UGG: Unified Generative Geometry Model

Amani Kiruga[1,*] **Zhiyi Li**[2,*] **Ruojin Cai**[1,*] **Hansen Lillemark**[3] **Xiaoshan Wu**[1]

Ayush Tewari[4] **Yilun Du**[1]

[1]Harvard University [2]MIT [3]UC San Diego [4]University 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. However, existing approaches are subdivided into feedforward models for direct geometric prediction, and generative models for visual synthesis. We introduce UGG, a unified framework for 4D scene understanding that jointly models RGB video and geometric representations such as depth and camera pose. By assigning different noise levels across modalities and time, UGG formulates visual generation and geometric prediction as conditional completion within a single generative model. This flexible formulation enables the model to learn from diverse, heterogeneous datasets with different subsets of geometric and visual annotations. Furthermore, our generative formulation enables inference-time search over multiple plausible scene hypotheses, using geometric evidence extracted from observed RGB frames to improve geometric prediction without ground truth geometry. We demonstrate UGG's effectiveness over non-unified baselines across camera-controlled video generation, depth estimation, and camera pose prediction, including in challenging sparse observation settings. Code and visualizations are available at ugg-model.github.io.

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 [11, 62, 4]. 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 [21, 34, 77], or by injecting geometric representations into generative models to improve 3D consistency [38, 58, 20]. 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 introduce UGG, a unified generative geometry model that treats visual generation and geometric prediction as different conditional queries of the same underlying 4D scene representation. The appearance (RGB video), 3D structure (depth), and viewpoint (camera geometry) all provide

*Equal contribution.

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

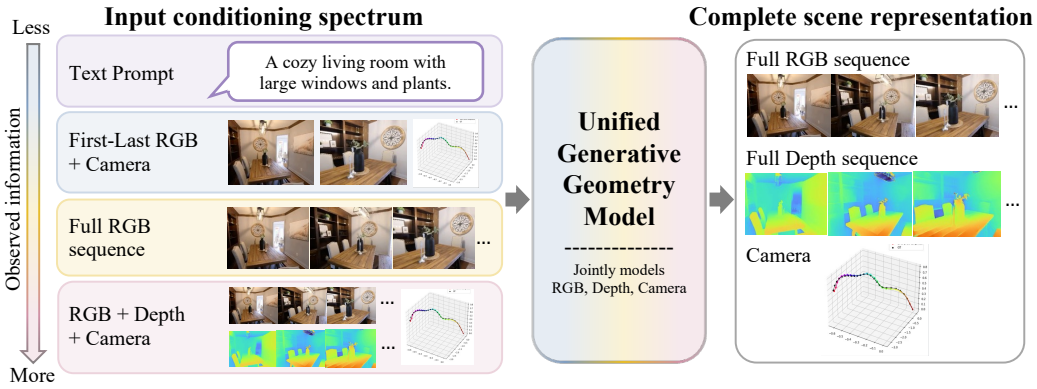


Figure 1: **A unified model for generation and prediction.** We formulate visual generation and geometric prediction as conditional completion in a single joint model over RGB, depth, and camera pose. Given arbitrary partial observations, ranging from a text prompt to RGB sequences with geometric cues,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

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 [10]: 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 [24, 66, 3, 35].

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 denoising 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 [53, 25, 68, 51, 40, 60, 55, 23]. 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 [9, 67], depth prediction [22, 17, 39], and dynamic 4D reconstruction [21, 77]. Generative models have also been used for novel view synthesis [65, 64]. However, these methods typically treat geometric prediction as a unidirectional mapping from RGB to specific geometric outputs and rely on strictly paired datasets.

Geometry-aware Video and 4D generation. Recent generative world models have advanced from purely visual synthesis [6, 49, 2, 11] toward geometry-aware generation. One line of work injects 3D scene structure or persistent spatial memory into video generation to improve spatial

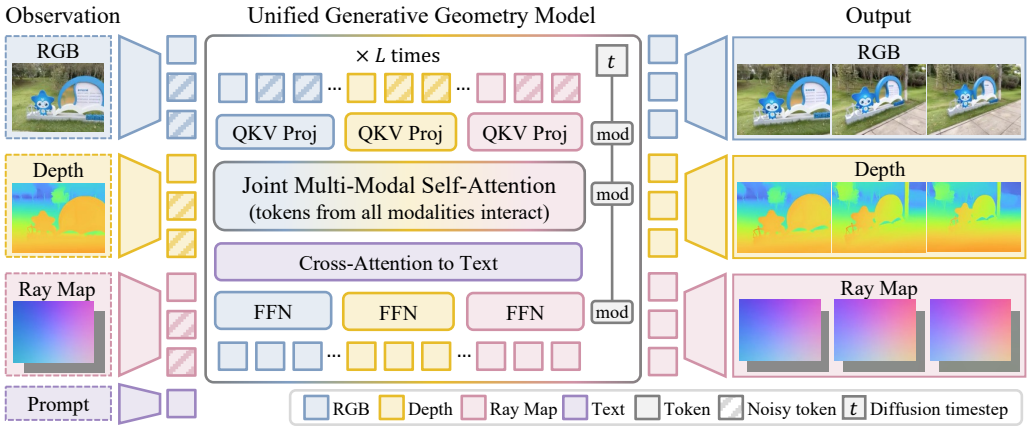


Figure 2: **Generative-predictive transformer.** We model RGB, depth, and camera pose as spatially aligned tokens, with camera pose represented as dense ray maps. Partial observations are zero-noise context tokens, while missing modalities are noisy tokens to be denoised. A unified transformer processes 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.

consistency [38, 69, 20, 59, 73, 28, 61]. Another line studies camera-conditioned generation, using camera trajectories to control synthesized videos [16, 72, 57]. More closely related, One4D [34] and WorldReel [14] 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 [27], such as Motus [4] and DreamZero [62], show that a single model can process heterogeneous modalities. For flexible conditioning, we build on Diffusion Forcing [10] and DFoT [43], which assign different noise levels to different tokens or frames. Recent works extend this paradigm for long-horizon video generation or embodied planning [12, 52]; 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 [71, 31], 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. 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 a text prompt. Our model is 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 [42, 67]. This representation turns camera geometry into an image-like modality. Each pixel (u, v) is associated

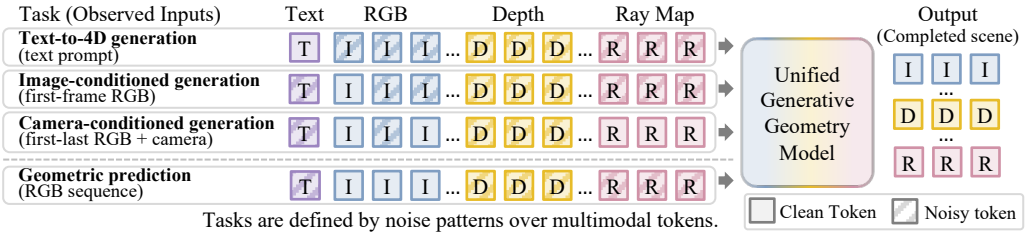


Figure 3: **Conditional completion with noise as masking.** We specify each task by the clean/noisy pattern over RGB, depth, camera-ray, and text tokens. Clean tokens provide observed context, while noisy tokens are denoised as missing targets, enabling generation and prediction with the same model architecture.

