
UGG: Unified Generative Geometry Model

Amani Kiruga^{1,*} Zhiyi Li^{2,*} Ruojin Cai^{1,*} Hansen Lillemark³ Xiaoshan Wu¹

Ayush Tewari⁴ Yilun Du¹

¹Harvard University ²MIT ³UC San Diego ⁴University of Cambridge

Abstract

Understanding 4D scenes is a core challenge in visual intelligence, requiring models to jointly capture scene geometry, appearance, and dynamics from partial observations. Existing approaches, however, are subdivided into feedforward models that are used for direct geometric prediction and generative models for visual synthesis. We propose a unified generative-predictive framework for 4D scene understanding that jointly models RGB video and geometric representations such as depth and camera pose with a single joint generative model. Our generative formulation enables us to learn from diverse, heterogeneous datasets with different subsets of geometric and visual annotations, and enables us to condition on arbitrary subsets of inputs, allowing a single model to perform cross-modal generation, geometric prediction, and future scene inference from partial observations. In addition, our generative approach enables test-time search over multiple candidate scene completions, improving geometric prediction from sparse observations. Overall, we illustrate the efficacy of our generative-predictive framework on a suite of generation and prediction tasks.

1 Introduction

For embodied agents to perceive and act intelligently in the physical world, they must understand its 3D structure and predict how it will evolve over time [9, 47, 3]. However, visual observations provide only partial 2D projections of the world, each capturing a limited set of viewpoints, visible surfaces, and changes over time. Robust 4D scene understanding therefore requires inferring what is not directly observed, namely the 3D geometry, camera viewpoint, scene appearance from unseen views, and scene evolution.

Current methods have made strong progress on different parts of this problem, but they are often separated into geometric prediction and visual generation. Feedforward models can directly estimate scene geometry, including depth, camera pose, and 3D structure, from images or videos. These methods are efficient and often accurate under the training distribution, but they typically produce a single deterministic estimate and can struggle when the observations are sparse or out of distribution. Generative models, in contrast, learn strong visual priors for completing missing observations, synthesizing novel views, and predicting plausible dynamic futures. However, they often lack explicit geometric grounding, which can lead to outputs that are visually plausible but inconsistent in 3D. Recent work has begun to narrow this gap by repurposing generative video models for geometric prediction [16, 25, 61], or by injecting geometric representations into generative models to improve 3D consistency [28, 43, 15]. Nevertheless, generation and prediction are still commonly treated as separate capabilities, rather than as two forms of inference in the same underlying scene model.

In this work, we take a different view: visual generation and geometric prediction are not separate problems. They can instead be different conditional queries of the same underlying 4D scene model.

*Equal contribution.

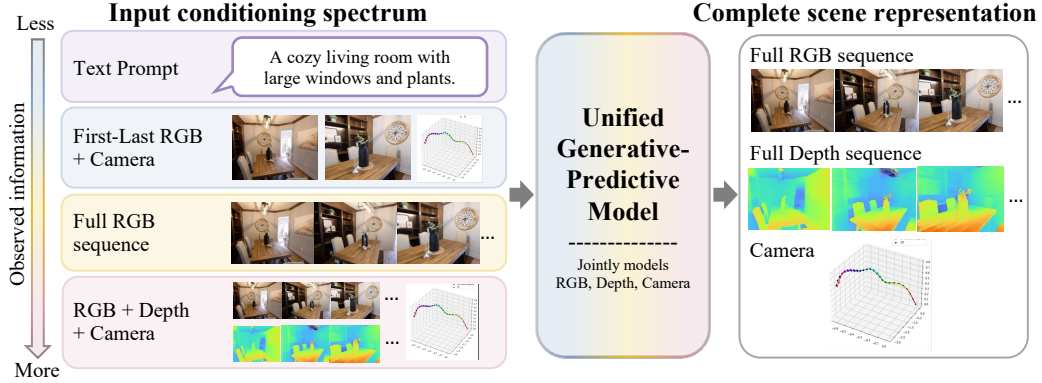


Figure 1: **A unified model for generation and prediction.** We formulate visual generation and geometric prediction as conditional completion in a single joint model over RGB, depth, and camera pose. Given arbitrary partial observations, ranging from a text prompt to RGB sequences with geometric cues, our model jointly uses these modalities to infer a complete 4D scene representation, enabling both generation from sparse inputs and geometric prediction from observations.

The appearance (RGB video), 3D structure (depth), and viewpoint (camera geometry) all provide complementary observations of a dynamic scene. Depending on what is observed, a joint model can infer missing scene components from the observed ones.

We instantiate this formulation by representing RGB, depth, and camera geometry as spatially aligned modalities, with camera pose encoded as dense ray maps. A multimodal transformer jointly processes these representation through shared self-attention and modality-specific layers. Modality flexibility is achieved through an extension of Diffusion Forcing [8]: different tasks are specified through noise patterns over both modality and time, where observed components serve as conditioning context and missing components are denoised. This conditional completion mechanism enables the same model to condition on a text prompt, a single RGB frame, an RGB sequence, camera poses, depth maps, or any combination of these, and complete the remaining modalities. Moreover, the model can learn from heterogeneous datasets by applying the loss only on available modalities, instead of requiring every example to contain complete annotations of pose and depth.

Our generative formulation also enables inference-time search over multiple plausible scene completions. Instead of committing to a single geometric prediction, we sample multiple candidates and select the one with the strongest cross-modal consistency, measured through photometric reprojection. This inference-time search capability is also consistent with the cognitive theory of perception as inference: human scene understanding requires inferring latent scene causes from incomplete observations, and may rely on internal simulation to evaluate possible scene explanations [19, 51, 2, 26].

Overall, our contributions are threefold: **i)** We formulate 4D scene understanding as joint generative modeling over appearance and geometry, treating visual generation and geometric prediction as missing-modality completion within a single conditional model. **ii)** We introduce a unified generative-predictive model that represents RGB, depth, and camera pose as spatially aligned modalities, enabling arbitrary conditioning and learning from heterogeneous datasets with incomplete supervision. **iii)** Finally, we leverage the generative formulation for inference-time search over multiple 4D scene hypotheses, using geometric consistency to improve geometric prediction.

2 Related Work

Generative Modeling for 4D Scene Understanding. Geometric prediction is largely dominated by feedforward discriminative models that regress depth, camera pose, and scene flow, or 3D structure from visual observations [39, 20, 53, 37, 30, 45, 41, 18]. While these methods achieve strong performance, they typically produce deterministic, single-pass estimates that struggle with inherent visual ambiguities, including sparse-view observations. To address these limitations, recent work has explored generative priors for pose estimation [7, 52], depth prediction [17, 13, 29], and dynamic 4D reconstruction [16, 61]. Generative models have also been used for novel view synthesis [50, 49]. However, these methods typically treat geometric prediction as a unidirectional mapping from RGB to specific geometric outputs and rely on strictly paired datasets.

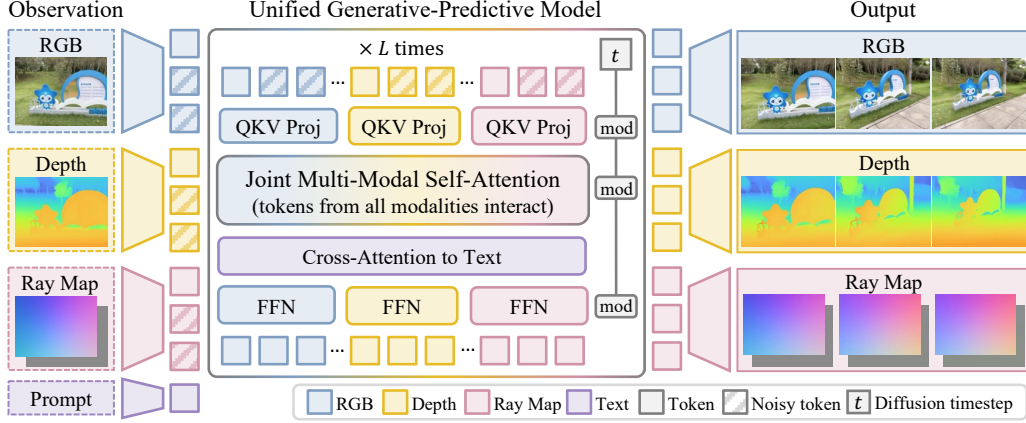


Figure 2: **Generative-predictive transformer.** We model RGB, depth, and camera pose as spatially aligned tokens, with camera pose represented as dense ray maps. Partial observations are zero-noise context tokens, while missing modalities are noisy tokens to be denoised. A unified transformer processes all tokens with modality-specific projections and feed-forward layers together with shared multimodal self-attention, enabling information exchange across modalities. The decoded RGB, depth, and ray maps form a complete 4D scene representation for cross-modal generation, geometric prediction, and inference-time consistency search.

