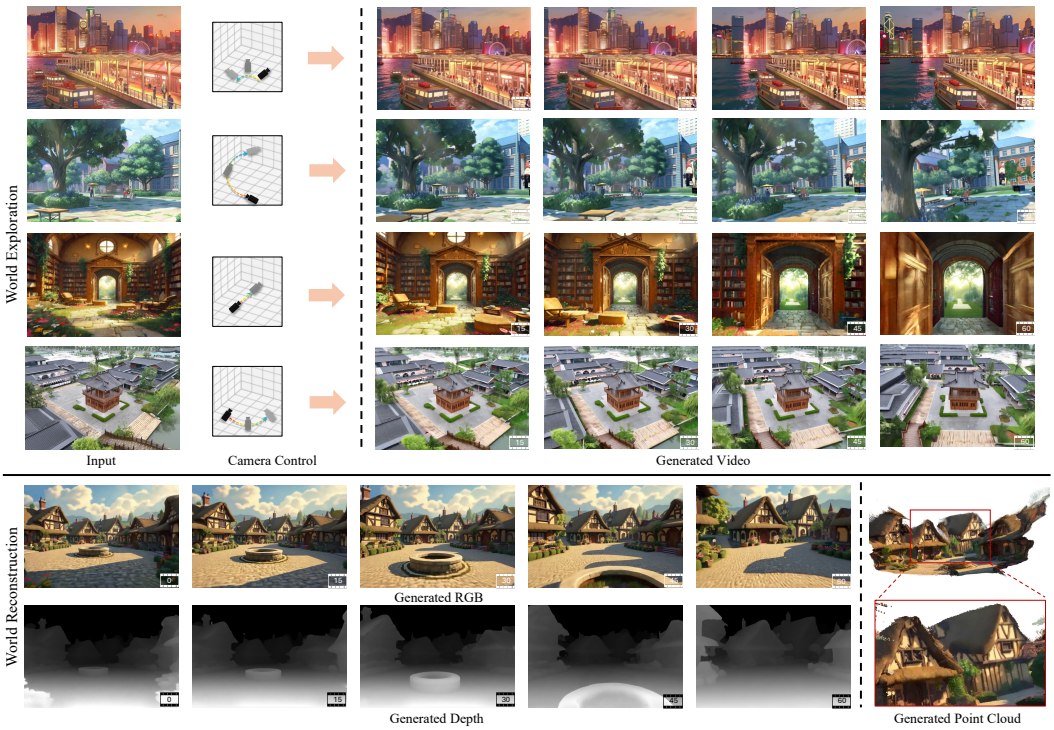

Figure 1: *Voyager* is a world-consistent video generation and reconstruction framework. Up: Voyager can generate 3D-consistent scene videos for world exploration following custom camera trajectories. Bottom: Voyager jointly generates aligned depth and RGB video for effective and direct 3D reconstruction.

1 Introduction

The creation of high-fidelity, explorable 3D scenes that users can navigate seamlessly, powers broad applications ranging from video gaming and film production to robotic simulation [33]. Yet, traditional workflows for constructing such 3D worlds remain bottlenecked by manual effort, requiring painstaking layout design, asset curation, and scene composition. While recent data-driven methods [40, 49, 25, 41, 21] have shown promise in generating objects or simple scenes, their ability to scale to complex scenes is limited by the scarcity of high-quality 3D scene data. This gap highlights the need for frameworks that enable scalable generation of user-navigable virtual worlds with 3D consistency.

Recently, a growing number of works [6, 8, 37, 48, 24, 50, 27, 3] have explored the use of novel view synthesis (NVS) and video generation as alternative paradigms for world modeling. These methods, while demonstrating impressive capabilities in generating visually appealing and semantically rich content, still face several challenges. **1) Long-Range Spatial Inconsistency.** Due to the absence of explicit 3D structural grounding, they often struggle to maintain spatial consistency and coherent viewpoint transitions during the generation process, especially when generating videos with long-range camera trajectories. **2) Visual Hallucination**. While several works [27, 3] have attempted to leverage 3D conditions to enhance geometric consistency, they typically rely on partial RGB images as guidance, *i.e.*, novel-view images rendered from point clouds reconstructed with input views. However, such representation may introduce significant visual hallucinations in complex scenes, such as the incorrect occlusions in Figure. 2, which may introduce inaccurate supervision during training. **3) Post-hoc 3D Reconstruction.** While these approaches can synthesize visually satisfying content, post-hoc 3D reconstructions are still required to obtain usable 3D content. This process is time-consuming and inevitably introduces geometric artifacts [38], making it inadequate for real-world applications.

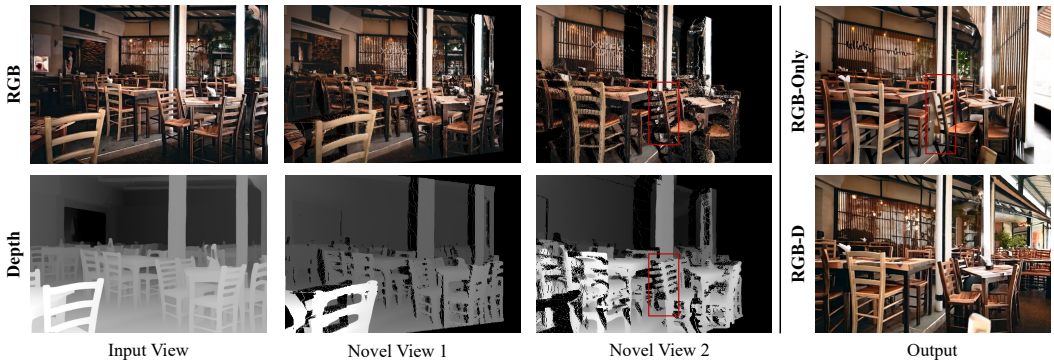

Figure 2: Partial RGB images and partial depth maps rendered from point clouds at different frames. In scenarios involving complex occlusion relationships, partial RGB images often exhibit significant visual artifacts. In contrast, partial depth maps can accurately represent occlusions.

To address these challenges, we introduce *Voyager* [13], a framework designed to synthesize long-range, world-consistent RGB-D(epth) videos from a single image and user-specified camera trajectories. At the core of Voyager is a novel **world-consistent video diffusion** that utilizes an expandable world caching mechanism to ensure spatial consistency and avoids visual hallucination. Starting from an image, we construct an initial world cache by unprojecting it into 3D space with a depth map. This 3D cache is then projected into target camera views to obtain partial RGB-D observations, which guides the diffusion model to maintain coherence with the accumulated world state. Crucially, the generated frames are fed back to update and expand the world cache, creating a closed-loop system that supports arbitrary camera trajectories while maintaining geometric coherence.

Unlike methods [48, 24, 27, 3] relying only on RGB conditioning, Voyager explicitly leverages depth information as a spatial prior, enabling more accurate 3D consistency during video generation. By simultaneously generating aligned RGB and depth sequences, our framework supports direct 3D scene reconstruction without requiring additional 3D reconstruction steps like structure-from-motion.

Despite promising performance, diffusion models struggle to generate long videos in a single pass. To enable **long-range world exploration**, we propose world caching scheme and smooth video sampling for auto-regressive scene extension. Our world cache accumulates and maintains point clouds from all previously generated frames, expanding as video sequences grow. To optimize computational efficiency, we design a point culling method to detect and remove redundant points with real-time rendering, minimizing memory overhead. Leveraging cached point clouds as a proxy, we develop a smooth sampling strategy that auto-regressively extends video length while ensuring smooth transitions between clips.

Training such a model requires large-scale videos with accurate camera poses and depth, but existing datasets often lack these annotations. To address this, we introduce a data engine for **scalable video reconstruction** that automatically estimates camera poses and metric depth for arbitrary scene videos. With metric depth estimation, our data engine ensures consistent depth scales across diverse sources, enabling high-quality training data generation. Using this pipeline, we compile a dataset of over 100,000 video clips, combining real-world captures and synthetic Unreal Engine renders.

