PVWANDER: 3D MEMORY-GUIDED PANORAMIC VIDEO WANDERING WITH PATH-FACING CANONICALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon panoramic video wandering aims to continuously synthesize scene-consistent 360° observations along user-specified camera trajectories from an initial equirectangular projection (ERP) panorama. Existing panoramic video models have achieved promising results in camera-controlled generation, while recent 3D-guided approaches further provide geometric guidance for improving scene consistency. However, stable long-horizon wandering remains challenging due to diverse motion–orientation relationships and imperfect memory guidance induced by accumulated generation artifacts and depth inaccuracies. To address these challenges, we propose **PVWander**, a 3D-memory-guided framework that accumulates generated observations into an online point-cloud memory and reprojects it to guide subsequent generation, with two complementary designs: *Path-Facing canonicalization* for motion–orientation alignment and *Timestep-aware gating modulation* for adaptive memory guidance. *Path-Facing canonicalization* exploits the horizontal cyclic symmetry of ERP panoramas to uniformly align each clip's ERP orientation with its principal translation direction, yielding more consistent, forward-like relative motions while preserving the prescribed camera path. *Timestep-aware gating modulation* exploits the coarse-to-fine nature of diffusion denoising to adaptively regulate reprojected memory guidance through timestep-dependent channel-wise gates, preserving its structural benefits while mitigating visual artifacts from imperfect projections. Extensive experiments demonstrate that PVWander achieves improved long-horizon generation quality, while ablations validate the effectiveness of both designs. Code will be available.

1 INTRODUCTION

Building generative models that can continuously unfold a 3D world along camera motion is an important step toward spatial intelligence. Such models can provide controllable virtual worlds for applications such as robotics and virtual reality Xie et al. (2025). Equirectangular panoramas are naturally suited to this setting by capturing the full 360° surrounding view at each camera location Yang et al. (2025); Sun et al. (2026). In this work, we focus on *long-horizon panoramic video wandering*: given an initial ERP panorama and a series of user-specified camera trajectories, the goal is to continuously generate panoramic observations along these trajectories.

Existing camera-controlled panoramic video generation methods have achieved promising results by synthesizing panoramic observations under user-specified camera trajectories Ji et al. (2025); Yin et al. (2026). However, maintaining scene appearance and structure becomes challenging when these methods are recursively applied over longer trajectories. Recent approaches Yang et al. (2025); Chen et al. (2026); Wang et al. (2026) introduce explicit 3D representations (e.g., point clouds or 3D Gaussians) derived from the input panorama to provide geometric guidance. Yet these methods primarily focus on single-segment generation, leaving closed-loop integration of newly generated observations during long-horizon rollout unexplored. In perspective video domain, recent works Wu et al. (2026); Shen et al. (2026) instead maintain an online 3D memory that is continuously updated and reprojected to condition subsequent video segments. Building on this paradigm, we propose **PVWander**, a 3D-memory-guided framework that continuously incorporates newly generated panoramic observations to guide long-horizon wandering.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

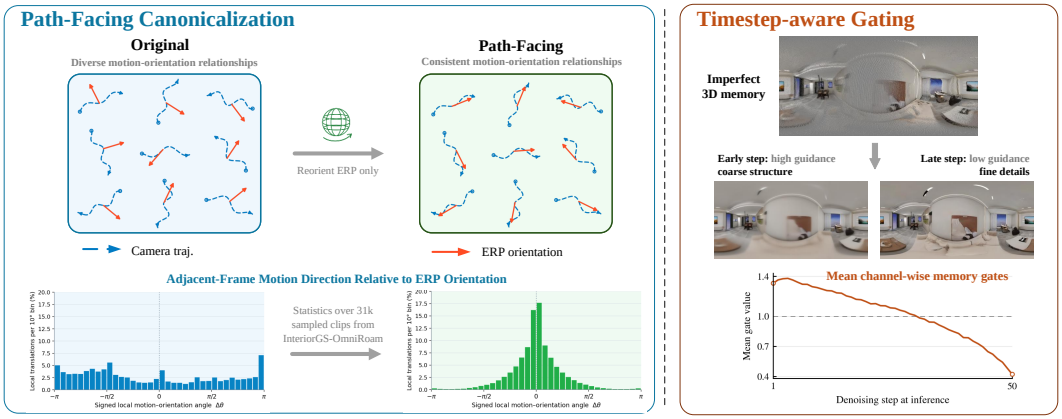


Figure 1: Two core designs in PVWander. Left: Path-Facing canonicalization reorients ERP to reduce variation in motion direction relative to per-frame ERP orientation. Right: Timestep-aware gating modulates reprojected memory features over 50 denoising steps, from high to low noise.

Even with online 3D memory, stable long-horizon panoramic video generation still faces two key challenges. First, free-form camera control involves highly varying motion directions relative to the viewing orientation, ranging from forward/backward to lateral motion. Such diverse motion–orientation relationships can exacerbate visual artifacts and temporal drift over long trajectories. Second, while 3D memory provides useful geometric guidance through reprojection, it is reconstructed from generated observations and estimated depth. Generation artifacts and depth inaccuracies can therefore introduce unreliable local details into reprojected views, potentially propagating errors across segments. To address these challenges, we introduce two complementary designs: *Path-Facing canonicalization* and *Timestep-aware gating modulation*, as illustrated in Fig. 1.

First, to reduce variation in motion–orientation relationships, we exploit the horizontal cyclic symmetry of ERP panoramas, which allows their orientation to change without altering scene content (see Fig. 9 in Appendix A.4). Based on this observation, we propose *Path-Facing canonicalization*, which estimates the principal translation direction of each segment and aligns all-frame ERP orientation with this direction. As illustrated in Fig. 1 (left), this reorientation transforms diverse motion–orientation relationships into a more consistent forward-like configuration while preserving the prescribed camera path. Second, to better exploit imperfect 3D memory projections, we aim to retain their useful coarse geometric guidance while reducing the influence of unreliable local details. Diffusion denoising generally follows a coarse-to-fine process, establishing global structure before refining local appearance Qian et al. (2024); Ren et al. (2026). This motivates our *Timestep-aware gating modulation*, which predicts timestep-dependent channel-wise gates to adapt memory guidance to different denoising stages. As shown in Fig. 1 (right), the learned mean gate is stronger early in denoising and generally decreases during later refinement.

Extensive experiments evaluate PVWander on long-horizon generation and zero-shot generalization, while ablations demonstrate the effectiveness of its key designs. Our main contributions are threefold: (1) We propose **PVWander**, a 3D-memory-guided closed-loop framework for long-horizon panoramic video wandering that combines motion–orientation canonicalization with adaptive memory guidance. (2) We propose *Path-Facing canonicalization*, which exploits ERP cyclic symmetry to reorient panoramas along the principal translation direction, yielding more consistent, forward-like motion–orientation relationships. (3) We propose *Timestep-aware gating modulation*, which adaptively modulates imperfect 3D memory guidance through timestep-dependent channel-wise gates.

2 Related Work

Camera-controlled panoramic video generation synthesizes observations along specified trajectories Ji et al. (2025); Yin et al. (2026). Methods that decouple camera rotation or use translation-only motion simplify trajectory conditioning Zhang et al. (2026); Cheng et al. (2026); Liu et al. (2026). Yet translation directions can vary widely relative to the ERP orientation, regardless of

whether that orientation is fixed or varies over time. Our *Path-Facing canonicalization* addresses this motion–orientation variability by aligning a shared ERP orientation for each segment with its principal translation direction, yielding more consistent forward-like motion. For long-horizon generation, existing methods reuse historical observations through view retrieval Yu et al. (2025a); Li et al. (2025; 2026a) or persistent 3D memory Wu et al. (2026); Wang et al. (2025b); Shen et al. (2026). PVWander adopts the online 3D-memory paradigm, with a focus on adapting the influence of reprojected memory across denoising stages through *Timestep-aware gating modulation*. A more comprehensive related work section is provided in Appendix A.1.

3 METHOD

Given an initial ERP panorama and user-specified long camera trajectories, **PVWander** generates panoramic video segment by segment with an online 3D point-cloud memory. As shown in Fig. 2, the memory is reprojected to target viewpoints to guide each segment, and newly generated observations are lifted into 3D to update the memory for subsequent generation. We first describe this memory-guided generation framework, followed by *Path-Facing canonicalization* for motion–orientation alignment and *Timestep-aware gating modulation* for adaptive guidance from imperfect memory projections. Preliminaries on the video diffusion backbone and 3D memory-guided generation are provided in Appendix A.2.

3.1 ONLINE 3D MEMORY-GUIDED PANORAMIC GENERATION

Online 3D Memory. We represent the memory at generation step s as a set of colored 3D points $\mathcal{M}^{(s)} = \{(\mathbf{p}_j, \mathbf{c}_j)\}_{j=1}^{N_s}$, where $\mathbf{p}_j \in \mathbb{R}^3$ denotes a world-coordinate point and $\mathbf{c}_j \in \mathbb{R}^3$ stores its RGB observation. The memory is initialized by lifting the input panorama into 3D using metric radial depth estimated by PVMetric (see Appendix A.9). After each segment, the generated observations are converted into 3D using their estimated metric depth, back-projected into the common world coordinate system, and added to the memory.

Before generating the next segment, following Shen et al. (2026), we select source observations based on their geometric coverage of target viewpoints, favoring complementary views that cover previously uncovered regions. The memory points associated with the selected observations are transformed into each target camera coordinate system and projected onto the ERP image plane, producing partial panoramic renders that provide spatial guidance for subsequent generation.

Memory-Guided Video Generation. Each segment is conditioned on its first frame, the target camera trajectory, and the reprojected memory renders. The first-frame condition is the input panorama for the initial segment and the final frame of the preceding segment thereafter. For camera conditioning, following Ji et al. (2025), we adapt Plücker ray encoding to ERP geometry and encode the ray maps using the existing camera adapter of the Wan backbone (see Appendix A.3 for details). For memory guidance, we encode the panoramic renders with the video VAE and map the resulting latents, together with a projection-validity mask indicating memory coverage, into render tokens using a lightweight render adapter. Following Matrix-3D Yang et al. (2025), we add the camera and render tokens to the patch-embedded video features before passing them to the diffusion backbone:

$$\mathbf{h}_t = \mathbf{h}_t^{\text{base}} + \mathbf{h}^{\text{cam}} + \mathbf{h}^{\text{render}}, \tag{1}$$

where t denotes the diffusion timestep, $\mathbf{h}_t^{\text{base}}$ denotes the patch-embedded features of the noisy video latent together with the first-frame condition, and $\mathbf{h}^{\text{cam}}$ and $\mathbf{h}^{\text{render}}$ denote the camera and memory-render conditioning tokens, respectively.