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 $\mathbf{o}_i$.

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

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, producing z^{R_d} for ray-direction tokens and z^{R_m} for ray-moment tokens. This yields the complete multimodal token sequence,

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

Each token is processed with modality-specific patch embedding layers and a spatiotemporal RoPE positional encoding over latent spatial coordinates and the frame index [45]. If the VAE encoder performs temporal downsampling, then the number of latents and frames could differ; we omit this from our notation for simplicity and continue to refer to the number of frames or latents as N.

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 [27] 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, enabling cross-modal information exchange. Modalities are separated to modality-specific feed-forward layers and output heads. Optional text prompts are injected through cross-attention.

Each token has its own diffusion timestep. These timesteps are embedded and injected into transformer blocks through adaptive layer normalization modulation [37], 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 temporal 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 [10]. We extend this formulation to multimodal sequences, allowing noise levels $t_i^m \in [0, 1]$ to vary across both modalities

(m) and temporal positions (i). We construct the noisy input $\tilde{z}_{i,t}^m$ through linear interpolation of the clean token $z_{i,0}^m$ and Gaussian noise:

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

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 modality tokens as context, and others as prediction targets. Context tokens share a single noise level, while target tokens share a separately sampled noise level. 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 [30], with target velocity $v_i^m = \partial \tilde{z}_{i,t}^m / \partial t_i^m = \epsilon_i^m - z_{i,0}^m$. Given the corrupted multimodal tokens $\tilde{z}$, noise levels t, and optional text conditioning c, the model predicts $\hat{v} = f_\theta(\tilde{z}_t, t, c)$, with the loss:

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

λ^m is a weight that balances 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.

Guided Inference Using Available Modalities. 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. We also use History Guidance [43] to strengthen consistency with the observed context.

3.4 Inference-time Search

The generative formulation also enables inference-time search, in contrast with a deterministic predictor that commits to one scene. Given the same observed context, we sample L candidate completions from different random initializations. Each sample yields a complete 4D scene hypothesis:

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

To select among these hypotheses without ground-truth depth or camera pose, we evaluate how well each hypothesis explains geometric evidence extracted from the observed RGB frames. Full equations and hyperparameters are given in App. B.8.

Geometric Evidence Derived from RGB. We extract pixel correspondences between observed RGB frames using RAFT [47]. To remove issues with occlusions and moving objects, we keep correspondences only if forward mapping followed by backward mapping returns a pixel to its original point. For dense observations, we select frame pairs with informative pixel motion, adapting the temporal separation according to the observed optical flow magnitude. For sparse observations, both generation and verification use only the first and last observed RGB frames. These correspondences provide geometric constraints against which candidate depth and camera predictions can be evaluated.

Depth Evaluation. We evaluate whether a candidate's predicted depth is consistent with the observed pixel motion. For each frame pair, we first estimate the relative camera pose from image correspondences using essential matrix estimation with RANSAC, shared across depth candidates. Each candidate's inverse depth is aligned to the observed parallax by fitting an affine transformation. The aligned depth is used to reproject pixels between views, which measures agreement with the observed correspondences. For sparse observations, evaluation uses correspondence residuals after epipolar filtering. For dense observations, we additionally incorporate photometric and DINOv2 feature consistency [36].

Pose Evaluation. For camera pose prediction, we compare each candidate's relative translation direction against the direction estimated from image correspondences using the candidate's predicted

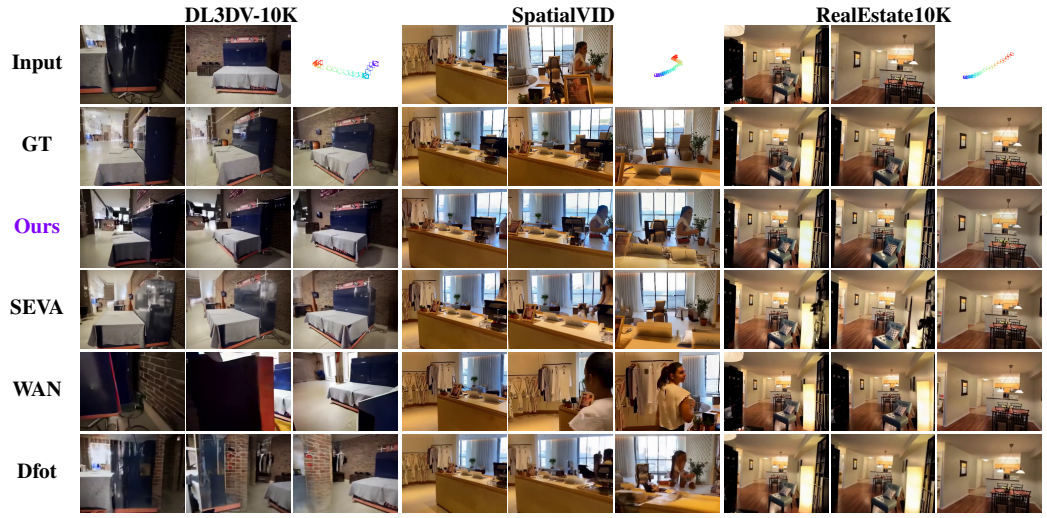


Figure 4: **Qualitative comparison of camera-controlled generation.** Given the first and last RGB frames and a target camera trajectory, we compare generated intermediate frames against ground truth and baseline video generation methods. UGG better preserves scene identity and dynamic content compared with the generic video completion baselines.

intrinsics. We compute the angular discrepancy over selected frame pairs, measuring whether the predicted camera motion agrees with the observed images.

Final Candidate Selection. Depth and pose are selected independently: there is no trade-off between the two scores, and the depth and pose results may come from different candidates. For depth prediction, we standardize and combine the verification residuals, rank candidates within each frame pair, and then aggregate these rankings across pairs. For pose prediction, we use the mean angular discrepancy between predicted and image-estimated relative translation directions. Finally, the model selects the candidate ℓ^* with the lowest task-specific verification score.

4 Experiments

We evaluate UGG on a suite of 4D scene understanding tasks, including camera-controlled generation (Sec. 4.2) and geometric prediction. Geometric prediction tasks under both dense and sparse observation settings include camera pose estimation (Sec. 4.3) and video depth estimation (Sec. 4.4). Our single unified model handles all tasks by changing only the clean/noisy masking pattern, without requiring task-specific architectures. We further show that UGG's generative formulation enables inference-time search, improving predicted scene consistency with additional samples (Sec. 4.5). Full code and additional visualizations are provided on the project page.