Extensive experiments demonstrate the effectiveness of Voyager in scene video generation and 3D world reconstruction. Benefiting from joint depth modeling, our results in Figure 1 exhibit more coherent geometry, which not only enable direct 3D reconstruction but also support infinite world expansion while preserving the original spatial layout. Additionally, we explore applications such as 3D generation, video transfer, and depth estimation, further showcasing the potential of Voyager in advancing spatial intelligence.

Our contributions can be summarized as:

- We introduce Voyager, a world-consistent video diffusion model for scene generation. To the best of our knowledge, Voyager is the first video model that jointly generates RGB and depth sequences with given camera trajectories.

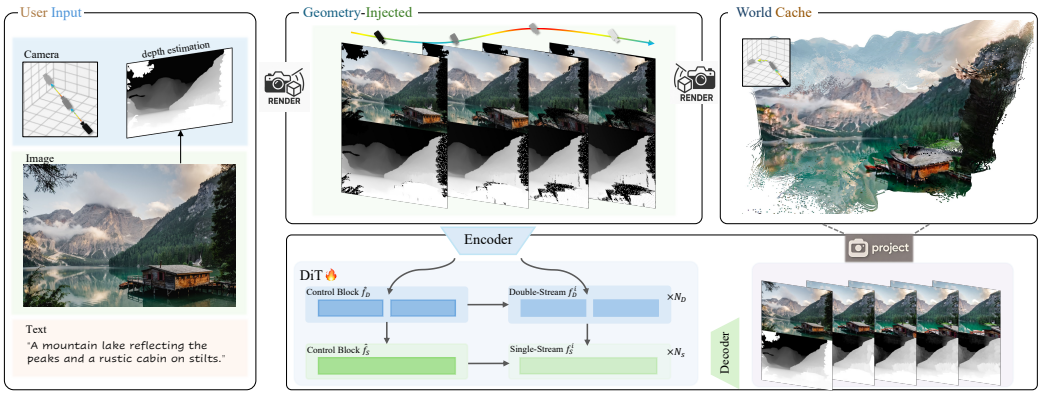

Figure 3: Overview of Voyager: Given the input image and camera trajectories, we first render partial RGB images and depth maps for each viewpoint as the condition for video generation. Our world-consistent video diffusion model is trained to generate RGB-D frames simultaneously, thus supporting the direct reconstruction of the 3D world. The projected points are store in our world cache efficiently, which can be rendered as condition for the next round generation.

- We propose an efficient world caching scheme and auto-regressive video sampling approach, extending Voyager to world reconstruction and infinite world exploration.
- We propose a scalable video data engine for camera and metric depth estimation, with over 100,000 training pairs prepared for the video diffusion model.

2 Technical Details

Given an image $I_0 \in \mathbb{R}^{3\times H\times W}$, our goal is to create an explorable world based on a user-defined camera trajectory. However, there is a gap between video generation and 3D world modeling, which mainly stems from three aspects: (1) the inconsistency of long-range video extension, (2) the hallucination of visual conditions for video generation, and (3) the incapability of reconstructing the world from video outputs. To address these issues, we propose ***Voyager***, a world-consistent video generation framework that can directly produce rgb-depth frames with corresponding camera parameters for long-range world exploration. In this section, we first introduce a geometry-injected frame condition to compensate for perceptual hallucination under visual conditions. (Sec. 2.1). With this input condition, we propose a depth-fused video diffusion model to ensure spatial consistency and context-based blocks to enhance its viewpoint control (Sec. 2.2). For 3D world reconstruction and long-range exploration, we propose world caching with point culling and smooth video sampling in the auto-regressive inference (Sec. 2.3). We further propose a scalable video data engine to prepare camera and metric depth for the training of the above model (Sec. 2.4).

2.1 Geometry-Injected Frame Condition

For the control of video viewpoint, camera parameter [50, 1] is a straightforward condition, but this implicit condition is nontrivial to the training of video models. Recent works [24, 27, 3] attempt to reconstruct the point cloud $p \in \mathbb{R}^{N\times 6}$ from videos as an explicit control, where N is the number of points and each point is represented by 6D coordinates (x, y, z, r, g, b). The warped RGB condition $\hat{I}_v$ for a novel view v can then be rendered according to the camera, which is a partial image with blank regions.

Nonetheless, such a partial RGB image is insufficient to ensure spatial consistency, *e.g.*, complex occlusion relationships in a scene may lead to visual hallucinations. To enforce spatially consistent control during training, we introduce an additional geometric condition partial depth map, which is aligned with the partial RGB image. Specifically, we first estimate the depth map D_k and corresponding camera parameters c_k for each frame I_k of the video. Since only the first frame is visible in video inference, we create a point cloud p_0 by projecting D_0 with c_0. For the k-th frame, its

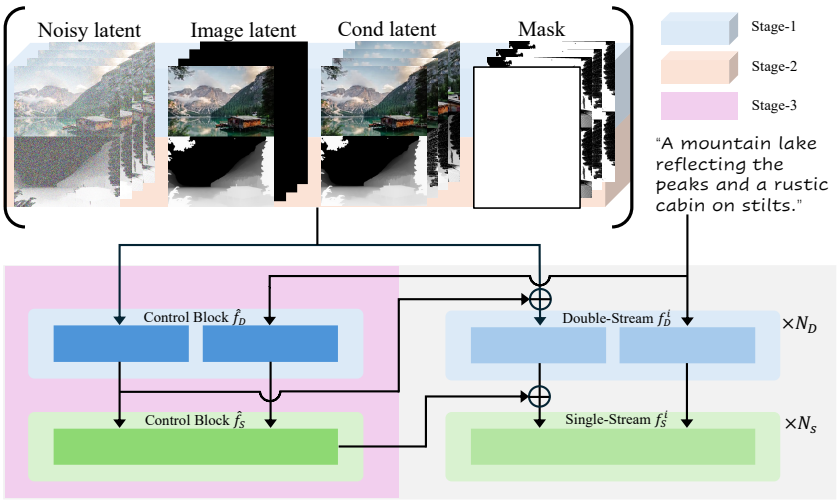

Figure 4: Details of world-consistent diffusion model. We concatenate RGB and depth in the spatial dimension and inject context-based control blocks to enhance the geometry consistency.

partial image $\hat{I}_k$ and partial depth $\hat{D}_k$ are acquired by masking the invisible region with the rendering mask $M_k = \text{render}(p_0, c_k)$.

2.2 World-Consistent Video Diffusion

Conditioned with partial RGB and depth maps, our intention is to generate plausible content for the invisible regions, ensuring consistency with the spatial information provided by the partial conditions.

For this purpose, the common practice [18, 27] is to concatenate the condition latents $\mathbf{z}_{\text{rgb}}$ and $\mathbf{z}_{\text{depth}}$ with original noisy latents $\mathbf{z}_t$ along the **channel** axis and project the concatenated latents back to the Transformer dimension via the patch-embedding layer f_{emb}: $\mathbf{z}'_{t,0} = f_{\text{emb}}(\text{concat}(\mathbf{z}_t, \mathbf{z}_{\text{rgb}}, \mathbf{z}_{\text{depth}}))$. Then, the projected latents $\mathbf{z}'_{t,0}$ are fed to double-stream and single-stream blocks sequentially, which is formulated as,

$$\mathbf{z}'_{t,i}, \mathbf{z}'_{y,i} = f_D^i(\mathbf{z}'_{t,i-1}, \mathbf{z}'_{y,i-1}, t), i = 1, ..., N_D, \tag{1}$$

$$\mathbf{z}''_{t,i} = f_S^i(\mathbf{z}''_{t,i-1}, t), i = 1, ..., N_S, \tag{2}$$

where $\mathbf{z}'_y$ is the text latents. N_D and N_S denote the block number of each stream. $\mathbf{z}''_{t,0}$ is initialized as the concatenation of $\mathbf{z}'_{t,N}$ and $\mathbf{z}'_{y,N}$.