Geometry-aware video and 4D generation. Recent generative world models have advanced from purely visual synthesis [4, 35, 1, 9] toward geometry-aware generation. One line of work injects 3D scene structure or persistent spatial memory into video generation to improve spatial consistency [28, 54, 15, 44, 58, 46]. Another line studies camera-conditioned generation, using camera trajectories to control synthesized videos [12, 57, 42]. More closely related, One4D [25] and WorldReel [11] jointly generate RGB frames and 4D geometric representations from text or RGB prompts. However, these methods typically follow fixed conditioning and output pathways. In contrast, we treat RGB, depth, and camera rays as symmetric modalities, enabling an any-to-any conditional completion framework for both generation and prediction.

Unified conditional modeling. Our work is related to flexible conditional generation, where diverse tasks are cast as missing-modality completion. Mixture-of-Transformers (MoT) architectures [21], such as Motus [3] and DreamZero [47], show that a single model can process heterogeneous modalities. For flexible conditioning, we build on Diffusion Forcing [8] and DFoT [33], which assign different noise levels to different tokens or frames. Recent works extend this paradigm for long-horizon video generation or embodied planning [10, 38]; in contrast, we apply it to multimodal 4D scene understanding, treating RGB, depth, and camera geometry as symmetric components. This probabilistic formulation also enables inference-time search. Motivated by test-time scaling [55, 24], we use additional inference compute to compare multiple candidate scene completions. We then select the candidate with stronger photometric reprojection consistency to improve geometric prediction.

3 Method

We propose a unified generative-predictive framework for 4D scene understanding. Given partial observations of a scene, our goal is to infer a complete scene representation containing appearance, depth, and camera geometry. We formulate both visual generation and geometric prediction as conditional completion in a joint generative model over spatially aligned RGB, depth, and camera-ray modalities. We specify each task by a noise pattern over multimodal tokens, with observed tokens as zero-noise context and missing tokens as noisy targets. The framework supports heterogeneous training data with incomplete modality supervision, arbitrary partial conditioning, and test-time search over multiple 4D scene hypotheses.

3.1 Unified 4D Scene Representation

We represent a 4D scene as a multimodal sequence of N frames. Each frame i contains an RGB image I_i , the depth map D_i , camera parameters $C_i = (K_i, T_i)$, where K_i denotes the camera intrinsics and T_i denotes the camera extrinsics. In practice, only a subset of this information may be observed. An example may contain one or more RGB frames, optionally with depth or camera parameters, or only

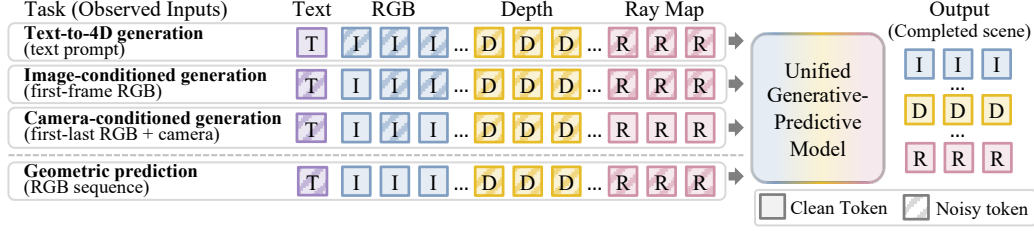


Figure 3: **Conditional completion with noise as masking.** We specify each task by the clean/noisy pattern over RGB, depth, camera-ray, and text tokens. Clean tokens provide observed context, while noisy tokens are denoised as missing targets, enabling generation and prediction with the same model architecture.

a text prompt. Our model is therefore designed to infer missing scene components from arbitrary partial observations.

To spatially align camera geometry with RGB and depth, we represent camera parameters C_i as a dense map of 6D Plücker rays, following prior ray-based scene and camera representations [32, 52]. This representation turns camera geometry into an image-like modality. Each pixel (u, v) is associated with a ray moment $\mathbf{m}_i(u, v)$ in a common world coordinate system, constructed from the viewing direction $\mathbf{d}_i(u, v)$ and the camera center $\mathbf{o}_i$.

$$R_i(u, v) = (\mathbf{d}_i(u, v), \mathbf{m}_i(u, v)), \quad \mathbf{m}_i(u, v) = \mathbf{o}_i \times \mathbf{d}_i(u, v). \quad (1)$$

The resulting scene representation $\mathcal{S}$ contains the appearance and local 3D geometry for each frame i :

$$\mathcal{S} = \{(I_i, D_i, R_i)\}_{i=1}^N.$$

3.2 Generative-Predictive Transformer

Given the spatially aligned representation $\mathcal{S}$, we design a multimodal diffusion transformer that jointly processes appearance, depth, and camera-ray geometry (Fig. 2).

Aligned Modality Tokenization. RGB frames and depth maps are encoded into latent tokens using a VAE encoder, producing z^I and z^D . Before encoding, we normalize depth and camera rays using a shared scale factor, making reprojection and cross-frame geometric consistency well defined. Ray maps are downsampled to the latent spatial resolution and projected directly into ray-map tokens. Since a ray map contains both ray directions and ray moments, we represent it with two token groups $z^R = [z^{R_d}; z^{R_m}]$, where z^{R_d} denotes ray-direction tokens and z^{R_m} denotes ray-moment tokens. The complete multimodal token sequence is

$$z = [z^I; z^D; z^{R_d}; z^{R_m}].$$

Each token is processed with modality specific patch embedding layers and a spatiotemporal RoPE positional encoding over latent spatial coordinates and the frame index [34].

At the output stage, RGB and depth tokens are decoded back to pixel space using the VAE decoder. Ray map tokens are directly mapped back to dense ray maps, which can be converted back to camera parameters C_i including rotation, translation, and intrinsics.

Diffusion Mixture-of-Transformers. The backbone uses a mixture-of-transformers [21] design. Each modality has its own input projections, allowing for modality-specific representation structure. After modality-specific projection, all modalities are concatenated and processed by shared self-attention, where information can be exchanged across modalities. Modalities are then separated again to modality-specific feed-forward layers and output heads. Optional text prompts are injected through cross-attention layers.

Each token has its own diffusion timestep. These timesteps are embedded and injected into transformer blocks through adaptive layer normalization modulation [27], allowing the model to process clean context tokens and noisy target tokens at different noise levels.

3.3 Conditional Completion with Noise as Masking

We formulate generation and geometric prediction as conditional denoising over a shared multimodal representation. Let $\mathcal{M} = \{I, D, R_d, R_m\}$ denote the set of modalities, and let $z_{i,0}^m$ denote the clean latent token at spatiotemporal position i for modality m . Which modalities are revealed or corrupted

specify different tasks without changing the architecture. For example, RGB-conditioned geometric prediction reveals RGB tokens while treating depth and camera-ray tokens as prediction targets.

Task Specification by Noise Pattern.

Diffusion forcing enables flexible conditional generation by assigning independently sampled noise levels across temporal positions [8]. We extend this formulation to multimodal sequences, allowing noise levels $t_i^m \in [0, 1]$ to vary across both modalities and temporal positions. We construct the noisy input $\tilde{z}_i^m$ through linear interpolation of the clean token and Gaussian noise

$$\tilde{z}_i^m = (1 - t_i^m)z_{i,0}^m + t_i^m \epsilon_i^m, \quad \epsilon_i^m \sim \mathcal{N}(0, I). \quad (2)$$

At $t_i^m = 0$, the token is fully clean, while at $t_i^m = 1$, it is pure noise. By varying the noise levels across both modality and temporal position, we realize different conditional generation and prediction tasks without modifying the model architecture.

