SPW-NAV: A STREAMING PANORAMIC WORLD MODEL FOR LANGUAGE-GUIDED NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-guided panoramic video generation benefits various downstream applications, such as interactive 3D scene exploration, virtual reality experiences, and embodied agent training. Existing panoramic generators follow predefined trajectories, and interactive world models act through low-level actions in perspective views. We propose SPW-Nav, a streaming panoramic world model that understands movement instructions and streams one minute of 2K 360° video in real time from a single panorama. SPW-Nav interprets each instruction in the previously generated panorama as camera motion. Spherical rotation decoupling applies rotation exactly on the sphere, pose-aligned conditioning keeps translation inputs bounded over long streams, and a multi-term memory with a few-step generator continues the scene as instructions change. We also build SPW-NavSet, panoramic videos with camera trajectories and verified instructions. Driven by language, SPW-Nav outperforms prior panoramic generators in camera-following accuracy and video quality, and supports on-the-fly instruction switching.

1 INTRODUCTION

Generative world models let users and agents explore virtual environments, and exploration requires control over where to move and what to observe next. Panoramic video covers the surroundings in all directions, and natural language expresses movement without camera poses. Language-guided exploration therefore calls for continuous panoramic video that follows a sequence of instructions and stays consistent as the route changes.

Prior approaches address only parts of this interaction. Text-to-video models (Ho et al., 2022; Singer et al., 2022) fix a whole video with one prompt. Camera-controlled panoramic generators (Ji et al., 2025; Liu et al., 2026c; Li et al., 2026b) follow predefined trajectories, while navigation and interactive world models (Bar et al., 2025; Hafner et al., 2020; Bruce et al., 2024; Zhang et al., 2025c; He et al., 2025b; Mao et al., 2025) and vision-and-language agents (Anderson et al., 2018; Savva et al., 2019) act through low-level actions (*e.g.*, keyboard inputs), mostly in forward views. None of these video generators understands instructions that refer to scene content (*e.g.*, walking to a door behind the camera), and this semantic gap motivates SPW-Nav.

Navigating a generated panoramic world raises three challenges. ***(i) Scene-dependent navigation.*** The meaning of an instruction depends on the panorama the model itself generates and therefore cannot be turned into motion in advance. ***(ii) Camera representation.*** A panoramic turn only rearranges visible content, yet existing generators synthesize it, and open-ended navigation drives world-frame poses beyond the training range. ***(iii) Translation following.*** The camera pose is a weak signal next to rich visual context, and 2K streaming leaves few network evaluations per frame.

In this paper, we propose SPW-Nav, a **S**treaming **P**anoramic **W**orld model for language-guided **Nav**igation. As illustrated in Figure 1, SPW-Nav understands the semantics of movement instructions (*e.g.*, walking to the door) and generates one minute of 360° video at 2K resolution (1024×2048) and 26.6 FPS from a single panorama. ***(i) Scene-aware navigation.*** A vision-language backbone reads the previously generated panorama and turns each instruction into an explicit displacement or a localized goal. ***(ii) Decoupled camera representation.*** Spherical rotation decoupling applies the commanded rotation exactly by resampling the sphere, leaving only translation to the video backbone. Pose-aligned conditioning expresses this translation relative to the previous chunk

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.

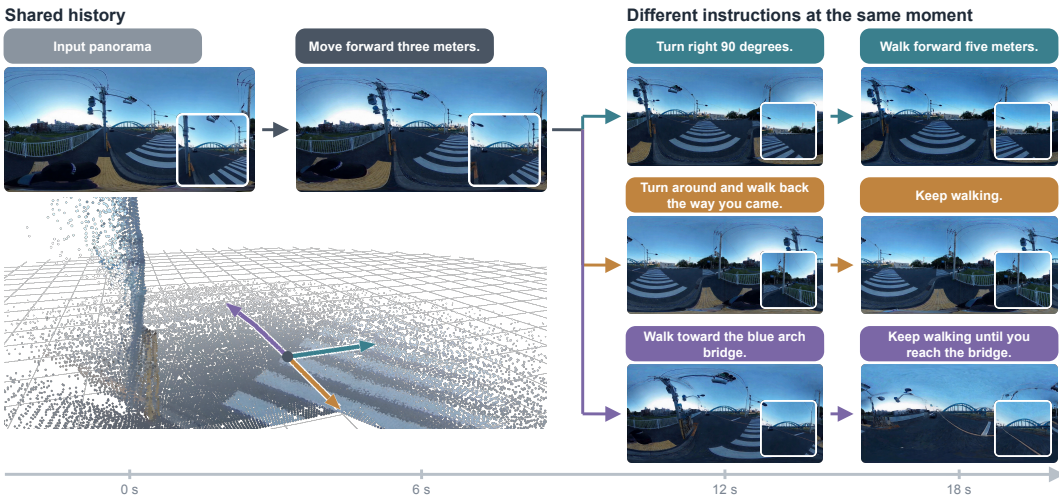

Figure 1: Language-guided exploration with SPW-Nav. ***Top left:*** a shared history generated from a single panorama. ***Bottom left:*** a point cloud reconstructed from the generated frames, showing the motion directions of the three candidate routes in the scene. ***Right:*** three different instructions given at the same moment continue this history, and each row shows the frames generated after its two instructions. Every frame shows the 360° panorama with its front view inset.

and keeps the pose inputs bounded however long the navigation runs. ***(iii) Enhanced translation following.*** Noise-tail trajectory learning focuses training and distillation on high noise levels, where the pose is the main motion cue, and opposite-trajectory guidance contrasts the commanded translation with its reverse at inference. A distilled pyramid generator with sparse attention (Li et al., 2026a) and a multi-term memory sustain these one-minute streams. The explicit motion representation lets SPW-Nav learn navigation and generation from complementary data in SPW-NavSet.

Our contributions are summarized as follows.

- We propose SPW-Nav, the first streaming panoramic world model for language-guided navigation, generating one-minute 2K 360° streams at 26.6 FPS. We also build SPW-NavSet, 233 hours of panoramic video with trajectories and verified instructions.
- We propose spherical rotation decoupling, which applies rotation exactly on the sphere such that generation is spent on translation alone, and pose-aligned conditioning, which keeps translation inputs bounded however far the navigation goes.
- We propose noise-tail trajectory learning and opposite-trajectory guidance to strengthen translation following, and enable streaming by making the video backbone autoregressive with a multi-term memory and distilling it to two steps per pyramid scale.

2 Related Work

Panoramic video generation. Spherical networks (Cohen et al., 2018; Coors et al., 2018; Su & Grauman, 2017) and 360° scene understanding (Sun et al., 2021) process equirectangular imagery. Diffusion models generate panoramic images (Zhang et al., 2024), surround-view driving video (Wen et al., 2024), and panoramic video (Zhang et al., 2025b), and have been extended to 360° video with latitude- and longitude-aware mechanisms (Xia et al., 2025) and to high-resolution cubemap autoregression (Li et al., 2026c). Panoramic depth estimation (Rey-Area et al., 2022) provides geometric priors, and video generation models in turn recover HDR environment maps of the full surrounding scene (Liang et al., 2025; Cai et al., 2026a). None of them follows the camera motion of a navigation route, whereas SPW-Nav treats each equirectangular frame as a function on the sphere, on which rotation becomes exact resampling.

Camera-controlled panoramic synthesis. CamPVG (Ji et al., 2025) conditions panoramic video on camera motion through spherical epipolar attention, OmniRoam (Liu et al., 2026c) combines trajectory-controlled preview with temporal refinement, and PanoWorld (Li et al., 2026b) uses dense ray conditioning with geometry-aware memory. These methods take predefined trajectories rather than inferring motion from language, and GenDoP (Zhang et al., 2025a) generates camera trajectories from text but no video. Perspective camera-controlled generators (He et al., 2025a; Zheng et al., 2024; Bai et al., 2025a; Zhou et al., 2025; Huang et al., 2025a) follow trajectories within a limited field of view, where a turn must synthesize new content, and LiVER (Cai et al., 2026b) conditions generation on layout, lighting, and camera controls rendered from an explicit 3D scene proxy. SPW-Nav instead infers the movement from language in the previously generated panorama and retains context in a streaming memory as instructions change.