While this closed-loop process enables scene information to be reused across segments, stable long-horizon generation still faces two challenges. First, free-form trajectories induce diverse motion directions relative to ERP orientation, increasing the difficulty of motion modeling and exacerbating visual degradation over long-horizon generation. Second, the online memory may accumulate generation artifacts and depth inaccuracies, which can propagate through subsequent reprojections. We address these issues with *Path-Facing canonicalization* and *Timestep-aware gating modulation*, respectively.

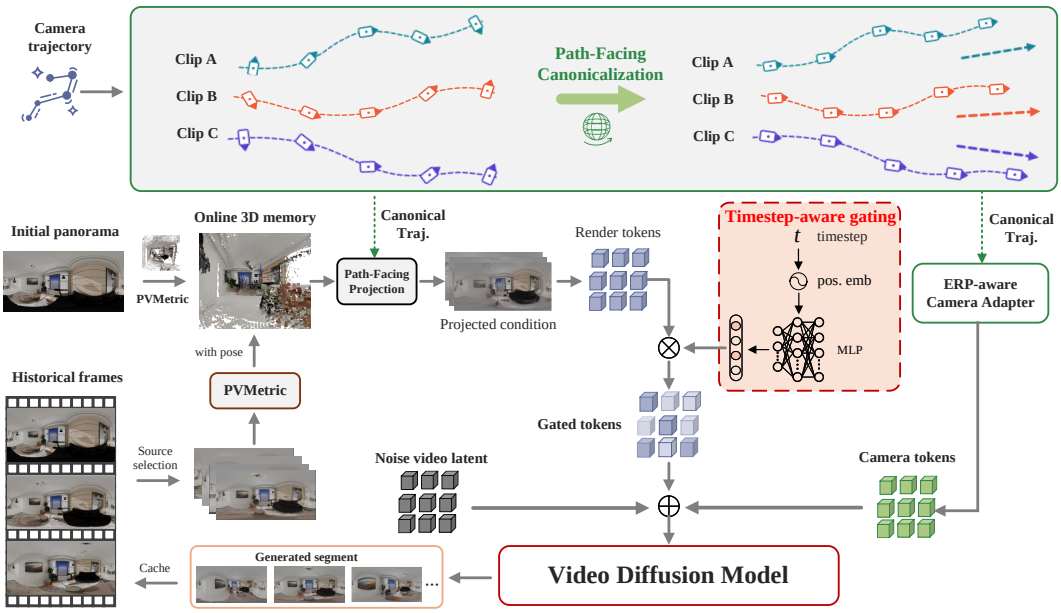


Figure 2: Overview of PVWander. Online 3D memory accumulates and reprojects generated observations to guide subsequent segments. Path-Facing canonicalization aligns ERP orientation with the principal translation direction for more consistent motion–orientation relationships, while Timestep-aware gating balances useful geometric guidance and unreliable local details across denoising stages.

3.2 Path-Facing Canonicalization

Path-Facing canonicalization reduces motion–orientation variation by aligning each segment's ERP orientation with its principal translation direction. It exploits the horizontal cyclic symmetry of ERP panoramas, which allows yaw reorientation through circular shifts while preserving scene coverage (see Fig. 9 in Appendix A.4).

Given a target trajectory segment with horizontal camera coordinates $\{\mathbf{q}_i\}_{i=1}^{L}$, we apply Principal Component Analysis (PCA) to estimate its principal translation axis:

$$\mathbf{v} = \mathrm{PCA}_1\left(\{\mathbf{q}_i - \bar{\mathbf{q}}\}_{i=1}^{L}\right), \tag{2}$$

where $\bar{\mathbf{q}}$ denotes the mean camera position. We orient this axis according to the temporal progression of the trajectory and use the resulting direction $\mathbf{v}$ as the shared ERP orientation for all frames in the segment, while leaving camera positions unchanged. As shown in Fig. 2, we reorient the first-frame condition and use the aligned camera poses for both memory reprojection and camera conditioning. Generated observations are then back-projected using these same poses, keeping the memory in a consistent world coordinate system. The generated panoramas can be rotated back to their original orientations if needed.

As shown by the statistics in Fig. 1 (left), Path-Facing makes frame-wise motion directions relative to the ERP orientation more concentrated around the forward direction. During training, we primarily use Path-Facing while retaining orientation augmentation for robustness. It requires no additional learnable parameters or architectural changes and incurs negligible computational overhead, allowing direct application to existing panoramic video generators.

3.3 Timestep-aware Gating Modulation

Reprojected memory may contain unreliable local details caused by generation artifacts and depth inaccuracies, while still retaining useful coarse spatial structure. This motivates us to adapt its guidance to the coarse-to-fine denoising process: early high-noise stages primarily establish global structure, whereas later stages refine local appearance Qian et al. (2024); Ren et al. (2026).

Simply disabling memory guidance during later stages may also discard useful information and applies the same cutoff to all feature channels (see Appendix A.7). We therefore introduce *Timestep-aware gating modulation*, which learns channel-wise gates conditioned on the denoising timestep, allowing memory features to be attenuated or amplified without a hard cutoff.

As shown in Fig. 2, we encode the current timestep t with a sinusoidal embedding $\mathbf{e}_t = \mathrm{PE}(t)$. A lightweight two-layer MLP with a SiLU activation predicts channel-wise gates that modulate the render tokens before fusion with the diffusion backbone:

$$\mathbf{g}_t = \mathbf{1} + \tanh\left(f_{\mathrm{gate}}(\mathbf{e}_t)\right), \qquad \mathbf{h}_t = \mathbf{h}_t^{\mathrm{base}} + \mathbf{h}^{\mathrm{cam}} + \mathbf{g}_t \odot \mathbf{h}^{\mathrm{render}}. \tag{3}$$

where f_{gate} denotes the timestep-conditioned MLP and $\mathbf{g}_t \in (0, 2)^C$, with C denoting the render token dimension. We zero-initialize the MLP's output layer so that $\mathbf{g}_t = \mathbf{1}$ at initialization, preserving the original render conditioning.

As shown in Fig. 6, the learned mean gate is stronger early in denoising and generally decreases during later refinement, consistent with retaining structural guidance while reducing reliance on imperfect local details.

4 Experiments

4.1 Experimental Settings

We conduct main experiments on InteriorGS-OmniRoam Liu et al. (2026), following its split of 876 training and 98 evaluation scenes. Since ERP panoramas provide full 360° coverage, the dataset uses a fixed ERP orientation across its diverse camera trajectories. Evaluation covers both the first 81-frame segment and the full eight-segment (641-frame) rollout using SSIM, PSNR, LPIPS, FID, and cubemap-based $\mathrm{FID}_{\mathrm{cube}}$. Trajectory fidelity is measured by TransErr and RotErr computed from poses estimated by ViPE Huang et al. (2025). We further evaluate zero-shot generalization on 20 RealSee3D scenes Li et al. (2026b) using 481-frame closed-loop trajectories. SSIM-Loop, PSNR-Loop, and LPIPS-Loop compare the initial and final panoramas, while trajectory fidelity is evaluated over the full sequence.

PVWander is initialized from *Wan2.1-Fun-V1.1-1.3B-Control-Camera* and trained on 81-frame clips using LoRA and progressively increasing resolutions of 640×320, 960×480, and 1440×720. We use PVMetric depth for memory construction during both training and inference.

Detailed evaluation protocols and implementation settings are provided in Appendix A.5.

4.2 Comparison with State-of-the-Art Methods

We compare PVWander with the ERP-adapted Wan2.1-Fun Wang et al. (2025a) baseline and recent representative panoramic video wandering methods, including Matrix-3D Yang et al. (2025), OmniRoam Liu et al. (2026), and PanoWorld Li et al. (2026a). For methods originally designed for single-segment generation, we extend them to long-horizon wandering through autoregressive rollout, using the last frame of each segment to initialize the next. Since the full OmniRoam pipeline follows a fixed 8-segment preview-and-refine protocol, we also evaluate its preview model, denoted as *OmniRoam-P*, under the same autoregressive wandering setting with other methods.

Quantitative Comparison As reported in Tab. 1, across all three resolutions, PVWander achieves the strongest overall appearance fidelity, consistently obtaining higher SSIM and PSNR and lower LPIPS than the competing methods for both the first segment and the full rollout. The advantage remains substantial after 641 frames, while trajectory fidelity remains competitive across resolutions.

Fig. 3 further examines how generation quality evolves throughout the rollout. PVWander maintains the highest SSIM among the compared methods at the corresponding resolution across all eight segments. Although generation quality decreases as the rollout proceeds, PVWander remains relatively stable in the later stages. In comparison, Matrix-3D at 1440×720, as well as PanoWorld and OmniRoam-P at 960×480, exhibit more pronounced degradation.

OmniRoam also achieves strong long-horizon performance with its full preview-and-refinement pipeline, where sparse frames spanning the entire 8-segment trajectory guide segment-wise refine-

Table 1: Quantitative comparison on InteriorGS-OmniRoam for the first segment and the full 8-segment rollout. Best results at each resolution are shown in bold. Gray entries are reference results: the ground-truth (GT) row reports ViPE trajectory estimates from GT frames, and OmniRoam-P$_{s=8}$ denotes the single-pass stride-8 sparse preview. Matrix-3D$^{\text{MoGe}}_{\text{adapted}}$ uses GT depth for training and GT-aligned MoGe depth at inference, while Matrix-3D$^{\text{PVMetric}}_{\text{adapted}}$ uses PVMetric consistently during both training and inference.

