WORLDEXPAND: BUILDING COHERENT 3D WORLDS VIA SPARSE JOINT PANORAMIC EXPANSION

Anonymous authors
Paper under double-blind review

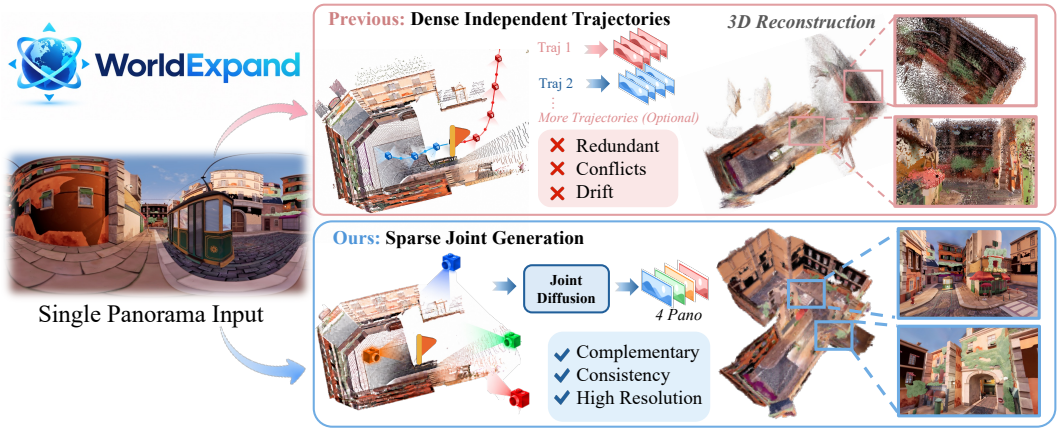

Figure 1: **WorldExpand: Sparse joint panorama generation for 3D world expansion.** Given a single panorama, previous panoramic-video methods (Yang et al., 2025) rely on independently generated dense trajectories to explore multiple directions. These trajectories contain redundant observations, may conflict in overlapping regions, and accumulate drift, making them unreliable for 3D reconstruction. In contrast, WorldExpand selects sparse, complementary viewpoints and jointly synthesizes four consistent, high-resolution panoramas, enabling more complete and detailed 3D reconstruction for world exploration.

ABSTRACT

Constructing complete 3D worlds from a single panorama is challenging because large portions of the scene remain unobserved. Existing panoramic-video methods expand spatial coverage through dense sequential generation, but neighboring frames contain substantial redundancy, long trajectories accumulate errors, and independently generated directions may be inconsistent. We present **WorldExpand**, which formulates single-panorama world expansion as the generation of sparse, complementary, and high-resolution panoramic observations. This representation concentrates the generation budget on informative scene regions and preserves the fine-grained details required for 3D construction. WorldExpand addresses two key challenges: **where to observe** and **how to synthesize**. Our *Coverage-Guided View Selection* identifies informative target locations using scene-adaptive quadrants and warp-based coverage scores, while our *Joint Panoramic Diffusion* uses Gaussian-splatting-based geometric warps to jointly synthesize four pose-grounded and cross-view-consistent panoramas. A diagnostic-guided global–local attention schedule retains global attention only in sensitive layers, reducing the training and inference costs of high-resolution joint generation. The resulting posed panoramas can be directly fed into an existing reconstruction pipeline to build an explicit 3D representation. Experiments show that WorldExpand recovers content behind occluders and around distant corners, improves both novel-view synthesis and downstream 3D reconstruction metrics across various datasets, and generalizes robustly to diverse in-the-wild scenes. Compared with the state-of-the-art panoramic-video method, WorldExpand achieves up to an $8.6\times$ speed-up with higher-resolution observations and stronger cross-view consistency.

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.

1 INTRODUCTION

Constructing explorable and geometrically coherent 3D worlds from text or a single image is a long-standing goal with broad applications in immersive content creation and virtual production (HunyuanWorld, 2025; HY-World et al., 2026; Lu et al., 2025; Zhou et al., 2024), games and embodied simulation (Gan et al., 2020; Savva et al., 2019), and spatial intelligence (Wu et al., 2026a; Liu et al., 2025b; 2026a). Video-based novel-view synthesis (NVS) methods extrapolate perspective observations from limited inputs (Zhang et al., 2026; Yu et al., 2024; Cao et al., 2025; Wan et al., 2025; Kong et al., 2024), but their narrow fields of view require multiple directional clips for full angular coverage. An initial panorama—captured directly or synthesized from text or an image by existing panorama generators (Zhang et al., 2024; Feng et al., 2025; HY-World et al., 2026)—provides full 360° angular coverage in the image domain, where scalable generative priors are readily available, but remains spatially incomplete: its single viewpoint cannot reveal surfaces behind occluders, around corners, or beyond doorways.

Expanding beyond this fixed viewpoint requires moving the observation center and synthesizing panoramas at new locations. Panoramic video generation (Lu et al., 2025; Yin et al., 2025; Wang et al., 2024a; Yang et al., 2025) offers a natural route, but existing approaches inherit fundamental limitations from dense sequential generation. In particular, they suffer from two limitations. (i) *Redundant information representation*: densely sampled consecutive frames encode largely overlapping scene information and thus provide redundant multi-view evidence for reconstruction. (ii) *Limited and inconsistent exploration*: a single rollout explores primarily along one direction, independently generated rollouts toward multiple directions may disagree in overlapping regions, and errors accumulate over long trajectories, resulting in inconsistent multi-view observations for 3D reconstruction. Consequently, directly adopting the output of video generation as the intermediate representation is not well aligned with the goal of 3D world construction. This raises a fundamental question: **What representation can capture maximal information at minimal generation cost while being tailored to 3D world construction?**

To address this question, we introduce **WorldExpand**, a framework that formulates single-panorama world expansion as the generation of a sparse set of high-resolution panoramic observations at novel locations. Each panorama captures the full angular context at one location, while a few displaced panoramas provide complementary spatial coverage. Compared with dense video, this representation avoids spending computation on redundant neighboring frames and instead uses the saved budget to increase observation resolution, which is crucial for preserving the fine-grained structures required for detailed 3D reconstruction.

Given this sparse panoramic representation, two key challenges remain. (1) **Where should the new observations be placed?** Fixed target directions ignore scene layout and may waste the limited view budget on blocked or redundant regions. Our *Coverage-Guided View Selection* adapts the quadrant orientation to scene boundaries and selects an informative target in each region. (2) **How can we jointly generate consistent high-resolution panoramas?** Our *Joint Panoramic Diffusion* uses Gaussian-splatting warps as geometric anchors and jointly synthesizes all four target panoramas to maintain cross-view consistency. To further eliminate unnecessary computation, we introduce a diagnostic-guided global–local attention schedule. Our diagnostic reveals that not all layers require global attention: sensitive layers retain cross-view communication, while the remaining layers use local attention, substantially reducing the training and inference costs of high-resolution joint generation. The resulting posed panoramas can be directly consumed by an existing 3D reconstruction pipeline (HY-World et al., 2026) to build the expanded world.

Experiments show that WorldExpand effectively expands scenes by generating content behind occluders and around distant corners. Across various datasets, it outperforms prior panoramic-video methods on both novel-view synthesis and downstream 3D reconstruction metrics, while generalizing robustly to diverse in-the-wild scenes. Owing to its efficient sparse representation, WorldExpand achieves up to an $8.6\times$ speed-up over the state-of-the-art method (Yang et al., 2025), while generating higher-resolution observations with stronger cross-view consistency. In practical applications, WorldExpand provides a useful building block for large-scale roaming, games, virtual environments, and embodied AI simulation.

2 Related Work

Novel-View Synthesis. Diffusion-based image NVS methods (Liu et al., 2023; Höllein et al., 2024; Shi et al., 2024) hallucinate novel views via pose conditioning or multi-view denoising, while camera-controllable video diffusion models (Voleti et al., 2024; Wang et al., 2024b; He et al., 2024; Sun et al., 2025; Gao et al., 2024; Wang et al., 2026d; Wu et al., 2025b; Li et al., 2026c) generate view-consistent sequences along specified trajectories; works that further couple multi-view generation with downstream 3D representations (Yu et al., 2024; Huang et al., 2025a; Cao et al., 2025; Zhang et al., 2026) show that generated views can be lifted into explicit 3D representations and motivate our use of point-cloud warping and joint multi-view denoising. However, their dense narrow-FoV sequences are redundant and prone to error accumulation. We instead generate a compact set of strategically placed *full-view (360°) panoramas* that provides complementary coverage and high-resolution observations for downstream reconstruction.