Navigation and interactive world models. World models predict future observations from past observations and actions, from latent dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; 2025; Micheli et al., 2023; Schrittwieser et al., 2020) to high-resolution video diffusion (Blattmann et al., 2023). Autoregressive video diffusion with few-step distillation further enables streaming generation (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025b; Yuan et al., 2026). NWM (Bar et al., 2025) predicts egocentric observations from navigation actions, and Genie (Bruce et al., 2024), Matrix-Game (Zhang et al., 2025c; He et al., 2025b), and Yume (Mao et al., 2025) generate explorable perspective worlds controlled by latent or keyboard actions. Such actions prescribe camera motion without scene semantics and cannot express instructions about scene content (*e.g.*, walking to a door). Streaming benchmarks (Liu et al., 2026b) report drift and slow interactive responses. SPW-Nav instead interprets each instruction semantically, localizing referenced objects and places in the generated panorama, and learns navigation and generation from complementary data.

3 SPW-NAVSET

Real-world panoramic videos (Tan et al., 2026) cover diverse scenes with estimated camera trajectories, while offering limited control over camera motion and few paired movement instructions. We therefore curate SPW-NavSet from filtered real outdoor videos and indoor videos rendered along controllable camera paths, and pair panoramas with movement instructions.

Panoramic videos. The video backbone is trained on panoramic videos with camera trajectories, which combine indoor videos rendered with Habitat (Savva et al., 2019) from HM3D (Ramakrishnan et al., 2021), Replica (Straub et al., 2019), and ReplicaCAD (Szot et al., 2021) scenes with exact camera poses and real outdoor street videos from MUGEN (Tan et al., 2026) with estimated poses. We remove clips whose camera stays static or only rotates in place, and resample each frame to the orientation of the first one. The videos total about 18K clips and 233 hours at 1024×2048 (appendix Figure D1).

Instruction annotations. The vision-language backbone is trained on panoramas paired with movement instructions and motion targets. Explicit instructions (*e.g.*, moving forward three meters or turning back) come from camera trajectories with varied phrasings. Goal instructions name a visible object or place (*e.g.*, a door), localized by a bounding box and verified by vision-language models. We manually check 200 random samples and hold out scenes for validation and testing.

SPW-NavBench. This test set contains 100 real-world outdoor clips, disjoint from all training data down to the source videos. Each clip provides an initial panorama, its ground-truth camera trajectory, and two instructions. A motion instruction describes this trajectory, and a navigation instruction is either an explicit instruction or a goal instruction (see appendix).

4 METHODOLOGY

As illustrated in Figure 2, SPW-Nav generates each chunk of video from the latest panorama, the active instruction, and a memory of generated frames.

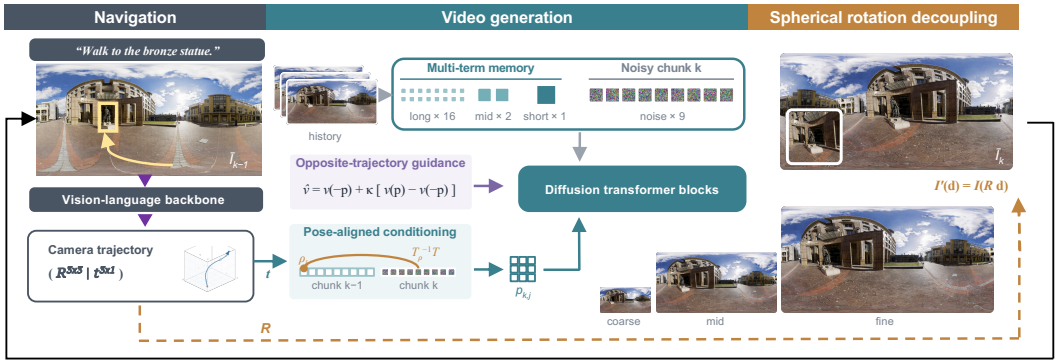

Figure 2: Overview of the SPW-Nav pipeline. (a) The vision-language backbone localizes the referenced target (box) in the latest panorama $\bar{I}_{k-1}$ and outputs a camera trajectory $(R, \mathbf{t})$. (b) Memory tokens at long, mid, and short range and the noisy latents of chunk k enter the diffusion transformer, and pose tokens $\mathbf{p}_{k,j}$ express $\mathbf{t}$ relative to the reference frame ρ_k. The chunk is denoised from coarse to fine, with guidance $\hat{v}$ at the coarsest scale. (c) The rotation R bypasses the video backbone (dashed line) and is applied to the rotation-free output by spherical resampling. The last frame $\bar{I}_k$ is fed back for the next instruction (solid line).

4.1 Preliminaries

Given a panorama I_0 at pose T_0 and a sequence of movement instructions $\{u_1, \ldots, u_K\}$, SPW-Nav outputs a stream of panoramic video segments $\{V_1, \ldots, V_K\}$ that extends as new instructions arrive. Segment V_k carries out u_k and is generated in chunks of $F = 33$ frames. Each frame is an equirectangular panorama I of resolution $H \times W = 1024 \times 2048$, whose pixel (x, y) observes the unit ray

$$\mathbf{d}(x, y) = (\cos\phi \sin\theta,\ \sin\phi,\ \cos\phi\cos\theta), \qquad \theta = 2\pi\left(\tfrac{x+1/2}{W} - \tfrac{1}{2}\right), \quad \phi = \pi\left(\tfrac{y+1/2}{H} - \tfrac{1}{2}\right), \tag{1}$$

with longitude θ, latitude ϕ, and camera axes pointing right, down, and forward. Frame I_n of the stream has camera pose $T_n = (R_n, \mathbf{t}_n) \in \mathrm{SE}(3)$.

4.2 Navigation in the current view

Some instructions (*e.g.*, moving forward or turning left) describe movement relative to the current viewpoint, while others (*e.g.*, walking to the door) refer to content that only the panorama reveals. SPW-Nav therefore interprets each instruction u_k in the latest generated panorama $\bar{I}_{k-1}$ (*i.e.*, the last frame of the previous segment, with $\bar{I}_0 = I_0$) and expresses it as one of two motion targets relative to the current camera (Figure 2(a)). A displacement target lists up to three waypoints (x, z, ψ) in meters and degrees relative to the current camera, which covers multi-step movement, stopping, and corrections. A goal target marks the named object or place with a bounding box, which may wrap across the image border, and an approach distance r. If c is the normalized horizontal center of the box, the camera turns to the bearing $\psi^\star = 2\pi(c - 1/2)$ and then moves forward by r. Either target is converted into the camera trajectory

$$A_k = \{T_{k,1}, \ldots, T_{k,L}\}, \qquad T_{k,j} = T_{k-1} \cdot \Delta T_{k,j}, \qquad \Delta T_{k,j} = \begin{bmatrix} \Delta R_{k,j} & \Delta\mathbf{t}_{k,j} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{2}$$

where $\Delta T_{k,j}$ interpolates the waypoints over the segment.

Architecture and training. The vision-language backbone is Qwen3-VL-8B (Bai et al., 2025b) with rank-16 LoRA on the attention layers of its language model and a frozen vision encoder. It is trained on the instruction annotations of SPW-NavSet (Section 3) with the next-token objective on the motion target, conditioned on the panorama and the instruction.