Resolution	Methods	Backbone	First Segment (81 frames)							8 Segments (641 frames)						
			SSIM↑	PSNR↑	LPIPS↓	FID↓	FID$_{\text{cube}}$↓	TransErr↓	RotErr↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	FID$_{\text{cube}}$↓	TransErr↓	RotErr↓
1440 × 720	GT	–	–	–	–	–	–	0.0017	0.1043	–	–	–	–	–	0.0162	0.3502
	OmniRoam	Wan 1.3B	0.6930	17.35	0.3217	22.46	39.75	0.1169	0.7338	0.6063	14.59	0.4662	**19.88**	38.38	0.2869	1.7781
	Matrix-3D	Wan 5B	0.7404	19.36	0.2709	23.22	45.85	0.0143	0.5015	0.5748	12.76	0.5765	82.49	100.76	1.1411	9.4044
	PVWander	Wan 1.3B	**0.7935**	**21.62**	**0.1750**	**16.18**	**27.15**	**0.0099**	**0.2513**	**0.6810**	**16.32**	**0.3983**	23.65	**36.92**	**0.1890**	**1.1388**
960 × 480	PanoWorld	Wan 5B	0.6000	15.73	0.3952	28.28	43.03	0.1390	0.3768	0.4546	11.96	0.5913	26.17	50.38	0.7299	1.5944
	OmniRoam-P$_{s=1}$	Wan 1.3B	0.6823	18.31	0.2536	19.31	30.53	0.0369	0.4757	0.5383	13.48	0.4994	32.57	48.15	0.3794	1.7844
	OmniRoam-P$_{s=8}$	Wan1.3B	–	–	–	–	–	–	–	0.5823	15.02	0.4392	23.14	55.87	0.2546	1.5701
	PVWander	Wan 1.3B	**0.7777**	**21.92**	**0.1656**	**16.79**	**29.29**	**0.0124**	**0.2789**	**0.6467**	**16.44**	**0.4038**	**23.07**	**37.56**	**0.1770**	**0.9859**
640 × 320	Wan2.1-Fun	Wan 1.3B	0.7181	20.88	0.1717	18.78	34.98	0.0225	0.4612	0.5492	15.36	0.4220	24.82	45.21	**0.2267**	1.5698
	Matrix-3D$^{\text{MoGe}}_{\text{adapted}}$	Wan 1.3B	0.7087	20.48	0.1838	18.84	33.77	**0.0146**	**0.3212**	0.5515	14.85	0.4732	29.39	47.09	0.6174	2.2268
	Matrix-3D$^{\text{PVMetric}}_{\text{adapted}}$	Wan 1.3B	0.7301	21.29	0.1686	18.95	35.50	0.0195	0.3685	0.5526	15.33	0.4304	27.02	47.69	0.2911	1.6191
	PVWander	Wan 1.3B	**0.7353**	**21.44**	**0.1615**	**17.92**	**33.08**	0.0194	0.3532	**0.5867**	**16.25**	**0.4032**	**23.75**	**38.13**	0.3521	**1.3363**

ment. When the same preview model is rolled out autoregressively at stride 1 as *OmniRoam-P*, performance degrades more noticeably. In contrast, its single-pass stride-8 preview remains relatively stable, suggesting that preview guidance spanning the full trajectory contributes to the pipeline's long-horizon performance.

Matrix-3D performs strongly on the first segment but degrades considerably during recursive long-horizon generation. Unlike PVWander, it reconstructs geometric guidance independently from each segment's initial frame without accumulating generated observations into a persistent scene memory. To reduce differences in backbone and training data, we further adapt Matrix-3D to our ERP Wan2.1-Fun backbone at 640 × 320. We evaluate two depth settings: Matrix-3D$^{\text{MoGe}}_{\text{adapted}}$ follows the original GT-training/MoGe-inference protocol with GT-aligned depth at inference, while Matrix-3D$^{\text{PVMetric}}_{\text{adapted}}$ uses PVMetric consistently during both training and inference without GT alignment, matching the depth protocol of PVWander. The latter improves first-segment appearance fidelity, indicating that reducing the train–inference depth discrepancy is beneficial. Nevertheless, after the full 8-segment rollout, both variants remain clearly below PVWander in appearance fidelity. These results suggest that matching the training and inference depth sources alone does not close the long-horizon performance gap. Detailed depth and alignment settings are provided in Appendix A.10.

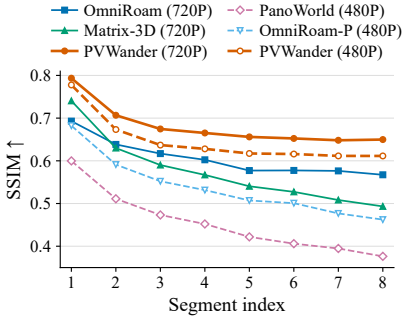


Figure 3: Segment-wise SSIM over the eight-segment rollout.

Qualitative Comparison. Fig. 4 compares panoramic frames at fixed intervals throughout the 641-frame rollout. Matrix-3D produces plausible results in the early stages but gradually develops severe structural distortions and visual artifacts under recursive generation, consistent with its quantitative degradation. PanoWorld maintains reasonable visual quality, but its generated views increasingly deviate from the corresponding GT viewpoints, indicating weaker trajectory adherence in long-horizon evaluation. Notably, the released PanoWorld model supports only single-segment generation and does not expose its full long-range memory mechanism; we therefore extend it autoregressively using the last frame of each segment, which likely contributes to its degradation over extended rollout. OmniRoam better preserves the overall scene layout, but exhibits discrepancies in the relative scale of scene elements. In contrast, PVWander maintains scene structure and object proportions closer to the GT throughout the trajectory, consistent with the spatial constraints provided by its continuously updated 3D memory.

Zero-Shot Evaluation. We further evaluate zero-shot generalization on RealSee3D under the protocol described above, with quantitative results reported in Tab. 2. OmniRoam-P is evaluated through autoregressive rollout. Since the full OmniRoam pipeline requires a fixed 8-segment input, we extend the 481-frame closed-loop trajectory to 641 frames by retracing the return path, generate

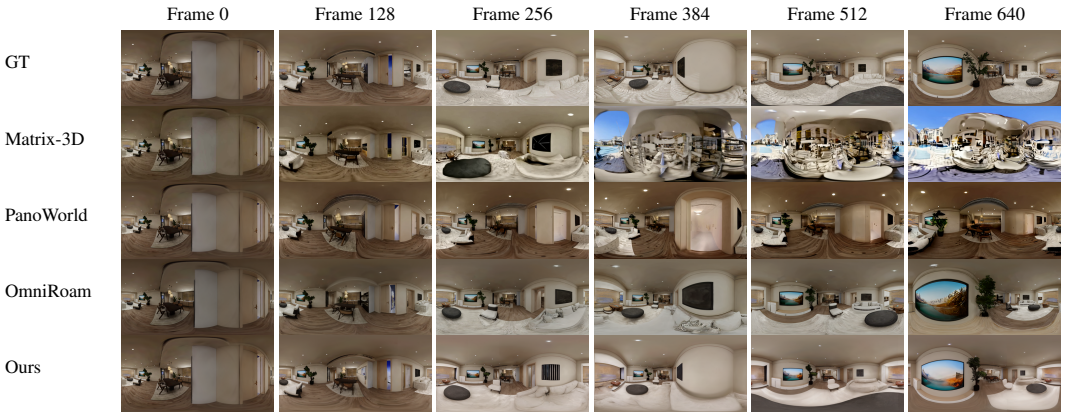


Figure 4: Qualitative comparison on InteriorGS-OmniRoam, showing frames sampled every 128 frames throughout the 641-frame rollout. Rows show GT, Matrix-3D, PanoWorld, OmniRoam, and PVWander (Ours).

Table 2: Zero-shot evaluation on RealSee3D with 481-frame closed-loop trajectories. Loop metrics compare the final generated panorama with the initial panorama. Best results at each resolution are shown in bold.

Method	Resolution	SSIM-Loop↑	PSNR-Loop↑	LPIPS-Loop↓	TransErr↓	RotErr↓
OmniRoam	1440×720	0.7052	18.63	**0.2881**	0.1692	0.8585
Matrix-3D	1440×720	0.6955	13.90	0.5324	0.3148	1.7719
PVWander	1440×720	**0.7883**	**19.42**	0.2983	**0.0343**	**0.5526**
PanoWorld	960×480	0.5501	14.96	0.5014	0.3948	0.6921
Matrix-3D	960×480	0.6427	13.48	0.6669	0.3742	0.7940
OmniRoam-P	960×480	0.5652	13.10	0.5367	0.4775	1.4250
PVWander	960×480	**0.7617**	**18.77**	**0.3295**	**0.0550**	**0.6895**

its stride-8 sparse preview over the extended trajectory, and refine the original 481 frames for evaluation. Across both resolutions, PVWander achieves the strongest overall zero-shot performance, demonstrating robust scene consistency upon revisitation and higher trajectory fidelity on unseen scenes.

Fig. 5 provides a qualitative comparison on one RealSee3D scene. We primarily visualize the 960×480 setting, which is supported by the largest set of compared methods, while additionally including the full OmniRoam result at its native 1440×720 resolution. Matrix-3D exhibits pronounced structural distortions, while OmniRoam-P suffers severe structural degradation under autoregressive rollout. With sparse preview guidance spanning the trajectory, the full OmniRoam pipeline better preserves the scene layout, although scene drift remains upon returning to the starting viewpoint. PanoWorld retains a plausible visual appearance, but the doorway and partition differ noticeably from those in the initial view. In comparison, PVWander better preserves the overall entrance and living-room layout upon revisitation.

4.3 Ablation Studies

Unless otherwise specified, experiments in this subsection are evaluated over the full eight-segment (641-frame) rollout. Component ablations use the first-stage training setting at 640×320 described in Appendix A.5.

Overall Ablation. We first evaluate the main design choices of PVWander, as shown in Tab. 3. Path-Facing improves long-horizon generation independently of 3D memory, demonstrating the benefit of canonicalizing diverse motion–orientation relationships. Timestep-aware gating further improves memory-guided generation, supporting its effectiveness in adaptively regulating imperfect reprojected guidance during denoising. Combining Path-Facing and Timestep-aware gating with online 3D memory provides the best overall balance across the evaluated metrics. We also observe that a fixed 3D memory constructed only from the initial panorama becomes increasingly limited over

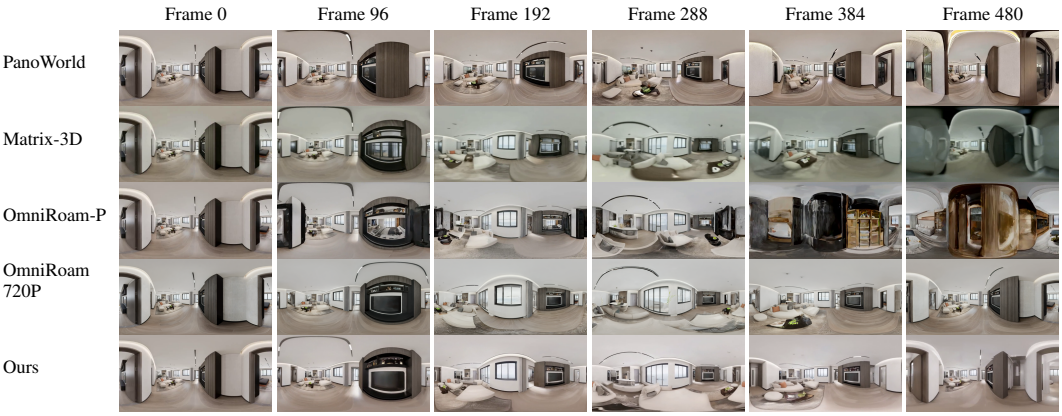


Figure 5: Zero-shot qualitative comparison on RealSee3D. We show every 96th frame along the 481-frame closed-loop trajectory. Frame 480 returns to the initial viewpoint, enabling direct assessment of scene consistency upon revisitation. OmniRoam is evaluated at 720P; all other methods are evaluated at 480P.