Panorama Generation. Panoramic image generation methods adapt 2D diffusion priors to the 360° domain via panorama–perspective coupling (Zhang et al., 2024; Feng et al., 2025; HY-World et al., 2026), cube-face synchronization (Huang et al., 2025b; Kalischek et al., 2025), sphere-aware attention (Ye et al., 2024), or fused overlapping diffusion paths (Bar-Tal et al., 2023). Panoramic video generation either lifts a perspective video to 360° playback (Li et al., 2026a; Luo et al., 2025; Wu et al., 2026b; Tan et al., 2026) or synthesizes panoramic video from text or images for explorable scenes and long-horizon roaming (Wang et al., 2024a; Xing et al., 2025; Lu et al., 2025; Yang et al., 2025; Yin et al., 2025; Li et al., 2025; Xia et al., 2025; Liu et al., 2026b). However, image-level methods produce only a single panorama from text or layout, while long video sequences accumulate appearance drift and geometric inconsistency, generate overlapping regions across roaming directions independently, and lack a co-designed reconstruction component. We instead jointly synthesize a compact set of mutually consistent *novel-view panoramas* that are simultaneously faithful to a source panorama and to specified target camera poses, and feed them into a tightly integrated reconstruction module.

3D Scene Generation. Recent 3D scene generation systems differ in the visual observations used for reconstruction. Panorama-centric methods generate a single 360° panorama or world proxy, providing full angular context but remaining tied to one observation center (Zhou et al., 2024; HunyuanWorld, 2025). To expand coverage, other methods generate camera-controlled videos or iterative perspective views before fusing them into 3D (Schneider et al., 2025; HY-World et al., 2026; Ainetter et al., 2026; Chung et al., 2023; Erkoç et al., 2026). However, covering a scene with these approaches requires generating a large number of video frames or perspective images, resulting in a redundant and inefficient intermediate representation. In contrast, WorldExpand uses a compact set of sparse panoramic observations: it selects complementary locations and jointly generates their high-resolution panoramas, retaining full angular context at each location while enforcing cross-view consistency for downstream reconstruction.

3 WorldExpand

As discussed above, dense panoramic video is an inefficient representation for 3D world construction: neighboring frames contain largely redundant observations, while long or independently generated trajectories can introduce cross-view inconsistencies. Given a single panorama I_0 and its depth D_0, estimated with DepthMaster (Wang et al., 2026b) for in-the-wild inputs, WorldExpand instead expands it into a coherent 3D world through a sparse panoramic representation designed around the needs of reconstruction (Figure 2). It first selects a limited set of informative viewpoints to maximize complementary scene coverage (Section 3.1), then jointly synthesizes their panoramas while maintaining pose grounding and cross-view consistency (Section 3.2), and finally fuses the resulting observations into a coherent 3D representation (Section 3.3).

3.1 Coverage-Guided View Selection

Why Coverage-Guided Selection? At inference time, we back-project the source panorama I_0 and its estimated depth D_0 into a point cloud $\mathcal{P}_0$. Although I_0 covers all viewing directions, its single observation center leaves large regions occluded. Random or naive target placement ignores scene

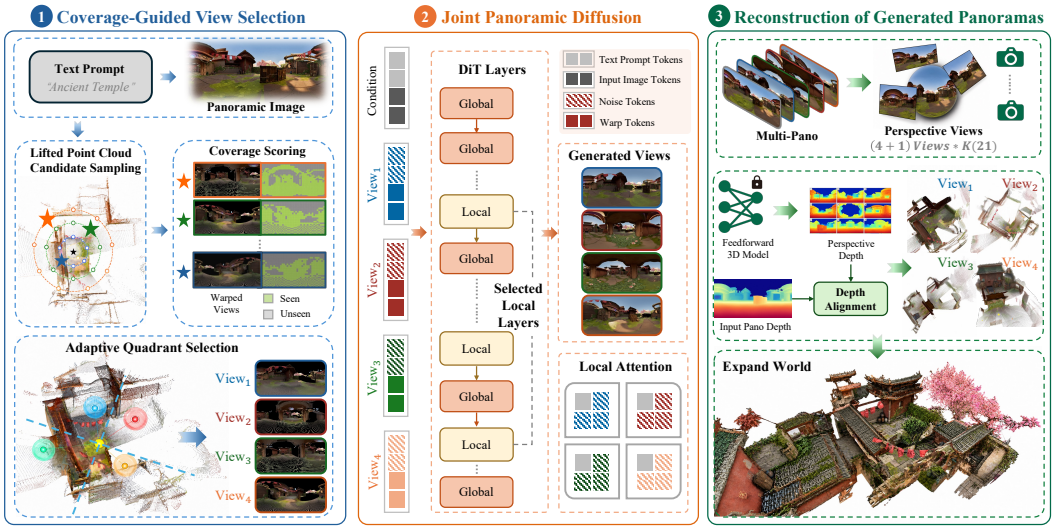

Figure 2: **Overview of WorldExpand.** Given one panorama, we select four viewpoints that expose complementary unseen regions, jointly synthesize their panoramic observations with selective cross-view communication, and align multi-view depths to reconstruct an expanded Gaussian scene.

layout and source visibility, and may therefore select blocked, redundant, or highly overlapping views that fail to reveal these unobserved regions. Given a limited budget of $N{=}4$ target views, we instead select spatially distributed poses according to their expected coverage of newly exposed content.

Candidate Sampling and Coverage Scoring. We extract the scene boundary from a horizontal mesh slice at camera height and sample candidate cameras on concentric rings around the source. Candidates that collide with scene geometry or lie beyond the boundary are removed. We then warp $\mathcal{P}_0$ into each candidate view v using Gaussian splatting. Pixels receiving projected source points form the seen region $\mathcal{S}_v$, whereas pixels receiving no points form the unseen region $\mathcal{U}_v$. The coverage score $c_v = |\mathcal{U}_v|$ measures how much new content candidate v can reveal and is used for the adaptive quadrant selection below.

Adaptive Quadrant Selection. To prevent the four targets from clustering in one open direction, we partition the azimuth into four 90° quadrants and select one target per quadrant. Since fixed axis-aligned quadrants ignore scene layout, we rotate the dividing cross to align its boundaries with the nearest walls:

$$\theta^\star = \arg\min_{\theta\in[0,\pi/2)} \sum_{k=0}^{3} d_{\mathrm{bnd}}\big(\theta + k\tfrac{\pi}{2}\big)\,, \tag{1}$$

where $d_{\mathrm{bnd}}(\phi)$ is the distance from the source to the scene boundary along azimuth ϕ. This places low-information wall directions on quadrant boundaries and open regions inside the quadrants. From the candidates $Q_i(\theta^\star)$ in each quadrant, we choose

$$v_i^\star = \arg\max_{v\in Q_i(\theta^\star)} c_v, \qquad i = 1,\dots,4. \tag{2}$$

Each selected candidate $v_i^\star$ specifies a target camera pose $\mathbf{T}_i$. Relative to the source pose $\mathbf{T}_0$, these target poses define the transformations $\{\Delta\mathbf{T}_{0\to i}\}_{i=1}^N$ used by the next stage to render full-resolution geometric conditions (Section 3.2).

3.2 Joint Panoramic Diffusion

Given the source panorama I_0 and the selected target relative poses $\{\Delta\mathbf{T}_{0\to i}\}_{i=1}^N$ from Section 3.1, our goal is to jointly synthesize the corresponding novel-view panoramas $\{\hat{I}_i\}_{i=1}^N$ using a flow-based image diffusion model (Wu et al., 2025a) as the generative backbone. Our main designs are geometry-aware multi-stream conditioning for precise pose grounding and a diagnostic-guided global–local attention schedule for efficient cross-view consistency.

Geometry-Aware Multi-Stream Conditioning. Text- or pose-only conditioning lacks spatial anchors, depth-as-color maps are sensitive to distribution shifts, and mesh warping stretches surfaces across depth discontinuities. We instead adopt Gaussian splatting to render geometric warps. Leveraging the accurate poses and depth provided by our synthetic dataset (Supplementary Section C), we unproject the source RGB-D panorama I_0 and D_0 into a colored point cloud $\mathcal{P}_0$, treat each point as an isotropic Gaussian splat, and render the resulting representation from each target pose:

$$(W_i, M_i) = \text{Render}_{\text{ERP}}(\Delta\mathbf{T}_{0\to i}\,\mathcal{P}_0), \quad i = 1, \ldots, N. \tag{3}$$

The warped image W_i provides pixel-level correspondences from the source panorama to the target view, while the mask M_i marks disoccluded regions that lack source observations and must be hallucinated. This conditioning directly translates the pose specification into a spatially explicit generation guide: visible regions are anchored by the warp, and the model's generative capacity is focused on the masked areas. In our N-view generator (N=4 in practice; see Supplementary Section E), we spatially concatenate all image tokens into one global sequence $\mathbf{H}^{global}$, which contains three semantic streams:

$$\mathbf{H}^{global} = [\mathbf{P}_{txt}; \mathbf{S}_0; \mathbf{Z}_1; \ldots; \mathbf{Z}_N; \mathbf{W}_1; \ldots; \mathbf{W}_N], \tag{4}$$

where $\mathbf{P}_{txt}$ indicates textual tokens while $\mathbf{S}_0$ is the source panorama token sequence; $\{\mathbf{Z}_i\}_{i=1}^N$ are noisy target-view tokens; $\{\mathbf{W}_i\}_{i=1}^N = \{\text{VAE}(W_i) + \phi(M_i)\}_{i=1}^N$ denote warped tokens, formed by channel-wise addition of the VAE-encoded warped image and a convolutional mask feature $\phi(M_i)$ that encodes per-pixel visibility information. To bridge the gap between clean training warps and estimated inference geometry, we independently perturb each target-view warp during training; details are provided in Supplementary Section G.