4.3 POSE-ALIGNED VIDEO GENERATION

Spherical rotation decoupling (SRD). A camera rotation changes which direction each pixel observes but reveals no new content in a panorama (Figure 2(c)). Rotating the camera by R maps a panorama I to $I'(\mathbf{d}) = I(R\,\mathbf{d})$, evaluated at every pixel ray (Eq. 1) with bilinear sampling that wraps around the longitude seam. In training, each frame is resampled to the orientation of the first frame, $\widetilde{I}_n(\mathbf{d}) = I_n(R_n^\top R_0\,\mathbf{d})$, and the trajectory reduces to the translation $\widetilde{\mathbf{t}}_n = R_0^\top(\mathbf{t}_n - \mathbf{t}_0)$. At inference, frames are generated in the orientation of the initial panorama and the rotation of A_k is restored after decoding. Rotation is thus exact up to interpolation (appendix Proposition 1), leaving the generator to model only the parallax of translation.

Pose-aligned conditioning (PAC). World-frame poses of the same motion differ across places and grow with the distance traveled. PAC therefore anchors every pose to a recent view (Figure 2(b)). Every camera pose of chunk k is expressed as $\widetilde{T}_n = T_{\rho_k}^{-1} T_n$ relative to ρ_k, the first frame of the preceding chunk (ρ_1 is the first frame of chunk 1). In the rotation-free frame, this pose has an identity rotation block and translation $s\,(\widetilde{\mathbf{t}}_n - \widetilde{\mathbf{t}}_{\rho_k})$ with a fixed scale s. Since ρ_k lies in the memory (Section 4.4), the motion is stated relative to a visible view. Each of the 9 latent frames of the chunk receives its flattened 3×4 pose $\mathbf{p}_{k,j} \in \mathbb{R}^{12}$, whose identity rotation block is kept for generality, and $\mathbf{p}_k$ stacks them. In every transformer block b, a zero-initialized linear layer maps $\mathbf{p}_{k,j}$ to a token offset

$$\mathbf{h}^{(b)}_{k,j,\ell} \leftarrow \mathbf{h}^{(b)}_{k,j,\ell} + W^{(b)}\mathbf{p}_{k,j} + \mathbf{c}^{(b)}, \tag{3}$$

which is added to every spatial token ℓ of latent frame j and leaves the pretrained backbone unchanged at initialization. A relative translation of $(1, 0, 0)$ thus always means moving right in the current view. Since the reference is at most two chunks earlier, the inputs are invariant to rigid transformations of the world frame and stay within the training range.

Noise-tail trajectory learning (NTL). High noise levels largely determine camera motion and coarse structure (Liu et al., 2026a; Zhao et al., 2026). NTL therefore emphasizes them both when the autoregressive teacher learns the trajectory conditioning and when the few-step student is distilled (Section 4.4). In distillation, each student sample $\hat{z}$ is re-noised to $\hat{z}_\sigma = (1-\sigma)\hat{z} + \sigma\epsilon$ and scored by a critic and the teacher. In both stages, a fraction $\lambda = 0.8$ of the noise levels comes from the tail $[\sigma_h, \sigma_{\max}] = [0.9, 0.98]$ and the rest from the default distribution p_0 of that stage,

$$\sigma \sim p_\lambda = \lambda\,\mathcal{U}[\sigma_h, \sigma_{\max}] + (1-\lambda)\,p_0. \tag{4}$$

At high noise the sample shows almost no layout. The training target and the score difference thus mainly concern where content should move under the relative poses. At low noise, in contrast, the layout is already visible and the pose adds little information. Beyond improving navigation, NTL also speeds up the convergence of trajectory learning.

Opposite-trajectory guidance (OTG). In a panorama, moving forward and moving backward produce mirrored flow fields, with expansion at the heading and contraction at its antipode. OTG therefore contrasts the commanded translation with its reverse at the coarsest scale of the generator (Section 4.4), where the velocity is

$$\hat{v} = v_\theta(z_\sigma, \sigma, \mathcal{M}_k, -\mathbf{p}_k) + \kappa\big(v_\theta(z_\sigma, \sigma, \mathcal{M}_k, \mathbf{p}_k) - v_\theta(z_\sigma, \sigma, \mathcal{M}_k, -\mathbf{p}_k)\big), \tag{5}$$

where $-\mathbf{p}_k$ negates the translation and keeps all other inputs. The difference cancels components shared by both directions (*e.g.*, scene appearance) and amplifies the one that depends on the direction of travel. Finer scales refine appearance with the plain conditional velocity, and we use $\kappa = 2$.

4.4 STREAMING GENERATION WITH MEMORY

When a new instruction arrives, the model must continue from the previous chunk without losing visual context and with few network evaluations per chunk. We build the streaming generator on the autoregressive recipe of Helios (Yuan et al., 2026) and detail the loop in Algorithm G.1.

Multi-term memory. The memory $\mathcal{M}_k$ (Figure 2(b)) holds the generated stream in three terms, the most recent latent frame, the two before it, and sixteen earlier ones, with older terms patchified more coarsely. A fixed anchor also keeps the first frame of the stream, which keeps the initial view visible throughout the stream. Memory tokens enter the transformer alongside the chunk tokens at noise level zero, and the first chunk starts from the input panorama repeated as a static clip. Since

generation runs in the rotation-free orientation and each remembered panorama covers all directions, rotation never disturbs the memory and recent appearance in every direction stays available.

Few-step generation. Each chunk is denoised from coarse to fine over a spatial pyramid of three scales, which splits the noise range into equal thirds. The student uses two steps per scale (four for the first chunk). A chunk thus takes six network evaluations plus two for the guidance of Eq. 5. The student is initialized by ODE regression onto the teacher and trained with distribution matching distillation (Yin et al., 2024), with the teacher as the real score and a critic tracking the student.

Streaming training. At inference the memory holds generated frames, whose errors differ from those of real video. ODE regression therefore uses histories from the model's own rollouts, and distillation corrupts ground-truth histories with noise, downsampling, and saturation shifts with probability 0.9, such that the student learns to continue from imperfect memory. Chunks do not overlap, and the memory alone carries continuity across chunks and instructions.

5 Experiments

5.1 Experimental setup

Implementation details. The video backbone is Wan2.2-TI2V-5B (Team Wan et al., 2025) fine-tuned with LoRA on panoramic video and generates 1024×2048 video at 16 FPS. It is trained without language on the rotation-free videos of SPW-NavSet in autoregressive, pyramid, and trajectory-conditioned stages, and then distilled into a few-step student (see appendix).

Baselines. We compare with generators that accept camera motion, each through its own interface. The panoramic baselines are NWM (Bar et al., 2025), CubeComposer (Li et al., 2026c), OmniRoam (Liu et al., 2026c), and the Action and Causal variants of PanoWorld (Li et al., 2026b), denoted PanoWorld-A. and PanoWorld-C. CubeComposer completes a perspective video into a panorama and takes neither trajectory nor language. It therefore appears only in camera-controlled generation. For camera-controlled and long video generation, we add Matrix-Game 3.0 (Wang et al., 2026), a streaming world model with long-horizon memory driven by keyboard and mouse actions in a perspective view. Matrix-Game 3.0 is evaluated in its native view and as a panorama completed by CubeComposer. The comparison with perspective generators Stable Virtual Camera (Zhou et al., 2025), ReCamMaster (Bai et al., 2025a), CamI2V (Zheng et al., 2024), HunyuanWorld-Voyager (Huang et al., 2025a), and CameraCtrl (He et al., 2025a) appears in the appendix.

Metrics. Camera-following accuracy is measured on the camera motion estimated from the generated video (Wang et al., 2025) by the trajectory direction error (TDE), the cumulative and per-interval rotation errors (RE and RPE_R), the error of the mean cumulative rotation (ΔRot.), a scale-free trajectory shape error (ATE), and motion accuracy (MA). Video quality is measured by PSNR, LPIPS, and FVD. The appendix gives the exact definitions.

5.2 Language-guided navigation

We first evaluate the full system, from a language instruction to panoramic video, and start every method from the same motion instruction of each SPW-NavBench clip. No prior panoramic generator understands movement instructions semantically, which is the gap SPW-Nav addresses. Each baseline therefore receives the instruction through the closest interface it offers. PanoWorld and OmniRoam take it in their text prompt with the motion preset matching its keywords, and NWM takes the actions planned by our vision-language backbone without fine-tuning.