4.1 Model Training

We train on a weighted mixture of real and synthetic video datasets, including DL3DV-10K [29], RealEstate10K [74], TartanAir [54], BEDLAM [5], Aria Synthetic Environments [1], OmniWorld-Game [75], Virtual-KITTI2 [8], and Mid-Air [15]. These datasets provide heterogeneous supervision over RGB video, depth, camera pose, and text; missing modalities are omitted from the loss. UGG is trained with multiple clean/noisy mask patterns over modality tokens to cover generation and geometric prediction tasks. UGG weights are initialized from Wan2.2-TI2V-5B, then finetuned on 50-frame clips at $[288 \times 384]$ resolution. The VAE and text encoder are frozen. Training and preprocessing details, including the full dataset list and mixture weights, 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, the context contains the target camera trajectory and either the first RGB frame or the first and last RGB frames. The model then generates

Table 1: **Camera-controlled generation results.** Generation is conditioned on the first RGB frame (1-view) or the first and last frames (2-view), together with the camera trajectory. We report PSNR (↑), LPIPS (↓), and SSIM (↑). Darker and lighter blue shading indicate the best and second-best results within each setting.

	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
	UGG	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
	UGG	21.79	0.150	0.779	14.86	0.428	0.405	13.92	0.385	0.346

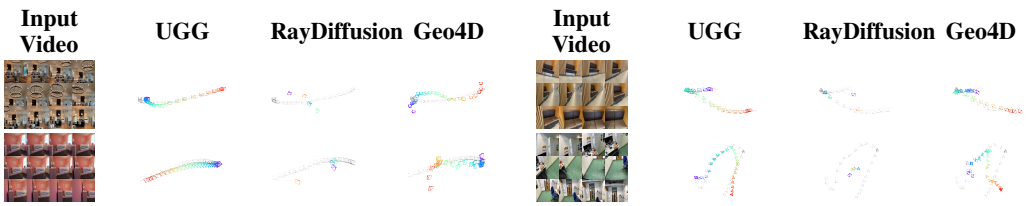


Figure 5: **Qualitative comparison of estimated camera trajectories.** The temporal evolution of each trajectory is encoded by a color gradient, from purple (start) to red (end). Ground-truth trajectory is shown in black. The leftmost column shows representative RGB frames from the input video. Compared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

the remaining frames from this context, which are evaluated against ground-truth videos using PSNR, SSIM, and LPIPS [56, 70].

Baselines. We compare quantitatively against Aether [76] and SEVA [72], and qualitatively against DFoT [43] and Wan2.1-FLF-14B in a first-last-frame conditioning setting [49]. These baselines represent unified generation and geometry prediction, diffusion-based novel view synthesis, diffusion-forcing-based video completion, and endpoint-conditioned video generation, respectively. In contrast to the task-specific baselines, UGG has the same multimodal architecture for both camera-controlled generation and geometry prediction.

Evaluation results. Camera-controlled generation results are shown in Table 1. Although not trained solely for novel-view synthesis, UGG 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 slightly better 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. We find that UGG also preserves dynamic content better; SEVA is primarily designed for static scenes, and DFoT distorts or removes 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 [7], TUM [44], ScanNetV2 [13], and ScanNet++. Following Geo4D [21], we align predicted trajectories to ground truth using Sim(3) Umeyama alignment in the dense setting, and report absolute trajectory error (ATE), relative pose error in translation (RPE_t), and in rotation (RPE_r). In the sparse setting, we evaluate the relative pose between the first and last frames using translation angular error (tAE) and RPE_r. Detailed metric definitions and evaluation protocols are provided in Appendix B.7.

Baselines. We compare quantitatively against Geo4D [21], Gen3R [18], Aether [76], and RayDiffusion [67]. 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 UGG predicts camera rays as part of a broader RGB-depth-camera scene representation.

Table 2: **Camera pose estimation results.** Dense: ATE, RPE_t, and RPE_r. Sparse: tAE and RPE_r, evaluated only on the two observed frames. Angular errors are in degrees; lower is better for all metrics. Darker and lighter blue shading mark the best and second-best distinct results in each column, including oracle results.

(a) Dense observations

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
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
RayDiffusion	0.139	0.081	1.809	0.471	0.293	14.495	0.088	0.058	2.942	0.170	0.090	8.909	0.344	0.148	25.941
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
UGG	0.010	0.004	0.076	0.098	0.034	0.708	0.018	0.007	1.052	0.038	0.016	0.938	0.087	0.036	3.861
UGG (TTS)	0.009	0.004	0.076	0.091	0.033	0.684	0.017	0.007	1.054	0.038	0.016	0.937	0.082	0.035	4.105
UGG (TTS Oracle)	0.008	0.004	0.076	0.072	0.030	0.695	0.015	0.006	1.053	0.033	0.015	0.939	0.072	0.035	4.078

(b) Sparse observations

Method	RealEstate10K		Sintel		TUM		ScanNetV2		ScanNet++	
	tAE	RPE_r	tAE	RPE_r	tAE	RPE_r	tAE	RPE_r	tAE	RPE_r
Geo4D	117.079	3.407	108.628	24.160	100.224	11.936	90.812	43.458	82.747	83.019
RayDiffusion	75.135	11.206	60.805	45.403	57.922	5.155	67.727	41.695	72.027	66.447
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
UGG	8.260	1.287	35.479	22.065	9.938	1.371	28.240	21.417	40.194	43.078
UGG (TTS)	3.475	1.393	18.733	20.646	8.520	1.444	20.057	18.911	31.835	37.875
UGG (TTS Oracle)	1.550	1.399	8.521	26.135	3.401	1.154	9.683	20.688	7.892	43.836

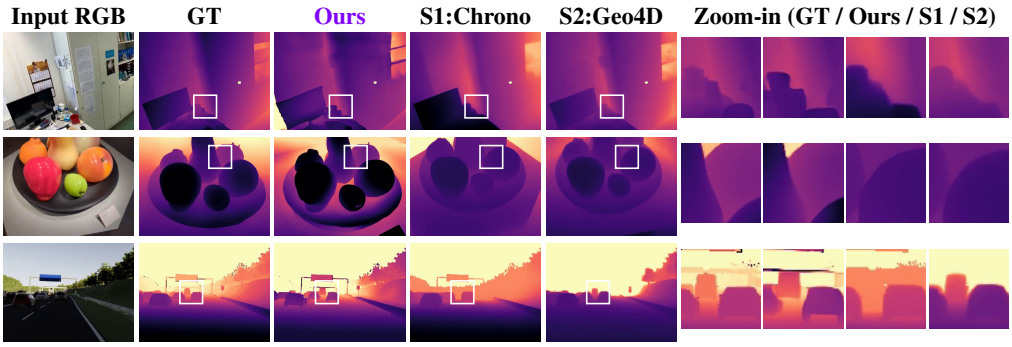


Figure 6: **Depth prediction qualitative comparison.** The white box marks the region shown in the zoom-in columns, ordered as GT, Ours, ChronoDepth, and Geo4D from left to right. Our predictions preserve sharper local structures and object boundaries in the highlighted regions.

Evaluation results. Table 2 reports camera pose estimation results. In the dense setting, UGG achieves lower ATE and RPE_t than the external baselines across all five datasets. For example, ATE is reduced from 0.190 to 0.087 on ScanNet++ compared with Geo4D. Figure 5 visualizes representative trajectories. Our predictions better follow the ground-truth motion, while RayDiffusion and Geo4D produce collapsed paths in challenging examples. In the sparse setting, UGG obtains the lowest tAE and RPE_r across all five datasets. These results suggest that the model can infer reasonable relative camera poses from only the first and last RGB frames, despite the ambiguity of sparse observations.