Diagnostic-Guided Global–Local Attention Schedule. Our global attention over $\mathbf{H}^{global}$ promotes cross-view consistency, but using it at every layer incurs unnecessary computation that grows quadratically with the number of views, especially at high resolution. Since layers serve distinct roles—some align views while others refine local texture—we retain global attention only where cross-view communication is essential and use efficient per-view local attention $\mathbf{H}_i^{local}$=$[\mathbf{P}_{txt}; \mathbf{Z}_i]$ elsewhere.

To resolve this, we perform a per-layer diagnostic on a fully global-attention model: each layer is individually switched to local attention without finetuning, and the ΔPSNR is measured against the full-global reference. As shown in Figure 3, the sensitivity is clearly heterogeneous—localizing certain layers causes severe degradation while many others incur negligible loss. We select the 30 layers with minimal degradation as local layers and finetune the resulting model, achieving comparable quality to the full-global baseline with significantly reduced memory and latency.

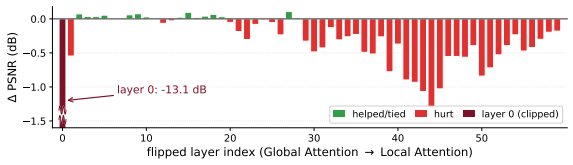

Figure 3: **Per-layer diagnostic for the global–local attention schedule.** Without finetuning, each bar reports the ΔPSNR when one transformer layer is switched from global to local attention at inference. We use these sensitivity scores to determine which 30 layers are switched to local attention, prioritizing the least-sensitive ones.

3.3 Reconstruction of Generated Panoramas

We reconstruct the scene using the WorldMirror 2.0 and 3DGS composition pipeline of HY-World 2.0 (HY-World et al., 2026), converting the source and generated posed panoramas into a unified Gaussian representation. Complete details of perspective decomposition, depth prediction and alignment, Gaussian initialization, and optimization are provided in Supplementary Section A.1.

4 Experiments

4.1 Datasets

We evaluate on two complementary panoramic RGB-D benchmarks and a diverse set of in-the-wild panoramas. The RGB-D benchmarks comprise our UE dataset for measuring in-domain fidelity and

the synthetic subset of RealSee3D for evaluating cross-dataset generalization. On each benchmark, we sample 64 anchor test cases and expand every anchor into 4 target views, yielding 256 generated panoramas for image- and geometry-level quantitative evaluation. We further evaluate on diverse in-the-wild panoramas to assess practical generalization when ground-truth depth is unavailable; in this setting, the source depth is estimated using DepthMaster (Wang et al., 2026b).

UE Dataset. Our primary benchmark is a synthetic panoramic RGB-D dataset built in Unreal Engine (Supplementary Section C). It contains diverse indoor and outdoor environments with accurate camera poses and dense metric depth, supporting both point-cloud warp conditioning and ground-truth 3D evaluation.

RealSee3D. To assess generalization to a distinct scene distribution, we use the synthetic subset of the public RealSee3D benchmark (Li et al., 2026b), which provides indoor multi-view panoramic RGB-D data with rendered camera poses and metric depth. This benchmark is held out from training and differs from our Unreal Engine training scenes.

4.2 Evaluation Metrics

We evaluate along two axes. For **per-view image quality**, we report PSNR, SSIM, and LPIPS against the ground-truth panorama at each target pose. For **3D reconstruction quality**, we measure (i) *camera-pose accuracy*: the poses of the four target panoramas are estimated with the off-the-shelf VGGT-Ω (Wang et al., 2026a) (each panorama sliced into perspective views) and compared to the ground truth via pose AUC at 3°/30°; and (ii) *point-cloud geometry*: for every method we reconstruct the point cloud with the *same* Panorama-Aware 3D Reconstruction module (Section 3.3) to ensure a fair comparison, and compare it to the ground-truth point cloud—obtained by unprojecting and merging the ground-truth RGB-D of the source (center) view and the four novel views—reporting Accuracy and Completeness (per-scene mean/median) and F-score@5cm. Full training and reconstruction implementation details are provided in Supplementary Section A.

4.3 Experimental Results

To our knowledge, no prior work shares the same setting of expanding a scene from a single panorama via multi-view panoramas. We therefore compare against the closest line of work—panoramic video generation—and select GenEx (Lu et al., 2025) and Matrix-3D (Yang et al., 2025) as representative baselines. Notably, both GenEx and Matrix-3D are trained on synthetic scenes rendered in Unreal Engine, so evaluation on held-out UE scenes does not give our method exclusive access to the synthetic-engine domain. Since our multi-view panoramas may correspond to target locations along arbitrary directions, requiring a single panoramic video to cover all of them is unrealistic. For a fair comparison, we instead run each baseline four times, generating panoramic videos along the four target directions independently, and extract the keyframes corresponding to our target viewpoints for evaluation against the ground truth.

Comparison on UE Synthetic. On our in-domain UE synthetic dataset, our method outperforms both baselines on all NVS metrics (Table 1); the complete per-view results in Supplementary Table 14 further show that this advantage holds across every target view. The PSNR/SSIM gains show our panoramas stay pixel- and structure-level faithful to the source-grounded reference, while the lower LPIPS reflects fewer hallucinated, cross-run-inconsistent structures that would otherwise corrupt downstream 3D reconstruction. This advantage carries over to 3D: because the baselines drift from the source geometry and disagree across independently generated views, both their estimated poses and reconstructed point clouds degrade, whereas our consistency-aware panoramas remain source-grounded and cross-view coherent.

Comparison on RealSee3D. We further evaluate on the synthetic subset of the public RealSee3D benchmark to probe cross-dataset generalization. Crucially, *none* of the compared models—neither the baselines nor ours—is trained or fine-tuned on RealSee3D; every method is applied in a purely zero-shot manner. This makes RealSee3D a fair, held-out testbed that measures generalization ability alone, isolating it from any advantage of in-domain training. Under the same protocol, the NVS and 3D reconstruction metrics in Table 1 follow the same trend as on our UE synthetic benchmark: our method improves both image fidelity and reconstructed geometry, showing that the consistency-

Table 1: **Novel-view synthesis and 3D reconstruction.** NVS metrics are averaged over the four target views. AUC and F@5 cm are percentages; Accuracy and Completeness are reported in meters as mean/median. Best per dataset is in **bold**.

		Novel-view synthesis			3D reconstruction				
Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	AUC@3° ↑	AUC@30° ↑	Acc.↓	Comp.↓	F@5 cm↑
UE synthetic	GenEx	15.18	0.461	0.468	4.58	20.96	0.86/0.46	3.70/0.55	12.95
	Matrix-3D	17.30	0.502	0.363	29.38	83.96	0.80/0.34	**2.50**/0.24	20.32
	Ours	**20.16**	**0.658**	**0.256**	**49.79**	**89.88**	**0.79/0.19**	2.87/**0.09**	**29.02**
RealSee3D	GenEx	15.85	0.644	0.455	0.42	19.04	0.21/0.14	0.31/**0.15**	24.42
	Matrix-3D	17.44	0.651	0.391	8.33	67.46	0.12/0.08	**0.30**/0.17	35.04
	Ours	**18.06**	**0.711**	**0.350**	**25.00**	**81.42**	**0.10/0.06**	**0.30**/0.17	**41.44**

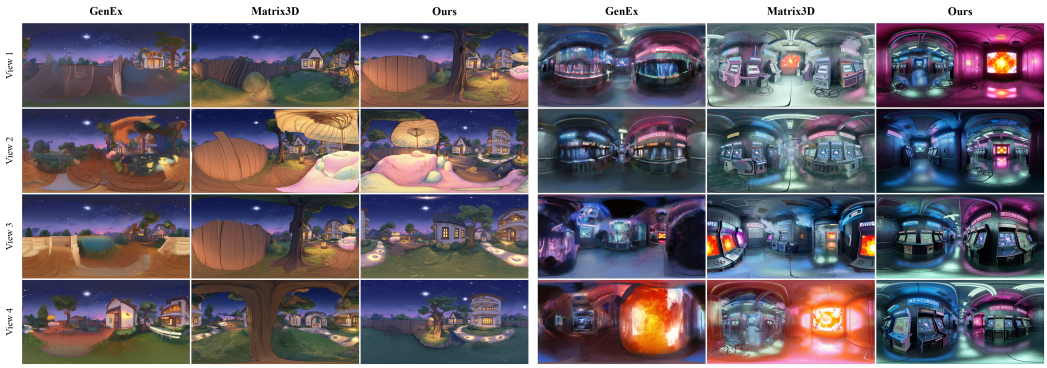

Figure 4: **Qualitative comparison on in-the-wild panoramas.** A single video trajectory cannot reach all four directional target viewpoints, so GenEx (Lu et al., 2025) and Matrix-3D (Yang et al., 2025) each generate four independent clips. Their results exhibit ERP distortions, content collapse on diverse wild inputs, and abnormal overexposure, whereas WorldExpand preserves coherent structure and geometric cues across different visual styles. Additional comparisons and analysis are provided in Supplementary Sections B, M and N and the supplementary video.

aware design transfers beyond the training distribution. Additional qualitative comparisons are provided in Supplementary Section N.