Table 1 shows that SPW-Nav leads on every measure except ATE, where the keyword-matched presets give PanoWorld a predefined trajectory shape while SPW-Nav infers the trajectory from the instruction and the scene. SPW-Nav reaches a TDE of 56.66° against at least 62.07° and an FVD of 351.6 against at least 628.8, and its RE is four times lower than that of any baseline, since SRD applies the interpreted rotation exactly while the baselines rotate only slightly. Both PanoWorld variants also lose about 5 dB of PSNR relative to Table 2, whereas SPW-Nav loses less than 1 dB. Replacing the fine-tuned backbone with its base model raises TDE to 100.74° and lowers MA to 0.028, since the base model often misreads the commanded direction.

Table 1: Language-guided navigation on SPW-NavBench. Each method receives the same instruction through its own interface. The italic row uses the base vision-language model and is not ranked. In all tables, the best is bold and the second best underlined.

Method	ΔRot.↓	TDE↓	RE↓	RPE_R↓	ATE↓	MA↑	PSNR↑	LPIPS↓	FVD↓
NWM (Bar et al., 2025)	82.74	84.79	83.87	19.18	23.48	0.064	9.90	N/A	1216.1
OmniRoam (Liu et al., 2026c)	13.00	<u>62.07</u>	13.46	1.603	4.75	0.256	<u>11.47</u>	0.654	694.9
PanoWorld-C. (Li et al., 2026b)	13.19	63.36	13.22	<u>1.517</u>	<u>4.25</u>	<u>0.264</u>	11.29	0.645	638.5
PanoWorld-A. (Li et al., 2026b)	<u>12.76</u>	63.03	<u>12.99</u>	1.545	**4.13**	0.257	11.14	<u>0.633</u>	<u>628.8</u>
SPW-Nav	**1.45**	**56.66**	**3.02**	**0.780**	6.28	**0.305**	**15.69**	**0.398**	**351.6**
W/o navigation fine-tuning	0.79	100.74	3.84	1.379	10.34	0.028	15.23	0.413	410.7

Table 2: Camera-controlled panoramic generation from ground-truth trajectories.

Method	ΔRot.↓	TDE↓	RE↓	RPE_R↓	ATE↓	MA↑	PSNR↑	LPIPS↓	FVD↓
NWM (Bar et al., 2025)	82.55	86.19	85.64	19.12	22.89	0.081	10.68	N/A	1176.1
CubeComposer (Li et al., 2026c)	13.97	91.40	13.97	<u>1.521</u>	22.60	0.073	14.61	0.565	973.1
OmniRoam (Liu et al., 2026c)	<u>12.30</u>	77.12	15.07	2.042	10.40	0.111	15.36	0.489	610.5
PanoWorld-C. (Li et al., 2026b)	13.46	26.28	<u>13.47</u>	1.528	<u>4.24</u>	<u>0.688</u>	16.12	0.460	462.0
PanoWorld-A. (Li et al., 2026b)	13.61	**24.26**	13.62	1.530	**3.93**	**0.726**	<u>16.24</u>	<u>0.425</u>	<u>384.2</u>
Matrix-Game 3.0 + CubeComposer	13.13	92.91	13.14	1.479	20.24	0.080	14.59	0.587	892.8
SPW-Nav	**0.42**	<u>24.30</u>	**1.84**	**0.282**	5.00	0.631	**16.48**	**0.382**	**308.0**

Navigation accuracy. Evaluated alone on the held-out test scene and on the navigation instructions of SPW-NavBench (40 explicit and 60 goal instructions, see appendix), the vision-language backbone interprets all explicit instructions correctly and localizes 62% of the visible targets. On the held-out test scene, fine-tuning raises the accuracy from 0.663 to 0.925 for explicit instructions and from 0.154 to 0.692 for goal instructions. Without the panorama, goal localization drops to zero, which indicates that goals are located in the image rather than guessed from the words. On the motion instructions, it matches the commanded direction for 98% of the instructions with a mean distance error of 0.03 m relative to the commanded distance.

5.3 Camera-controlled generation

We then isolate the video backbone by giving every method the ground-truth camera trajectory instead of language. SPW-Nav and OmniRoam take the full trajectory, the PanoWorld variants take its translation with a fixed heading, NWM takes the corresponding actions with three ground-truth past views, and CubeComposer takes a perspective video rendered along the commanded rotation.

As Table 2 shows, SPW-Nav rotates by $13.59°$ for a commanded rotation of $14.01°$, while the panoramic baselines rotate by less than $2°$ and NWM overshoots. Its RE is thus seven times lower than that of PanoWorld-A. This margin holds by construction, since SRD applies the turn by resampling, whereas the baselines either fix the heading or reproduce little of the rotation they receive. For translation, SPW-Nav matches PanoWorld-A. in direction, while PanoWorld-A. reaches a lower ATE and a higher MA. PanoWorld-A. specializes in fixed-heading 960×480 clips, whereas SPW-Nav streams 2K panoramas with arbitrary turns and a few-step generator. This gap therefore reflects a trade-off between specialization and generality. SPW-Nav achieves the best video quality on all three measures, and Figure 3 shows qualitative comparisons. Against the perspective generators, evaluated on a perspective view cropped from the panorama (see appendix), SPW-Nav reaches the lowest RE and RPE_R and the second-best PSNR, LPIPS, and FVD.

5.4 Applications

Long video generation. SPW-Nav streams one minute of continuous 2K panoramic video (960 frames at 16 FPS) from a single panorama, including panoramas outside its training data (appendix Figures A1 and A2). Such long streams must stay consistent when a route returns to its start. For each of 20 scenes, we run two closed routes (*i.e.*, the final pose equals the initial one) of twelve



Figure 3: Qualitative comparison with ground-truth trajectories. Each row shows the panoramic video generated by one method at six evenly spaced timestamps from left to right. The last row shows the ground truth, and SPW-Nav is outlined in green.

Table 3: Long-horizon consistency on 40 closed routes of twelve chunks. †Native perspective view.

Method	Memory gain ↑	Closure rot. (°) ↓	Closure trans. (%) ↓
OmniRoam (Liu et al., 2026c)	0.026	11.02	57.5
PanoWorld-A. (Li et al., 2026b)	0.084	39.52	59.0
HunyuanWorld-Voyager† (Huang et al., 2025a)	0.087	<u>8.91</u>	40.7
Matrix-Game 3.0† (Wang et al., 2026)	**0.249**	28.41	32.7
+ CubeComposer (Li et al., 2026c)	<u>0.204</u>	22.26	<u>32.0</u>
SPW-Nav	0.173	**1.62**	**30.7**

chunks (see appendix). Memory gain measures how much closer a revisited view is to its earlier observation than a non-revisiting view, and closure errors compare the final and initial poses. Table 3 compares SPW-Nav with OmniRoam, PanoWorld-A., HunyuanWorld-Voyager, and Matrix-Game 3.0, where the last two are evaluated in their native perspective view. SPW-Nav closes the loop with a rotation error of $1.62°$, against $8.91°$ for HunyuanWorld-Voyager, the next best, and up to $39.52°$ for PanoWorld-A. It also reaches the lowest translation closure error, 30.7%. Matrix-Game 3.0 reaches a higher memory gain than SPW-Nav (0.249 in its perspective view and 0.204 as a panorama, against 0.173), but its closure rotation error exceeds $22°$ in both settings.

Instruction switching. New instructions may arrive at any time. We run 20 scenes through four consecutive instructions spanning twelve chunks (about 25 seconds), where the vision-language backbone reads the last generated frame before every instruction and the memory holds only generated frames. Figure 4 shows examples, and Table A2 lists per-segment metrics. Instruction following does not degrade with the segment index, while video quality drops after the first segment. Switches cause no visible jump, as inter-frame LPIPS across instruction boundaries is no higher than within segments. With eight H200 GPUs and sparse attention, SPW-Nav generates 26.6 frames per second, faster than the 16 FPS playback rate of its output.