Although the video model can best preserve the pre-trained parameters in this way, the spatial conditions is only used in channel-wise. The missing parts in our partial maps can range from small cracks to large blank areas, depending on the extent of the viewpoint change. This trivial solution struggles to handle variable situations. Therefore, as shown in Figure 4, we propose depth-fused video generation and context-based control enhancement to improve the video model.

Depth-Fused Video Generation. Instead of relying solely on partial depth as the input condition for completing the missing regions in the RGB frames, we propose to simultaneously generate both complete RGB and depth frames. As a result, the video model can take advantage of DiT's full-attention structure, allowing for the interaction of visual and geometric information at the pixel level. To this end, we concatenate the rgb and depth images along the **height** axis as $\mathcal{I}_k = [I_k, \Phi, D_k]_h$, as well as condition maps $\hat{\mathcal{I}}_k = [\hat{I}_k, \Phi, \hat{D}_k]_h$ and masks $M_k = [M_k, \Phi, M_k]_h$. Here, we add a placeholder row Φ between the rgb and depth images to help the model separate these two types of content. The new video latents are presented as $\mathbf{z}'_{t,0} = f_{\text{emb}}(\text{concat}(\mathbf{z}_t, \hat{\mathbf{z}}_i, \hat{\mathbf{z}}_0, m))$, where $\hat{\mathbf{z}}_0$ is the latent of $[\hat{\mathcal{I}}_k]_{k=0}^{T-1}$ and m is the down-sampled map of $[M_k]_{k=0}^{T-1}$ via max-pooling. To further support image conditioning, we concatenate the image latent $\hat{\mathbf{z}}_i$, which encodes the first frame of

Table 1: Quantitative comparison of novel view synthesis on *RealEstate10K* and *Tanks and Temples*. **Bold** indicates the 1st, underline indicates the 2nd.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
RealEstate10K			
SEVA	16.648	0.613	0.349
ViewCrafter	16.512	0.636	0.332
See3D	18.189	0.694	0.290
FlexWorld	18.278	0.693	0.281
Voyager	**18.751**	**0.715**	**0.277**
Tanks and Temples			
SEVA	10.946	0.461	0.603
ViewCrafter	12.205	0.406	0.544
See3D	12.241	0.456	0.602
FlexWorld	12.494	0.451	0.541
Voyager	**12.684**	**0.482**	**0.539**

the ground-truth video, while filling the latents of the remaining frames with zeros. Accordingly, $\mathbf{z}'_{t,0}$ is fed to the diffusion model similar to Eq. 1-2. The diffusion model is thus trained to generate rgb-depth video frames.

Context-Based Control Enhancement. The above concatenation mechanism incorporates conditional information only at the input of the DiT model, leading to weak enforcement of the geometric conditions and resulting in misalignment between generated frames and input conditions.

To enhance the geometric-following capabilities, following [2], we further inject the diffusion model with lightweight modules. Concretely, we replicate the first block from the double-stream and single-stream modules as the Control blocks $\hat{f}_D$ and $\hat{f}_S$. Given the input video latent $\mathbf{z}'_{t,0}$, we have the following operations for each Transformer block i:

$$\mathbf{z}_D = \hat{f}_D(\mathbf{z}'_{t,0}), \mathbf{z}_S = \hat{f}_S(\mathbf{z}_D), \tag{3}$$

$$\mathbf{z}'_{t,i} = \mathbf{z}'_{t,i} + l_D(\mathbf{z}_D), \mathbf{z}''_{t,i} = \mathbf{z}''_{t,i} + l_S(\mathbf{z}_S), \tag{4}$$

where l_D and l_S are zero-initialized linear layers. Early-stage latent features preserve more contextual information, so that the integration into each block can strengthen pixel-level controllability.

2.3 Long-Range World Exploration

For long-range or even infinite video generation, auto-regressive is a natural choice. This paradigm recursively generates future frames or clips based on previously generated content, maintaining temporal continuity over time. However, due to the limited memory capacity of video diffusion models, auto-regressive methods are often restricted to conditioning on only a few preceding frames or clips. This limited context leads to inevitable information loss, making it fundamentally infeasible to retain and propagate the full scene history. In contrast to previous auto-regressive methods, Voyager exploits point-cloud conditions for generation, which is a scalable representation to store the whole history information. To enable infinite generation, we propose world caching with point culling to efficiently store spatial information and adopt smooth video sampling to ensure the consistency of consecutive clips.

World Caching with Point Culling. With input camera parameters and corresponding RGB-D video frames, point clouds can be projected to 3D space as $\hat{p} \in \mathbb{R}^{(T\times H\times W)\times 3}$, where T is the number of total frames. As the video continues to extend, the number of points can easily grow to millions, posing significant challenges in terms of memory and computational efficiency. To address that, we propose to maintain a world cache, which eliminates redundant points while preserving essential geometric information. Specifically, we incrementally add new points to the cache on a per-frame basis: given the accumulated point cloud $\hat{p}$ from previous frames, we render a visibility mask $M = \text{render}(\hat{p}, c_i)$ from the current camera view c_i. Points in the invisible regions are added to $\hat{p}$ first. For the visible regions, if the angle between the surface normal of existing points and the current view direction exceeds 90 degrees, the new point is also updated into the cache, because these

Table 2: Quantitative comparison of Gaussian Splattig reconstruction on *RealEstate10K*. Baselines require additional reconstruction step [35], while Voyager performs better with our generated depth.

Method	Post Rec.	PSNR ↑	SSIM ↑	LPIPS ↓
SEVA	VGGT	15.581	0.602	0.452
ViewCrafter	VGGT	16.161	0.628	0.440
See3D	VGGT	16.764	0.633	0.440
FlexWorld	VGGT	17.623	0.659	0.425
Voyager	VGGT	<u>17.742</u>	<u>0.712</u>	<u>0.404</u>
Voyager	-	**18.035**	**0.714**	**0.381**

existing points cannot be seen at the current viewpoint. This strategy reduces the number of stored points by approximately 40% and avoids noise accumulation caused by multi-frame aggregation.

Smooth Video Sampling. Conditioned on the above world cache, our video model can access the complete spatial information from previous frames. However, although each independently generated video clip is spatially consistent, there can still be color discrepancies, making them unsuitable for direct concatenation. We adopt two strategies to ensure smoother transitions between adjacent clips. (1) We first divide the input video into overlapping segments, where the length of the overlapping region is half of one segment. For each segment, the overlapping region is initialized with the generated results from the previous segment, serving as the noise initialization for the current segment's overlap region. (2) After completing inference for the consecutive two segments, we apply averaging across the overlapping regions and perform a few steps of denoising to refine transitions. In this way, we ensure the efficient generation of multiple clips while maintaining visual consistency across consecutive video frames.

2.4 Scalable Video Data Engine

Training such a video model demands large-scale video frames with corresponding camera parameters and depth maps. We carefully curate over 100,000 video clips from both real-captured videos and 3D renderings, and propose a scalable video data engine to automatically annotate required 3D information for arbitrary scene videos.

Data Curation. We selected two open-source real-world datasets, *i.e.*, RealEstate [51] and DL3DV [19] for the training. RealEstate contains 74,766 video clips related to real estate scenes, primarily featuring indoor home scenes, along with some outdoor environments. DL3DV provides 10K real-scene videos, but most of them suffer from rapid or shaky camera movements. We curate 3,000 high-quality videos from this dataset and segment them into approximately 18,000 video clips. Additionally, to increase the diversity of generation content, we collected 1,500 Unreal Engine scene models and rendered over 10,000 video samples to augment the dataset. In the end, we collected over 100,000 video clips from these datasets.

Data Annotation. Accurate camera parameters and depth are crucial for model training, but RealEstate and DL3DV do not provide such ground-truth data. Existing methods [3, 27, 30] adopt dense stereo models [32] to prepare training pairs, struggling to produce geometrically consistent depth. We propose a more robust data processing engine as shown in Figure 5. Specifically, we first use VGGT [35] to estimate camera parameters and depth for all video frames. The depth estimated by VGGT is not accurate enough, but it is aligned with camera poses. To further improve the depth estimation, we then employ MoGE [36] as a robust depth estimator and align the two depth maps with least squares optimization.