Qualitative Comparison and Reconstruction Analysis. Figure 4 evaluates in-the-wild inputs with diverse scene styles. Because the video baselines require a separate trajectory for each direction, their four clips are generated independently and can suffer ERP distortion, severe appearance drift, content collapse, or unstable exposure. WorldExpand instead generates the four target panoramas jointly and remains robust across these styles while preserving clearer structure and geometric cues. The downstream point-cloud comparison in Figure 5 further shows that these consistent observations produce stable, well-defined 3D structures, whereas Matrix-3D exhibits local collapse and distorted geometry. Controlled comparisons on UE synthetic and RealSee3D, together with further system-level analysis, are provided in Supplementary Sections B, M and N.

4.4 Ablation Study

We ablate the key components of WorldExpand to assess their contributions to view coverage, cross-view consistency, generation quality, and efficiency. We summarize the main findings below and provide complete quantitative results and additional analyses in the Supplementary.

Diagnostic-Guided Global–Local Attention Schedule. Prior panoramic-video methods allocate their computational budget to densely sampled, highly redundant frames, leaving limited memory for high spatial resolution. Our sparse panoramic keyframes instead concentrate this budget on a few high-resolution observations, yielding up to an $8.6\times$ speed-up for four-direction generation over the panoramic-video baseline; detailed system-level timings are provided in Supplementary Section B. However, joint 2K synthesis remains expensive if every transformer layer performs full cross-view

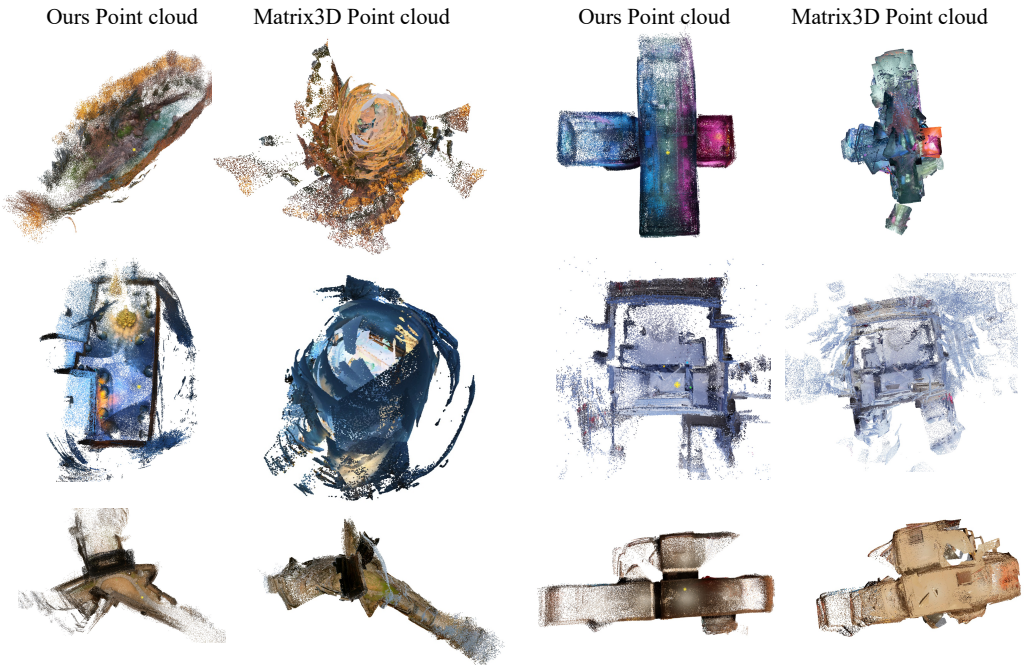

Figure 5: **Additional in-the-wild 3D reconstruction comparisons.** Across diverse panoramic inputs, WorldExpand produces stable, well-defined point-cloud structures. In contrast, Matrix-3D exhibits local point-cloud collapse and distorted geometric details, reflecting inconsistencies in its generated observations.

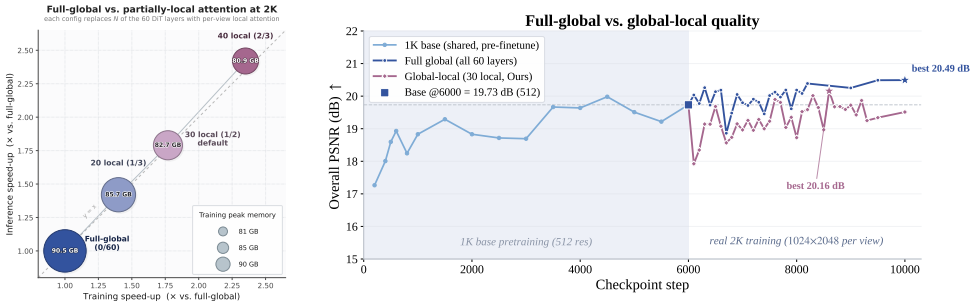

Figure 6: **Global–local attention is an efficiency–quality trade-off.** *Left*: training vs. inference speed-up over the full-global baseline at 2K; bubble size is the training peak memory (localizing more layers is faster and lighter). **Our default 30-local schedule reaches nearly 2× speed-up in both training and inference (1.77×/1.79×) while cutting peak memory.** *Right*: PSNR vs. checkpoint step for full-global vs. our default 30-local schedule—the two track within ∼0.33 dB. Full analysis in Supplementary Section I.

attention. We therefore ablate how the number of diagnostic-selected local layers affects training and inference speed, memory, and generation quality. As shown in Figure 6, localizing more layers steadily reduces cost with only a modest quality trade-off; using 30 of 60 layers as local provides a balanced operating point. Supplementary Sections I to K provide the complete layer-count analysis and selection controls.

Joint Four-View Generation. In *w/o four-view*, we train a single-view variant that accepts only one target warp at a time and independently denoises the four target panoramas without cross-view attention. Qualitatively, joint generation produces more consistent appearance and structure across overlapping regions, as shown in Figure 7; additional examples are provided in Supplementary Figure 14. Quantitatively, it raises the four-view average PSNR from 18.27 to 20.16 dB (+1.89 dB)

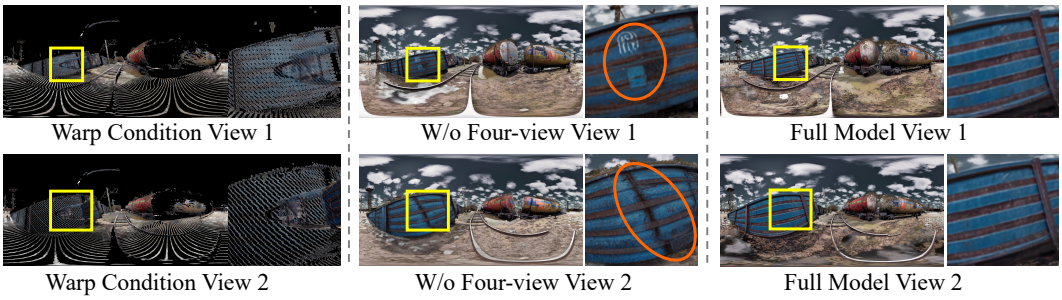

Figure 7: **Joint four-view generation improves consistency.** Independently generated views disagree in an overlapping structure (orange ellipses), whereas joint generation preserves consistent content across the two views.

Table 2: **Point-cloud warp condition rendered by Gaussian splatting.** *Left*: render cost for the $N{=}4$ target views at 1K/2K per-view resolution (splatting is much cheaper and point-count independent). *Right*: synthesis quality of splatting vs. plain point rendering (same point cloud, identical protocol). Full mesh and depth-as-color comparison in Supplementary Section F. Best in **bold**.