5.5 Ablation Study

Table 4 ablates NTL, OTG, and SRD. Without NTL, neither the teacher nor the student uses the noise tail. Disabling OTG means $\kappa = 1$ in Eq. 5.

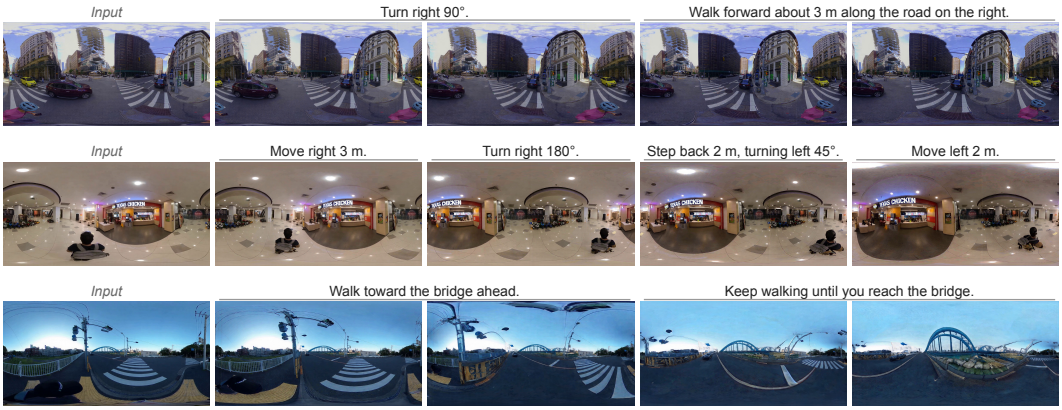

Figure 4: Streaming generation with instruction switching. The first column shows the input panorama, and each instruction above spans the frames generated while following it.

Table 4: Ablation on SPW-NavBench. Each row of the lower block changes one component of the full model.

NTL	OTG	TDE↓	MA↑	ATE↓	RE↓	RPE_R↓	PSNR↑	LPIPS↓	FVD↓
		29.15	0.469	**4.80**	26.58	3.157	13.69	0.494	512.2
	✓	27.85	0.517	5.64	28.13	3.350	13.26	0.514	562.3
✓		43.53	0.395	12.33	**0.99**	**0.160**	**17.05**	**0.361**	395.6
✓	✓	**24.30**	**0.631**	5.00	1.84	0.282	16.48	0.382	**308.0**
W/o SRD		27.99	0.521	5.99	31.31	3.801	12.97	0.518	596.5

Impact of OTG. With NTL but without OTG, the video often moves in an unintended direction. Adding OTG cuts TDE from $43.53°$ to $24.30°$, raises motion accuracy, and lowers FVD. Larger guidance weights no longer improve TDE but raise RE and degrade appearance (appendix Table A4). OTG adds only two network evaluations per chunk, both at the coarsest scale.

Impact of NTL. Without NTL, RE rises from $0.99°$ to $26.58°$, PSNR drops by more than 3 dB, and FVD rises from 395.6 to 512.2. OTG then lowers TDE only from $29.15°$ to $27.85°$ and further degrades appearance. NTL is therefore needed both for motion accuracy and for the gain from OTG.

Impact of SRD. Without SRD, the model is trained on videos that keep their rotation and generates the rotation itself. The available model of this kind is also trained without NTL. Its RE rises from $1.84°$ to $31.31°$, and its FVD from 308.0 to 596.5. Generating the rotation thus costs both rotation accuracy and video quality, whereas resampling on the sphere applies it exactly.

6 Conclusion

We present SPW-Nav, a streaming panoramic world model that interprets each language instruction in the previously generated panorama and streams one minute of 2K 360° video at 26.6 FPS. Spherical rotation decoupling makes rotation exact, pose-aligned conditioning keeps translation inputs bounded, and noise-tail trajectory learning with opposite-trajectory guidance strengthens translation following. SPW-Nav outperforms existing panoramic generators in camera-following accuracy and video quality and closes long routes with near-exact rotation, making language a practical interface for exploring generated worlds. Since generation runs faster than playback, new instructions can be issued while the stream plays. Consistency on distant revisits and the trajectory shape of translation remain open. In future work, a pose-indexed 3D memory could make distant revisits more consistent, and more diverse training trajectories could improve the shape of translation. Motion targets could also come from other interfaces, such as clicks on the panorama or external planners, since the video backbone consumes only camera trajectories and never the instruction itself.

AI USE STATEMENT

We used large language model assistants to polish the writing, to draft scripts for figure layout and experiment bookkeeping, and to help organize and monitor experiment runs. All research ideas, method designs, experiment designs, and reported results are the authors' own. The authors reviewed every piece of AI-assisted text and code and verified every reported number against the experiment logs, and they take full responsibility for the content of this paper.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *IEEE/CVF International Conference on Computer Vision*, 2025a.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

Ziqi Cai, Shuchen Weng, Kaiqi Liu, Zifeng Wang, Zhiquan Zhang, Minggui Teng, Han Jiang, and Boxin Shi. Video generation models are inherent lighting estimators. *arXiv preprint arXiv:2607.04674*, 2026a.

Ziqi Cai, Taoyu Yang, Zheng Chang, Si Li, Han Jiang, Shuchen Weng, and Boxin Shi. Lighting-grounded video generation with renderer-based agent reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *European Conference on Computer Vision*, 2018.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.

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*, 2025a.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025b.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3D scene generation. *ACM Transactions on Graphics*, 2025a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025b.

Chenhao Ji, Chaohui Yu, Junyao Gao, Fan Wang, and Cairong Zhao. CamPVG: Camera-controlled panoramic video generation with epipolar-aware diffusion. In *ACM SIGGRAPH Asia Conference*, 2025.

Haopeng Li, Yitong Li, Junsong Chen, Tian Ye, Haozhe Liu, Jincheng Yu, Duomin Wang, Ruihua Zhang, Zeke Xie, Enze Xie, and Song Han. Sol-Attn: Accelerating video generation inference via on-the-fly attention sparsification. *arXiv preprint arXiv:2607.24027*, 2026a.

Haoyuan Li, Dizhe Zhang, Yuemei Zhou, Xiangkai Zhang, Haoran Feng, Xiaofan Lin, Wenjie Jiang, Bo Du, Ming-Hsuan Yang, and Lu Qi. PanoWorld: Real-world panoramic generation. *arXiv preprint arXiv:2607.09661*, 2026b.

Lingen Li, Guangzhi Wang, Xiaoyu Li, Zhaoyang Zhang, Qi Dou, Jinwei Gu, Tianfan Xue, and Ying Shan. CubeComposer: Spatio-temporal autoregressive 4K 360° video generation from perspective video. *arXiv preprint arXiv:2603.04291*, 2026c.

Ruofan Liang, Kai He, Zan Gojcic, Igor Gilitschenski, Sanja Fidler, Nandita Vijaykumar, and Zian Wang. LuxDiT: Lighting estimation with video diffusion transformer. In *Advances in Neural Information Processing Systems*, 2025.

Jiwen Liu, Shujuan Li, Xiaohan Li, Zijie Meng, Xinyue Liu, Yulong Xu, Yan Zhou, and Guoxin Zhang. TARS: Timestep-aware data scaling for 3D-free video re-shooting. *arXiv preprint arXiv:2607.28261*, 2026a.

Kaiqi Liu, Haoxuan Zeng, Jingqi Liu, Jiacong Fang, Ziqi Cai, Yunyao Mao, Henglin Liu, Yu Sheng, Shuchen Weng, and Boxin Shi. StreamAV-Bench: A comprehensive benchmark for streaming audio-video generation. *arXiv preprint arXiv:2608.26336*, 2026b.

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. In *ACM SIGGRAPH Conference*, 2026c.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360MonoDepth: High-resolution 360° monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 2020.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360° imagery. In *Advances in Neural Information Processing Systems*, 2017.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. HoHoNet: 360° indoor holistic understanding with latent horizontal features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimír Vondruš, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, 2021.