Finally, since our UE data provides metric depth values, we need to align all the estimated depth to a standard scale. We estimate the metric depth range of the scene using Metric3D [12] and map the previous depths into this range. This way, we can automatically annotate camera and depth for videos from any source.

3 Model Evaluation

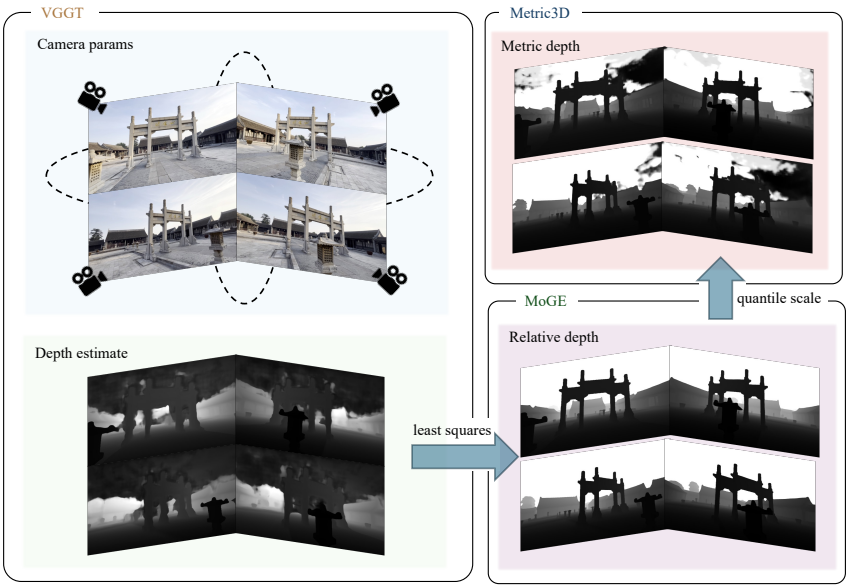

Figure 5: Overview of our scalable video data engine. We first estimate depth and corresponding camera parameters with VGGT. Estimated depth is then aligned with MoGE to obtain a more accurate value. Finally, we employ Metric3D to rescale the depth value into world coordinates.

Algorithm 1 World Caching

Input: RGB-D frames $[I_i, D_i]_{i=0}^{T-1}$, and corresponding cameras $[c_i]_{i=0}^{T-1}$
initialize world cache $\hat{p} = \emptyset$
for $i = 0$ to $T-1$ **do**
  Project point map $p \in R^{H\times W\times 3}$ from D_i and c_i;
  Project color map $c \in N^{H\times W\times 3}$ from I_i and c_i;
  Estimate normal map $n \in R^{H\times W\times 3}$ from p; ▷ estimate with o3d
  Render visibility mask M and normal angle θ with $\hat{p}$ and c_i;
  $\hat{p} \leftarrow \hat{p} \cup \{(p_{i,j}, c_{i,j}, n_{i,j}) \,|M_{i,j} = 0 \quad \text{or} \quad \theta_{i,j} > \frac{\pi}{2}\}$;
end for
return world cache $\hat{p}$

3.1 Implementation Details

Our training basically follows the image-to-video model of HunYuan-Video [17]. We divide the training into three stages: the first stage only trains the RGB video model; in the second stage, depth is introduced into the training; and in the third stage, the DiT parameters are frozen and ControlNet blocks are incorporated for training. We use all three datasets in the first training stage. However, DL3DV is removed in the second stage due to its fast camera motion, which makes it unsuitable for depth training. In the third stage, we train solely on the UE dataset with its ground-truth depth. During training, we randomly select a width-height ratio from $[1, 1.25, 1.5, 1.75]$ to support the generation of videos with multiple aspect ratios. To improve the model's robustness to varying camera motion speeds, we randomly select frame intervals to sample training videos, thereby exposing the model to trajectories of different temporal densities. The learning rate is set to 1×10^{-5} with a warm-up phase of 100 iterations, and is scheduled to decay to a minimum value of 1×10^{-6}. For inference, depth of the input image is first estimated by MoGE [36]. The number of sampling steps is set to 50 by default. Peak GPU memory consumption during single-card inference is approximately 60 GB. Using four GPUs in parallel, the end-to-end generation of a single video segment completes in approximately 4 minutes. The number of generation frames for a single pass is 49. We provide pseudocode of our world cache and smooth video sampling in Algorithm 1 and 2, respectively.

Algorithm 2 Smooth Video Sampling

Input: The former video clip V_0, overlap frames N
Load: DiT ϕ, VAE Encoder E and Decoder D
▶ initialize the next clip noisy latent n_1
  $z_0 = E(V_0), n_1 = [z_0[-N:], \epsilon]$
▶ denoise the latent n_1 with 50 steps
  $z_1 = \phi(n_1, \text{step=50})$
▶ average overlapped latent
  $z_1[:N] = (z_1[:N] + z_0[-N:]) * 0.5$
▶ further denoise the latent with 5 steps
  $z_1 = \phi(z_1, \text{step=5})$
return video $V_1 = D(z_1)$

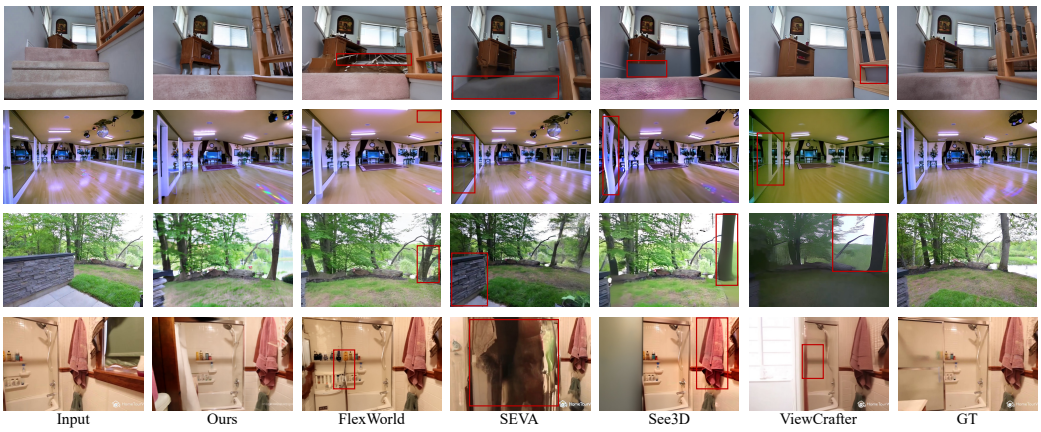

Figure 6: Qualitative results on video generation. Compared to the baselines, our model can generate a more reasonable unseen region and meanwhile preserve the content in the input view.

3.2 Video Generation

We evaluate the video generation quality of Voyager by comparing four open-source camera-controllable video generation methods on image-to-video generation, including SEVA [50], ViewCrafter [48], See3D [24], and FlexWorld [3]. Among these methods, ViewCrafter, See3D, and FlexWorld control the viewpoints with point cloud conditions, which are similar to our method. SEVA directly takes camera parameters as input conditions.

Dataset and Metrics. We randomly select 150 video clips from the test set of RealEstate [51] as our test dataset. To further evaluate our model with unseen data, we sample 50 videos from Tanks and Temples [16] as an out-of-domain dataset. Since the video clips do not provide ground-truth cameras, we estimate the camera parameters and depth maps with the same pipeline in our data engine. To evaluate the visual quality of generated videos, we adopt PSNR, SSIM, and LPIPS to measure the similarity between the generated frames and the ground truth.

Results. We report the quantitative results on Table 1. Our method outperforms all the baselines, demonstrating the high generation quality of our video model. The qualitative comparison in Figure 6 also showcases our capability of generating photorealistic videos. Especially in the last case of Figure 6, only our method can preserve the details of products in the input image. However, other methods are prone to generating artifacts, *e.g.*, in the first example of Figure 6, these methods fail to provide reasonable predictions when the camera movement is too large.