Render engine	1K	2K
Gaussian splatting (ours)	**0.72 s**	**0.89 s**
Point cloud (1M pts)	4.21 s	4.51 s
Mesh	60 s	∼73 s

Renderer	PSNR↑	SSIM↑	LPIPS↓
Point render	19.40	0.593	**0.252**
Splatting (ours)	**20.16**	**0.658**	0.256

and SSIM from 0.560 to 0.658 ($+0.098$), while reducing LPIPS from 0.284 to 0.256 (-0.028; Supplementary Section H and Table 9). Because the ground-truth panoramas are mutually consistent across viewpoints, these gains provide indirect evidence that the generated set more faithfully preserves shared scene content.

Geometric Conditioning Design. We condition on a colored point cloud rendered by Gaussian splatting. Splatting is $\sim$5–6$\times$ faster than plain point rendering and orders of magnitude faster than meshing (Table 2, left), and its cleaner warp—especially where the point cloud is locally sparse—also yields better synthesis quality than plain point rendering ($+0.76$ dB PSNR; Table 2, right). We prefer a point cloud over a reconstructed mesh, which stretches "rubber-sheet" triangles across depth discontinuities and cannot be augmented, and over depth-as-color, which loses foreground contrast under the large outdoor depth range. The full comparison, including qualitative artifacts, is in Supplementary Section F.

5 Conclusion

We presented WorldExpand, a framework that formulates single-panorama world expansion through a sparse set of complementary, high-resolution panoramic observations. This reconstruction-oriented representation avoids the redundancy of dense panoramic video while retaining full angular context at each selected location. To realize it, Coverage-Guided View Selection places a limited number of observations according to scene layout, while Joint Panoramic Diffusion uses Gaussian-splatting warps and selective cross-view communication to jointly generate four pose-grounded, consistent panoramas. The resulting observations can be directly integrated by an existing reconstruction pipeline to form an explicit 3D representation. Experiments across multiple benchmarks demonstrate improvements in both novel-view synthesis and downstream 3D reconstruction metrics, together with robust generalization to diverse in-the-wild scenes. WorldExpand recovers content behind occluders and around distant corners, produces higher-resolution and more consistent observations, and achieves up to an $8.6\times$ speed-up over the state-of-the-art panoramic-video method. Overall, WorldExpand provides a compact and efficient representation for constructing more complete and coherent 3D worlds from a single panorama.

AI USE STATEMENT

We used generative AI tools to assist with language editing and manuscript formatting. All AI-assisted text and formatting changes were reviewed by the authors, who verified the technical claims and take responsibility for the final content.

REFERENCES

Stefan Ainetter, Thomas Deixelberger, Edoardo A Dominici, Philipp Drescher, Konstantinos Vardis, and Markus Steinberger. Scene generation at absolute scale: Utilizing semantic and geometric guidance from text for accurate and interpretable 3d indoor scene generation. In *European Conference on Computer Vision*, pp. 132–150. Springer, 2026.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

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

Ziya Erkoç, Angela Dai, and Matthias Nießner. Worldagents: Can foundation image models be agents for 3d world models? In *European Conference on Computer Vision*, pp. 474–491. Springer, 2026.

Haoran Feng, Dizhe Zhang, Xiangtai Li, Bo Du, and Lu Qi. Dit360: High-fidelity panoramic image generation via hybrid training. *arXiv preprint arXiv:2510.11712*, 2025.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

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

Lukas Höllein, Aljaž Božič, Norman Müller, David Novotny, Hung-Yu Tseng, Christian Richardt, Michael Zollhöfer, and Matthias Nießner. Viewdiff: 3d-consistent image generation with text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Tianyu Huang, Guandong Zhengwang, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2025a.

Yukun Huang, Yanning Zhou, Jianan Wang, Kaiyi Huang, and Xihui Liu. Dreamcube: 3d panorama generation via multi-plane synchronization. *arXiv preprint arXiv:2506.17206*, 2025b.

Team HunyuanWorld. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint*, 2025.

Team HY-World, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026.

Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. Cubediff: Repurposing diffusion-based image models for panorama generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Beomyoung Kim, Chanyong Shin, Joonhyun Jeong, Hyungsik Jung, Se-Yun Lee, Sewhan Chun, Dong-Hyun Hwang, and Joonsang Yu. Zim: Zero-shot image matting for anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23828–23838, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

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*, 2026a.

Linyuan Li, Yan Wu, Xi Li, Lingli Wang, Tong Rao, Jie Zhou, Cihui Pan, and Xinchen Hui. Realsee3d: A large-scale multi-view rgb-d dataset of indoor scenes (version 1.1), 2026b. URL https://doi.org/10.5281/zenodo.17826242.

Renjie Li, Panwang Pan, Bangbang Yang, Dejia Xu, Shijie Zhou, Zeming Li, Achuta Kadambi, Zhangyang Wang, Zhengzhong Tu, Zhiwen Fan, et al. 4k4dgen: Panoramic 4d generation at 4k resolution. In *International Conference on Learning Representations*, volume 2025, pp. 100692–100713, 2025.

Xinyang Li, Tengfei Wang, Zixiao Gu, Shengchuan Zhang, Chunchao Guo, and Liujuan Cao. Flashworld: High-quality 3d scene generation within seconds. In *ICLR*, 2026c.

Fangfu Liu, Diankun Wu, Jiawei Chi, Yimo Cai, Yi-Hsin Hung, Xumin Yu, Hao Li, Han Hu, Yongming Rao, and Yueqi Duan. Spatial-ttt: Streaming visual-based spatial intelligence with test-time training. *arXiv preprint arXiv:2603.12255*, 2026a.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025a.

Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv preprint arXiv:2501.10074*, 2025b.

Yuheng Liu, Xin Lin, Xinke Li, Baihan Yang, Chen Wang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Hao Tan, Kai Zhang, Xiaohui Xie, et al. Omniroam: World wandering via long-horizon panoramic video generation. *arXiv preprint arXiv:2603.30045*, 2026b.

Taiming Lu, Tianmin Shu, Alan Yuille, Daniel Khashabi, and Jieneng Chen. Genex: Generating an explorable world. In *International Conference on Learning Representations*, volume 2025, pp. 52310–52335, 2025.

Rundong Luo, Matthew Wallingford, Ali Fahardi, Noah Snavely, and Wei-Chiu Ma. Beyond the frame: Generating 360 panoramic videos from perspective videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14336–14345. IEEE, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

Manuel-Andreas Schneider, Lukas Höllein, and Matthias Nießner. Worldexplorer: Towards generating fully navigable 3d scenes. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International conference on learning representations*, volume 2024, pp. 39838–39859, 2024.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Jing Tan, Shuai Yang, Tong Wu, Jingwen He, Yuwei Guo, Ziwei Liu, and Dahua Lin. Imagine360: Immersive 360 video generation from perspective anchor. *Advances in Neural Information Processing Systems*, 38:139856–139874, 2026.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pp. 439–457. Springer, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Pengfei Wang, Shihao Wang, Liyi Chen, Zhiyuan Ma, Guowen Zhang, and Lei Zhang. Depthmaster: Unified monocular depth estimation for perspective and panoramic images. *arXiv preprint arXiv:2606.12368*, 2026b.

Qian Wang, Weiqi Li, Chong Mou, Xinhua Cheng, and Jian Zhang. 360dvd: Controllable panorama video generation with 360-degree video diffusion model. *arXiv preprint arXiv:2401.06578*, 2024a.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026c.

Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, Chunchao Guo, and Zhou Zhao. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026d.

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

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in Neural Information Processing Systems*, 38: 13569–13597, 2026a.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26057–26068, 2025b.

Ziyi Wu, Daniel Watson, Andrea Tagliasacchi, David J Fleet, Marcus A Brubaker, and Saurabh Saxena. 360anything: Geometry-free lifting of images and videos to 360 {\deg}. *arXiv preprint arXiv:2601.16192*, 2026b.

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 {\deg} with latitude/longitude-aware mechanisms. *arXiv preprint arXiv:2505.22016*, 2025.

Ke Xing, Hanwen Liang, Dejia Xu, Yuyang Yin, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Tip4gen: Text to immersive panorama 4d scene generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 9267–9276, 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.

Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

Weicai Ye, Chenhao Ji, Zheng Chen, Junyao Gao, Xiaoshui Huang, Song-Hai Zhang, Wanli Ouyang, Tong He, Cairong Zhao, and Guofeng Zhang. Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. *arXiv preprint*, 2024.

Yuyang Yin, HaoXiang 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. *arXiv preprint arXiv:2509.24997*, 2025.

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

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. Worldstereo: Bridging camera-guided video generation and scene reconstruction via 3d geometric memories. *arXiv preprint arXiv:2603.02049*, 2026.

Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*, pp. 324–342. Springer, 2024.