Jiaming Tan, Mingliang Zhai, Zhen Li, Yuwei Wu, Chuanhao Li, and Kaipeng Zhang. AlayaVista: Streaming world modeling from panoramic states to perspective video. *arXiv preprint arXiv:2609.14462*, 2026.

Team Wan, 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, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing 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 preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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. In *Advances in Neural Information Processing Systems*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming stable diffusion for text to 360° panorama image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. GenDoP: Auto-regressive camera trajectory generation as a director of photography. In *IEEE/CVF International Conference on Computer Vision*, 2025a.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025b.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, and Yahui Zhou. Matrix-Game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025c.

Pu Zhao, Juyi Lin, Timothy Rupprecht, Arash Akbari, Chence Yang, Rahul Chowdhury, Elaheh Motamedi, Arman Akbari, Yumei He, Chen Wang, Geng Yuan, Weiwei Chen, and Yanzhi Wang. PhyWorld: Physics-faithful world model for video generation. *arXiv preprint arXiv:2605.19242*, 2026.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. CamI2V: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024.

Jensen 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. In *IEEE/CVF International Conference on Computer Vision*, 2025.

A ADDITIONAL RESULTS

Upward and downward arrows mark metrics where larger and smaller values are better.

A.1 ZERO-SHOT GENERALIZATION TO UNSEEN SCENES

Figure A1 shows streams generated from panoramas that are not part of the training data, covering indoor halls, city streets, waterfronts, and natural scenes.

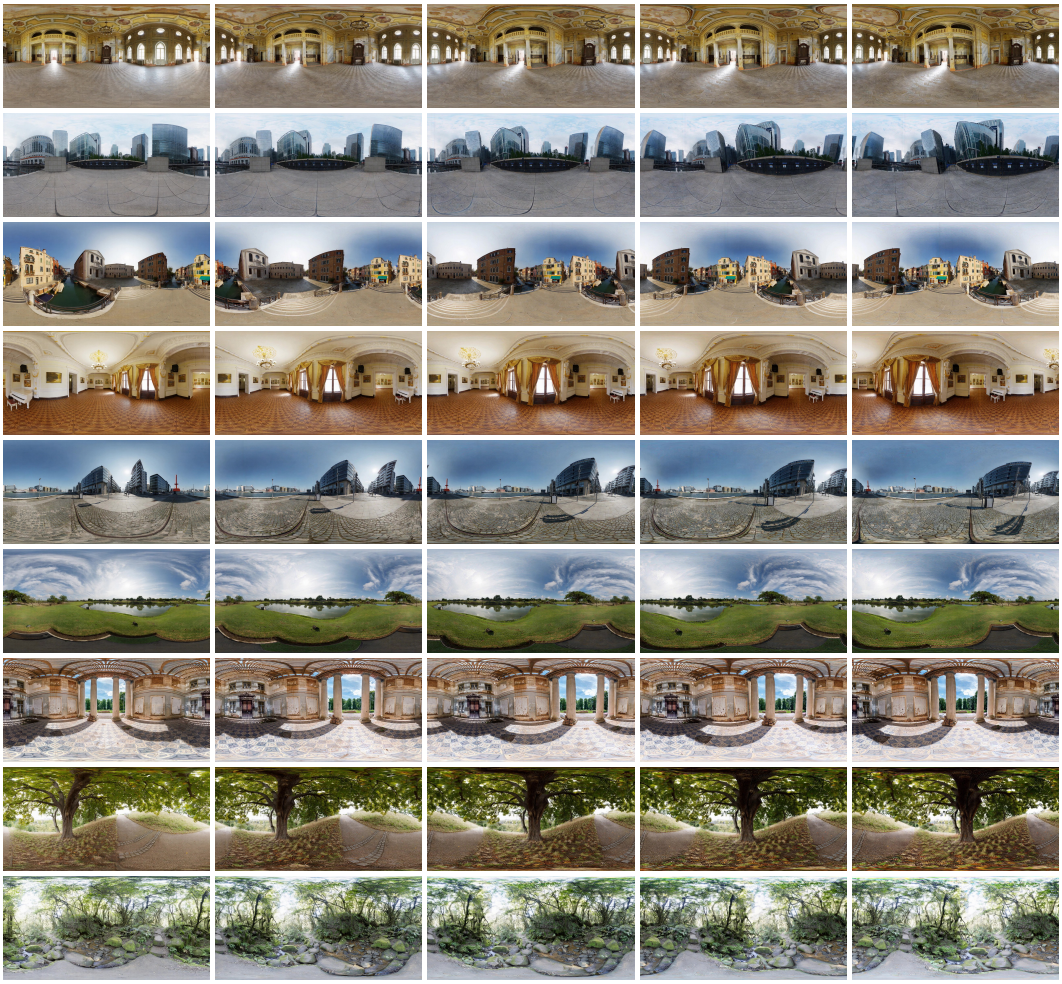

Figure A1: Zero-shot streams generated by SPW-Nav from panoramas outside its training data. Each row is one stream of 14 to 30 seconds with evenly spaced frames.

A.2 ONE-MINUTE STREAMS

Figure A2 shows streams of about one minute, each generated continuously from a single panorama.



Figure A2: One-minute streams from SPW-Nav. Each row spans 60 to 74 seconds.

A.3 Navigation accuracy

Table A1 evaluates the vision-language backbone in isolation on the held-out test scene and on all navigation instructions of SPW-NavBench.

Table A1: Navigation accuracy. Accuracy of the vision-language backbone on the held-out test scene and on the navigation instructions of SPW-NavBench.

	Test scene		SPW-NavBench	
Method	Action	Goal	Action	Goal
Template rules	0.100	0.000	0.375	0.000
W/o fine-tuning	0.663	<u>0.154</u>	<u>0.850</u>	<u>0.450</u>
W/o image	<u>0.900</u>	0.135	**1.000**	0.000
Ours	**0.925**	**0.692**	**1.000**	**0.617**

A.4 Streaming generation

Table A2 reports metrics for each of four consecutive instructions in the streaming experiment.

Table A2: Streaming generation. Metrics over four consecutive instructions in 20 scenes. Since segment 2 is a pure turn, its TDE is undefined.

Segment	1	2	3	4
TDE (°) ↓	11.2	N/A	31.8	19.1
RE (°) ↓	0.9	2.8	1.1	0.6
MA ↑	0.86	1.00	0.51	0.69
FVD ↓	224.9	364.4	316.3	435.5

A.5 Comparison with perspective generators

Perspective camera-controlled generators take no language instructions and cannot produce panoramas. We therefore compare them only in this setting, on a perspective view cropped from SPW-Nav's output (Table A3). SPW-Nav has the lowest rotation errors, and its video quality is second only to CameraCtrl. The remaining gap follows from how the evaluated view is produced. The 60° crop covers one sixth of the panorama width, about 340 pixels of SPW-Nav's output, while each perspective baseline spends its full output resolution on this single view and generates nothing outside it. HunyuanWorld-Voyager follows translation more accurately through depth-based reprojection.

Table A3: Comparison with perspective generators on SPW-NavBench.

Method	PSNR ↑	LPIPS ↓	FVD ↓	TDE ↓	RE ↓	MA ↑	RPE_R ↓
Stable Virtual Camera (Zhou et al., 2025)	12.52	0.531	553.6	26.3	6.95	0.677	1.987
ReCamMaster (Bai et al., 2025a)	15.54	0.440	353.7	48.9	5.19	0.336	0.836
CamI2V (Zheng et al., 2024)	16.54	0.364	332.8	30.4	3.29	0.666	0.483
HunyuanWorld-Voyager (Huang et al., 2025a)	16.57	0.388	301.0	**17.1**	3.45	**0.768**	0.583
CameraCtrl (He et al., 2025a)	**17.31**	**0.340**	**269.8**	25.5	3.28	0.713	0.493
Matrix-Game 3.0 (Wang et al., 2026)	16.10	0.355	844.2	93.2	8.69	0.057	0.943
SPW-Nav	17.12	0.352	278.8	27.0	**2.23**	0.617	**0.321**

A.6 Guidance weight

Table A4 sweeps the OTG weight κ with and without NTL.