3.3 Scene Generation

To evaluate the quality of scene generation, we further compare the quality of scene reconstruction with generated videos based on Sec. 3.2. Since the compared baselines only produce RGB frames, we first exploit VGGT [35] to estimate camera parameters and initialize the point clouds for the generated videos of these methods. Thanks to the capability of generating RGB-D content, our results can be directly used in 3DGS reconstruction.

Figure 7: Qualitative results on Gaussian Splatting reconstruction. Our results present much more details than the compared baselines.

Table 3: Quantitative comparison on *WorldScore Benchmark*. **Bold and underline** indicates the 1st, **Bold** indicates the 2nd, underline indicates the 3rd.

Method	WorldScore Average	Camera Control	Object Control	Content Alignment	3D Consistency	Photometric Consistency	Style Consistency	Subjective Quality
WonderJourney	63.75	84.60	37.10	35.54	80.60	79.03	62.82	**66.56**
LucidDreamer	70.40	**88.93**	41.18	**75.00**	**90.37**	**90.20**	48.10	58.99
WonderWorld	**72.69**	**92.98**	51.76	**71.25**	**86.87**	85.56	70.57	49.81
EasyAnimate	52.85	26.72	54.50	50.76	67.29	47.35	73.05	50.31
Allegro	55.31	24.84	57.47	51.48	70.50	69.89	65.60	47.41
Gen-3	60.71	29.47	**62.92**	50.49	68.31	87.09	62.82	63.85
CogVideoX-I2V	62.15	38.27	40.07	36.73	86.21	**88.12**	**83.22**	62.44
Voyager	**77.62**	85.95	**66.92**	68.92	81.56	85.99	**84.89**	**71.09**

In Table 2, our reconstruction results with VGGT post-hoc outperform the compared baselines, indicating that our generated videos are more consistent in aspect of geometry. The results are even better when initializing point clouds with our own depth output, which demonstrates the effectiveness of our depth generation for scene reconstruction. The qualitative results in Figure 3 illustrate the same conclusion. Particularly in the last case, our method retains most details of the chandelier, while baseline methods even fail to reconstruct a basic shape.

3.4 World Generation

Besides the in-domain comparison on RealEstate, we test Voyager on WorldScore [5] static benchmark on world generation. WorldScore consists of 2,000 static test examples that span diverse worlds, *e.g.*, indoor and outdoor, photorealistic and stylized. In each example, an input image and a camera trajectory are provided. The metrics evaluate the controllability and quality of video generation. Specifically, we use "Camera Control", "Object Control", and "Content Alignment" to judge how the video model adhere to viewpoint instructions and text prompts. We use "3D Consistency", "Photometric Consistency", "Style Consistency", and "Subjective Quality" to evaluate the consistency and quality of generated content. Finally, an average score is presented to show the overall performance. Please refer to WorldScore for the details of these metrics.

We compare seven top methods in the existing benchmark, including three 3D methods WonderJourney [47], LucidDreamer [4], and WonderWorld [46], and four video methods EasyAnimate [42], Allegro [52], Gen-3 [29], and CogVideoX [43]. The scores are reported in Table 3. Voyager achieves the highest score on this benchmark. The score shows that voyager has competitive performance on camera control and spatial consistency, compared with 3D-based methods. Compared with warping methods like LucidDreamer and WonderWorld, our method strikes a better balance between generation consistency and quality. Our style consistency and subjective quality are the highest among all methods, further demonstrating the visual quality of our generated videos. Notably, since our video condition is constructed with metric depth, the camera movement in our results are larger than other methods, which is much harder to generate.

3.5 Long-Range Video Generation

As explained in Sec. 2.3, our method allows long-range video generation with efficient world caching and smooth video sampling. In Figure 8, we provide generation results that consist of three video clips, with totally different camera trajectories among clips.

The results present camera controllability and spatial consistency of the generated video, demonstrating that our method is capable of long-range world exploration. For example, the mountain in clip-3 of Figure 8(a) is exactly the same as the input image, which benefits from spatial information preserved by our world cache. In Figure 8(d), the blue chair on the far left remains consistent with the input image, even though it has moved out of the field of view during the generation of clip-1.

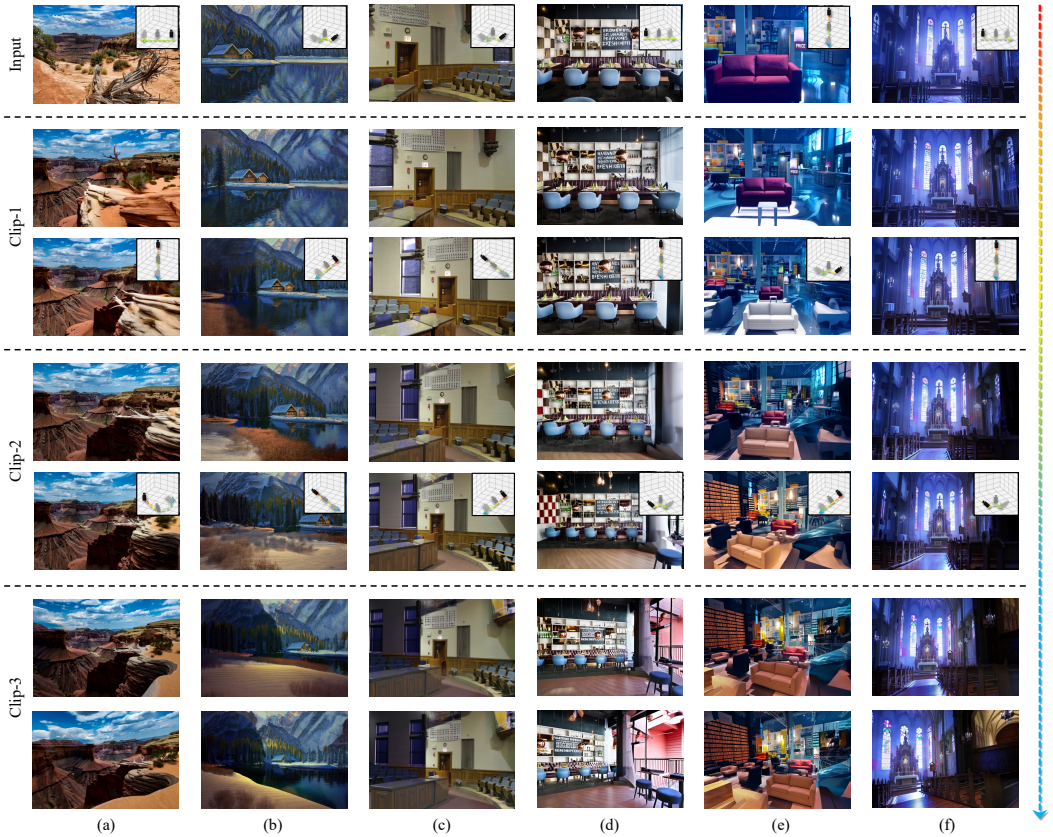

Figure 8: Visualization of long-range video generation. Thanks to our world cache, Voyager can maintain the content in the input image even when it has moved out of the viewpoint in some generated video clips.

3.6 Ablation Studies

To verify the effectiveness of our proposed designs, we conduct ablation studies on our world-consistent video diffusion and long-range world exploration.

World-Consistent Video Diffusion We evaluate our video models trained in the three stages separately on both RealEstate10K testset and Worldscore benchmark, *i.e.*, (a) model trained only on RGB conditions, (b) model trained on RGB-D conditions, and (c) model attached with additional control blocks. As shown in Table 4, fusing depth conditions in training can significantly enhance the generation quality and camera control. The control blocks can further improve the spatial consistency of generated results. We also provide qualitative results in Figure 9. The RGB-only model may generate inconsistent content when the camera moves to an unseen region. The results of RGB-D

Table 4: Ablation studies under two benchmarks. Left: *RealEstate10K* test results. Right: *WorldScore* metrics. Best results are in **bold**.