4.4 Depth Prediction

We evaluate depth prediction from RGB observations in the dense and sparse settings described in the previous section. In both cases, depth tokens are initialized from noise and denoised, testing whether the unified conditional-completion model learns a useful geometric posterior over scene depth.

Benchmarks and Metrics. We evaluate on ScanNet++, 7-Scenes [41], KITTI, Sintel, and Bonn. These datasets cover driving scenes, synthetic video, and indoor RGB-D scans. Following Geo4D [21], we apply video-level scale and shift alignment before evaluation, and report absolute relative error between the true and predicted depth (Abs Rel) and threshold accuracy, whether the predicted depth differs from the ground truth by less than a factor of 1.25 ($\delta < 1.25$).

Baselines. We compare with Geo4D, ChronoDepth [39], Gen3R, Aether, and Marigold [22]. 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

Table 3: **Depth prediction results.** We report Abs Rel (↓) and $\delta < 1.25$ (↑) on ScanNet++, 7-Scenes, KITTI, Sintel, and Bonn in dense and sparse settings. Darker and lighter blue shading indicate the best and second-best distinct results within each setting, including oracle results.

	Method	ScanNet++		7-Scenes		KITTI		Sintel		Bonn	
		Abs Rel	$\delta_{1.25}$	Abs Rel	$\delta_{1.25}$	Abs Rel	$\delta_{1.25}$	Abs Rel	$\delta_{1.25}$	Abs 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
	UGG	0.073	0.952	0.083	0.921	0.064	0.960	0.303	0.654	0.060	0.963
	UGG (TTS)	0.072	0.954	0.084	0.920	0.063	0.962	0.293	0.661	0.059	0.969
	UGG (TTS Oracle)	0.063	0.963	0.074	0.923	0.055	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
	UGG	0.101	0.884	0.087	0.916	0.099	0.917	0.258	0.660	0.049	0.977
	UGG (TTS)	0.082	0.917	0.086	0.916	0.087	0.934	0.219	0.678	0.047	0.979
	UGG (TTS Oracle)	0.055	0.966	0.075	0.923	0.068	0.946	0.177	0.781	0.043	0.979

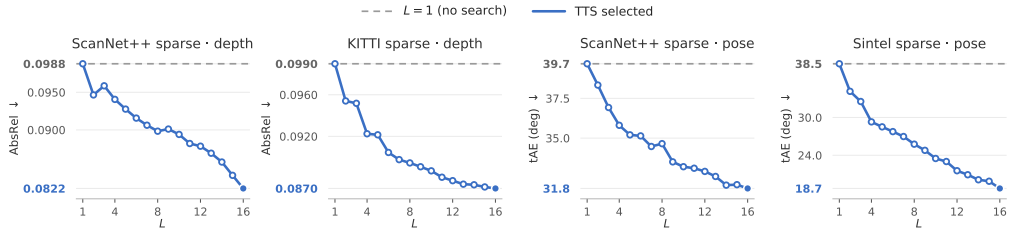


Figure 7: **Test-time search with increasing sample budget.** We sample up to 16 candidate completions and select one using the image-based depth or pose verifier. The dashed line shows the single-sample result, and the solid curve shows the selected result as L increases. Search improves several pose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

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 4 out of 5 datasets. At the same time, Geo4D remains stronger on Sintel and 7-Scenes metrics for the dense setting, indicating that specialized geometry models may 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 from different noise initializations and select depth and pose independently using their respective verification scores, as described in Section 3.4.

Tables 3 and 2 show that *UGG (TTS)* improves sparse depth Abs Rel and pose translation angular error on all five benchmarks. It also improves dense depth Abs Rel and pose ATE on four of five benchmarks. These results demonstrate that the verifier selects lower-error depth and camera translation estimates from the model's sampled hypotheses using image evidence alone. Figure 7 studies how pose and depth accuracy vary with the number of sampled candidates. Larger search budgets often improve the selected result, but the gains are not uniform because the verification score is only a proxy for ground-truth geometric accuracy.

UGG (TTS Oracle) selects one of the same 16 candidates per input by minimizing ground-truth scale-and-shift-aligned Abs Rel for depth, ATE for dense pose, or tAE for sparse pose. All metrics are reported for the selected candidate. Overall, these results show that inference-time search is a promising benefit of the generative formulation, while leaving room for stronger search criteria and verification.

Table 4: **Ablations.** Each row changes one training task, dataset, architecture, or camera representation. We report Abs Rel (↓) and $\delta_{1.25}$ (↑) for depth, ATE (↓) and RPE_r (↓, degrees) for pose, and PSNR (↑) for generation. RE10K: RealEstate10K; Spatial: SpatialVID; gen.: RGB generation; tr.: transformer; Dense rays: dense ray maps; Global cam.: global camera tokens. Shading marks the best result within each block.

(a) Depth prediction

	Bonn		KITTI	
Variant	Rel	$\delta_{1.25}$	Rel	$\delta_{1.25}$
Full model	0.135	0.843	0.152	0.791
w/o gen.	0.178	0.782	0.322	0.486
w/o RE10K	0.239	0.651	0.196	0.732
Single tr.	0.129	0.870	0.248	0.640

(b) Camera pose estimation

	ScanNetV2		ScanNet++	
Variant	ATE	RPE_r	ATE	RPE_r
Full model	0.074	2.91	0.193	6.18
w/o depth	0.081	4.30	0.234	7.57
Single tr.	0.083	3.02	0.204	6.60
Dense rays	0.071	2.80	0.206	6.09
Global cam.	0.103	3.44	0.233	6.61

(c) Camera-controlled generation

	PSNR		
Variant	RE10K	Spatial	DL3DV
Full model	20.18	13.67	12.60
Global cam.	20.03	12.89	11.64
Single tr.	19.68	12.75	11.61
w/o pose	13.26	11.87	10.95

4.6 Ablation Studies

Setup. We study the effects of joint training, training data, architecture, and camera representation using the smaller Wan2.1-1.3B instead of the final Wan2.2-5B model. Unless otherwise specified, we train on TartanAir and RE10K for 7,500 optimizer steps at batch size 8, with the same optimization settings. Additional details on the ablation setup are provided in Appendix B.4.

Joint training. Learning to generate RGB video also helps the model recover geometry. On KITTI, depth Abs Rel more than doubles when RGB generation is removed from training, rising from 0.152 to 0.322. Geometry supervision also benefits the other tasks: depth prediction improves camera estimation, while pose estimation improves camera-controlled generation by 6.92 dB on RE10K (Table 4a–c). Joint training therefore improves what the model learns about both scene appearance and geometry.

Heterogeneous training data. To test whether these benefits extend to data with incomplete annotations, we add RE10K, which provides RGB and camera annotations but no depth supervision. Adding it to TartanAir training improves depth prediction on both Bonn and KITTI, reducing Abs Rel from 0.239 to 0.135 and from 0.196 to 0.152, respectively (Table 4a). The model can therefore use RGB and camera supervision from additional datasets to improve depth prediction without requiring depth labels from those datasets.