APPENDIX OVERVIEW

This supplementary material is organized as follows:

- Section A: training and reconstruction implementation details. . . . . p. 14
- Section A.1: perspective decomposition, known-camera depth alignment, Gaussian initialization, and optimization details. . . . . p. 14
- Section B: system-level comparison with Matrix-3D. . . . . p. 16
- Section C: details of scene collection, roaming-trajectory generation, and intersection-centric anchor–target sampling for the synthetic dataset. . . . . p. 16
- Section D: comparison of view-selection strategies by reconstructable scene coverage.p. 18
- Section E: why $N{=}4$ target views are enough. . . . . p. 20
- Section F: discussion of geometric condition modalities (point-cloud warp vs. mesh vs. depth-as-color) and why we render the point cloud with Gaussian splatting. . . . . p. 21
- Section G: warp-side augmentation as an in-domain-accuracy vs. robustness trade-off.p. 23
- Section H: quantitative and qualitative ablation on the joint four-view generation design. p. 24
- Section I: trade-off between the number of local-attention layers and the training memory/speed. . . . . p. 25
- Section J: analysis showing that the global–local layer selection is stable across training checkpoints. . . . . p. 27
- Section K: control experiment showing that *which* layers are localized (least- vs most-sensitive) is what matters, not merely how many. . . . . p. 28
- Section L: full per-view NVS results on UE synthetic and RealSee3D. . . . . p. 30
- Section M: qualitative comparisons on held-out UE synthetic data. . . . . p. 30
- Section N: additional qualitative novel-view synthesis comparisons on the RealSee3D dataset. . . . . p. 31

A IMPLEMENTATION DETAILS

Setup. We build our novel-view panorama generator on a pretrained image diffusion model (Wu et al., 2025a) and train it in two stages for joint multi-view panoramic synthesis, with each training and inference step jointly producing four novel-view panoramas. We first train at 1K resolution (512×1024 per view) with full global attention for 6,000 steps, which teaches the model to follow the point-cloud warp condition and perform novel-view synthesis; we then fine-tune for 2,000 steps at 2K resolution (1024×2048 per view) with the global–local attention schedule that switches 30 of the 60 DiT layers to per-view local attention. Thus, all reported main-model trajectories switch from 1K to 2K at step 6,000. Both stages use 8 GPUs with a per-GPU batch size of 1. For the feed-forward reconstruction stage, we adopt WorldMirror2 (HY-World et al., 2026) as the perspective backbone. Each input panorama is sliced into 21 perspective views that uniformly cover the 360° field of view, which are fed into the feed-forward model to obtain globally consistent depth predictions. After known-camera calibration and RANSAC-based scale alignment, the resulting point cloud is used to initialize a Gaussian Splatting (Kerbl et al., 2023) representation, and the final scene is optimized using a gsplat-based (Ye et al., 2025) implementation.

A.1 RECONSTRUCTION DETAILS

We follow the WorldMirror 2.0 and 3DGS composition pipeline of HY-World 2.0 (HY-World et al., 2026) to convert the generated posed panoramas into an explicit 3D representation. The details below are included for completeness; reconstruction is not a contribution of our method.

Perspective Decomposition and Depth Prediction. Let the full panorama set be $\mathcal{I} = \{I_0\} \cup \{\hat{I}_i\}_{i=1}^{N}$ with known camera poses $\{\mathbf{T}_i\}_{i=0}^{N}$. For each panorama, we sample $K{=}21$ tangent cameras

that uniformly cover the sphere and render dense perspective crops. We jointly feed all crops into WorldMirror 2.0 (Liu et al., 2025a), which performs cross-view reasoning and predicts per-crop depths $\hat{D}_{i,k}$, where $k \in \{1, \ldots, K\}$. We discard its predicted cameras and retain the globally coherent depth predictions because the panorama poses are already known.

Known-Camera Depth Alignment and Gaussian Initialization. Each tangent view has a known camera pose $\mathbf{T}_{i,k}$, obtained by composing panorama pose $\mathbf{T}_i$ with the k-th tangent-camera rotation. Let $\Omega_{i,k}^{\text{align}}$ denote pixels with valid correspondences between predicted depth $\hat{D}_{i,k}$ and anchor depth $D_{i,k}^{\text{anchor}}$. For the source panorama (i=0), the anchor is obtained by projecting its depth D_0 into the tangent view; for generated panoramas ($i \geq 1$), it is obtained by warping the source geometry into the corresponding target tangent view. We pool all valid correspondences and estimate one scene-level scale shared by all crops:

$$s^\star = \underset{\substack{i\in\{0,\ldots,N\},\, k\in\{1,\ldots,K\} \\ \mathbf{x}\in\Omega_{i,k}^{\text{align}}}}{\text{RANSAC}} \frac{D_{i,k}^{\text{anchor}}(\mathbf{x})}{\hat{D}_{i,k}(\mathbf{x})}. \tag{5}$$

Using a shared scale preserves the relative geometry predicted across panoramas and avoids independently scaled local reconstructions. Let $\Omega_{i,k}^{\text{recon}}$ contain all valid, non-sky depth predictions retained for reconstruction, including newly revealed regions. We apply the scale to all predicted depths and back-project them using the known cameras:

$$\tilde{D}_{i,k} = s^\star \hat{D}_{i,k}, \quad \mathcal{P}_{i,k} = \left\{ \mathbf{R}_{i,k}^\top \left(\tilde{D}_{i,k}(\mathbf{x})\, \mathbf{K}^{-1} \bar{\mathbf{x}} - \mathbf{t}_{i,k} \right) \mid \mathbf{x} \in \Omega_{i,k}^{\text{recon}} \right\}, \tag{6}$$

where $\bar{\mathbf{x}}$ is the homogeneous pixel coordinate, $\mathbf{K}$ is the tangent-camera intrinsic matrix, and $(\mathbf{R}_{i,k}, \mathbf{t}_{i,k})$ are the rotation and translation components of $\mathbf{T}_{i,k}$. Merging all local point clouds gives $\mathcal{P}_{\text{init}} = \bigcup_{i,k} \mathcal{P}_{i,k}$, which initializes the Gaussian representation and includes geometry inferred in newly exposed regions.

Evaluation Depth Protocol. For a controlled comparison in Table 1, every method uses the same source-view depth wherever geometry is required for view planning, point-cloud warp construction, or final scale alignment. We use rendered ground-truth source depth on both UE synthetic and the RealSee3D synthetic subset. GenEx, Matrix-3D, and WorldExpand are therefore evaluated with identical source geometry and identical selected target poses; no target-view depth is provided to any method, and target RGB-D is used only to compute evaluation metrics. For the in-the-wild qualitative examples, where ground-truth depth is unavailable, we instead predict the source depth using DepthMaster (Wang et al., 2026b). This separates the controlled benchmark protocol from the practical deployment setting with estimated geometry.

Pano-Aware Gaussian Optimization. Starting from the initialized point cloud $\mathcal{P}_{\text{init}}$, we follow the Gaussian optimization design of HY-World 2.0 (HY-World et al., 2026) and optimize a 3D Gaussian Splatting representation $\mathcal{G}$ under sparse panoramic supervision. For each tangent training view, the differentiable rasterizer outputs rendered color $\bar{I}_{i,k}$, rendered inverse depth $\bar{D}_{i,k}^{-1}$, and rendered normals $\bar{\mathbf{N}}_{i,k}$. The optimization combines color, geometry, and reliability terms (view subscripts omitted for brevity):

$$\begin{aligned} \mathcal{L} = \lambda_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \lambda_{\text{lpips}} \mathcal{L}_{\text{lpips}} + \lambda_{\text{disp}} \left\| \bar{D}^{-1} - \tilde{D}^{-1} \right\|_1 \\ + \lambda_{\text{n}} \left(1 - \left\langle \bar{\mathbf{N}}, \mathbf{N}^{\text{pseudo}} \right\rangle \right) + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}. \end{aligned} \tag{7}$$

We use inverse-depth supervision because panoramic scenes span extreme depth ranges, and derive $\mathbf{N}^{\text{pseudo}}$ from the aligned depth $\tilde{D}$ by finite differences. The mask loss is $\mathcal{L}_{\text{mask}} = (\text{mean}(\bar{M}))^2$, where $\bar{M}$ is the rendered Gaussian mask, and suppresses diffuse opacity. We additionally place virtual cameras between selected panorama pairs and supervise their rendered inverse depth using pseudo-depth warped from nearby training views, regularizing geometry between the sparse observations.

We optimize the Gaussian representation for T=1,500 steps. The color term is $\mathcal{L}_{\text{rgb}} = 0.8\|\bar{I} - I\|_1 + 0.2(1 - \text{SSIM}(\bar{I}, I))$ throughout optimization. For the remaining terms, each weight is linearly interpolated as $\lambda(t) = (1 - t/T)\lambda^{\text{start}} + (t/T)\lambda^{\text{end}}$: λ_{lpips} decreases from 0.2 to 0.1, λ_{disp} decreases from 0.2 to 0.01, and λ_{n} increases from 0.1 to 0.5. The inverse-depth residual is additionally normalized by the scene scale. We keep $\lambda_{\text{mask}} = 0.003$ and activate the mask term only

Table 3: **System-level comparison with Matrix-3D.** Despite using a larger model and higher resolution, WorldExpand is approximately $2.3\times$/$8.6\times$ faster than Matrix-3D at 480p/720p for four-direction generation. Times are single-H20 inference measurements; Matrix-3D timings are multiplied by four for four independently generated directions. The denoising row is stage-matched, whereas WorldExpand end-to-end time was not separately profiled. Render costs follow the controlled comparison in Table 6.