Table 3: Overall ablation of PVWander. The first row denotes the ERP-adapted Wan2.1-Fun camera-only baseline. Path-Facing, when enabled, is used during both training and inference.

Components				8 Segments (641 frames)				
Path-Facing	Fixed Memory	Online Memory	Timestep-aware gating	SSIM↑	PSNR↑	LPIPS↓	FID↓	FID_{cube} ↓
–	–	–	–	0.5492	15.36	0.4220	24.82	45.21
✓	–	–	–	0.5761	15.91	0.4094	24.72	42.98
–	✓	–	–	0.5280	14.76	0.4460	29.80	51.39
–	–	✓	–	0.5806	15.95	0.4132	26.69	42.67
–	–	✓	✓	0.5770	16.06	0.4042	23.79	40.06
✓	–	✓	–	0.5767	15.95	0.4107	**23.09**	40.50
✓	–	✓	✓	**0.5867**	**16.25**	**0.4032**	23.75	**38.13**

extended rollout. This may reflect reduced reprojection coverage as the camera explores beyond the initial observation, whereas online updates incorporate newly observed regions (see Appendix A.8).

Table 4: Param-only uses timestep-independent learnable modulation; Scalar-wise uses a shared timestep-dependent scalar, while Channel-wise predicts a gate per channel.

Memory	Gating	SSIM↑	PSNR↑	LPIPS↓	FID↓	FID_{cube} ↓
–	–	0.5492	15.36	0.4220	24.82	45.21
✓	None	**0.5806**	15.95	0.4132	26.69	42.67
✓	Param-only	0.5605	15.81	0.4126	26.18	47.53
✓	Scalar-wise	0.5750	15.99	0.4092	26.61	46.26
✓	Channel-wise	0.5770	**16.06**	**0.4042**	**23.79**	**40.06**

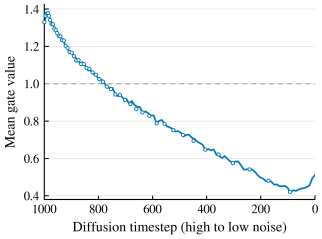


Figure 6: Learned gate values averaged across channels. Open circles mark the 50-step inference schedule with a shift of 5.

Timestep-aware gating Modulation. We examine whether the benefit of memory modulation arises from timestep awareness rather than simply introducing additional learnable parameters. As shown in Tab. 4, the proposed Timestep-aware gating achieves the strongest overall performance among the gating variants. The parameter-only variant provides no consistent improvement over ungated memory, indicating that the gain is not merely due to the additional modulation parameters. Using a timestep-dependent scalar provides only limited improvement, whereas the proposed channel-wise formulation allows the modulation to adapt more flexibly across denoising stages. We further compare against manually specified render-conditioning cutoffs in Appendix A.7, where the learned gating also outperforms the tested hard-cutoff configurations.

Fig. 6 visualizes the learned modulation across denoising timesteps. The mean gate is stronger during the early high-noise stage and generally decreases during later refinement. This learned trend is consistent with our motivation to emphasize memory-based structural guidance early while reducing reliance on imperfect rendered details at later stages.

Zero-Shot Transfer of Path-Facing Canonicalization. Since Path-Facing canonicalization introduces no additional parameters or architectural changes, it can be applied to existing panoramic generators without additional fine-tuning. Here we evaluate its zero-shot transfer on InteriorGS-OmniRoam using Wan2.1-Fun, OmniRoam, Matrix-3D, and PanoWorld.

As shown in Tab. 5, Path-Facing improves SSIM in all five configurations, with gains in PSNR and LPIPS in most cases. These results show that the benefits of Path-Facing are not limited to models trained with this canonicalization. Notably, both Wan2.1-Fun and OmniRoam have been fine-tuned on InteriorGS-OmniRoam with fixed ERP orientations. Their improvements support the benefit of trajectory-dependent alignment beyond fixed orientations alone.

Table 5: Zero-shot transfer of Path-Facing on InteriorGS-OmniRoam. Each entry reports performance before → after applying Path-Facing at inference. All evaluated checkpoints are trained without Path-Facing. OmniRoam (Preview) denotes the stride-8 sparse preview; all others use the full 641-frame rollout.

Method	SSIM↑	PSNR↑	LPIPS↓
Wan2.1-Fun	0.5492 → **0.5600**	15.36 → **15.64**	0.4220 → **0.4201**
OmniRoam	0.6063 → **0.6092**	**14.59** → 14.58	0.4662 → **0.4608**
OmniRoam (Preview)	0.5823 → **0.5854**	15.02 → **15.04**	0.4392 → **0.4330**
Matrix-3D	0.5748 → **0.5776**	12.76 → **12.95**	**0.5765** → 0.5770
PanoWorld	0.4546 → **0.4650**	11.96 → **12.21**	0.5913 → **0.5903**

Limitations. PVWander relies on estimated depth to construct and update its 3D memory. Depth estimation errors, particularly in geometrically ambiguous regions such as reflective surfaces, can lead to severely corrupted memory reprojections. Although train–inference consistency and Timestep-aware gating make the model more robust to imperfect projected guidance, they cannot fully compensate when depth errors substantially alter the underlying scene structure. As shown in Fig. 7, the model recovers plausible appearance from distorted memory projections, but structural deviations remain. Future improvements in geometric estimation could further strengthen structural fidelity.

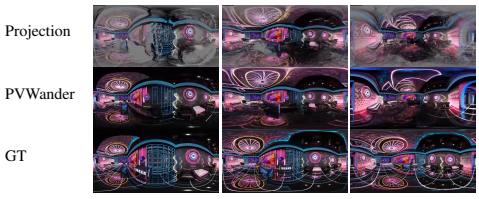


Figure 7: Failure case with severely corrupted 3D-memory geometry. PVWander preserves plausible appearance but shows structural deviations.

5 Conclusion

This paper introduced PVWander, a framework for long-horizon panoramic video wandering that continuously accumulates generated observations into an online 3D memory and reuses the accumulated scene information to guide subsequent generation. Path-Facing canonicalization simplifies diverse motion–orientation relationships without introducing additional parameters and with negligible computational overhead, while Timestep-aware gating modulation adaptively regulates imperfect memory guidance across denoising stages. Experiments on InteriorGS-OmniRoam and zero-shot evaluation on RealSee3D demonstrate improved long-horizon generation quality, scene consistency, and trajectory fidelity. Ablation studies validate the effectiveness of the proposed designs, while transfer experiments further show that Path-Facing canonicalization can benefit existing panoramic generators without additional training. While the current framework focuses on static-scene exploration, extending memory-guided panoramic generation to dynamic environments with object motion and interactions remains an important direction toward more general world models.

AI USE STATEMENT

In this work, we used generative AI tools to improve grammar, clarity, conciseness, and readability, as well as to refine phrasing and the presentation of the paper. We have not used generative AI tools for research ideas, methodological design, data analysis, and interpretation of results. We have reviewed all AI-assisted work. All AI-assisted text was reviewed and revised by the authors to ensure its accuracy and consistency with the underlying research. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work builds on the publicly available open-source Wan2.1-Fun-V1.1-1.3B-Control-Camera model and does not involve human subjects or private data. We do not release any new dataset containing personal or sensitive information. We acknowledge the general ethical considerations associated with generative models and emphasize their responsible use in accordance with applicable licenses and regulations.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed descriptions of the proposed method, implementation settings, training and inference procedures, datasets, and evaluation protocols in the main paper and appendix. Source code and trained models are intended to be made publicly available.

REFERENCES

Iro Armeni, Sasha Sax, Amir Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *ArXiv*, abs/1702.01105, 2017.

Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias NieBner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *2017 International Conference on 3D Vision (3DV)*, pp. 667–676, 2017.

Sili Chen, Hengkai Guo, Shengnan Zhu, Fei-Hu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22831–22840, 2025.

Ting-Hsuan Chen, Ying-Huan Chen, Tao Tu, Jie-Ying Lee, Cho-Ying Wu, Fangzhou Lin, Hengyuan Zhang, David Paz, Xinyu Huang, Yuliang Guo, Yu-Lun Liu, Yue Wang, and Liu Ren. Pantheon360: Taming digital twin generation via 3d-aware 360deg video diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11138–11149, June 2026.

Xinhua Cheng, Haiyang Zhou, Wangbo Yu, Tanghui Jia, Bin Lin, Yunyang Ge, Weiqi Li, and Li Yuan. 360Explorer: Exploring 4D controllable world in panoramic videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(5):3300–3308, 2026. ISSN 2374-3468.

Hao-Yi Duan, Hong-Xing Yu, Si-Rui Chen, Fei-Fei Li, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 01–12, 2025.

Xian Ge, Yulin Pan, Yuhang Zhang, Xiang Li, Weijun Zhang, Dizhe Zhang, Zhaoliang Wan, Xin Lin, Xiangkai Zhang, Juntao Liang, Jason Li, Wenjie Jiang, Bo Du, Ming-Hsuan Yang, and Lu Qi. Airsim360: A panoramic simulation platform within drone view. *ArXiv*, abs/2512.02009, 2025.

Yi-Jing Guo, Meng Chao, Luozhou Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. Panovggt: Feed-forward 3d reconstruction from panoramic imagery. *ArXiv*, abs/2603.17571, 2026.

Hao He, Ying-Hao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *ArXiv*, abs/2404.02101, 2024.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jia-Wei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. *ArXiv*, abs/2508.10934, 2025.

Chenhao Ji, Chaohui Yu, Junyao Gao, Fan Wang, and Cairong Zhao. Campvg: Camera-controlled panoramic video generation with epipolar-aware diffusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Haoyuan Li, Dizhe Zhang, Yuemei Zhou, Xiangkai Zhang, Hao li Feng, Xiaofan Lin, Wenjie Jiang, Bo Du, Ming-Hsuan Yang, and Lu Qi. Panoworld: Real-world panoramic generation. *ArXiv*, abs/2607.09661, 2026a.

Runjia Li, Philip H. S. Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699, 2025.

Xi Li, Linyuan Li, Yan Wu, Tong Rao, Kai Zhang, Xin-Chen Hui, and Cihui Pan. Argus: Metric panoramic 3d reconstruction for indoor scenes. *ArXiv*, abs/2606.30047, 2026b.

Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Q. Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation. *ArXiv*, abs/2512.16913, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2023.

Yuheng Liu, Xin Lin, Xinke Li, Baihan Yang, Chen Wang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Hao Tan, Kai Zhang, Xiaohui Xie, Zifan Shi, and Yiwei Hu. Omniroam: World wandering via long-horizon panoramic video generation. *SIGGRAPH*, 2026.

Bingchen Miao, Rong Wei, Zhiqi Ge, Xiao-Quan Sun, Shiqi Gao, Jingzhe Zhu, Renhan Wang, Siliang Tang, Jun Xiao, Rui Tang, and Juncheng Li. Towards physically executable 3d gaussian for embodied navigation. *ArXiv*, abs/2510.21307, 2025.

Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8911–8920, 2024. doi: 10.1109/CVPR52733.2024.00851.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, and Liang-Chieh Chen. Frequency-aware flow matching for high-quality image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9074–9083, June 2026.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Muller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132, 2025.

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

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Chuanxin Song and Peixi Peng. PVDepth: Panoramic video depth estimation via geometry-aware spatiotemporal adaptation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=HwYgnZjxUB.

Yongxin Su, Linjie Hou, Feng Wang, Jialin Tang, Zhijun Li, Qian Wang, and Maoqing Yao. Genie sim panoworld: An infinite indoor 3d world generation pipeline via panoramic scene modeling and simulation. *ArXiv*, abs/2607.26646, 2026.

Mingchao Sun, Luyang Tang, Yu Liu, Xu Yan, Zhan Li, Yunwei Zhang, Fei Yu, Zengye Ge, Yumin Liu, Jiacheng Zhang, et al. Abot-3dworld 0: A universal world model to explore any 3d space. *arXiv preprint arXiv:2607.11673*, 2026.

Xiancheng Sun, Mai Xu, Sheng-Xi Li, Senmao Ma, Xin Deng, Lai Jiang, and Gang Shen. Spherical manifold guided diffusion model for panoramic image generation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5824–5834, 2025.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningying Zhang, Pandeng Li, Ping Wu, Rui-Hang Chu, Rui Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tian-Yi Gui, Ting-Yu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wen-Chao Zhou, Wente Wang, Wen Shen, Wenyuan Yu, Xianzhong Shi, Xiaomin Huang, Xin Xu, Yan Kou, Yan-Mei Lv, Yifei Li, Yi-Jing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025a.

Dehui Wang, Rong Wei, Yue Shi, Congsheng Xu, Shoufa Chen, Dingxiang Luo, Tianshuo Yang, Xiaokang Yang, Wei Sui, Yusen Qin, et al. Rein3d: Reinforced 3d indoor scene generation with panoramic video diffusion models. *arXiv preprint arXiv:2604.10578*, 2026.

Jiahao Wang, Luoxin Ye, Taiming Lu, Junfei Xiao, Jiahan Zhang, Yuxiang Guo, Xi-Jun Liu, Rama Chellappa, Cheng Peng, Alan L. Yuille, and Jie-Neng Chen. Evoworld: Evolving panoramic world generation with explicit 3d memory. *ArXiv*, abs/2510.01183, 2025b.

Qian Wang, Weiqi Li, Chong Mou, Xinhua Cheng, and Jian Zhang. 360dvd: Controllable panorama video generation with 360-degree video diffusion model. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6913–6923, 2024a.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271, 2025c.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Shan Ying. Motionctrl: A unified and flexible motion controller for video generation. *ACM SIGGRAPH 2024 Conference Papers*, 2024b.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems*, 38:49371–49393, 2026.

Yifei Xia, Shuchen Weng, Siqi Yang, Jingqi Liu, Chengxuan Zhu, Minggui Teng, Zijian Jia, Han Jiang, and Boxin Shi. Panowan: Lifting diffusion video generation models to 360° with latitude/longitude-aware mechanisms, 2025. URL https://arxiv.org/abs/2505.22016.

Haozhe Xie, Beichen Wen, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 3d scene generation: A survey. *arXiv preprint arXiv:2505.05474*, 2025.

Zhongqi Yang, Wenhang Ge, Yuqi Li, Jiaqi Chen, Haoyuan Li, Mengyin An, Fei Kang, Hua Xue, Baixin Xu, Yuyang Yin, et al. Matrix-3d: Omnidirectional explorable 3d world generation. *arXiv preprint arXiv:2508.08086*, 2025.

Yuyang Yin, Hao-Xiang Guo, Fangfu Liu, Mengyu Wang, Hanwen Liang, Eric Li, Yikai Wang, Xiaojie Jin, Yao Zhao, and Yunchao Wei. Panoworld-x: Generating explorable panoramic worlds via sphere-aware video diffusion, 2026. URL https://openreview.net/forum?id=iZyBEbq6jR.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Peng-Fei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025a.

Wangbo Yu, Jinbo Xing, Li Yuan, Wen-Bo Hu, Xiaoyu Li, Zhi-Peng Huang, Xiangjun Gao, Tien-Tsin Wong, Shan Ying, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2025b.

Cheng Zhang, Hanwen Liang, Donny Y Chen, Qianyi Wu, Konstantinos N Plataniotis, Camilo Cruz Gambardella, and Jianfei Cai. Panflow: Decoupled motion control for panoramic video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Muyang Zhang, Yuzhi Chen, Rongtao Xu, Changwei Wang, Jinming Yang, Weiliang Meng, Jianwei Guo, Huihuang Zhao, and Xiaopeng Zhang. Panodit: Panoramic videos generation with diffusion transformer. In *AAAI Conference on Artificial Intelligence*, 2025.

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

A Appendix

A.1 Extended Related Work

Panoramic Video Generation. Panoramic video generation has been explored from both content synthesis and controllable exploration perspectives. For content synthesis, existing methods extend pretrained text-to-video models with panorama-specific designs to generate immersive videos with a full 360° field of view Wang et al. (2024a); Xia et al. (2025); Zhang et al. (2025). For controllable exploration, camera-controlled approaches synthesize panoramic videos along specified camera trajectories Ji et al. (2025); Yin et al. (2026). Several methods further simplify trajectory conditioning by decoupling camera rotation or using translation-only motion Zhang et al. (2026); Cheng et al. (2026); Liu et al. (2026). However, regardless of whether these simplifications are applied, free-form translations can exhibit highly diverse directions relative to the ERP orientation. In contrast, our *Path-Facing canonicalization* exploits the horizontal cyclic symmetry of panoramas to align each segment's ERP orientation with its principal translation direction, yielding more consistent, forward-like motion–orientation relationships. Explicit 3D representations have also been introduced to provide geometric guidance using meshes, point clouds, or 3D Gaussians Yang et al. (2025); Chen et al. (2026); Wang et al. (2026); Sun et al. (2026). However, most do not maintain a persistent 3D memory that evolves with newly generated observations, limiting their ability to continuously accumulate scene information over long horizons. Similar to our formulation, EvoWorld Wang et al. (2025b) maintains an evolving 3D point-cloud memory that is iteratively updated and reprojected to guide subsequent panoramic generation. However, repeated memory updates can also accumulate generation artifacts and depth estimation errors. PVWander further emphasizes robustness to these accumulated errors through *Timestep-aware gating modulation*.

Camera-Controlled Long-Horizon Video Generation. Camera-controlled video diffusion models inject camera trajectories through pose parameters or dense ray embeddings Wang et al. (2024b); He et al. (2024). Recent methods further incorporate rendered 3D guidance to strengthen geometric control Yu et al. (2025b); Ren et al. (2025). However, most models operate within a fixed temporal context, which can compromise long-horizon consistency when earlier observations fall outside the model's effective context window. Memory-aware methods address this limitation by retrieving relevant historical views Yu et al. (2025a); Li et al. (2025; 2026a) or accumulating observations into persistent 3D scene representations that can be reprojected to future viewpoints Wu et al. (2026); Wang et al. (2025b). Lyra 2.0 Shen et al. (2026) further combines geometry-aware retrieval with per-frame 3D representations to establish correspondences with relevant historical observations. Geometry-grounded memories improve long-horizon consistency, but continuously updating them from generated observations can introduce a closed-loop error-propagation problem, as visual artifacts and depth inaccuracies may be repeatedly written into the memory and affect subsequent

predictions. We therefore introduce *Timestep-aware gating modulation*, which exploits the coarse-to-fine nature of diffusion denoising Ren et al. (2026) to adaptively regulate memory guidance with timestep-dependent gates, preserving its benefits for structural consistency while mitigating visual artifacts.

A.2 Preliminaries

Video Diffusion Model. We adopt Wan2.1 Wang et al. (2025a) as our video generation backbone. Given a video $\mathbf{x}$, a spatiotemporal VAE encodes it into a latent representation $\mathbf{z}_{\text{clean}} = \mathcal{E}(\mathbf{x})$. Following flow matching Lipman et al. (2023), we construct noisy latents from Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the clean latent as $\mathbf{z}_\sigma = (1-\sigma)\mathbf{z}_{\text{clean}} + \sigma\boldsymbol{\epsilon}$, where $\sigma \in [0, 1]$. The model uses the scheduler timestep $t = 1000\sigma$ and predicts the target velocity $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{z}_{\text{clean}}$ with

$$\mathcal{L} = \mathbb{E}\left[\|\mathbf{u}_\theta(\mathbf{z}_\sigma, \mathbf{c}, t) - \mathbf{v}\|_2^2\right], \tag{4}$$

where $\mathbf{c}$ denotes the generation conditions.

To enable camera-controllable video generation, our model is initialized from *Wan2.1-Fun-V1.1-1.3B-Control-Camera*. Given a sequence of camera poses, the corresponding pixel-wise Plücker ray maps are used as camera conditions. These ray maps are encoded by a lightweight camera adapter and additively fused with the patch-embedded latent features, providing explicit conditioning for camera-motion control.

3D Memory-Guided Video Generation. 3D-guided video generation provides explicit spatial context by lifting observed images into 3D representation, and reprojecting them to future viewpoints as additional conditions for video generation Yu et al. (2025b); Ren et al. (2025). For long-horizon generation, some recent methods Wu et al. (2026) further maintain an online 3D memory to accumulate newly generated observations, which can then be reprojected and reused as conditioning for subsequent videos.

A.3 ERP-aware Camera Control

We build our camera-controllable panoramic video generation baseline on *Wan2.1-Fun-V1.1-1.3B-Control-Camera*. Although this model provides an effective camera-conditioning interface, its original Pl"ucker representation is constructed from perspective camera rays, which cannot be directly applied to ERP panoramas. We therefore adapt its camera representation to ERP videos by constructing pixel-wise spherical rays, similar to CamPVG Ji et al. (2025).