Architecture. MoT allows the modalities to share information through attention while retaining separate processing branches. Compared with a single shared transformer, this design improves camera estimation on both benchmarks and generation on all three. Its effect on depth is less uniform: MoT performs better on KITTI, while the shared transformer is slightly better on Bonn (Table 4a–c).

Camera representation. We compare dense ray maps with global camera tokens in models trained only for pose estimation (Table 4b). Dense ray maps improve both trajectory and rotation accuracy on ScanNetV2 and ScanNet++. They also improve camera-controlled generation, yielding higher PSNR than global camera tokens on all three benchmarks (Table 4c). Aligning the camera representation with image locations is thus useful both for recovering cameras from images and for generating images from cameras. Additional results on history guidance are provided in Appendix B.4.

5 Conclusion

We present a unified generative geometry model for 4D scene understanding, formulating visual generation and geometric prediction as conditional completion over RGB, depth, and camera rays. Through noise patterns over modality tokens, a single model supports camera-controlled generation and geometric prediction under dense and sparse observations, while learning from heterogeneous datasets with varying modality availability. Across diverse datasets, UGG achieves strong video generation, depth prediction, and pose estimation performance, and further improves geometric consistency 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.

Acknowledgments and Disclosure of Funding

We thank the Kempner Institute at Harvard University for providing computational resources that supported this work. Ayush Tewari acknowledges the NVIDIA Academic Grant Program.

References

[1] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024. 6, 17, 24

[2] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025. 2

[3] 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

[4] 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

[5] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6, 17, 24

[6] 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. 2

[7] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012. 7

[8] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 6, 17, 24

[9] 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

[10] 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, 4

[11] 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, 2

[12] 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

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7

[14] 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

[15] Michaël Fonder and Marc Van Droogenbroeck. Mid-Air: A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 6, 17, 24

[16] 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

[17] 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

[18] Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3r: 3d scene generation meets feed-forward reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25358–25369, June 2026. 7

[19] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 17, 24

[20] 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

[21] 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, 21

[22] 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

[23] 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 Kontschieder. Mapanything: Universal feed-forward metric 3d reconstruction, 2025. 2

[24] Daniel Kersten, Pascal Mamassian, and Alan Yuille. Object perception as bayesian inference. *Annu. Rev. Psychol.*, 55(1):271–304, 2004. 2

[25] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. 2

[26] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 17, 24

[27] 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

[28] Hansen Jin Lillemark, Benhao Huang, Fangneng Zhan, Yilun Du, and Thomas Anderson Keller. Flow equivariant world models: Structured memory for dynamic environments. In *International Conference on Machine Learning (ICML)*, 2026. 3

[29] Lu Ling, Yicheng Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6, 17, 24

[30] 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

[31] 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

[32] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. SceneNet RGB-D: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 17, 24

[33] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 17, 24

[34] 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

[35] 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

[36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 5, 21

[37] 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

[38] 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

[39] 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

[40] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 2

[41] 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, 24

[42] 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. 3

[43] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. 3, 5, 7

[44] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. 7, 21

[45] 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

[46] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *European Conference on Computer Vision (ECCV)*, 2010. 21

[47] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Computer Vision – ECCV 2020*, pages 402–419. Springer, 2020. 5, 20, 21

[48] Joachim Tesch, Giorgio Becherini, Prerana Achar, Anastasios Yiannakidis, Muhammed Kocabas, Priyanka Patel, and Michael J. Black. BEDLAM2.0: Synthetic humans and cameras in motion. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. 17, 24

[49] 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. 2, 7, 24

[50] 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. 24

[51] 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

[52] 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

[53] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023. 2

[54] 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. 6, 17, 24

[55] 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

[56] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, April 2004. 7

[57] 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

[58] 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

[59] 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

[60] 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

[61] 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

[62] 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

[63] 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. 24

[64] 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

[65] 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

[66] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006. 2

[67] 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, 3, 7

[68] 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

[69] 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

[70] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7

[71] 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

[72] 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, 7

[73] 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

[74] 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, 17, 24

[75] 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, 17, 24

[76] Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-Aware Unified World Modeling. In *International Conference on Computer Vision*, pages 8535–8546, 2025. 7

[77] 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

Comprehensive visualizations across inference settings are available on our project page, including camera-conditioned generations, 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 and synthetic video datasets: RealEstate10K [74], DL3DV-10K [29], TartanAir [54], TartanAir V2, Spring [33], BEDLAM [5], BEDLAM2.0 [48], Virtual KITTI 2 [8], Aria Synthetic Environments [1], OmniWorld-Game [75], SceneNet RGB-D [32], MVS-Synth [19], Mid-Air [15], and MatrixCity [26]. 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, we use the provided RGB frames and camera poses; for DL3DV-10K, we use RGB frames and the official COLMAP camera poses. Neither provides depth. For the synthetic datasets, we use the rendered RGB, depth, and camera-pose annotations, except for OmniWorld-Game and MatrixCity, where we use RGB and depth only. We use language conditioning throughout: every training clip is paired with a scene-level text description, and a null prompt is used when no description is available.

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

$$R_i(u, v) = (\mathbf{d}_i(u, v), \mathbf{m}_i(u, v)), \qquad \mathbf{m}_i(u, v) = \mathbf{o}_i \times \mathbf{d}_i(u, v), \tag{4}$$

where $\mathbf{o}_i$ is the camera center and $\mathbf{d}_i(u, v)$ is the unit viewing direction in the world coordinate frame, which we define by the first camera of the clip. 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 $[\mathbf{d}, \mathbf{m}]$ at 36×48 resolution and are bilinearly 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, the maximum ray-moment magnitude over the clip (floored at 1), and use it consistently for camera ray moments and depth; for clips without camera supervision, s is the median valid depth. Ray directions are already unit-normalized, while ray moments are divided by s and clamped to $[-1, 1]$. For depth, we use a disparity-like parameterization

$$\tilde{D} = \frac{2}{1 + D/s} - 1,$$

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 Wan2.2-TI2V-5B 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,

$$[z^I,\ z^{R_d},\ z^{R_m},\ z^D], \tag{5}$$

corresponding to RGB video, ray directions, ray moments, and depth, respectively. The two ray latents are concatenated along the channel dimension and embedded as a single ray token stream, so the transformer processes three token streams: RGB, ray, and depth.