Property	WorldExpand	Matrix-3D
Generation backbone	20B	14B
Resolution	1024×2048	480×960 or 720×1440
Representation	4 joint panoramas	81-frame video per direction
Four directions	1 run; 4 panoramas	4 runs; 324 frames
Denoising (4 directions)	$\sim$45 min 26 s (1 joint run)	1 h 45 min 42 s (4 clips, 480p) / 6 h 29 min 53 s (4 clips, 720p)
Condition	Gaussian-splat point cloud	MoGe-derived mesh
Render cost	GS: 0.72 s / 0.89 s (1K / 2K)	Mesh ref.: 60 s / $\sim$73 s (1K / 2K)
Consistency	Joint across views	Independent across directions
Reconstruction	Arbitrary posed panoramas	Single trajectory; PanoLRM/Pano-GS

from step 300 onward. These schedules were selected empirically to balance appearance fitting and geometric refinement rather than through exhaustive tuning. In practice, the reconstruction is insensitive to modest perturbations of these values, which produce no material change in the resulting geometry or rendering quality.

B System-Level Comparison with Matrix-3D

We compare WorldExpand with Matrix-3D (Yang et al., 2025) in Table 3; all runtime measurements use a single NVIDIA H20 GPU. WorldExpand uses a 20B-parameter backbone and generates panoramas at 1024×2048, whereas Matrix-3D uses a 14B-parameter backbone and operates at 480×960 or 720×1440. Despite using the larger model and higher output resolution, WorldExpand completes four-direction denoising approximately $2.3\times$ faster than Matrix-3D at 480p and $8.6\times$ faster at 720p. This advantage comes from its reconstruction-oriented design: it spends computation on sparse complementary observations, generates them jointly, and directly fuses their posed panoramas. Matrix-3D instead generates a dense 81-frame video for each direction. Most adjacent frames provide little new scene coverage, yet substantially increase generation and downstream processing cost; exploring four directions further requires four independent runs without cross-trajectory communication. Its generation stage is therefore not co-designed to provide a compact multi-direction observation set for reconstruction, and its official reconstruction interface processes one video trajectory at a time.

Additional In-the-Wild Comparisons. Figure 8 extends the main-paper comparison to more architectural, game-like, stylized, indoor, and outdoor panoramas. Across these diverse inputs, independently generated video trajectories frequently introduce ERP distortion, unstable exposure, or content collapse, whereas WorldExpand retains clearer structure and geometric cues across the four target views.

C Dataset Construction Details

Training a pose-controllable multi-pano generation model requires scenes with multiple walking directions, moderate anchor–target disparity, accurate camera poses, and dense depth for warp-based conditioning. Real-world panoramic data cannot reliably provide these properties due to motion artifacts, missing metric depth, and limited trajectory diversity; we therefore construct a large-scale synthetic dataset in Unreal Engine. Each training tuple is

$$\mathcal{S} = \left(I_0, D_0, \mathbf{T}_0, \{I_i, \Delta\mathbf{T}_{0\to i}\}_{i=1}^{N}\right), \tag{8}$$

where I_0, D_0, and $\mathbf{T}_0$ are the anchor panorama, depth, and camera pose, while I_i and $\Delta\mathbf{T}_{0\to i}$ are the i-th target panorama and its relative pose. We next detail the scene collection, trajectory generation, and anchor–target sampling used to construct these tuples.

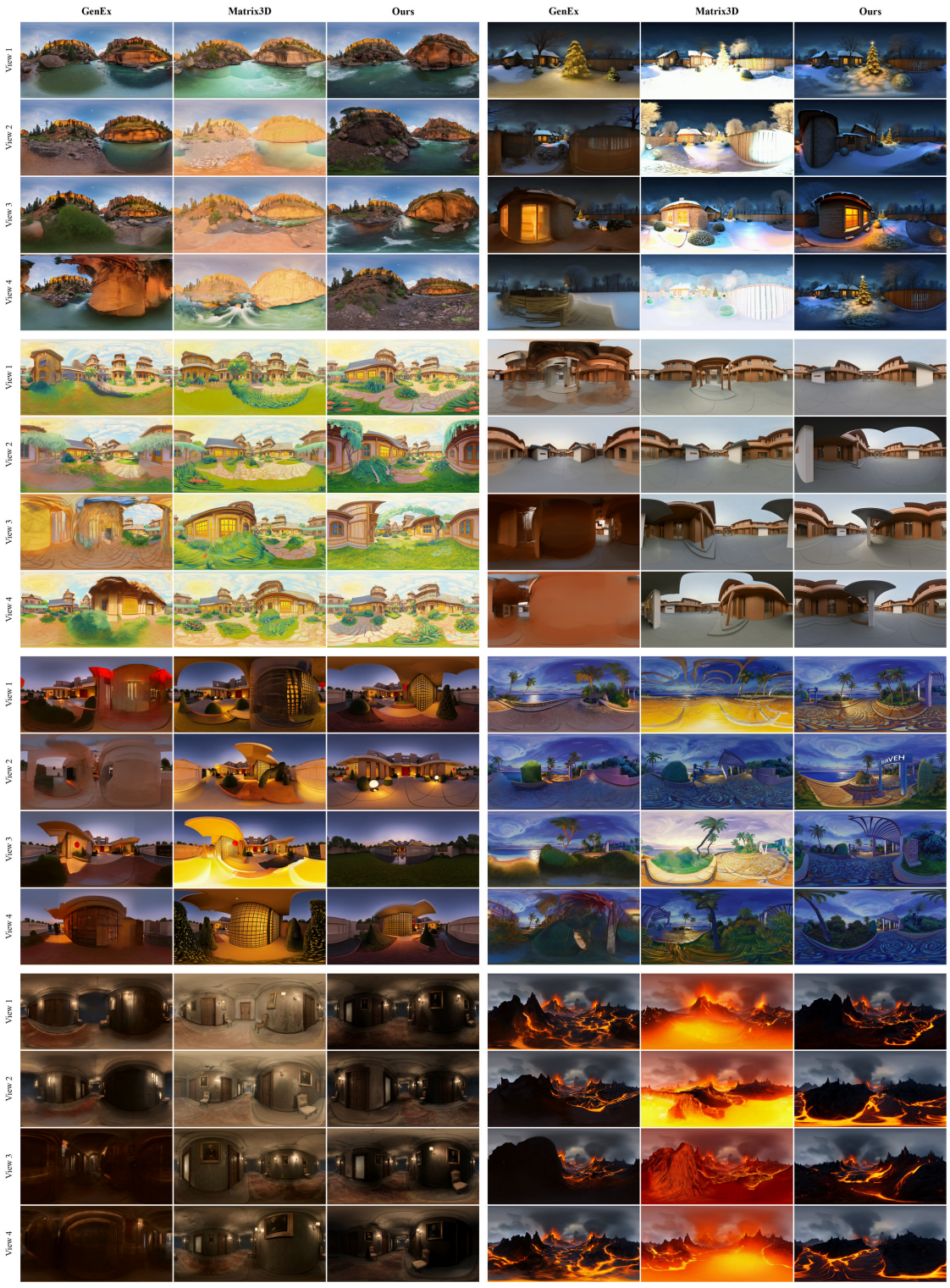

Figure 8: **Additional in-the-wild comparisons.** The four groups are arranged on one page; method headers are shared at the top. WorldExpand remains robust across diverse styles, while independent video generation is more susceptible to distortion, exposure shifts, and structural degradation.

Scene and Roaming Trajectory Collection. We curate Unreal Engine scenes with complex meshes and diverse layouts, and build a trajectory generation pipeline. For each scene, we construct navigable regions via NavMesh-based collision avoidance, generate camera paths with Unreal Engine's random roaming, and filter out invalid cases such as collisions, mesh penetration, short movements, and low-parallax trajectories. We then render equirectangular RGB and depth from a panoramic camera along each valid trajectory, mark sky regions as infinite depth, and record synchronized RGB, depth, and pose annotations. The final dataset contains 76,721 individual trajectories, corresponding to approximately 19.2K four-trajectory groups, covering diverse camera displacements and reliable geometric correspondences. We reserve 30 scenes for validation, with the remaining scenes used for training, ensuring no scene-level overlap between splits.