For ERP panoramas, pixel-wise ray directions are determined by spherical coordinates in the local camera frame. Given a pixel (u, v) in an ERP frame of resolution $H \times W$, we define its longitude and latitude as $\lambda = 2\pi(u/W - 1/2)$ and $\varphi = \pi(1/2 - v/H)$, respectively. The corresponding unit spherical ray direction is

$$\mathbf{d}(u, v) = [\cos\varphi \sin\lambda,\ \sin\varphi,\ \cos\varphi\cos\lambda]^\top. \tag{5}$$

Let $\mathbf{c}_t$ denote the camera position at frame t. We define its relative position with respect to the first frame as $\mathbf{o}_t = \mathbf{c}_t - \mathbf{c}_0$, and construct the panoramic Pl"ucker embedding as

$$\mathbf{p}_t(u, v) = [\mathbf{o}_t \times \mathbf{d}(u, v),\ \mathbf{d}(u, v)]. \tag{6}$$

This formulation retains the pixel-wise Pl"ucker conditioning interface of the pretrained camera adapter while incorporating the spherical projection geometry of ERP panoramas.

A.4 Panoramic Representation and ERP Reorientation

Cubemap–ERP representation. A panoramic observation captures the complete 360° surrounding view of a camera. Such spherical content can be represented using different projection formats. Cubemap and equirectangular projection (ERP) are two commonly used parameterizations of the viewing sphere.

Cubemap represents the spherical view using multiple perspective images, where each view corresponds to a limited field of view. In contrast, ERP unwraps the entire sphere into a single rectangular image by mapping longitude and latitude to the horizontal and vertical image axes, respectively.

Therefore, the full panoramic content, including front, back, left, right, up, and down directions, is stored in one continuous image.

Due to this property, ERP provides a convenient representation for panoramic video generation models, as temporal modeling and image-based generative models can directly operate on a single image sequence. Fig. 8 illustrates the correspondence between Cubemap and ERP representations.

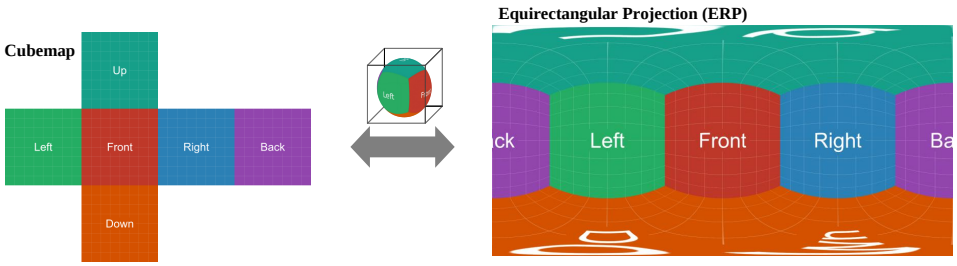


Figure 8: Cubemap and ERP represent the same viewing sphere with different sampling grids. Colors identify the six Cubemap faces and their corresponding regions in ERP.

ERP reorientation. Because ERP covers the complete viewing sphere, changing its horizontal reference orientation does not alter the underlying scene content. A yaw rotation around the vertical axis only shifts the longitude coordinate while preserving latitude, which corresponds to a horizontal circular shift in ERP space.

For an ERP image I, a yaw offset α can be applied as

$$I_\alpha(u, v) = I\left(\left(u - \frac{W\alpha}{2\pi}\right) \bmod W, v\right),$$

where W denotes the ERP width and α is the yaw offset in radians.

Therefore, yaw reorientation preserves spherical coverage and only changes the horizontal reference orientation of the panorama. This cyclic property enables **Path-Facing** to canonicalize each clip's ERP reference orientation according to its principal translation direction, without modifying the prescribed camera trajectory.

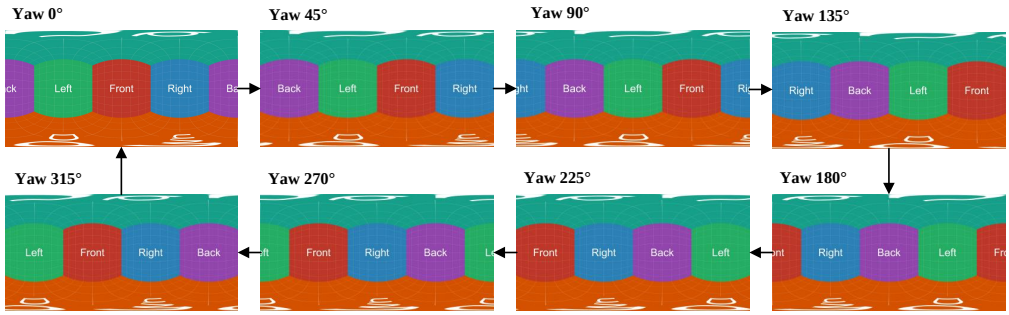


Figure 9: ERP panoramas under different yaw reorientations. Horizontal circular shifts only change the horizontal reference orientation while preserving the complete spherical content.

A.5 Experimental Settings and Implementation Details

Datasets and Evaluation Protocols. We conduct our main experiments on the *InteriorGS-OmniRoam* panoramic video dataset Liu et al. (2026), which is rendered from around 1,000 3D Gaussian Splatting scenes collected from InteriorGS Miao et al. (2025). We follow the train–test split of OmniRoam, using 876 scenes for training and 98 scenes for evaluation. Each scene contains a sequence of 800 panoramic frames rendered along a wandering camera trajectory. The dataset uses

a fixed ERP orientation across its diverse camera trajectories, taking advantage of the full 360° coverage of panoramic observations. For evaluation, following OmniRoam's protocol, we recursively generate 8 consecutive segments, resulting in a 641-frame sequence. We report results for both the first segment and the full rollout. For generation with Path-Facing canonicalization, we restore the generated panoramas to the GT orientations before evaluation to ensure a fair comparison with other methods. Appearance fidelity is evaluated using PSNR, SSIM, and LPIPS, while perceptual quality is measured using FID. Since directly resizing panoramic frames to square images distorts ERP geometry, we additionally report $\mathrm{FID}_{\mathrm{cube}}$ following SMGD Sun et al. (2025) and PanoWorld Li et al. (2026a), by projecting each panorama onto six cubemap faces, computing FID for each face, and averaging the results. For camera controllability, following Li et al. (2026a); Duan et al. (2025), we further evaluate trajectory fidelity by using ViPE Huang et al. (2025) to reconstruct frame-wise camera poses from the generated videos and compare the recovered trajectories with the prescribed camera trajectories. We report rotation error (*RotErr*, degrees) and scale-aligned translation error (*TransErr*). Appendix A.11 provides the detailed pose alignment and metric definitions.

To further evaluate zero-shot generalization, we conduct experiments on RealSee3D Li et al. (2026b). Since RealSee3D consists of static RGB-D captures rather than video sequences, following Su et al. (2026), we randomly sample initial panoramas from 20 scenes and use the GT depth of each initial panorama to construct a 481-frame camera trajectory that returns to the starting viewpoint at the final frame. This enables loop-closure evaluation without requiring GT roaming videos. We therefore report *SSIM-Loop*, *PSNR-Loop*, and *LPIPS-Loop* between the initial and final frames to evaluate loop-closure consistency, while *TransErr* and *RotErr* are computed over the full recovered camera trajectory to evaluate trajectory fidelity. All models are evaluated on RealSee3D without any fine-tuning.

Implementation Details. We initialize PVWander from *Wan2.1-Fun-V1.1-1.3B-Control-Camera* and train it on four NVIDIA A100 GPUs using 81-frame clips. We use rank-64 LoRA for the attention layers and jointly train the conditioning modules and input projections. We use AdamW with BF16 mixed precision and an effective batch size of 32. Training is conducted in three stages with progressively increasing ERP resolution. We first train at 640×320 for 5,000 steps with a learning rate of 10^{-4}, using camera conditioning only for the first 1,000 steps and introducing memory-render guidance for the remaining 4,000 steps. We then train at 960×480 for 3,500 steps with a learning rate of 10^{-5}, where the temporal stride is fixed to 1 for the first 1,500 steps and uniformly sampled from $\{1, 2, 3\}$ for each clip during the remaining 2,000 steps to improve robustness to varying temporal dynamics. Finally, we train at 1440×720 for 500 steps with a learning rate of 5×10^{-6} while retaining random temporal-stride sampling. We use PVMetric to estimate depth for 3D memory construction during both training and inference, maintaining train–inference consistency and allowing the generator to adapt to its prediction errors during training. Appendix A.6 further demonstrates the importance of this consistency for generation quality. For Path-Facing, we choose the sign of the first principal component so that projected camera positions have positive correlation with frame indices. When this correlation is numerically negligible, we resolve the sign using the displacement from the first to the last frame. During training, we apply Path-Facing to 90% of clips. For the remaining 10%, we use the horizontal direction from the first camera position to a randomly selected target frame's position as the common ERP reference orientation for all frames to improve robustness to orientation variation. For memory reprojection, we randomly sample the number of source frames from 1 to 4 during training. At inference, we use four source frames for each segment except the first.

A.6 Training–Inference Depth Consistency

We examine how the depth source used for geometric conditioning affects generation performance. For the Pose + 3D Memory setting, using GT depth during both training and inference provides an oracle upper bound. However, when a model trained with GT depth is evaluated using PVMetric predictions, performance drops substantially even with GT scale alignment. Using PVMetric during both training and inference improves SSIM from 0.7170 to 0.7357, PSNR from 20.83 to 21.41, and LPIPS from 0.1761 to 0.1654. This suggests that exposing the generator to prediction errors during training is beneficial when the same estimated depth is required at inference. Based on these results, PVWander uses PVMetric for 3D memory construction during both training and inference.

Table 6: Depth-source ablation for geometric conditioning. All results are evaluated on the first 81-frame segment. "w/ align" denotes aligning the predicted depth to the GT depth scale. The GT/GT setting serves as an oracle reference.

Method	Train Depth	Inference Depth	SSIM↑	PSNR↑	LPIPS↓
Pose + 3D Memory	GT	GT (Oracle)	0.7716	22.68	0.1424
	GT	PVMetric (w/ align)	0.7170	20.83	0.1761
	PVMetric	PVMetric	**0.7357**	**21.41**	**0.1654**
Matrix-3D	GT	MoGe (w/ align)	0.7087	20.48	0.1838
	PVMetric	PVMetric	**0.7301**	**21.29**	**0.1686**

We also observe a similar trend for the adapted Matrix-3D setting: replacing its GT-training/MoGe-inference configuration with PVMetric for both stages improves all three metrics. We do not additionally train Matrix-3D with MoGe-predicted depth because applying MoGe to the panoramic training set is prohibitively slow.