Tokenization. All clips contain 50 frames resized to 288×384. RGB frames are encoded by the frozen Wan2.2 VAE into latent tensors with 48 channels and spatial size 18×24. 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 first and last frames each occupy their own latent frame and the 48 interior frames are compressed $4\times$ in time, so the temporal latent length per modality is $T_{\text{lat}} = 14$. Thus, each modality stream has shape $[B, 48, 14, 18, 24]$, and the concatenated four-stream latent has shape $[B, 48, 56, 18, 24]$. We use a 3D patch size of $[1, 2, 2]$, giving a per-stream patch grid of $14 \times 9 \times 12$, or 1512 tokens per stream. Since the two ray latents share one token stream, the transformer operates on $3 \times 1512 = 4536$ 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 48-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. The 12 packed channels of the direction and moment latents are concatenated into a 24-channel input to the ray patch embedding. 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, output projections, and query/key normalization layers 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 feed-forward networks, while text cross-attention is shared across streams. Each stream has its own output head. 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 Wan2.2-TI2V-5B configuration: hidden dimension 3072, feed-forward dimension 14336, 24 attention heads, and 30 transformer layers. The VAE latent channel dimension is 48, the VAE stride is $[4, 16, 16]$ in time, height, and width, and the transformer patch size is $[1, 2, 2]$. Text conditioning uses the Wan umT5 text encoder with prompt embeddings of dimension 4096 and a maximum length of 512 tokens.

Initialization of new modality branches. When initializing from the RGB-only Wan2.2 checkpoint, all compatible transformer weights are copied from the pretrained model. The RGB patch embedding initializes the depth patch embedding, and the RGB output head initializes the ray and depth heads. The ray patch embedding has a different number of input channels and is trained from scratch. Attention, cross-attention, feed-forward, time-embedding, and text-conditioning weights are copied into the corresponding modality streams. 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. We sample raw diffusion times $t \in [0, 1]$ for the context and target groups, as described below, and apply the shifted schedule

$$t' = \frac{\alpha t}{1 + (\alpha - 1)t}, \qquad \alpha = 3.0. \tag{6}$$

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

$$z_{t'} = (1 - t')z_0 + t'\epsilon, \tag{7}$$

and the model predicts the flow target

$$v^\star = \epsilon - z_0. \tag{8}$$

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

$$\mathcal{L} = \frac{1}{4}\left(\mathcal{L}_I + \mathcal{L}_{R_d} + \mathcal{L}_{R_m} + \mathcal{L}_D\right). \tag{9}$$

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 share a single uniformly sampled noise level. Target tokens share a separately sampled noise level. 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 288×384 resolution, sampled at a dataset-specific frame rate between 8 and 30 FPS. Optimization uses AdamW with betas $(0.9, 0.95)$, weight decay 0.05, and gradient clipping at 1.0. The learning rate warms up over 250 steps to a peak of 4×10^{-5} and is decayed to 6×10^{-6} by the end of training. Training is performed with `bf16` mixed precision and FSDP on 128 NVIDIA H200 GPUs for two weeks, or approximately $100{,}000$ optimizer steps. We use batch size 8 per GPU without gradient accumulation, giving a global batch size of 1024. We use gradient checkpointing for memory efficiency.

Dataset sampling weights. The training mixture uses relative dataset weights, which are multiplied by dataset size and normalized by the mixture loader. The resulting sampling proportions are 18.8% TartanAir, 15.0% RealEstate10K, 10.0% TartanAir V2, 8.3% Aria Synthetic Environments, 7.3% BEDLAM, 7.3% BEDLAM2.0, 7.2% DL3DV-10K, 6.2% SceneNet RGB-D, 5.2% OmniWorld-Game, 4.2% Virtual KITTI 2, 4.2% MatrixCity, 2.6% MVS-Synth, 2.6% Mid-Air, and 1.0% Spring.

B.4 Ablation Setup

Task-removal ablations. When we remove a task from training, we keep the training clips and optimization schedule unchanged. We disable its loss and replace its input tokens with noise, keeping the remaining loss weights fixed. When removing RGB generation, RGB remains available as clean conditioning input. The global-camera variant uses camera poses as conditioning without predicting them. The no-pose variant is an image-only baseline, without camera conditioning or pose supervision. Depth and pose estimation use the dense evaluation setting where all RGB frames are provided as input.

Table 5: **History-guidance weight.** We report ATE (↓) and RPE_r (↓, degrees) on ScanNetV2 and ScanNet++. Shading marks the best result in each column.

	ScanNetV2		ScanNet++	
w_{hist}	ATE	RPE_r	ATE	RPE_r
0	0.0796	2.46	0.2170	5.77
1	0.0741	2.83	0.2089	6.09
2	0.0740	3.24	0.2091	6.45
4	0.0765	4.22	0.2141	7.70

History guidance. History guidance improves trajectory alignment at the cost of rotation accuracy. Moderate weights ($w_{\text{hist}} \in \{1, 2\}$) reduce ATE relative to unguided inference, but rotational RPE increases with guidance strength (Table 5). Increasing the weight to 4 worsens both metrics relative to moderate guidance.

B.5 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, the denoising loop for one 50-frame clip takes approximately 20 seconds on a single NVIDIA H100 or H200 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

$$\hat{v}_{\text{HG}} = \hat{v}_H + w_{\text{hist}} \left(\hat{v}_H - \hat{v}_\emptyset \right). \tag{10}$$

This is analogous to classifier-free guidance, but the dropped condition is the observed history/context rather than the text prompt. 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 against RAFT [47] correspondences between the observed frames, without ground-truth depth or camera pose: depth candidates by how well their scale-and-shift-aligned depth explains the observed correspondences, and pose candidates by the agreement between their relative translation directions and those estimated from the correspondences. This implements the inference-time search procedure described in Sec. 3.4.

B.6 Depth Evaluation

We decode our disparity predictions to depth before evaluation. All methods use the same affine-invariant evaluation protocol: we fit a single scale and shift per clip by minimizing an L1 objective over valid pixels pooled across the scored frames,

$$(s^*, t^*) = \arg\min_{s,t} \sum_{p \in \mathcal{V}} |s\hat{d}_p + t - d_p|, \qquad \widetilde{d}_p = \max(s^* \hat{d}_p + t^*, 10^{-6}), \tag{11}$$

where $\hat{d}_p$ and d_p are predicted and ground-truth depths. Ground-truth depths are used in metres without additional normalization. The shared fit across frames preserves sensitivity to temporal depth consistency.

We report absolute relative error and threshold accuracy:

$$\text{AbsRel} = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} \frac{|\widetilde{d}_p - d_p|}{d_p}, \tag{12}$$

$$\delta_{<1.25} = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} \mathbf{1}\left[\max\left(\frac{\widetilde{d}_p}{d_p}, \frac{d_p}{\widetilde{d}_p} \right) < 1.25 \right]. \tag{13}$$

Both metrics average valid pixels jointly across the clip, using a strict threshold for $\delta_{<1.25}$.

Dense evaluation scores all 50 frames; sparse evaluation fits and scores only the first and last observed frames. On Sintel, Bonn, and KITTI, dense evaluation uses a 240×320 grid, while sparse evaluation uses the cropped ground-truth resolution. We report an unweighted mean of per-clip scores for each dataset.

B.7 Camera Pose Evaluation

Dense setting. In the dense setting, models condition on the full RGB sequence. Following Geo4D [21], we align predicted trajectories to ground truth using Sim(3) Umeyama alignment. We report absolute trajectory error (ATE), relative pose error in translation ($\mathrm{RPE_t}$), and in rotation ($\mathrm{RPE_r}$). ATE measures global trajectory agreement after alignment, while RPE measures local frame-to-frame motion consistency.