Training with Heterogeneous Data. During training, we sample conditioning patterns that designate subsets of multimodal tokens as context, and the remainder as prediction targets. Context tokens receive either zero or reduced noise levels, while target tokens have independently sampled noise levels. With probability p_{clean} , context noise levels are set to $t_i^m = 0$ to allow for fully clean context conditioning.

The model is trained using conditional Flow Matching [23], with target velocity $v_i^m = \partial \tilde{z}_i^m / \partial t_i^m = \epsilon_i^m - z_{i,0}^m$. Given the corrupted multimodal tokens $\tilde{z}$, noise levels t , and optional text conditioning c , the model predicts $\hat{v} = f_\theta(\tilde{z}, t, c)$, with the loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, \epsilon} \left[\sum_{i, m \in \mathcal{M}} a_i^m \lambda^m \|\hat{v}_i^m - v_i^m\|_2^2 \right]. \quad (3)$$

λ^m is a weight balancing the losses for each modality. Modality availability is denoted as $a_i^m \in \{0, 1\}$. The availability mask allows a single model to be trained using heterogeneous mixes of RGB-only, RGB-camera, RGB-depth, and RGB-depth-camera data.

Inference with History Guidance. At inference time, the same noise pattern is used, but conditioning is stricter. Observed tokens are clamped to their clean values at every denoising step and assigned $t_i^m = 0$, while unknown tokens are initialized from Gaussian noise and progressively denoised. For example, in a first-last-frame camera-controlled setting, the first and last RGB tokens and all camera ray tokens are clamped as context, while intermediate RGB and depth tokens are generated. If depth is not provided as an input, it is not clamped and is inferred jointly with the generated RGB sequence.

We also use History Guidance [33] to strengthen consistency with the observed context. Let $\mathcal{H}$ denote the observed history context tokens, including RGB frames, ray maps, depth maps, or any combination of modalities. We compute a history-conditioned flow $\hat{v}_{\mathcal{H}}$ with tokens in $\mathcal{H}$ clamped to their observed values, and a history-masked flow $\hat{v}_{\emptyset}$ by replacing the same context tokens with random noise. The guided flow is

$$\hat{v}_{\text{HG}} = \hat{v}_{\mathcal{H}} + w(\hat{v}_{\mathcal{H}} - \hat{v}_{\emptyset}), \quad (4)$$

where w is the history-guidance strength. This is analogous to classifier-free guidance, but the dropped condition is the observed history/context rather than the text prompt.

3.4 Inference-time Search

The generative formulation also enables inference-time search. Given the same observed context, we sample L candidate completions by running the denoising process from different random initializations. Each sample yields a complete 4D scene hypothesis:

$$\hat{\mathcal{S}}^{(\ell)} = \{(\hat{I}_i^{(\ell)}, \hat{D}_i^{(\ell)}, \hat{R}_i^{(\ell)})\}_{i=1}^N, \quad \ell = 1, \dots, L.$$

From this set, we then choose the candidate that is most geometrically self-consistent. For each candidate, we evaluate a photometric reprojection error on a lightweight set of frame pairs, measuring whether a candidate’s RGB, depth, and camera predictions are mutually consistent. For each pair (i, j) , we reproject the RGB from frame i to the perspective of frame j using the predicted depth and camera geometry, and then compare it to the candidate’s RGB prediction of frame j . Candidates with inconsistent geometry or appearance produce inaccurate warps and have higher error.

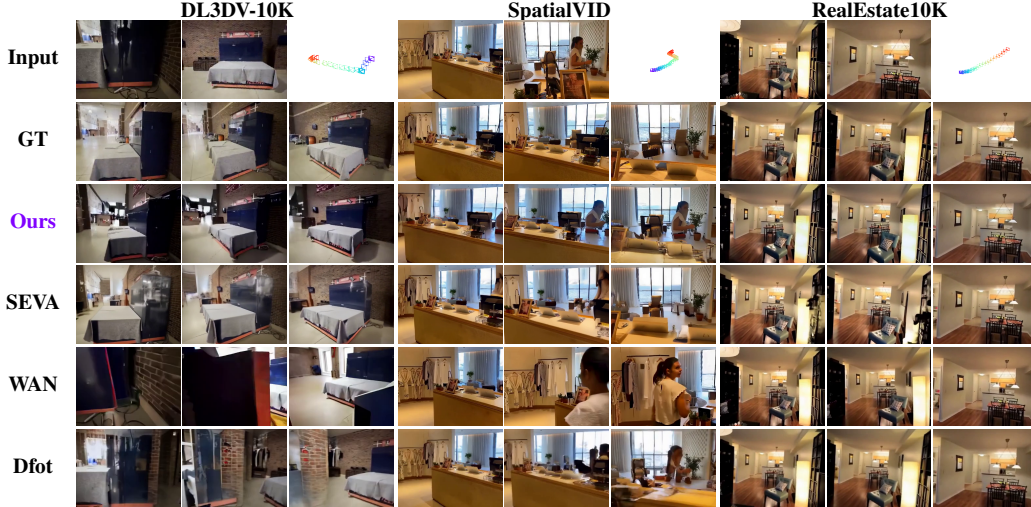


Figure 4: **Camera-controlled generation qualitative results.** The first row shows input samples with start/end images and target camera trajectories, and each following row shows synthesized novel views from one method. Our model produces competitive camera-conditioned generations, better preserving scene identity and dynamic content under camera motion compared with generic video-completion baselines.

4 Experiments

We evaluate the proposed unified generative-predictive model on a suite of 4D scene understanding tasks, including camera-controlled generation (Sec. 4.2) and geometric prediction under both dense and sparse observation settings, covering camera pose estimation (Sec. 4.3) and video depth estimation (Sec. 4.4). We show that a single unified model handles all tasks by changing only the clean/noisy masking pattern, without task-specific architectures. We further show that its generative formulation enables inference-time search, improving prediction quality (Sec. 4.5). Additional visualizations are provided in the supplementary material.

4.1 Model Training

We train on a weighted mixture of real, synthetic, and robotic video datasets, including DL3DV-10K [22], RealEstate10K [59], SpatialVID [36], ScanNet++ [48], AgiBot World [5], OmniWorld [60], Virtual-KITTI2 [6], and PointOdyssey [56]. These datasets provide heterogeneous supervision over RGB video, depth, camera pose, and text, and missing modalities are omitted from the loss. We train with multiple clean/noisy patterns over RGB, depth, and camera-ray tokens to cover both generation and geometric prediction. We initialize from Wan2.1-T2V-1.3B, finetune the diffusion transformer while freezing the VAE and text encoder, and train on 50-frame clips at 240×320 resolution and 10 FPS. Additional optimizer, compute, and preprocessing details are provided in the appendix.

4.2 Camera-Controlled Generation

We evaluate the unified model on camera-controlled video generation, testing whether camera-ray conditioning can guide visual synthesis while preserving appearance and scene layout across viewpoint changes.

Benchmarks and Metrics. We evaluate on RealEstate10K, SpatialVID, and DL3DV, covering diverse indoor and outdoor scenes. For each sequence, either the first RGB frame or the first and last RGB frames, together with the target camera trajectory, are provided as context, and the model generates the remaining frames. We evaluate the generated frames against ground-truth videos using PSNR, SSIM, and LPIPS.

Baselines. We compare quantitatively against Aether and SEVA [57], and qualitatively against Wan2.1-FLF-14B in a first-last-frame conditioning setting [35] and DFoT [33]. These baselines represent unified generation and geometry prediction, diffusion-based novel view synthesis, endpoint-conditioned video generation, and diffusion-forcing-based video completion, respectively. In contrast to the task-specific baselines, we use the same multimodal architecture for both camera-controlled generation and geometry prediction.

Table 1: **Camera-controlled generation results.** We report PSNR ($\uparrow$), LPIPS ($\downarrow$), and SSIM ($\uparrow$) on RealEstate10K, SpatialVID, and DL3DV.