Table A4: Full sweep of the OTG weight κ with and without NTL on SPW-NavBench.

NTL	κ	TDE ↓	MA ↑	ATE ↓	RE ↓	RPE_R ↓	PSNR ↑	LPIPS ↓	FVD ↓
	1	29.15	0.469	**4.80**	26.58	3.157	13.69	0.494	512.2
	2	27.85	0.517	5.64	28.13	3.350	13.26	0.514	562.3
✓	1	43.53	0.395	12.33	**0.99**	**0.160**	**17.05**	**0.361**	395.6
✓	2	24.30	0.631	5.00	1.84	0.282	16.48	0.382	**308.0**
✓	3	**24.09**	**0.649**	5.25	3.14	0.506	15.67	0.412	333.4
✓	4	24.80	0.642	6.24	5.71	0.959	14.99	0.442	381.5
✓	5	24.82	0.634	6.88	4.90	1.089	14.42	0.469	436.9

B Evaluation protocols

B.1 Protocol

The navigation table evaluates the vision-language backbone on a held-out test scene and on SPW-NavBench, which contains 40 displacement instructions and 60 visible-target instructions. The streaming and long-horizon experiments use 20 scenes. The long-horizon experiment runs two closed routes of twelve chunks per scene whose final pose equals the initial one. The first moves forward, turns around, walks back, and turns around again. The second moves forward and back, then turns right and left by the same angle. Matrix-Game 3.0 receives each route as keyboard and

mouse actions under its official calibration, and CubeComposer completes its perspective video into a panorama for the panoramic evaluation. Unless noted otherwise, video metrics are computed at the source timestamps after aligning each prediction to the evaluated view.

B.2 METRIC DEFINITIONS

We use two groups of metrics. For the video-generation metrics, camera motion is estimated from forward crops of the generated video with a VGGT-based estimator. Trajectory direction error (TDE) is the angle between the commanded and estimated translation directions, rotation error (RE) is the error of the cumulative rotation, and $\mathrm{RPE_R}$ is the rotation error per interval. ΔRot. is the absolute difference between the mean estimated cumulative rotation and the mean commanded rotation. ATE is a scale-free trajectory shape error, and motion accuracy (MA) is the fraction of evaluation intervals whose motion follows the commanded direction. PSNR on the forward view, LPIPS averaged over four viewing directions, and FVD measure video quality. Memory gain compares a revisited view with its earlier observation, relative to a non-revisiting view at the same time gap, and closure errors compare the final estimated pose with the initial one. The navigation-target metrics evaluate the vision-language backbone alone. A displacement target is correct when its path is within $\max(0.15\,\mathrm{m}, 10\%)$ of the commanded excursion and its heading within 5°. A goal target is correct when its box overlaps the annotated target with an IoU of at least 0.3 and its center bearing is within 20°. The Action and Goal columns of the navigation table report these accuracies.

C LIMITATIONS AND FUTURE WORK

The multi-term memory of generated frames already keeps revisited views consistent and closes long routes with near-exact rotation. A more explicit memory (*e.g.*, a pose-indexed 3D scene representation) could make consistency on distant revisits more reliable, although advancing memory mechanisms for video generation may also achieve this within the video itself. More diverse training trajectories could further improve the trajectory shape of translation.

D DATASET DETAILS

Indoor videos. We render 8,527 panoramic videos with Habitat from 29 indoor scene configurations, three from HM3D, ten from Replica, and sixteen from ReplicaCAD, along camera paths that cover translation in all four horizontal directions, turns, and composed routes.

Outdoor videos. The outdoor part keeps 9,609 MUGEN clips from 2,856 source videos after removing static and in-place rotating cameras. The original captions often describe turning, which the rotation-free video no longer shows. We therefore replace them with a neutral scene prompt. The distillation stage uses a direction-balanced subset of 2,500 of these videos.

Instruction annotations. Explicit instructions cover translation in four directions, pure turns, stopping, corrections, out-and-back routes, and two-leg routes, with varied distances, English and Chinese phrasings, and varied background panoramas. Each goal is labeled with a bounding box and an approach distance, verified by two vision-language teachers, and augmented by rotating the panorama and its box together. The annotations span 68 unique goals from four indoor scenes. Two scenes provide the training set, and validation and test sets of 128 and 132 instructions come from the two remaining scenes, whose action wording and distances are also absent from training.

SPW-NavBench. The 100 clips come from 94 MUGEN source videos, each with an 81-frame window. Of the navigation instructions, 40 are explicit instructions and 60 are goal instructions.

Curation pipeline. For the outdoor videos, we discard MUGEN clips whose camera stays nearly static and clips that only rotate in place, resample the remaining clips to be rotation-free, and replace their captions with a neutral scene prompt, which leaves 9,609 clips. For the instruction annotations, explicit instructions are derived from camera trajectories, and their motion targets are therefore exact by construction. A goal candidate is kept only after two vision-language teachers verify it.

Data splits. The instruction annotations use disjoint scenes for training, validation, and testing. SPW-NavBench shares neither clips nor source videos with any training data of the video backbone, including the distillation subset, or with the instruction annotations.

Statistics. SPW-NavSet has 18,136 clips (9,609 outdoor, 8,527 indoor), about 233 hours at 30 FPS. Figure D1 shows sample clips.



Figure D1: Sample clips of SPW-NavSet. Each row is one clip, with frames evenly spaced from the first to the last frame. The label on the left gives the source of the clips.

E IMPLEMENTATION DETAILS

Backbone and representation. The video backbone is Wan2.2-TI2V-5B (Team Wan et al., 2025) fine-tuned on panoramic video, with LoRA of rank 256 on its linear layers. Its VAE compresses frames by $16\times$ spatially and $4\times$ temporally, giving a 64×128 latent grid. Decoding pads the latents by two columns copied from the opposite side, preserving continuity at the left and right panorama borders. Generated videos have resolution 1024×2048 and 16 FPS.

Training schedule. The video backbone is trained without language on the rotation-free videos in three stages: autoregressive training with the multi-term memory (10,000 steps), pyramid training (4,000 and 9,500 steps), and trajectory-conditioned training (28,832 steps). The student is initialized by 10,000 ODE-regression steps and distilled for 3,500 steps, 1,500 over the full noise range followed by 2,000 with noise-tail sampling, with learning rates of 2×10^{-6} (generator) and 4×10^{-7} (critic), five critic updates per generator step, and an EMA decay of 0.99.

Vision-language training and compute. The vision-language backbone is fine-tuned for three epochs with learning rate 5×10^{-5} and batch size 16 on four NVIDIA H200 GPUs.

F BASELINE DETAILS

Panoramic interfaces. NWM (Bar et al., 2025) predicts 224×224 perspective views from navigation actions and additionally receives three ground-truth past views, as required by its interface. CubeComposer (Li et al., 2026c) completes a perspective video into a panorama. Since it takes no trajectory, we repeat the first panorama as a static clip and render its perspective input along the commanded rotation. It thus receives no translation. OmniRoam (Liu et al., 2026c) takes the full camera trajectory. The Action and Causal variants of PanoWorld (Li et al., 2026b) fix the heading and receive translation only. All panoramic baselines generate 81 frames at 960×480, except CubeComposer, which generates 27 frames at 1536×3072. All are evaluated at matching timestamps.

Perspective interfaces. Stable Virtual Camera (Zhou et al., 2025), ReCamMaster (Bai et al., 2025a), CamI2V (Zheng et al., 2024), HunyuanWorld-Voyager (Huang et al., 2025a), and CameraCtrl (He et al., 2025a) receive the same 60° crop of the first panorama as the conditioning image. Each output is aligned to its actual field of view before evaluation.