Metric	*RealEstate10K*			*WorldScore*		
	PSNR ↑	SSIM ↑	LPIPS ↓	Camera Control	Content Alignment	3D Consistency
Ours (RGB-only)	17.644	0.652	0.303	74.98	48.92	68.86
Ours (RGB-D)	18.355	0.696	0.279	85.04	65.72	78.58
Ours (full)	**18.751**	**0.715**	**0.277**	**85.95**	**68.92**	**81.56**

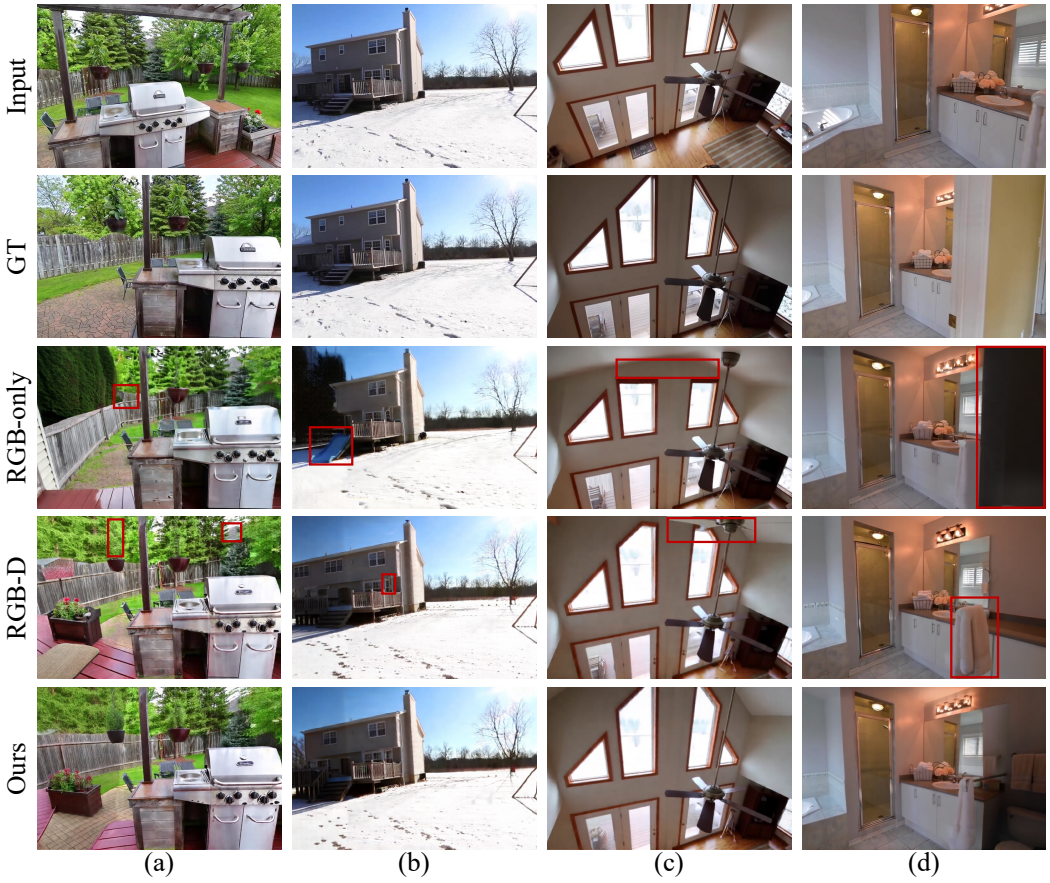

Figure 9: Qualitative results on ablation study. We compare the video models in our three training stages. Our final model achieves the highest quality.

model is more consistent with the input image, but it could still produce some minor artifacts. Our final model generates the most reasonable results.

Long-range video generation. We evaluate the quality of point culling and smooth sampling in Figure 10. For point culling, storing all points introduces noise, while storing points in the invisible region is insufficient. Results with additional normal check have comparable visual performance with storing all points, but save almost 40% storage. For smooth sampling, the video clip without sampling may exhibit inconsistencies compared to the first clip. Smooth sampling ensures a seamless transition between two consecutive segments.

4 Applications

Benefiting from our depth-fused video generation, Voyager supports various 3D-related applications. Besides, we provide more visualization results in Figure 12 and 13.

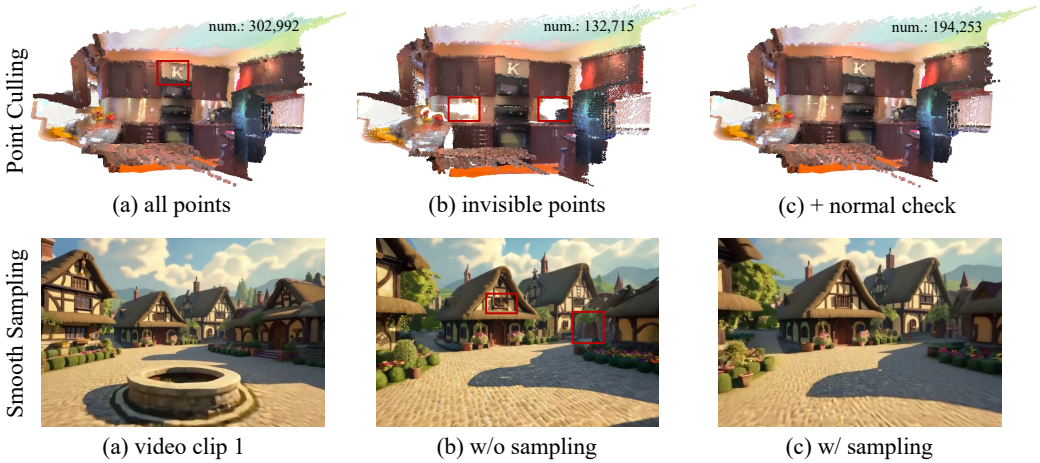

Figure 10: Qualitative results on ablation study. We compare the video models in our three training stages. Our final model achieves the highest quality.

Table 5: User study on depth projection.

Scene	VGGT	Ours	Diff
Indoor	47%	53%	+6%
Outdoor	32%	68%	+36%

Image-to-3D Generation. In Figure 11(a), we use three state-of-the-art 3D generation methods Trellis [40], Rodin v1.5 [28], and Hunyuan-3D v2.5 [14] to generate an orbit trajectory of input images. Compared with Voyager, 3D methods generally fall short in capturing fine-grained texture details and ensuring the plausibility of novel views. As shown in the generation of the ancient character, our method presents more detailed textures, especially in the back-view generation. Moreover, native 3D generative models can hardly handle the generation of multiple objects. In the simple combination where a car leans against a tent, Rodin failed to generate the tent, while Trellis produced a tent with missing parts. Hunyuan successfully generated two complete objects, but the spatial relationship was inaccurate, with the tent being too far from the car. Our method not only generates the correct content, but also produces more realistic visual effects. The tent is even visible through the car window in the side view.

Depth-Consistent Video Transfer. Generating a spatially consistent video with a different style typically requires training a stylized video model. However, to achieve the desired effect with our model, we only need to replace the reference image while retaining the original depth condition. As shown in Figure 11(b), we can change the original video to American-style or to the night, while maintain the spatial information of the original video.

Video Depth Estimation. Our video model is naturally capable of estimating video depth. In Figure 11(c), our predicted depth can preserve the details on the architectures. We further project the estimated depth to 3D point cloud in Figure 11(d). Compared with VGGT, Voyager's projection looks more plausible. Since ground-truth depth is unavailable for generated videos, it is infeasible to evaluate depth estimation using metrics like Absolute Relative Error(AbsRel). Therefore, we conducted a user study on projected points from two methods. We randomly select 10 indoor and 10 outdoor scenes from RealEstate10k and ask 10 participants to choose the more plausible one. Results are shown in Table 5. Our results are preferred by participants, particularly in outdoor scenes with a larger depth range.