	Method	RealEstate10K			SpatialVID			DL3DV		
		PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
2-view	Aether	20.02	0.196	0.624	13.84	0.489	0.334	13.14	0.399	0.243
	SEVA	25.25	0.081	0.838	15.44	0.344	0.466	17.61	0.211	0.552
	Ours	25.73	0.061	0.849	16.99	0.288	0.535	16.61	0.239	0.495
1-view	Aether	15.79	0.335	0.469	11.81	0.632	0.283	11.96	0.506	0.240
	SEVA	19.58	0.194	0.703	13.35	0.472	0.343	13.50	0.393	0.335
	Ours	21.79	0.150	0.779	14.86	0.408	0.405	13.92	0.385	0.346



Figure 5: **Qualitative comparison of estimated camera trajectories.** The temporal evolution of each trajectory is encoded by a color gradient, from purple (start) to red (end). The leftmost column shows representative RGB frames from the input video. Compared with Geo4D and RayDiffusion, our predicted trajectories more closely follow the ground-truth trajectory, with better overall structure in challenging cases.

Evaluation results. Camera-controlled generation results are shown in Table 1. Although not trained solely for novel-view synthesis, our model remains competitive across camera-conditioned generation benchmarks. It achieves the best PSNR, LPIPS, and SSIM on all three datasets in the one-view setting and on RealEstate10K and SpatialVID in the two-view setting; SEVA remains stronger on two-view DL3DV. This shows that ray-map conditioning supports accurate camera control, even within a broader RGB-depth-camera completion model. We visualize generated examples in Figure 4. Wan2.1-FLF does not take camera trajectories as input, and therefore struggles with complex camera motion. Compared with SEVA and DFoT, our model better preserves dynamic content, while SEVA is primarily designed for static scenes and DFoT can distort or remove moving humans.

4.3 Camera Pose Estimation

We evaluate camera pose estimation from RGB observations in dense and sparse settings, conditioning on either the full video or only the first and last frames. Camera-ray tokens are denoised and converted back into camera parameters, testing whether camera geometry can be predicted as a missing modality within the same conditional-completion framework.

Benchmarks and Metrics. We report results on RealEstate10K, Sintel, TUM, ScanNetV2, and ScanNet++. Following Geo4D [16], we align predicted trajectories to ground truth using Sim(3) Umeyama alignment and report ATE, RPE_t, and RPE_r.

Baselines. We compare quantitatively against Geo4D [16], Gen3R, and Aether, and qualitatively against RayDiffusion [52]. Geo4D estimates 4D geometry using a generative video prior, while RayDiffusion directly models camera rays through a diffusion denoising process. The geometry-focused baselines are more specialized for pose estimation, whereas our model predicts camera rays as part of a broader RGB-depth-camera scene representation.

Evaluation results. Table 2 reports camera pose estimation results. In the dense setting, where the full RGB sequence is observed, our model obtains the lowest ATE and translation RPE on all five datasets. For example, ATE is reduced from 0.190 to 0.089 on ScanNet++ compared with Geo4D. These improvements show that predicting camera rays as dense spatial tokens provides accurate global trajectory estimates and locally consistent translations. Figure 5 visualizes representative trajectories. Our predictions better follow the ground-truth motion, while RayDiffusion and Geo4D often produce collapsed paths in challenging settings.

The sparse setting is more challenging, since only the first and last RGB frames are provided. Here, the advantage of the generative formulation becomes more pronounced, with our model obtaining the lowest ATE and translation RPE across all five datasets. This suggests that the model can infer reasonable camera pose from sparse observations, where pose estimation is ambiguous and cannot rely on local matching. By training with diverse conditioning patterns, the model learns a joint RGB-camera prior over plausible trajectories between the observed views.

Table 2: **Camera pose estimation results.** For dense evaluation, we report ATE ($\downarrow$), RPE_t ($\downarrow$), and RPE_r ($\downarrow$, degrees). For sparse evaluation, where only the two observed frames are scored, we report translation angular error (tAE, $\downarrow$) and RPE_r ($\downarrow$, degrees). Ours uses shared intrinsics.

Method	RealEstate10K			Sintel			TUM			ScanNetV2			ScanNet++		
	ATE	RPE _t	RPE _r	ATE	RPE _t	RPE _r	ATE	RPE _t	RPE _r	ATE	RPE _t	RPE _r	ATE	RPE _t	RPE _r
Dense															
Geo4D	0.026	0.028	0.147	0.160	0.097	0.911	0.052	0.029	1.414	0.074	0.044	1.762	0.190	0.090	5.591
Gen3R	0.030	0.011	0.141	0.233	0.066	1.538	0.046	0.016	0.943	0.075	0.024	1.018	0.232	0.066	5.760
Aether	0.018	0.014	0.851	0.180	0.867	1.354	0.041	0.016	1.462	0.075	0.030	1.617	0.225	0.068	5.944
Ours	0.010	0.004	0.075	0.107	0.034	0.680	0.017	0.007	1.051	0.039	0.016	0.943	0.089	0.037	3.858
Method	RealEstate10K			Sintel			TUM			ScanNetV2			ScanNet++		
	tAE	RPE _r		tAE	RPE _r		tAE	RPE _r		tAE	RPE _r		tAE	RPE _r	
Sparse															
Geo4D	117.079	3.407		108.628	24.160		100.224	11.936		90.812	43.458		82.747	83.019	
Gen3R	44.004	4.197		75.984	24.984		82.875	12.374		79.871	43.481		92.324	82.540	
Aether	39.834	14.969		78.012	33.640		48.436	6.234		70.925	36.720		77.234	81.681	
Ours	4.287	1.332		28.543	16.138		7.439	1.613		24.363	21.177		34.149	31.594	

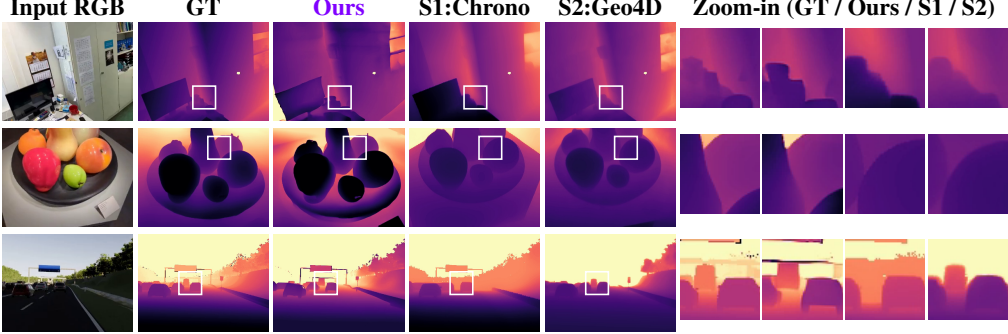


Figure 6: **Depth prediction qualitative comparison.** The white box marks the region shown in the zoom-in columns, ordered as GT, Ours, ChronoDepth, and Geo4D from left to right. Our predictions preserve sharper local structures and object boundaries in the highlighted regions.

4.4 Depth Prediction

We evaluate depth prediction from RGB observations in dense and sparse settings. The dense setting conditions on the full RGB video, while the sparse setting uses only the first and last frames. In both cases, depth tokens are initialized from noise and denoised, testing whether the model learns a useful geometric posterior over scene depth within the same conditional-completion framework.

Benchmarks and Metrics. We evaluate on ScanNet++, 7-Scenes [31], KITTI, Sintel, and Bonn. These datasets cover driving scenes, synthetic video, and indoor RGB-D scans. Following Geo4D [16], we apply video-level scale and shift alignment before evaluation, and report Abs Rel and $\delta < 1.25$.

Baselines. We compare with Geo4D, ChronoDepth [29], Gen3R, Aether, and Marigold [17]. These baselines include specialized depth and 4D geometry models as well as a unified model, while ours is trained for RGB-depth-camera completion.