Sparse setting. In the sparse setting, models condition on only the first and last RGB frames, and we evaluate their relative pose. With only two camera centers, Sim(3) alignment can match both positions exactly, making the standard aligned ATE uninformative. For sparse setting results we therefore report translation angular error (tAE), measuring the angular difference between the predicted and ground truth relative translation directions, and $\mathrm{RPE_r}$ for relative rotation error.

Let $T_i = [R_i \mid \mathbf{c}_i]$ denote camera-to-world poses. For dense evaluation, we align each predicted trajectory to ground truth using a single Sim(3) Umeyama fit over all camera centers, yielding aligned predictions $\widetilde{T}_i$. We define

$$\Delta_{ij} = T_i^{-1} T_j, \qquad E_{ij} = \Delta_{ij}^{-1} (\widetilde{T}_i^{-1} \widetilde{T}_j). \tag{14}$$

Following [44], we report

$$\mathrm{ATE} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \|\widetilde{\mathbf{c}}_i - \mathbf{c}_i\|_2^2}, \tag{15}$$

$$\mathrm{RPE_t} = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-2} \|\mathbf{t}(E_{i,i+1})\|_2^2}, \tag{16}$$

$$\mathrm{RPE_r} = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-2} \theta(R(E_{i,i+1}))^2}, \tag{17}$$

where $\mathbf{t}(E)$ and $R(E)$ are the translation and rotation components, and $\theta(R) = \frac{180}{\pi} \arccos((\mathrm{tr}(R) - 1)/2)$. Translation errors remain in ground-truth units, without additional normalization.

For sparse evaluation, we use the unaligned first–last relative poses $\Delta_{0,N-1} = [R \mid \mathbf{t}]$ and $\widehat{\Delta}_{0,N-1} = [\widehat{R} \mid \widehat{\mathbf{t}}]$:

$$\mathrm{RPE_r} = \theta(R^\top \widehat{R}), \qquad \mathrm{tAE} = \frac{180}{\pi} \arccos\left(\frac{\widehat{\mathbf{t}}^\top \mathbf{t}}{\|\widehat{\mathbf{t}}\|_2 \|\mathbf{t}\|_2} \right). \tag{18}$$

Both are invariant to global similarity transformations, and tAE lies in $[0^\circ, 180^\circ]$. All inverse-cosine arguments are clipped to $[-1, 1]$. We report an unweighted mean of per-clip scores for each dataset.

B.8 Inference-time Search Details

This section gives the equations and hyperparameters of the search in Sec. 3.4. The verifier uses only the observed RGB frames, RAFT-Large optical flow [47], DINOv2 ViT-S/14 features [36], and each candidate's own predicted depth, cameras and intrinsics. It never uses ground-truth depth or pose, never uses another depth or pose model, and never averages or compares candidates against each other except when ranking them. All computation is at the model resolution (288×384), and residuals are evaluated on a stride-2 pixel grid.

Conventions. Cameras follow the OpenCV convention (x right, y down, z forward) and pixel coordinates are at 288×384. For a frame pair (i, j), the relative camera $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ maps camera-i coordinates to camera-j coordinates, $\mathbf{X}_j = \mathbf{R}_{ij}\mathbf{X}_i + \mathbf{t}_{ij}$. For a candidate this is $\hat{\mathbf{T}}_{ij} = \hat{\mathbf{T}}_j^{-1} \hat{\mathbf{T}}_i$, where $\hat{\mathbf{T}}_i$ is its predicted camera-to-world matrix. Image-estimated translations are defined only up to scale and have unit norm.

Forward-backward consistency. RAFT is run in both directions to give flows $\mathbf{F}_{ij}$ and $\mathbf{F}_{ji}$. The correspondence $\mathbf{x} \mapsto \mathbf{x}' = \mathbf{x} + \mathbf{F}_{ij}(\mathbf{x})$ is kept only if $\mathbf{x}'$ lies inside frame j and the round trip returns close to $\mathbf{x}$ [46]:

$$\left\| \mathbf{F}_{ij}(\mathbf{x}) + \mathbf{F}_{ji}(\mathbf{x}') \right\|^2 \leq 0.01 \left(\|\mathbf{F}_{ij}(\mathbf{x})\|^2 + \|\mathbf{F}_{ji}(\mathbf{x}')\|^2 \right) + 0.5. \tag{19}$$

A pixel whose match in frame j maps back somewhere else is occluded, moving, or a flow failure. Such pixels are discarded before any camera estimate or residual is computed.

Frame-pair selection. A fixed frame gap gives almost no parallax on slow video and breaks RAFT on fast video. For dense observations, we therefore choose each gap so that the pair has a fixed target motion τ:

$$g(i) = \mathrm{clip}\Big(\mathrm{round}\big(\tau \,/\, \mathrm{med}_{\mathbf{x}} \|\mathbf{F}_{i,i+1}(\mathbf{x})\|\big),\, 1,\, 12\Big), \tag{20}$$

with $\tau = 16$ px for every task except dense pose, where $\tau = 32$ px. The same values are used on every benchmark. *Dense depth*: 12 anchors i spaced evenly over the clip, each paired with $i + g(i)$. *Dense pose*: 16 anchors, each used in both directions (32 pairs). *Sparse depth*: only the two given frames are used. Correspondences are computed directly between them, and only the candidate's depth predictions at the observed frames are scored. *Sparse pose*: only the two given frames are used, testing the candidate's predicted first–last relative camera.

Relative camera from RGB (RANSAC). On up to 6,000 consistent correspondences (subsampled uniformly) we estimate the essential matrix with the five-point algorithm inside RANSAC (OpenCV `findEssentialMat`: confidence 0.999, 1 px threshold). We then decompose it with `recoverPose`, which picks the solution that places points in front of both cameras. For *depth*, this camera is computed once per pair and shared by all candidates, so the candidates differ only in their depth. Using each candidate's own predicted camera instead put the ranking at chance, because a wrong camera can be compensated by a wrong depth. For *pose*, the reference camera is recomputed with each candidate's own predicted intrinsics $\hat{\mathbf{K}}$.