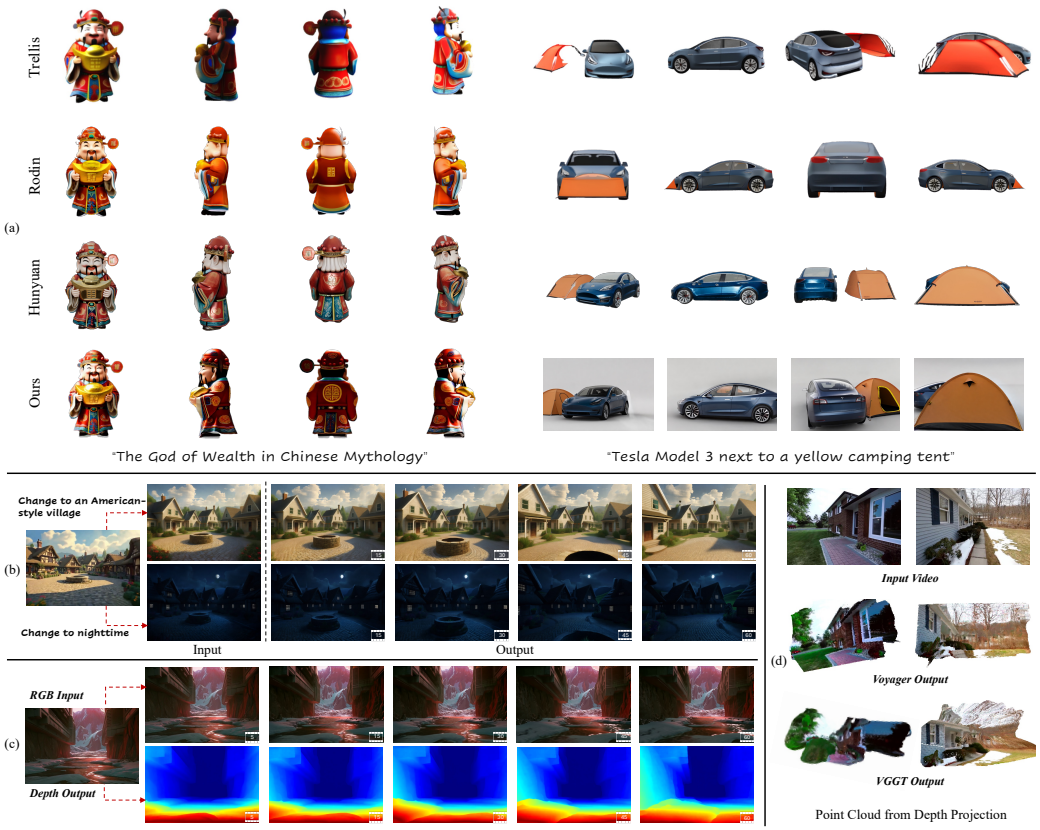

Figure 11: Applications: (a) Image-to-3D generation. (b) World-consistent video style transfer. (c) Monocular video depth estimation. (d) Point cloud projection.

5 Related Work

5.1 Camera-Controllable View Generation

Existing camera-controllable generation models can be categorized into three types: novel view synthesis [15, 26, 39, 11] produces new viewpoints through multi-view reconstruction. These methods rely on dense viewpoints and struggle to handle single-view inputs. The second method [7, 20, 37, 8, 50, 22] implicitly incorporates camera parameters into the model, training it to generate images from the corresponding viewpoints. For example, SEVA [50] injected plucker embeddings of camera trajectories to the attention layers, enforcing viewpoint following of video generation. However, these methods often suffer from spatial inconsistency, where content from different viewpoints is spatially misaligned, leading to noticeable artifacts and misalignments in the generated output. The third method [31, 24, 27, 3] leverages point clouds obtained by warping the input view as conditions for novel view generation as an explicit condition, significantly reducing the generation complexity. LucidDreamer [4] proposed an inpainting model to fill in the missing regions of warped images. See3D [24] trained a multi-view generation model conditioned with warped images. ViewCrafter [48], Gen3C [27], and FlexWorld [3] proposed to introduce video models to this task, further improving the consistency of generated content. However, the warped images still contain artifacts that negatively affect model training, as discussed in Figure 2. In this work, we introduce warping depth as an additional conditioning input and generate both RGB and depth content.

5.2 Long-Range Video Generation

Current video models are limited in their ability to generate long videos in a single pass. To extend video length, existing research explores training-free methods [34, 23], hierarchical strategies [44, 9],

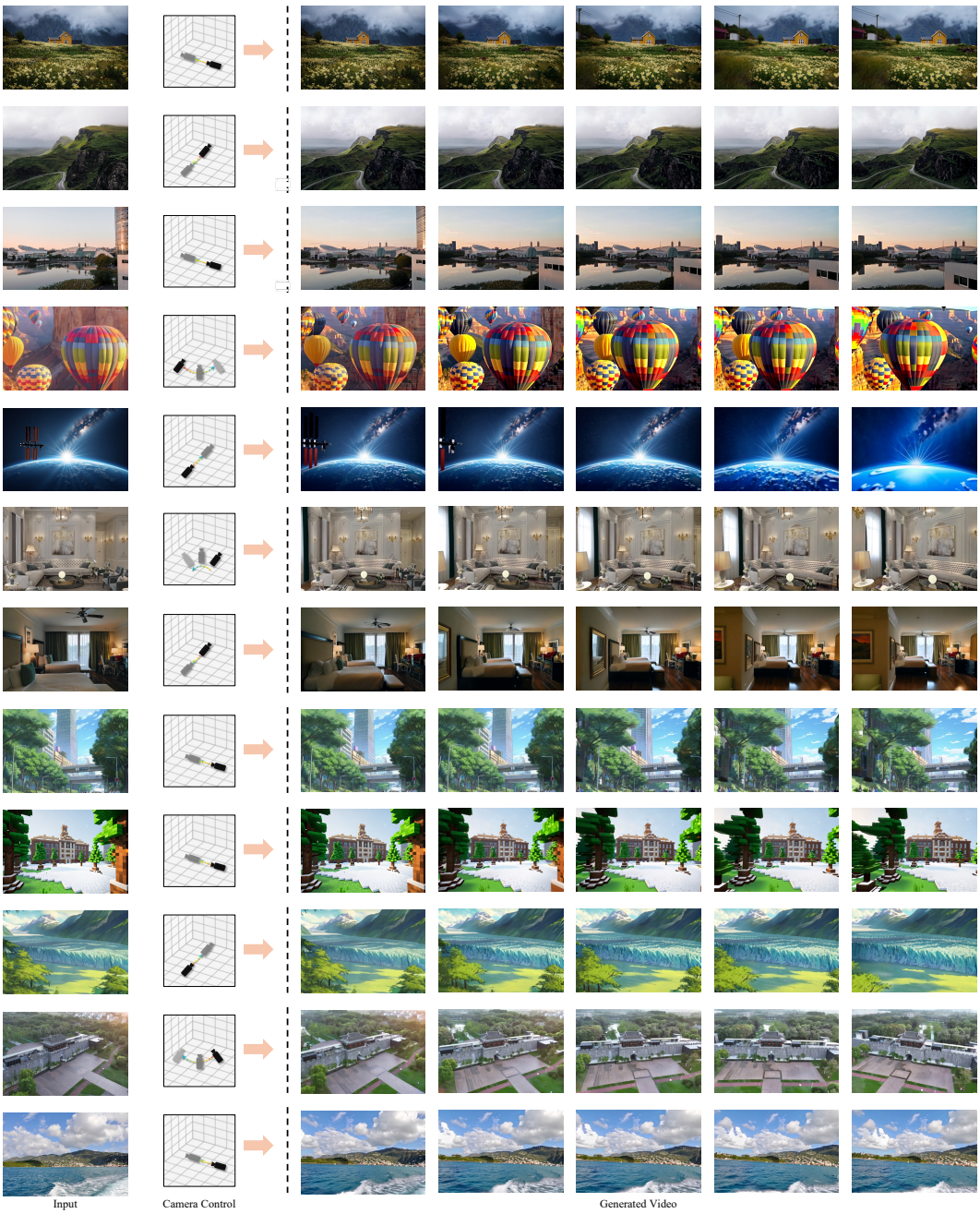

Figure 12: More Results.