Language interfaces. In the end-to-end setting, PanoWorld and OmniRoam receive the instruction appended to their text prompts together with the native motion preset matching the instruction's keywords. NWM has no text input and receives the actions planned by our vision-language backbone without fine-tuning. CubeComposer takes no language and is excluded.

G STREAMING GENERATION LOOP

Algorithm G.1 makes the state update explicit. Each instruction is interpreted once, while generation, memory update, and exact spherical rotation happen once per chunk.

Algorithm G.1 Streaming panoramic generation

Require: Initial panorama I_0, instructions $\{u_1, \ldots, u_K\}$
  $\mathcal{M} \leftarrow$ Bootstrap(I_0), $T_{\text{cur}} \leftarrow T_0$, $\bar{I} \leftarrow I_0$
  for $k = 1$ **to** K **do**
    $A_k \leftarrow$ Trajectory(Interpret$(u_k, \bar{I}), T_{\text{cur}}$)
    for each chunk c of A_k **do**
      $\mathbf{p}_c \leftarrow$ translations of chunk c relative to its reference ρ_c
      $\widetilde{V}_c \leftarrow$ PyramidGenerate$(\mathcal{M}, \mathbf{p}_c)$ with Eq. 5 at the coarsest scale
      $\mathcal{M} \leftarrow$ Update$(\mathcal{M}, \widetilde{V}_c)$, $V_c \leftarrow$ Rotate(Decode$(\widetilde{V}_c), A_k)$
    end for
    $T_{\text{cur}} \leftarrow T_{k,L}$, $\bar{I} \leftarrow$ last frame of V_c
  end for

H CORRECTNESS OF SPHERICAL ROTATION DECOUPLING

We show that spherical rotation decoupling reproduces the video of the full camera trajectory exactly, up to the interpolation of discrete resampling. We model the scene as static during a chunk and describe it by its light field $L(\mathbf{c}, \mathbf{w})$, the color seen from a camera center $\mathbf{c} \in \mathbb{R}^3$ in the world direction $\mathbf{w} \in \mathbb{S}^2$. A camera with pose $T = (R, \mathbf{t})$, where $R \in \mathrm{SO}(3)$ maps camera axes to world axes, records the panorama

$$I(\mathbf{d}) = L(\mathbf{t}, R\,\mathbf{d}), \qquad \mathbf{d} \in \mathbb{S}^2, \tag{6}$$

where $\mathbf{d}$ is the ray of a pixel in camera coordinates (Eq. 1). A panorama therefore records every world direction, and the camera orientation only decides which pixel stores which direction.

Lemma 1 (Rotation is resampling)**.** *Let two cameras share the center* $\mathbf{t}$ *and have orientations* R_a *and* R_b*. Their panoramas satisfy* $I_b(\mathbf{d}) = I_a(R\,\mathbf{d})$ *for all* $\mathbf{d} \in \mathbb{S}^2$*, with* $R = R_a^\top R_b \in \mathrm{SO}(3)$*.*

Proof. By Eq. 6, $I_b(\mathbf{d}) = L(\mathbf{t}, R_b\,\mathbf{d}) = L(\mathbf{t}, R_a R_a^\top R_b\,\mathbf{d}) = I_a(R_a^\top R_b\,\mathbf{d})$, since $R_a R_a^\top = I$. □

Since R is a bijection of $\mathbb{S}^2$ onto itself, the resampling neither creates nor removes content. A pure rotation of a panoramic camera thus requires no generation, in contrast with a perspective camera, whose field of view covers only part of the sphere.

Proposition 1 (Decoupling)**.** *Let a video* $\{I_n\}$ *be recorded along poses* $T_n = (R_n, \mathbf{t}_n)$*, and let* $\widetilde{I}_n(\mathbf{d}) = I_n(R_n^\top R_0\,\mathbf{d})$ *and* $\widetilde{\mathbf{t}}_n = R_0^\top(\mathbf{t}_n - \mathbf{t}_0)$ *as in Section 4.3. Then*

- *(i)* $\widetilde{I}_n$ *is the panorama recorded at center* $\mathbf{t}_n$ *with the fixed orientation* R_0*, so* $\{\widetilde{I}_n\}$ *is a physically valid video of the same scene along a pure translation, which is* $\widetilde{\mathbf{t}}_n$ *in the camera axes of the first frame;*
- *(ii) the original video is recovered exactly as* $I_n(\mathbf{d}) = \widetilde{I}_n(R_0^\top R_n\,\mathbf{d})$*;*
- *(iii) the rotation-free video depends on the trajectory only through* $\{\widetilde{\mathbf{t}}_n\}$*, so any two trajectories with the same translations and different rotations share the same rotation-free video.*

Proof. (i) By Eq. 6, $\widetilde{I}_n(\mathbf{d}) = L(\mathbf{t}_n, R_n R_n^\top R_0\,\mathbf{d}) = L(\mathbf{t}_n, R_0\,\mathbf{d})$, which is the panorama of a camera at $\mathbf{t}_n$ with orientation R_0. Its displacement from the first frame is $\mathbf{t}_n - \mathbf{t}_0$ in world axes, or $R_0^\top(\mathbf{t}_n - \mathbf{t}_0) = \widetilde{\mathbf{t}}_n$ in the axes of a camera with orientation R_0. (ii) Applying Lemma 1 with $R_a = R_0$ and $R_b = R_n$ at the common center $\mathbf{t}_n$ gives $I_n(\mathbf{d}) = \widetilde{I}_n(R_0^\top R_n\,\mathbf{d})$. Directly, $\widetilde{I}_n(R_0^\top R_n\,\mathbf{d}) = L(\mathbf{t}_n, R_0 R_0^\top R_n\,\mathbf{d}) = L(\mathbf{t}_n, R_n\,\mathbf{d}) = I_n(\mathbf{d})$. (iii) By (i), $\widetilde{I}_n(\mathbf{d}) = L(\mathbf{t}_n, R_0\,\mathbf{d})$ contains no R_n, and $\mathbf{t}_n = \mathbf{t}_0 + R_0\,\widetilde{\mathbf{t}}_n$ is determined by $\widetilde{\mathbf{t}}_n$ once the first pose is fixed. □

Proposition 1 justifies both halves of spherical rotation decoupling. By (i), the rotation-free training videos are genuine translation-only videos of the original scenes, so the video backbone learns real parallax rather than an artificial warp. By (ii) and (iii), a model that generates the rotation-free video for the commanded translations and then applies $R_0^\top R_n$ frame by frame produces the video of the full trajectory, and the rotation never needs to be generated.

Proposition 2 (Invariance of the pose inputs)**.** *Let* $G \in \mathrm{SE}(3)$ *be any rigid transformation of the world frame, which maps every pose* T_n *to* $G\,T_n$*. The relative pose of pose-aligned conditioning is unchanged,* $(G\,T_{\rho_k})^{-1}(G\,T_n) = T_{\rho_k}^{-1} T_n$*. For rotation-free poses, its rotation block is the identity.*

Proof. $(G\,T_{\rho_k})^{-1}(G\,T_n) = T_{\rho_k}^{-1} G^{-1} G\,T_n = T_{\rho_k}^{-1} T_n$. All rotation-free poses share the orientation R_0, so the relative rotation is $R_0^\top R_0$, the identity. □

Remark. In practice, $\mathbf{d} \mapsto R\,\mathbf{d}$ is evaluated at the pixel rays of Eq. 1 with bilinear sampling that wraps around the longitude seam, so the equalities above hold up to interpolation error. For a rotation about the vertical axis by an angle α, the resampling reduces to a circular horizontal shift $I'(\theta, \phi) = I(\theta + \alpha, \phi)$, which is exact whenever α is a multiple of the pixel width $2\pi/W$. We apply the rotation to decoded frames rather than to latents, because the video autoencoder is not equivariant to general rotations of the sphere, while generation and memory stay in the initial orientation.