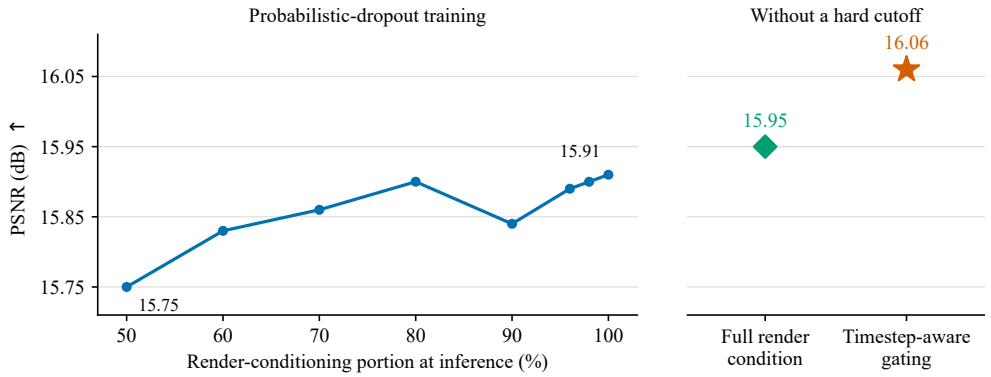


Figure 10: Hard render-conditioning cutoffs versus Timestep-aware gating on eight-segment rollouts. The curve reports PSNR for a model trained with probabilistic render-condition dropout, evaluated with different render-conditioning portions during denoising. All eight tested settings (50, 60, 70, 80, 90, 96, 98, and 100%) are shown. The diamond and star denote full render conditioning and Timestep-aware gating, respectively; they share the PSNR axis but are displayed separately because neither uses a hard cutoff. Higher PSNR is better.

A.7 Hard Render-Conditioning Cutoffs vs. Timestep-aware Gating

We compare manually disabling the render condition during denoising with full render conditioning and Timestep-aware gating on eight-segment rollouts. For the hard-cutoff baseline, training uses timestep-dependent probabilistic dropout: the render condition is disabled for timesteps below 50% of the timestep range and retained with probability 0.5 for timesteps above 50%. At inference, we vary the portion of the denoising process that uses the render condition from 50% to 100%, retaining it in the initial high-noise portion and disabling it thereafter. Full render conditioning uses the condition throughout both training and inference, whereas Timestep-aware gating learns channel-wise modulation without a hard cutoff.

Fig. 10 shows that none of the tested hard cutoffs improves over using the render condition throughout inference, suggesting that simply disabling memory guidance at later denoising stages is suboptimal. The 100% cutoff setting also remains slightly below full render conditioning, suggesting that introducing hard condition dropout during training provides no additional benefit in this setting. Timestep-aware gating further improves performance by retaining memory guidance throughout denoising while adaptively modulating its strength across timesteps, providing more flexible control.

A.8 FIXED VS. ONLINE 3D MEMORY

To examine the practical difference between fixed and evolving memory configurations, we compare the camera-only baseline with fixed and online 3D memory under both short- and long-horizon evaluation. The fixed-memory variant constructs its 3D memory only from the initial observation and keeps it unchanged throughout the rollout. During training, its rendered condition is likewise always obtained from the initial frame. In contrast, the online-memory variant is trained with 1–4 randomly sampled source observations for memory reprojection and progressively incorporates newly generated observations during inference. Except for the first segment, where only the initial observation is available, we use four historical source observations for memory reprojection during rollout.

Table 7: Comparison of fixed and online 3D memory. We report generation quality on both the first 81-frame segment and the full eight-segment (641-frame) rollout.

Method	First Segment (81 frames)					8 Segments (641 frames)				
	SSIM↑	PSNR↑	LPIPS↓	FID↓	FID_{cube} ↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	FID_{cube} ↓
Wan2.1-Fun	0.7181	20.88	0.1717	18.78	34.98	0.5492	15.36	0.4220	24.82	45.21
+ Fixed Memory	0.7263	21.19	0.1691	18.87	35.51	0.5280	14.76	0.4460	29.80	51.39
+ Online Memory	**0.7357**	**21.41**	**0.1654**	**18.60**	**33.74**	**0.5806**	**15.95**	**0.4132**	26.69	**42.67**

As shown in Tab. 7, fixed 3D memory provides modest improvements over the camera-only baseline in the first segment, indicating that reprojected geometric guidance is useful when the target views remain well covered by the initial observation. The online-memory variant performs slightly better at this stage. Since no memory update has yet occurred in the first segment, this difference is mainly associated with its multi-source training strategy, which exposes the model to more diverse reprojection conditions than the single-source training used by the fixed-memory variant.

The difference becomes more pronounced over extended rollout. After eight segments, the fixed-memory variant falls below the camera-only baseline across all reported metrics. As the camera explores regions beyond the initial observation, fixed-memory reprojections can provide less coverage, which may contribute to the observed degradation.

In contrast, the online-memory configuration continuously incorporates newly generated observations as exploration proceeds, allowing the memory to expand with the explored scene and provide guidance from more relevant historical views. This leads to substantially better long-horizon performance than the fixed-memory configuration and supports the use of an evolving 3D memory for extended panoramic wandering.

A.9 PVMETRIC

To estimate depth for online 3D memory construction, we train PVMetric, a feed-forward panoramic video metric-depth estimator. It predicts depth at metric scale and is used during both training and inference of PVWander.

A.9.1 ARCHITECTURE DESIGN

Overview. PVMetric extends the pretrained Depth Any Panoramas (DAP) Lin et al. (2025) image model into a feed-forward panoramic video metric-depth estimator. It retains the DINOv3 encoder Siméoni et al. (2025) and DPT multi-scale decoder of DAP, and introduces Cube-rectified Temporal Modeling (CRTM) modules at selected feature scales to capture cross-frame dependencies. This design builds on DAP's pretrained panoramic metric-depth representation while adapting its image-based decoder to video inputs. The overall architecture is summarized in Fig. 11.

Frame-wise Spatial Encoding. Given an ERP video $\mathbf{I} \in \mathbb{R}^{B\times T\times 3\times H\times W}$, we first fold the frame dimension into the batch dimension and process the resulting $(BT) \times 3 \times H \times W$ images with a shared DINOv3 encoder. Following DAP, four intermediate encoder features are passed to the DPT decoder for multi-scale spatial refinement. At this stage, each frame is encoded independently; cross-frame interactions are introduced by the temporal modules described below.

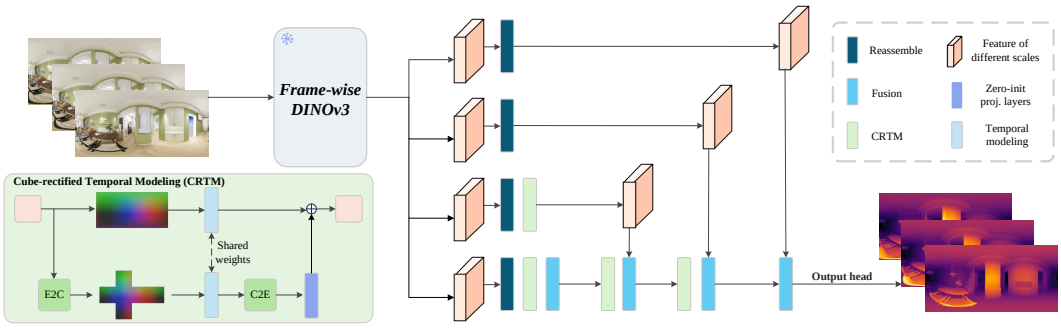


Figure 11: Architecture of PVMetric. The frame-wise DINOv3 backbone extracts multi-scale spatial features, while Cube-rectified Temporal Modeling (CRTM) modules perform temporal modeling at selected feature scales of the DPT decoder.

Multi-scale Temporal Modeling. Following Video Depth Anything Chen et al. (2025), we insert temporal layers at several lower-resolution positions of the DPT head to model cross-frame interactions while limiting the additional computational cost. Specifically, temporal modeling is introduced at four intermediate feature scales, whereas the higher-resolution features remain on the original frame-wise spatial pathway. For a feature at scale s, we gather the features at each spatial location across all frames into a temporal sequence:

$$\mathbf{F}^{(s)} \in \mathbb{R}^{B\times C_s\times T\times H_s\times W_s} \xrightarrow{\text{reshape}} \mathbf{F}^{(s)}_{\text{seq}} \in \mathbb{R}^{(BH_sW_s)\times T\times C_s}. \tag{7}$$

Self-attention is performed only along the T dimension, whereas spatial interaction is handled by the DINOv3 encoder and DPT refinement blocks. Each temporal Transformer is incorporated as a residual update initialized to preserve the pretrained spatial pathway.

Cube-rectified Temporal Modeling. The latitude-dependent stretching of ERP causes smooth motion to manifest as increasingly nonlinear temporal dynamics toward the polar regions. We therefore incorporate Cube-rectified Temporal Modeling (CRTM) from PVDepth Song & Peng (2026) into the multi-scale temporal modules of PVMetric. At each temporal scale, CRTM processes the feature through parallel ERP and Cubemap branches:

$$\begin{aligned}
\mathbf{Y}^{(s)}_E &= \mathcal{T}^{(s)}\left(\mathbf{F}^{(s)}\right),\\
\mathbf{Y}^{(s)}_C &= \mathcal{P}_{C\to E}\left(\mathcal{T}^{(s)}\left(\mathcal{P}_{E\to C}\left(\mathbf{F}^{(s)}\right)\right)\right),\\
\mathbf{F}^{(s)}_{\text{out}} &= \mathbf{Y}^{(s)}_E + \mathcal{G}^{(s)}_C\left(\mathbf{Y}^{(s)}_C\right).
\end{aligned} \tag{8}$$

Here, $\mathcal{P}_{E\to C}$ and $\mathcal{P}_{C\to E}$ denote ERP-to-Cubemap and Cubemap-to-ERP projection, respectively. The six Cubemap faces are folded into the batch dimension and modeled independently along the temporal dimension. Within each feature scale, the ERP and Cubemap branches share the same temporal Transformer $\mathcal{T}^{(s)}$, while different scales use independent parameters. The fusion projection $\mathcal{G}^{(s)}_C$ is zero-initialized, allowing the Cubemap branch to be progressively learned as a residual correction to the ERP temporal features.