Depth residuals. For pair (i, j) and candidate inverse depth w at frame i, let $\mathbf{f}_{\mathrm{rot}}$ be the flow induced by rotation alone (depth $\to \infty$). We linearize reprojection around zero inverse depth as $\mathbf{x}'(w) \approx \mathbf{x} + \mathbf{f}_{\mathrm{rot}} + w\mathbf{b}$, where $\mathbf{b} = \partial \mathbf{x}'(w)/\partial w|_{w=0}$ depends only on the shared camera and pixel position. Let $\mathbf{c} = w\mathbf{b}$ be the candidate's predicted parallax, used for parallax filtering, and let $\hat{\mathbf{b}} = \mathbf{b}/\|\mathbf{b}\|$. Let $\mathbf{p} = \mathbf{F}_{ij} - \mathbf{f}_{\mathrm{rot}}$ be the observed parallax. We split $\mathbf{p}$ into $a = \mathbf{p} \cdot \hat{\mathbf{b}}$, the part a depth can explain, and the perpendicular remainder $e_\perp = \|\mathbf{p} - a\hat{\mathbf{b}}\|$, the part no depth can explain. *Gauge*: with $\hat{w} = w/\mathrm{med}(w)$, a closed-form weighted least-squares fit gives the scale and shift $(\alpha, \beta) = \arg\min_{\alpha,\beta} \sum_{\mathbf{x}} \omega(\mathbf{x}) \|\mathbf{p} - (\alpha\hat{w} + \beta)\,\mathbf{b}\|^2$, where the sum is over kept pixels and $\omega(\mathbf{x}) = \|\nabla I_i(\mathbf{x})\|$. We write the aligned inverse depth as $\tilde{w} = \alpha\hat{w} + \beta$. *Epipolar filtering*: we keep only pixels that pass Eq. 19, have predicted parallax $\|\mathbf{c}\| > 0.5$ px, and have $e_\perp$ at or below its 80th percentile in the pair.The epipolar residual is used only for filtering. Kept pixels are weighted by image gradient magnitude $\|\nabla I_i\|$. The aligned warp $\mathbf{x}' = \mathbf{x} + \mathbf{f}_{\mathrm{rot}} + \tilde{w}\,\mathbf{b}$ defines three pointwise residuals:

$$\begin{aligned} r_{\mathrm{corr}}(\mathbf{x}) &= \big|a - \tilde{w}\|\mathbf{b}\|\big|, \\ r_{\mathrm{photo}}(\mathbf{x}) &= \big\|I_i(\mathbf{x}) - I_j(\mathbf{x}')\big\|_1, \\ r_{\mathrm{feat}}(\mathbf{x}) &= 1 - \cos\big(\phi_i(\mathbf{x}), \phi_j(\mathbf{x}')\big). \end{aligned} \tag{21}$$

Each residual is aggregated as an image-gradient-weighted mean over kept pixels.

where ϕ are DINOv2 ViT-S/14 patch features, bilinearly upsampled to image resolution and sampled at $\mathbf{x}'$. Sparse depth uses r_{corr}.. Dense depth uses $\{r_{\mathrm{corr}}, r_{\mathrm{feat}}, r_{\mathrm{photo}}\}$.

Pose residual. For each pair, $\theta_{ij}^{(\ell)} = \angle\big(\hat{\mathbf{t}}_{ij}^{(\ell)}, \mathbf{t}_{ij}^{\mathrm{RGB}}\big)$ in degrees, where $\mathbf{t}_{ij}^{\mathrm{RGB}}$ is the RANSAC translation computed with candidate ℓ's intrinsics.

Selection score. Depth and pose are selected *independently*. There is no weighting between them, and the two results can come from different candidates. *Depth.* For pair p, each residual k in the rule set $\mathcal{K}$ is z-normalized across the L candidates. The normalized residuals are averaged into $s_p^{(\ell)} = \frac{1}{|\mathcal{K}|} \sum_k z_p^{(\ell)}(r_k)$, and candidates are ranked within the pair, $\rho_p^{(\ell)} \in [0, 1]$. Pairs are weighted by how strongly they separate the candidates, $\omega_p = \mathrm{IQR}_\ell(s_p^{(\ell)})$, because a pair on which all candidates score alike carries no information:

$$E_{\mathrm{depth}}^{(\ell)} = \frac{\sum_p \omega_p\, \rho_p^{(\ell)}}{\sum_p \omega_p}. \tag{22}$$

Using ranks means a single pathological pair cannot dominate. *Pose.*

$$E^{(\ell)}_{\text{pose}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \theta^{(\ell)}_{ij}. \tag{23}$$

The selected candidate is $\ell^* = \arg\min_\ell E^{(\ell)}_{\text{task}}$. If no pair survives the filtering for a scene, the verifier abstains and the scene is scored as a uniformly random pick from the pool.

Choosing the rules. The residual set and the target τ for each task and protocol were chosen by leave-one-benchmark-out: the rule is selected on the other benchmarks and evaluated on the held-out one. Sparse depth and sparse pose select the same rule in every fold.

C Existing Assets and Licenses

We use existing public datasets and pretrained model assets. Table 6 summarizes the role of each asset in this work and the corresponding license or access terms. All datasets are used for academic research purposes only. We do not redistribute the original datasets, videos, game assets, or other third-party data. Model checkpoints are released under terms compatible with the upstream data and model licenses.

Table 6: **Datasets and pretrained assets.** We summarize the role of each asset and its license or access terms. License names are abbreviated, original agreements and third-party restrictions apply.

Asset	Use in this work	License / access terms
Training datasets		
DL3DV-10K [29]	RGB and pose; generation evaluation	CC BY-NC 4.0; dataset terms; research and education only
RealEstate10K [74]	RGB and pose; generation and pose evaluation	CC BY 4.0
TartanAir / TartanAir V2 [54]	Synthetic RGB, depth, and pose	CC BY 4.0
Spring [33]	Synthetic RGB, depth, and pose	CC BY 4.0
BEDLAM [5]	Human-centric RGB, depth, and pose	BEDLAM license; noncommercial research and education
BEDLAM2.0 [48]	Human-centric RGB, depth, and pose	Perceiving Systems License 1.0; noncommercial research; third-party terms
Virtual KITTI 2 [8]	Driving RGB, depth, and pose	CC BY-NC-SA 3.0
Aria Synthetic Environments [1]	Indoor RGB, depth, and pose	Aria dataset agreement; noncommercial research
OmniWorld-Game [75]	Game-scene RGB and depth	CC BY-NC-SA 4.0; additional game-asset terms and notices
SceneNet RGB-D [32]	Indoor RGB, depth, and pose	SceneNet terms; noncommercial use
MVS-Synth [19]	Urban RGB, depth, and pose	Research release; GTA V terms apply
Mid-Air [15]	Drone RGB, depth, and pose	CC BY-NC-SA 4.0
MatrixCity [26]	City-scale RGB and depth	CC BY-NC 4.0
Evaluation-only datasets		
SpatialVID [50]	Camera-controlled generation	CC BY-NC-SA 4.0
ScanNet++ [63]	Indoor pose and depth prediction	ScanNet++ terms; noncommercial research and education; redistribution restricted
7-Scenes [41]	Indoor RGB-D depth prediction	Microsoft 7-Scenes agreement; noncommercial use
Pretrained model		
Wan2.2-TI2V-5B [49]	Backbone initialization; transformer finetuning; frozen VAE and text encoder	Apache 2.0

CC: Creative Commons; BY: attribution; NC: noncommercial; SA: share-alike.

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: [Yes]

   Justification: The current submission provides all open-source code, model checkpoints, reproduction scripts at the project website. 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, preprocessing, 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 5B-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: [Yes]

    Justification: The release includes intended-use documentation and responsible-use guidance, warning about inaccurate geometry and synthetic content and advising against deceptive, privacy-invasive, or unvalidated safety-critical uses.

    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. Our release of code or checkpoints is 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: [Yes]

    Justification: We release UGG source code and model weights at `https://github.com/amanikiruga/UGG`. The accompanying README and model card document installation, dependencies, supported tasks, inference examples, training configurations, hardware requirements, licensing, and known 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.