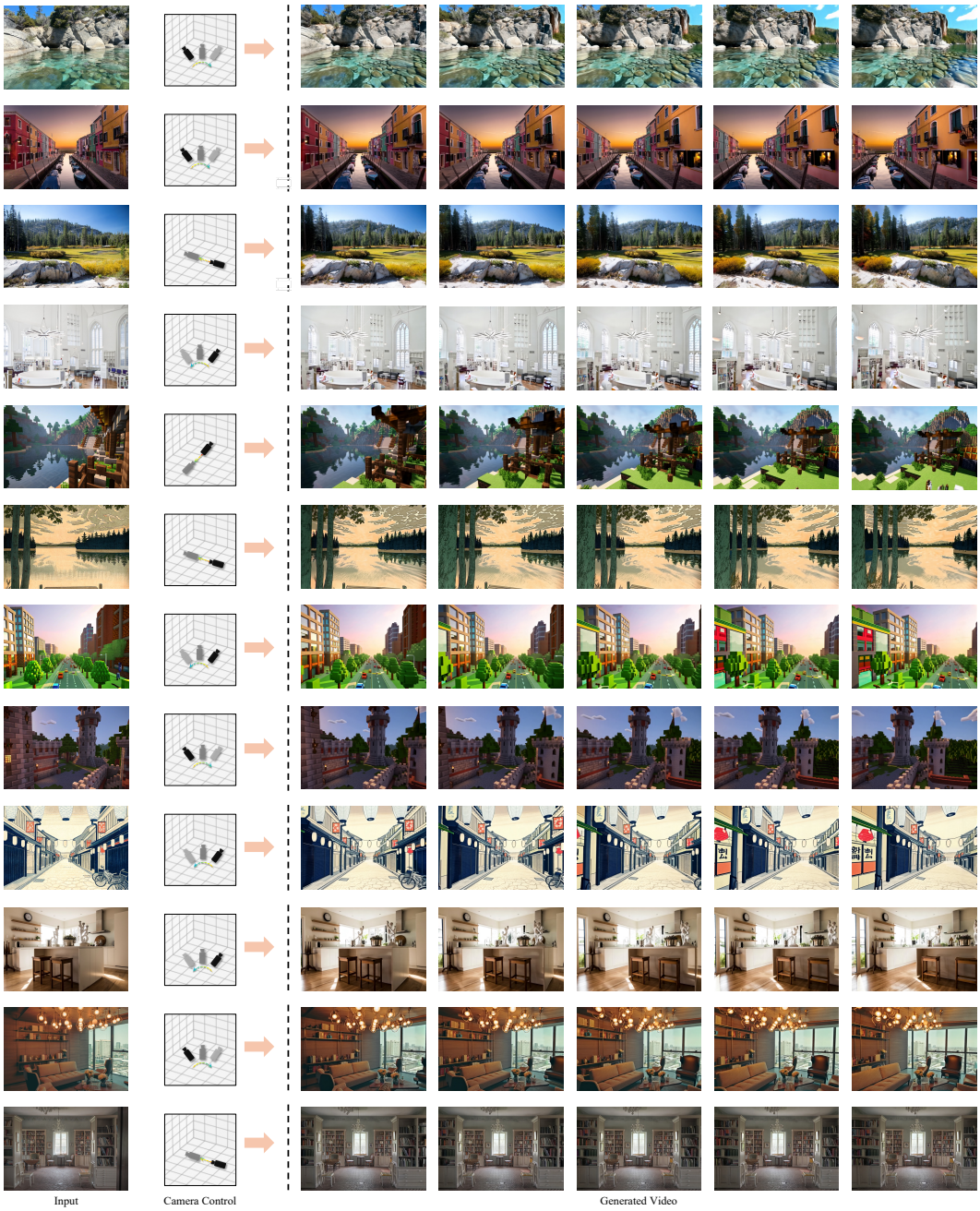

Figure 13: More Visualization Results.

and auto-regressive frameworks [45, 10]. However, the first two approaches cannot scale to infinitely long videos, while the auto-regressive strategy relies on memory caches that struggle to retain information from distant past frames. To address this limitation, we propose world cache with point culling in this work that efficiently preserves spatial information and enables the generation of arbitrarily long videos with smooth video sampling in an auto-regressive inference.

6 Conclusion

In this paper, we present **Voyager**, a world-consistent video generation framework for long-range world exploration. The proposed RGB-D video diffusion model can produce spatially consistent video sequences that align with the input camera trajectories, allowing direct 3D scene reconstruction. This supports auto-regressive and consistent world expansion. Experiments demonstrate high visual fidelity and strong spatial coherence in both generated videos and point clouds.

Contributors

- **Project Sponsors:** Jie Jiang, Linus, Yuhong Liu, Peng Chen
- **Project Leaders:** Tengfei Wang, Chunchao Guo
- **Core Contributors:** Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Junta Wu, Zhenwei Wang, Yuhao Liu
- **Contributors:** Xuhui Zuo, Lifu Wang, Yixuan Tang, Yonghao Tan, Chao Zhang

References

[1] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. Recammaster: Camera-controlled generative rendering from a single video, 2025.

[2] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. *arXiv preprint arXiv:2503.05639*, 2025.

[3] Luxi Chen, Zihan Zhou, Min Zhao, Yikai Wang, Ge Zhang, Wenhao Huang, Hao Sun, Ji-Rong Wen, and Chongxuan Li. Flexworld: Progressively expanding 3d scenes for flexiable-view synthesis, 2025.

[4] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

[5] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.

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

[7] 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. *arXiv preprint arXiv:2307.04725*, 2023.

[8] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[9] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

[10] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.

[11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[12] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson WH Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *arXiv preprint arXiv:2506.04225*, 2025.

[14] Hunyuan3D. Hunyuan-3d. 2025.

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[17] 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.

[18] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[19] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[20] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[21] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024.

[22] Yuhao Liu, Tengfei Wang, Fang Liu, Zhenwei Wang, and Rynson WH Lau. Shape-for-motion: Precise and consistent video editing with 3d proxy. *arXiv preprint arXiv:2506.22432*, 2025.

[23] Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. Freelong: Training-free long video generation with spectralblend temporal attention. *arXiv preprint arXiv:2407.19918*, 2024.

[24] Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *IEEE/CVF conference on computer vision and pattern recognition*, 2025.

[25] Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. *arXiv preprint arXiv:2409.08215*, 2024.

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[27] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[28] RodinAI. Rodin. 2025.

[29] Runway. Introducing gen-3 alpha: A new frontier for video gneration. 2024.

[30] Katja Schwarz, Denys Rozumnyi, Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. A recipe for generating 3d worlds from a single image. *arXiv preprint arXiv:2503.16611*, 2025.

[31] Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. *arXiv preprint arXiv:2405.17251*, 2024.

[32] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[33] HunyuanWorld Team Tencent. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels, 2025.

[34] Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*, 2023.

[35] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.

[36] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024.

[37] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[38] Ethan Weber, Riley Peterlinz, Rohan Mathur, Frederik Warburg, Alexei A Efros, and Angjoo Kanazawa. Toon3d: Seeing cartoons from a new perspective. *arXiv preprint arXiv:2405.10320*, 2024.

[39] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21551–21561, 2024.

[40] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[41] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer: Compositional generative model of unbounded 3d cities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9666–9675, 2024.

[42] Jiaqi Xu, Xinyi Zou, Kunzhe Huang, Yunkuo Chen, Bo Liu, MengLi Cheng, Xing Shi, and Jun Huang. Easyanimate: A high-performance long video generation method based on transformer architecture. *arXiv preprint arXiv:2405.18991*, 2024.

[43] 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. *arXiv preprint arXiv:2408.06072*, 2024.

[44] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023.

[45] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv preprint arXiv:2412.07772*, 2024.

[46] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T. Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv:2406.09394*, 2024.

[47] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. Wonderjourney: Going from anywhere to everywhere. *arXiv preprint arXiv:2312.03884*, 2023.

[48] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

[49] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

[50] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025.

[51] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[52] Yuan Zhou, Qiuyue Wang, Yuxuan Cai, and Huan Yang. Allegro: Open the black box of commercial-level video generation model. *arXiv preprint arXiv:2410.15458*, 2024.