Evaluation results. We report depth prediction results in Table 3. Overall, our model is competitive with specialized depth and 4D geometry baselines across both dense and sparse settings. In the dense setting, it obtains the lowest Abs Rel and highest $\delta < 1.25$ on ScanNet++ among the external baselines. In the sparse setting, where only an image pair is available, our model improves over the baselines on both metrics on 7-Scenes and Bonn. At the same time, Geo4D remains stronger on Sintel and some 7-Scenes metrics, indicating that specialized geometry models still have advantages in some settings. Figure 6 provides qualitative comparisons. The zoom-in crops show that our predictions preserve sharper object boundaries and more coherent local structures in highlighted regions. This suggests that joint RGB-depth training, together with the generative completion objective, helps the model recover fine-scale depth discontinuities while maintaining globally coherent depth maps.

4.5 Test-Time Search on Geometric Prediction

We evaluate whether test-time search (TTS) improves geometric prediction by sampling multiple RGB-depth-camera completions at inference time. For each input, we generate 16 candidates and select the one with the lowest RGB reprojection error.

Table 3 reports the selected depth results as *Ours* (TTS). TTS improves several depth metrics, especially in sparse settings, showing that the generative model can produce useful alternative scene hypotheses. Figure 7 studies the effect of the number of sampled candidates on both pose and depth. Larger search budgets often improve the selected result, but the gains are not uniform because RGB

Table 3: **Depth prediction results.** We report Abs Rel ($\downarrow$) and $\delta < 1.25$ ($\uparrow$) on ScanNet++, 7-Scenes, KITTI, Sintel, and Bonn in dense and sparse settings.

	Method	ScanNet++		7-Scenes		KITTI		Sintel		Bonn	
		Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$
Dense	ChronoDepth	0.177	0.724	0.116	0.871	0.120	0.857	0.394	0.530	0.092	0.920
	Geo4D	0.106	0.894	0.080	0.923	0.097	0.923	0.247	0.676	0.057	0.972
	Gen3R	0.157	0.772	0.079	0.937	0.185	0.777	0.414	0.520	0.168	0.805
	Aether	0.166	0.769	0.109	0.898	0.056	0.975	0.351	0.607	0.095	0.911
	Marigold	0.175	0.730	0.094	0.910	0.098	0.910	0.388	0.517	0.082	0.933
	Ours	0.076	0.950	0.078	0.883	0.067	0.967	0.326	0.612	0.056	0.971
	Ours (TTS)	0.075	0.949	0.086	0.916	0.068	0.956	0.297	0.645	0.077	0.955
	Ours (TTS Oracle)	0.067	0.959	0.076	0.923	0.057	0.967	0.263	0.692	0.056	0.973
Sparse	ChronoDepth	0.199	0.700	0.130	0.844	0.089	0.905	0.424	0.487	0.072	0.960
	Geo4D	0.177	0.737	0.098	0.912	0.100	0.897	0.310	0.554	0.067	0.966
	Gen3R	0.238	0.620	0.124	0.867	0.104	0.888	0.347	0.510	0.127	0.847
	Aether	0.269	0.567	0.167	0.772	0.149	0.786	0.487	0.424	0.138	0.849
	Marigold	0.169	0.733	0.106	0.890	0.087	0.921	0.345	0.527	0.071	0.970
	Ours	0.103	0.881	0.087	0.917	0.102	0.914	0.273	0.638	0.051	0.975
	Ours (TTS)	0.104	0.875	0.086	0.919	0.093	0.917	0.237	0.668	0.048	0.978
	Ours (TTS Oracle)	0.056	0.965	0.080	0.923	0.069	0.946	0.183	0.760	0.043	0.979

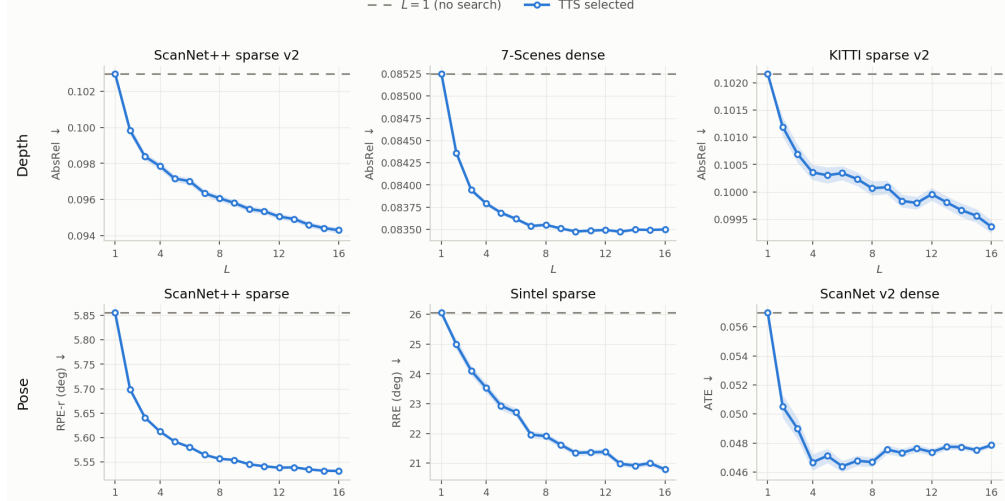


Figure 7: **Test-time search with increasing sample budget.** We sample 16 candidate completions and select the one with the lowest RGB reprojection error. The dashed line shows the single-sample result, and the solid curve shows the selected result as L increases. Search improves several pose and depth settings, showing that the generative model produces useful candidate diversity.

reprojection error is only a proxy for ground-truth geometric accuracy. Overall, these results show that inference-time search is a promising benefit of the generative formulation, while leaving room for stronger search criteria.

4.6 Ablation Studies

Setup. All ablations fine-tune Wan2.1-1.3B on TartanAir [40] and RE10K under a matched budget: 7,500 optimizer steps at batch size 8 (60,000 clips), the same sampled clips, and identical hyperparameters. Each variant changes a single factor. To remove a task, we set its loss weight to zero and replace its input tokens with noise; the remaining loss weights are not renormalized, so every retained task receives the same supervision as in the full model. For the variant without RGB generation, RGB remains a clean conditioning input.

Cross-task transfer. Removing one task degrades a different one (Table 4a-c). Without RGB generation, KITTI depth Rel rises from 0.152 to 0.322; without depth prediction, ScanNetV2 RPE_r rises from 2.91° to 4.30° ; and without pose estimation, RE10K PSNR drops from 20.18 to 13.26 dB. Joint training therefore helps each task, beyond merely sharing one network.

Heterogeneous data. RE10K provides RGB and camera supervision but no depth labels. Removing it still worsens depth on both Bonn (Rel 0.135 to 0.239) and KITTI (0.152 to 0.196), so the availability mask lets depth prediction benefit from depth-free data.

Table 4: **Ablations.** Each row changes one training task, dataset, architecture, camera representation, or inference setting. We report Rel ($\downarrow$) and $\delta_{1.25}$ ($\uparrow$) for depth, ATE ($\downarrow$) and RPE_r ($\downarrow$, degrees) for pose, and PSNR ($\uparrow$) for generation. RE10K: RealEstate10K. Shading marks the best result within each block.

(a) Depth prediction					(b) Camera pose estimation				
Variant	Bonn		KITTI		Variant	ScanNetV2		ScanNet++	
	Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$		ATE	RPE _r	ATE	RPE _r
Full model	0.135	0.843	0.152	0.791	Full model	0.074	2.91	0.193	6.18
w/o RGB generation	0.178	0.782	0.322	0.486	w/o depth prediction	0.081	4.30	0.234	7.57
w/o RE10K data	0.239	0.651	0.196	0.732	Single transformer	0.083	3.02	0.204	6.60
Single transformer	0.129	0.870	0.248	0.640	Dense ray maps	0.071	2.80	0.206	6.09
					Global camera tokens	0.103	3.44	0.233	6.61

(c) Camera-controlled generation				(d) History-guidance weight				
Variant	PSNR			w_{hist}	ScanNetV2		ScanNet++	
	RE10K	SpatialVID	DL3DV		ATE	RPE _r	ATE	RPE _r
Full model	20.18	13.67	12.60	0	0.0796	2.46	0.2170	5.77
w/o pose estimation	13.26	11.87	10.95	1	0.0741	2.83	0.2089	6.09
Single transformer	19.68	12.75	11.61	2	0.0740	3.24	0.2091	6.45
Global camera tokens	20.03	12.89	11.64	4	0.0765	4.22	0.2141	7.70

Architecture and camera representation. Replacing the Mixture-of-Transformers with a single shared transformer lowers pose accuracy, generation quality, and KITTI depth accuracy, although it is slightly better on Bonn. Replacing dense ray maps with global camera tokens degrades pose on both benchmarks and PSNR on all three, supporting image-aligned rays as the interface between camera geometry and visual tokens.