Metric Depth Prediction. The DPT decoder progressively fuses the temporally enhanced multi-scale features and predicts depth results for each input frame. During training, GT metric depth is clipped at $D_{\max} = 100\,\text{m}$ and divided by $D_{\max}$ to obtain values in $[0, 1]$. To account for ERP's nonuniform spherical sampling, all loss terms are computed over valid pixels with cosine-latitude weighting. The training objective combines an absolute depth loss, a log-depth loss, and a temporal depth-difference loss Chen et al. (2025):

$$\mathcal{L} = \mathcal{L}_{\text{L1}} + 0.2\mathcal{L}_{\log} + \mathcal{L}_{\text{temp}}. \tag{9}$$

Here, $\mathcal{L}_{\text{L1}}$ and $\mathcal{L}_{\log}$ supervise the scaled metric depth in the linear and logarithmic domains, respectively, while $\mathcal{L}_{\text{temp}}$ matches predicted and GT depth changes between adjacent frames. The

temporal term is set to zero for single-frame samples. During inference, the prediction is multiplied by $D_{\max}$ to recover metric depth. For long-video inference, we follow the overlapping-window inference strategy of Video Depth Anything Chen et al. (2025), but omit the inter-window scale–shift alignment used for affine-invariant depth. The resulting metric depth sequence can be directly integrated into the online 3D memory for geometric reconstruction.

A.9.2 Implementation Details

Training Setup. We initialize PVMetric from the pretrained DAP model and freeze the DINOv3 encoder, while optimizing the DPT decoder and the four temporal modules. We use learning rates of 1×10^{-5} and 5×10^{-5} for the spatial and temporal layers, respectively. The model is optimized with AdamW using a weight decay of 10^{-2} and gradient clipping at 1.0 under BF16 mixed-precision training. We use effective batch sizes of 8 video clips and 160 images for the video and image modes, respectively, and train PVMetric for 150K optimization steps on four NVIDIA A100 GPUs, requiring approximately 636 GPU-hours. Both training and inference operate on ERP inputs resized to 1024×512.

Mixed Image–Video Training. Inspired by VDA Chen et al. (2025), we jointly train PVMetric with 32-frame video clips and single-frame images to learn temporal depth evolution while retaining the single-frame prior of DAP. At each training step, we select the video mode with probability 0.8 or the image mode with probability 0.2. The video mode uses PanoCARLA Song & Peng (2026), PanoCity Guo et al. (2026), and InteriorGS-OmniRoam Liu et al. (2026), which together contain approximately 975K training frames. The image mode samples single frames from these video datasets and additionally includes Structured3D Zheng et al. (2020) and Omni360-Scene Ge et al. (2025); the latter two provide approximately 227K additional training images. Both modes share the same network and metric depth prediction target.

A.9.3 Depth Estimation Evaluation

PanoCARLA ablation and evaluation. To isolate the contribution of CRTM, we conduct a controlled ablation on the official PanoCARLA training and test splits. PVMetric variants with and without CRTM are trained only on PanoCARLA for 20K optimization steps with an effective batch size of 8. We additionally evaluate the selected architecture trained with the complete recipe alongside the frame-wise DAP baseline and the two controlled variants. We report absolute relative error (Abs Rel) and δ_1, the fraction of predictions satisfying a depth ratio below 1.25, under three video settings: 2 FPS / 50 frames, 10 FPS / 90 frames, and 20 FPS / 110 frames.

Table 8: Panoramic video metric-depth evaluation on PanoCARLA. The two 20K PVMetric variants are trained only on PanoCARLA under identical settings to isolate the effect of CRTM; PVMetric (full, 150K) denotes the final model trained with the complete recipe in Section A.9.

	2 FPS / 50 frames		10 FPS / 90 frames		20 FPS / 110 frames	
Method	Abs Rel ↓	δ_1 ↑	Abs Rel ↓	δ_1 ↑	Abs Rel ↓	δ_1 ↑
DAP	0.314	0.619	0.297	0.660	0.294	0.664
PVMetric w/o CRTM (20K)	0.199	0.702	0.180	0.749	0.178	0.752
PVMetric (20K)	0.194	0.727	**0.173**	0.779	**0.171**	0.785
PVMetric (full, 150K)	**0.190**	**0.794**	**0.173**	**0.832**	0.172	**0.836**

Tab. 8 shows that all PVMetric variants outperform the frame-wise DAP baseline across the three video settings. Under the controlled 20K-step protocol, adding CRTM consistently improves both metrics, motivating its inclusion in the final architecture. The full 150K model further improves δ_1 at all frame rates while maintaining comparable Abs Rel.

Zero-shot single-frame generalization. To examine whether the final PVMetric preserves the single-frame generalization ability inherited from DAP, we evaluate the final PVMetric model on Stanford2D3D Armeni et al. (2017) and Matterport3D Chang et al. (2017) without dataset-specific fine-tuning, following the evaluation protocol of DAP Lin et al. (2025).

Table 9: Zero-shot single-frame metric-depth evaluation of DAP and PVMetric.

	Stanford2D3D			Matterport3D		
Method	Abs Rel ↓	RMSE ↓	δ_1 ↑	Abs Rel ↓	RMSE ↓	δ_1 ↑
DAP	**0.1038**	**0.1760**	0.9217	**0.1290**	**0.2589**	0.8485
PVMetric	0.1163	0.1892	**0.9225**	0.1304	0.2614	**0.8490**

Tab. 9 shows that PVMetric achieves slightly higher δ_1 on both datasets, whereas DAP retains lower Abs Rel and RMSE.

InteriorGS-OmniRoam evaluation. The video generation experiments in the main paper are conducted on InteriorGS-OmniRoam Liu et al. (2026), where PVMetric provides the depth estimates used to construct the online 3D memory. We therefore evaluate PVMetric's metric-depth accuracy on the InteriorGS-OmniRoam test set.

Table 10: Metric-depth evaluation on the InteriorGS-OmniRoam test set.

Method	Abs Rel ↓	δ_1 ↑
DAP (zero-shot)	0.1516	0.8462
PVMetric	**0.0623**	**0.9541**

As shown in Tab. 10, PVMetric achieves an Abs Rel of 0.0623 and a δ_1 of 0.9541, supporting its use as the depth provider for online 3D memory in the downstream video generation experiments.

A.10 Baseline Details

We provide additional implementation and evaluation details for all baselines shown in Tab. 1.

Wan2.1-Fun. We adapt *Wan2.1-Fun-V1.1-1.3B-Control-Camera* Wang et al. (2025a) to panoramic generation by replacing perspective rays with ERP spherical rays in its Plücker camera encoding, as detailed in Appendix A.3. We fine-tune this camera-only baseline on InteriorGS-OmniRoam for a fair comparison. At inference, it performs autoregressive rollout by using the last frame of each segment as the initial-frame condition for the next, together with the corresponding camera trajectory.

OmniRoam. OmniRoam Liu et al. (2026) adopts a two-stage preview-and-refinement pipeline. Given the full eight-segment trajectory, its preview model first generates sparse panoramic frames at 960×480 with a temporal stride of 8. These frames provide guidance across the full trajectory for subsequent segment-wise refinement at 1440×720. For comparison with other autoregressive methods, we also evaluate the released preview model without refinement, using each segment's last frame to initialize the next. We denote this variant as *OmniRoam-P* in Tab. 1.

Matrix-3D. Matrix-3D Yang et al. (2025) estimates depth from the input panorama using MoGe Wang et al. (2025c) and constructs a scene mesh whose renders along the target trajectory provide geometric conditioning. For long-horizon rollout, we extend its single-segment generation autoregressively by using the last frame of each segment to initialize the next. At each segment, geometry is reconstructed from the current initial frame.

In addition to the released Matrix-3D model, we train two adapted variants on our dataset using the same ERP Wan2.1-Fun backbone as PVWander at 640×320, enabling a more controlled comparison. Matrix-3D$^{\mathrm{MoGe}}_{\mathrm{adapted}}$ follows the original depth-source protocol, using GT depth for geometric conditioning during training and MoGe-estimated depth during inference. The predicted depth of the initial frame is aligned to GT depth before projection. Matrix-3D$^{\mathrm{PVMetric}}_{\mathrm{adapted}}$ instead uses PVMetric predictions during both training and inference without GT alignment, matching the depth protocol used by PVWander.

As shown in Tab. 1, the PVMetric variant improves SSIM, PSNR, and LPIPS over the MoGe variant for both the first segment and the full rollout, supporting the benefit of maintaining a consistent depth source across training and inference. The MoGe variant achieves higher trajectory fidelity in the first segment, likely benefiting from the GT-aligned depth used for geometric projection. However, this advantage does not persist over the full rollout, where the PVMetric variant yields lower translation and rotation errors. This suggests that accurate depth alignment can benefit short-horizon geometric guidance, whereas maintaining a consistent depth-estimation pipeline becomes increasingly important under recursive long-horizon generation. Appendix A.6 further analyzes the effect of training–inference depth consistency.

PanoWorld. PanoWorld Li et al. (2026a) incorporates panoramic camera conditioning and long-range memory for controllable panoramic generation. However, the released code and weights support only single-segment generation. We therefore extend it to long-horizon evaluation through autoregressive rollout, using the last frame of each segment as the initial-frame condition for the next.

A.11 Camera Trajectory Evaluation

Pose Estimation. Following the camera-controllability evaluation of WorldScore Duan et al. (2025), we use ViPE Huang et al. (2025) to recover frame-wise camera poses from generated panoramic videos and compare them with the prescribed GT trajectories. The estimated and GT poses are converted to a common camera and world-coordinate convention.

Alignment. We left-multiply all ViPE rotations and positions by the GT first-frame rotation to express them in the GT world frame. Let $\hat{\mathbf{t}}_i$ denote the resulting estimated positions and $\mathbf{t}_i^{\mathrm{gt}}$ the GT positions for N frames indexed by $i = 0, \ldots, N-1$. We subtract each trajectory's first-frame position and estimate one global scale per sequence:

$$s^* = \arg\min_s \sum_{i=0}^{N-1} \left\| \left(\mathbf{t}_i^{\mathrm{gt}} - \mathbf{t}_0^{\mathrm{gt}}\right) - s\left(\hat{\mathbf{t}}_i - \hat{\mathbf{t}}_0\right) \right\|_2 . \tag{10}$$

Metrics and Aggregation. TransErr measures the Euclidean distance between the scale-aligned camera positions, while RotErr measures the geodesic angular distance between camera rotations in degrees. TransErr uses the same unit as the GT camera positions. We first average each metric over all frames within a scene and then equally average the scene-level results over all scenes. Lower values indicate better trajectory fidelity.