History guidance. Moderate guidance ($w_{\text{hist}} \in \{1, 2\}$) gives the lowest ATE, whereas RPE_r increases monotonically with w_{hist} and $w_{\text{hist}} = 4$ degrades both (Table 4d). Figure 7 reports sensitivity to the test-time search budget.

5 Conclusion

We present a unified generative-predictive framework for 4D scene understanding, formulating visual generation and geometric prediction as conditional completion over RGB, depth, and camera rays. By changing the clean/noisy pattern over multimodal tokens, a single model supports camera-controlled generation and geometric prediction under dense and sparse observations, while learning from heterogeneous datasets with different modality annotations. Across diverse datasets, our model achieves competitive generation and depth prediction results, strong pose estimation performance, and improved geometric prediction through test-time search. These results suggest that jointly modeling appearance and geometry with generative models is a promising direction toward more general 4D scene understanding.

Limitations. Our current model is not uniformly stronger than specialized methods: task-specific NVS models still perform better on pixel-aligned metrics, and specialized geometry models remain stronger on several indoor depth metrics. Future work could address these gaps with better dataset balancing, stronger geometric supervision, and improved depth-camera consistency during training. Our test-time search also relies on RGB reprojection error, which is useful but not uniformly reliable; learned or multimodal consistency scores may provide stronger search criteria. Finally, training at larger scale, higher resolution, and with more diverse data could further improve both visual detail and geometric accuracy.

References

- [1] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttmore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika

- Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025. 3
- [2] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the national academy of sciences*, 110(45):18327–18332, 2013. 2
- [3] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. 1, 3
- [4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3
- [5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 6, 15, 18
- [6] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 6, 15, 18
- [7] Ruojin Cai, Jason Y Zhang, Philipp Henzler, Zhengqi Li, Noah Snavely, and Ricardo Martin-Brualla. Can generative video models help pose estimation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16764–16773, 2025. 2
- [8] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024. 2, 3, 5
- [9] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025. 1, 3
- [10] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. 3
- [11] Shaoheng Fang, Hanwen Jiang, Yunpeng Bai, Niloy J Mitra, and Qixing Huang. Worldreel: 4d video generation with consistent geometry and motion modeling. *arXiv preprint arXiv:2512.07821*, 2025. 3
- [12] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025. 3
- [13] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *CVPR*, 2025. 2
- [14] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers arXiv:2508.10934*, 2025. 15

- [15] Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44(6):1–15, 2025. 1, 3
- [16] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20658–20671, 2025. 1, 2, 7, 8
- [17] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 8
- [18] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kotschieder. Mapanything: Universal feed-forward metric 3d reconstruction, 2025. 2
- [19] Daniel Kersten, Pascal Mamassian, and Alan Yuille. Object perception as bayesian inference. *Annu. Rev. Psychol.*, 55(1):271–304, 2004. 2
- [20] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. 2
- [21] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024. 3, 4
- [22] Lu Ling, Yicheng Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. D13dv-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 (CVPR)*, 2024. 6, 15, 18
- [23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. 5
- [24] Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18671–18681, 2025. 3
- [25] Zhenxing Mi, Yuxin Wang, and Dan Xu. One4d: Unified 4d generation and reconstruction via decoupled lora control. *arXiv preprint arXiv:2511.18922*, 2025. 1, 3
- [26] Bruno A. Olshausen. Perception as an inference problem. In Michael S. Gazzaniga and George R. Mangun, editors, *The Cognitive Neurosciences*. The MIT Press, 10 2014. 2
- [27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182. IEEE, 2023. 4
- [28] 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*, pages 6121–6132, 2025. 1, 3
- [29] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22841–22852, 2025. 2, 8

- [30] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 2
- [31] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 8, 18
- [32] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in neural information processing systems*, 34:19313–19325, 2021. 4
- [33] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. 3, 5, 6
- [34] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4
- [35] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3, 6, 18
- [36] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025. 6, 15, 18
- [37] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 2
- [38] Ruixiang Wang, Qingming Liu, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. Eva: Aligning video world models with executable robot actions via inverse dynamics rewards. *arXiv preprint arXiv:2603.17808*, 2026. 3
- [39] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023. 2
- [40] Wenshan Wang, DeLong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 9
- [41] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3 : Scalable permutation-equivariant visual geometry learning, 2025. 2
- [42] 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. 3
- [43] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *The Fourteenth International Conference on Learning Representations*, 2026. 1
- [44] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025. 3
- [45] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025. 2
- [46] Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026. 3

- [47] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 1, 3
- [48] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6, 15, 18
- [49] Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. *arXiv preprint arXiv:2405.15364*, 2024. 2
- [50] 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. 2
- [51] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006. 2
- [52] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 4, 7
- [53] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 2
- [54] Qihang Zhang, Shuangfei Zhai, Miguel Ángel Bautista, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *arXiv*, 2024. 3
- [55] Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *arXiv preprint arXiv:2505.23614*, 2025. 3
- [56] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19855–19865, 2023. 6, 15, 18
- [57] 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. 3, 6
- [58] Siyuan Zhou, Yilun Du, Yuncong Yang, Lei Han, Peihao Chen, Dit-Yan Yeung, and Chuang Gan. Learning 3d persistent embodied world models. *arXiv preprint arXiv:2505.05495*, 2025. 3
- [59] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG), SIGGRAPH*, 37(4):1–12, 2018. 6, 15, 18
- [60] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025. 6, 15, 18
- [61] Ruijie Zhu, Jiahao Lu, Wenbo Hu, Xiaoguang Han, Jianfei Cai, Ying Shan, and Chuanxia Zheng. Motioncrafter: Dense geometry and motion reconstruction with a 4d vae. *arXiv preprint arXiv:2602.08961*, 2026. 1, 2

A Visualizations

The supplementary zip file includes an HTML page with additional qualitative results for camera-conditioned generation, camera pose estimation, and depth estimation across datasets.

B Implementation Details

B.1 Datasets and Preprocessing

Datasets. We train a single model on a heterogeneous mixture of real, synthetic, and robotic video datasets: DL3DV-10K [22], RealEstate10K [59], the SpatialVID subset [36], ScanNet++ [48], AgiBot World [5], OmniWorld [60], Virtual KITTI 2 [6], and PointOdyssey [56]. The datasets provide different subsets of RGB, depth, camera pose, and language supervision. This heterogeneity is handled directly by diffusion forcing: unavailable modalities are treated as unobserved targets and their losses are masked out, so datasets with incomplete annotations can still contribute to the shared RGB and cross-modal representation.

Available modalities. For RealEstate10K and the SpatialVID subset, we use the provided RGB frames and camera poses. For DL3DV-10K and ScanNet++, we use RGB frames and obtain depth and camera-pose annotations using the VIPE [14] annotation pipeline. For AgiBot World, we use RGB video supervision only. For synthetic datasets, including OmniWorld, Virtual KITTI 2, and PointOdyssey, we use the available RGB, depth, and camera-pose annotations. We use language conditioning when captions or scene descriptions are available: SpatialVID and AgiBot World provide language descriptions, and we annotate DL3DV-10K and RealEstate10K with scene-level description prompts. For the remaining datasets, we train without text conditioning.

Camera representation. We represent camera pose as a dense ray map aligned with the image plane. For each pixel (u, v) , the ray map stores a Plücker ray

$$R_i(u, v) = (\mathbf{d}_i(u, v), \mathbf{m}_i(u, v)), \quad \mathbf{m}_i(u, v) = \mathbf{o}_i \times \mathbf{d}_i(u, v), \quad (5)$$

where $\mathbf{o}_i$ is the camera center and $\mathbf{d}_i(u, v)$ is the unit viewing direction in the world coordinate frame. The ray direction encodes orientation and intrinsics, while the ray moment encodes the camera center up to the common scene scale. In implementation, ray maps are stored as six-channel tensors with channel order $[\mathbf{d}, \mathbf{m}]$ and are downsampled to the latent spatial resolution.

Cross-dataset normalization. To make supervision from different datasets comparable, we normalize all geometric quantities with a shared clip-level scale. For each video clip, we compute a mutual scale s and use it consistently for camera ray moments and depth. Ray directions are already unit-normalized, while ray moments are divided by s and clamped to the normalized range. For depth, we use a disparity-like parameterization

$$\tilde{D} = \frac{2}{1 + D/s} - 1, \quad (6)$$

where D is the raw metric or reconstructed depth. This maps near depths to values close to 1 and far depths toward -1 , yielding a bounded representation that is numerically compatible with RGB and latent diffusion training. With this convention, RGB, depth, and ray-map channels are all represented in approximately the same normalized range $[-1, 1]$.

Missing-modality handling. Each training sample is represented in a common format containing RGB, ray maps, and depth. When a dataset does not provide ray maps or depth, the corresponding tensors are filled with random Gaussian placeholders and marked by modality-availability flags. During noise injection, missing ray-map and depth blocks are forced to the maximum noise level, so the model never receives structured false geometry. During loss computation, unavailable ray-map and depth targets are masked out. This prevents RGB-only or RGB-plus-pose datasets from corrupting geometric supervision while still allowing them to train the shared video-generation backbone.

B.2 Model Architecture

Base model. We initialize from Wan2.1-T2V-1.3B and extend its diffusion transformer into a multimodal Mixture-of-Transformers backbone. The pretrained video VAE and text encoder are kept frozen, and we finetune only the diffusion transformer. The model jointly processes four latent streams,

$$[z^I, z^{R_d}, z^{R_m}, z^D], \quad (7)$$

corresponding to RGB video, ray directions, ray moments, and depth, respectively. The two ray streams use separate latent blocks but share the same ray-stream transformer weights.

Tokenization. All clips contain 50 frames resized to 240×320 . RGB frames are encoded by the frozen Wan VAE into latent tensors with 16 channels and spatial size 30×40 . Depth maps are first

repeated to three channels and are encoded by the same frozen VAE. Ray maps bypass the VAE: we split each ray map into direction and moment fields and pack them directly into latent tensors at the same spatial resolution as the VAE latents. For the default 50-frame setting, the temporal latent length per modality is $T_{\text{lat}} = 14$. Thus, each modality stream has shape $[B, 16, 14, 30, 40]$, and the concatenated four-stream latent has shape $[B, 16, 56, 30, 40]$. We use a 3D patch size of $[1, 2, 2]$, giving a per-modality patch grid of $14 \times 15 \times 20$, or 4200 tokens per modality and 16800 tokens in total.

Ray latent packing. For ray directions and ray moments, four consecutive 3-channel ray frames are packed into the first 12 channels of a 16-channel latent tensor, with the remaining channels padded by zeros. In the first-last-frame setting, the first latent slot repeats the first frame, the middle slots pack groups of interior frames, and the final latent slot repeats the last frame. This packing keeps ray-map tokens spatially and temporally aligned with RGB and depth VAE tokens while avoiding unnecessary VAE compression of camera geometry.

Mixture-of-Transformers backbone. The transformer uses modality-specific input projections and feed-forward layers, together with joint multimodal self-attention. Concretely, we use separate 3D patch embeddings for RGB, ray maps, and depth. Queries, keys, values, and output projections are stream-specific for the RGB, ray, and depth streams, but the resulting tokens are concatenated and attend to each other in a single multimodal attention operation. Each transformer block also contains stream-specific normalization layers and feed-forward networks. This design preserves modality-specific feature statistics while allowing RGB, depth, and camera geometry to exchange information through shared attention.

Model dimensions. The diffusion transformer follows the Wan2.1-T2V-1.3B configuration: hidden dimension 1536, feed-forward dimension 8960, 12 attention heads, and 30 transformer layers. The VAE latent channel dimension is 16, the VAE stride is $[4, 8, 8]$ in time, height, and width, and the transformer patch size is $[1, 2, 2]$. Text conditioning uses the Wan text-conditioning path with prompt embeddings of dimension 4096 and a maximum length of 512 tokens.

Initialization of new modality branches. When initializing from the RGB-only Wan2.1 checkpoint, all compatible transformer weights are copied from the pretrained model. The RGB patch embedding and output-head weights are used to initialize the new ray and depth patch embeddings and heads. Attention, cross-attention, normalization, feed-forward, time-embedding, and text-conditioning weights are copied into the corresponding modality streams when shapes are compatible. This initialization preserves the pretrained video prior while allowing the new modality branches to specialize during finetuning.

B.3 Training Objective and Diffusion Forcing

Continuous flow matching. We train with continuous flow matching. For each latent token, we sample a raw diffusion time $t \in [0, 1]$ and apply the shifted schedule

$$t' = \frac{\alpha t}{1 + (\alpha - 1)t}, \quad \alpha = 3.0. \quad (8)$$

Given a clean latent z_0 and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, the noisy input is

$$z_{t'} = (1 - t')z_0 + t'\epsilon, \quad (9)$$

and the model predicts the flow target

$$v^* = \epsilon - z_0. \quad (10)$$

The loss is the mean-squared error between the predicted and target flow, computed separately for RGB, ray direction, ray moment, and depth:

$$\mathcal{L} = \frac{1}{4} (\mathcal{L}_I + \mathcal{L}_{R_d} + \mathcal{L}_{R_m} + \mathcal{L}_D). \quad (11)$$

For samples without ray-map or depth supervision, the corresponding losses are masked out.

Diffusion-forcing modes. We specify training tasks by assigning different diffusion levels to different modality-frame groups. Clean or low-noise tokens serve as conditioning context, while high-noise tokens are prediction targets. We use five conditioning patterns:

1. First RGB latent as history; remaining RGB, ray maps, and depth as targets;

2. All RGB latents as history; ray maps and depth as targets;
3. First and last RGB latents together with all ray-map latents as history; middle RGB and depth as targets;
4. First RGB latent together with all ray-map latents as history; remaining RGB and depth as targets;
5. First RGB latent together with all depth latents as history; remaining RGB and ray maps as targets.

These patterns correspond to image-conditioned scene completion, RGB-to-geometry prediction, camera-conditioned video generation, camera-conditioned extrapolation, and depth-conditioned generation/prediction. We sample these modes during training and restrict them according to the modalities available in each dataset. For example, depth-conditioned modes are used only for datasets with depth supervision, and ray-conditioned modes are used only for datasets with camera-pose or ray-map supervision.

History noise. For history tokens, we use clean context with probability 0.5. Otherwise, history tokens receive a random lower noise level. Target tokens receive independently sampled future noise levels. This teaches the model to use both perfectly clean context and imperfect context, which improves robustness when conditioning on generated or partially corrupted histories.

Optimization. We train on 50-frame clips at 240×320 resolution with a reference frame rate of 10 FPS. Optimization uses AdamW with learning rate 1×10^{-5} , betas (0.9, 0.95), weight decay 0.05, gradient clipping at 5.0, and a constant learning-rate schedule with 1000 warmup steps. Training is performed with mixed precision on 4 NVIDIA H200 GPUs for approximately 30,000 optimizer steps. We use batch size 1 per GPU and gradient accumulation of 4, giving an effective batch size of 16. We use gradient checkpointing for memory efficiency.

Dataset sampling weights. The training mixture uses relative dataset weights. In our final configuration, the relative weights are 0.5 for RealEstate10K, 5.0 for DL3DV-10K, 1.0 for SpatialVID, 1.5 for OmniWorld, 30.0 for ScanNet++, 50.0 for Virtual KITTI 2, 50.0 for PointOdyssey, and 0.04 for AgiBot World. The weights are multiplied by dataset size and normalized by the mixture loader.

B.4 Inference

Sampling. At inference time, the same clean/noisy pattern specifies the task. Observed RGB, depth, or ray-map tokens are clamped to their clean values at every denoising step and assigned zero diffusion level. Unknown tokens are initialized from Gaussian noise and progressively denoised. We use UniPC sampling with 40 denoising steps and the same shifted time schedule used during training. Under this setting, generating one 50-frame clip takes approximately one minute on a single NVIDIA H100 GPU.

History guidance. We use history guidance to strengthen consistency with observed context. For a set of observed history tokens H , we compute a history-conditioned flow $\hat{v}_H$ with H clamped to the observed values and a history-dropped flow $\hat{v}_\emptyset$ in which the same context is replaced by noise. The guided flow is

$$\hat{v}_{\text{HG}} = \hat{v}_H + w_{\text{hist}} (\hat{v}_H - \hat{v}_\emptyset). \quad (12)$$

We set $w_{\text{hist}} = 2.0$ in our default configuration.

Decoding. After denoising, RGB and depth latents are decoded with the frozen Wan VAE. Ray-direction and ray-moment latents are unpacked into dense ray maps. For camera-pose evaluation, the predicted ray maps are converted back to camera parameters using the same ray-map-to-camera recovery procedure across all datasets.

Inference-time search. For test-time search, we sample multiple candidate completions from different random seeds while clamping the same observed context. Each candidate produces RGB, depth, and ray-map predictions. We score candidates by cross-frame RGB reprojection error using their own predicted depth and camera geometry, and select the candidate with the lowest reprojection error. This implements the inference-time search procedure described in Sec. 3.4.

C Existing Assets and Licenses

We use existing public datasets and pretrained model assets. Table 5 summarizes the role of each asset in this work and the corresponding license or access terms. All datasets are used for academic

Asset	Type	Use in this work	License / access terms
DL3DV-10K [22]	Dataset	Training and evaluation for RGB video, camera pose, and depth/camera-conditioned generation.	DL3DV-10K Terms of Use and Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0); non-commercial research and educational use.
RealEstate10K [59]	Dataset	Training and evaluation for camera trajectories, camera-controlled generation, and pose prediction.	Creative Commons Attribution 4.0 International (CC BY 4.0), following the official RealEstate10K release.
SpatialVID [36]	Dataset	Training and evaluation for in-the-wild video with spatial annotations.	Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).
ScanNet++ [48]	Dataset	Training and evaluation on indoor RGB-D scenes with camera and geometry annotations.	ScanNet++ Terms of Use; non-commercial research and educational use only, with restricted redistribution.
AgiBot World [5]	Dataset	Training on robotic manipulation videos with multimodal embodied-scene observations.	Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).
OmniWorld [60]	Dataset	Training on multi-domain 4D world-modeling data.	Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0), with additional restrictions and notices for third-party game content.
Virtual KITTI 2 [6]	Dataset	Training and evaluation for synthetic driving scenes with RGB, depth, camera, and motion annotations.	Non-commercial use only; Creative Commons Attribution-NonCommercial-ShareAlike 3.0 (CC BY-NC-SA 3.0).
PointOdyssey [56]	Dataset	Training on synthetic dynamic scenes with camera, depth, segmentation, and long-term point/geometry annotations.	Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) for the released dataset.
7-Scenes [31]	Dataset	Evaluation-only dataset for indoor RGB-D depth prediction.	Microsoft Research 7-Scenes license agreement; non-commercial use only.
Wan2.1-T2V-1.3B [35]	Pretrained model	Initialization for the video diffusion backbone; we finetune the diffusion transformer while using the pretrained VAE and text encoder components.	Apache License 2.0, following the official Wan2.1-T2V-1.3B release.

Table 5: Summary of existing datasets and pretrained model assets used in this work. We cite the original creators of each asset and use each asset according to its official license or access terms. We do not redistribute third-party datasets or restricted source media.

research purposes only. We do not redistribute the original datasets, videos, game assets, or other third-party data. Any released code will provide dataset preprocessing and evaluation scripts, while requiring users to obtain the underlying datasets from their official sources and comply with the corresponding terms of use. If model checkpoints are released in the future, they will be released only under terms compatible with the upstream data and model licenses.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: The abstract and introduction state the paper’s intended scope: a unified generative-predictive model over RGB, depth, and camera pose for cross-modal generation, geometric prediction, and inference-time search. These claims are evaluated in Sec. 4 through camera-controlled generation, camera pose estimation, depth prediction, and test-time search experiments.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: The paper includes a limitations discussion in Sec. 5. It states that specialized methods remain stronger on some pixel-aligned novel-view-synthesis metrics and indoor depth metrics, that RGB reprojection is an imperfect proxy for test-time search, and that the current model is trained at moderate scale and resolution.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [N/A]

Justification: The paper does not present theoretical results, theorems, or formal proofs. The mathematical notation is used to define the scene representation, diffusion objective, and inference procedure.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: The paper discloses the model representation, architecture, training objective, conditioning patterns, datasets, evaluation settings, metrics, baselines, and test-time search procedure. Appendix B further specifies preprocessing, tokenization, model dimensions, optimization hyperparameters, sampling, and inference details needed to reproduce the main experimental setup.

Guidelines:

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

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

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

Answer: [No]

Justification: The current submission does not provide open-source code, model checkpoints, reproduction scripts, or an anonymized repository. The experiments use existing datasets and pretrained model assets cited in the paper; access to those assets follows the terms of their original providers.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: Sec. 4 and Appendix B specify the training datasets, modality availability, pre-processing, normalization, model architecture, diffusion-forcing modes, optimizer, learning rate, batch size, gradient accumulation, sampling procedure, evaluation datasets, metrics, and baselines. These details are sufficient to understand the experimental protocol and the reported results.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [No]

Justification: The paper reports standard benchmark metrics across multiple datasets and tasks, but does not report error bars, confidence intervals, or statistical significance tests. Repeated full training runs are computationally expensive for the 1.3B-parameter generative model, so the paper instead supports its claims through evaluation across diverse datasets, dense/sparse settings, and multiple tasks.

Guidelines:

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: Appendix B reports the training resolution, clip length, optimizer settings, mixed-precision training, number and type of GPUs, optimizer steps, batch size, gradient accumulation, gradient checkpointing, and inference-time sampling cost. These details describe the main compute resources needed to reproduce training and inference.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics. It uses existing datasets and pretrained model assets, does not involve new human-subject experiments or crowdsourcing, and reports assumptions, limitations, and evaluation results transparently.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: This work is a technical study of unified RGB, depth, and camera-pose modeling for 4D scene understanding, evaluated on standard research benchmarks. It does not introduce a deployed system, collect new human-subject data, or release a new dataset. We therefore do not anticipate substantial direct negative societal impacts beyond general risks associated with generative visual models and geometric scene reconstruction.

Guidelines:

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

11. Safeguards

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

Answer: [N/A]

Justification: The current submission does not release a new dataset or high-risk pretrained model checkpoint. If model checkpoints or generated assets are released in the future, we will include documentation, intended-use restrictions, and responsible-use guidance to reduce risks from misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

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

Answer: [\[Yes\]](#)

Justification: Appendix [C](#) summarizing all existing datasets and pretrained model assets used in the work, including their citations, role in the experiments, and license or access terms. We use the assets for academic research, comply with non-commercial and attribution requirements where applicable, and do not redistribute third-party datasets, videos, game assets, or restricted source media. Any future release of code or checkpoints will be made only under terms compatible with the upstream asset licenses.

Guidelines:

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

13. New assets

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

Answer: [\[N/A\]](#)

Justification: The paper does not introduce or release a new dataset, benchmark, codebase, or model checkpoint as part of the current submission. If assets are released later, they will be documented with license, usage terms, training details, and limitations.

Guidelines:

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

14. Crowdsourcing and research with human subjects

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or newly conducted research with human subjects. It uses existing datasets and pretrained assets.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not conduct new human-subject studies or crowdsourcing experiments, so IRB or equivalent review is not applicable.

Guidelines:

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

16. Declaration of LLM usage

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

Answer: [N/A]

Justification: The core method does not use LLMs or agents as an important, original, or non-standard component. Any writing, editing, or formatting assistance does not affect the scientific methodology, rigor, or originality of the work.

Guidelines:

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