Intersection-Centric NVS Pair Collection. Exhaustively enumerating anchor–target combinations would produce many redundant pairs: very close views have little parallax, while very distant views encourage hallucination. We instead use an intersection-aware sampling strategy. For each scene, we project all camera centers onto the ground plane and detect intersections between trajectory segments. Nearby intersections are merged, and the frame closest to each merged point is used as an anchor panorama. These intersections naturally connect multiple walking directions and provide a reconstruction-oriented center for scene expansion. Around each anchor, we sample four target views from the incoming and outgoing trajectory directions, subject to a valid baseline range and a minimum temporal gap. For each anchor–target pair, we unproject the anchor RGB-D panorama into a colored point cloud, transform it by the relative pose, and render the point cloud at the target view to obtain a warped image and a disocclusion mask that explicitly marks regions requiring synthesis. This intersection-centered design yields approximately 120K effective NVS training tuples.

D View Selection Strategy Comparison

We compare our Coverage-Guided View Selection (Section 3.1) against four target-view baselines that share the same boundary constraint and generation/reconstruction pipeline, so only *where* the targets are placed differs: (i) *Fixed orthogonal*, four axis-aligned front/back/left/right targets; (ii) *Random*, four positions sampled uniformly in the navigable region; (iii) *Forward walk*, four targets at increasing distance along one forward direction, as in panoramic-video baselines; and (iv) *Fixed quadrant (w/o rotation)*, our one-target-per-quadrant selection with axis-aligned quadrants, ablating the adaptive cross rotation of Equation (1).

Since view planning aims to enlarge the reconstructable scene, we measure expanded area on the final 3D result. For each strategy we reconstruct the scene point cloud (Section 3.3), project both the *source-only* and *source+novel* clouds onto the ground (xz) plane, and rasterize each into an occupancy grid with per-scene adaptive cell size ($1/40$ of the 90th-percentile source radius). A cell is occupied if it holds at least m points, which discards floaters; we sweep $m \in \{10, 20, 30, 50\}$. We first subsample both clouds to the *same* point count, since the source+novel cloud fuses five panoramas and is far denser—otherwise its raw density alone would cross the threshold and be miscounted as new coverage. Let $\tau_0(m)$ and $\tau_{\mathcal{S}}(m)$ be the sets of cells occupied under threshold m by the source-only and source+novel reconstructions; we report the *expansion ratio*

$$E(m) \;=\; \frac{|\tau_0(m) \cup \tau_{\mathcal{S}}(m)|}{|\tau_0(m)|} \;\geq\; 1, \tag{9}$$

where larger $E(m)$ means a larger reconstructable scene (coverage increase $E(m) - 1$), averaged over all scenes.

As reported in Table 4, our planning attains the largest E at *every* threshold, expanding the floor plan by $\sim$41% at m=20 vs. Random (33%), Fixed-orthogonal (39%), and Forward-walk (40%); the last concentrates targets in one direction and expands along a single axis. The gap over Fixed-quadrant (w/o rotation) isolates the adaptive cross rotation of Equation (1)—a small but consistent gain, which we analyze below. Overall, WorldExpand expands the reconstructable floor-plan area to up to $1.53\times$ the source-only coverage. Qualitatively, Figure 11 shows that the four selected views recover complementary geometry in different directions, which merges into a broader and more complete scene reconstruction.

Why the Adaptive Rotation Yields only a Small Average Gain. The rotation is a safeguard for a specific failure configuration rather than a generally active component, so a small mean gap is the ex-

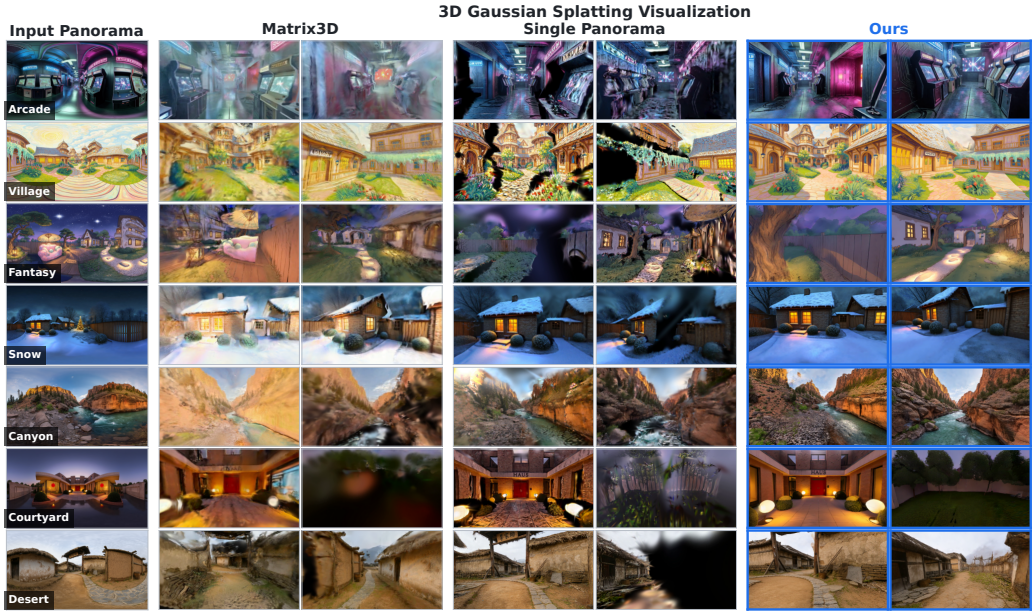

Figure 9: **Additional in-the-wild 3D Gaussian Splatting comparisons.** Each row shows the input panorama followed by reconstructions obtained with Matrix-3D, the single panorama alone, and WorldExpand. Across diverse scenes, WorldExpand provides broader scene coverage and more stable geometry, while single-panorama reconstructions remain incomplete and Matrix-3D exhibits distorted or inconsistent structure.

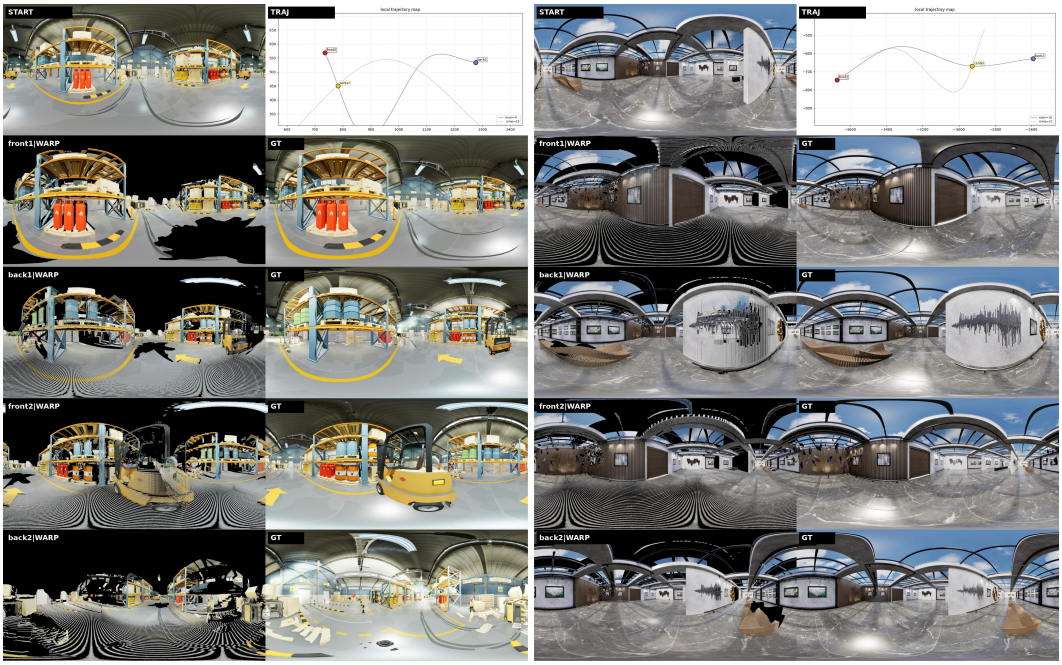

Figure 10: **Dataset examples.** Two training samples from our synthetic Unreal Engine dataset, an outdoor *Colonial City* scene (left) and an indoor *gallery* scene (right). For each example we show the anchor panorama (START) and its local roaming trajectory (TRAJ), followed by the four target views: the point-cloud warp condition (WARP) and the corresponding ground-truth panorama (GT). The warps expose disocclusions (black regions) that the model must synthesize while staying consistent with the anchor.

Table 4: **View-selection strategy comparison.** Expansion ratio $E(m)$ (≥ 1; higher is better) at density threshold m (occupancy min-count), computed from the reconstructed point cloud projected onto the xz plane and averaged over all evaluation scenes. Best per column in **bold**.

Strategy	m=10	m=20	m=30	m=50
Fixed orthogonal	1.317	1.387	1.435	1.485
Random	1.285	1.333	1.362	1.388
Forward walk	1.318	1.396	1.452	1.504
Fixed quadrant (w/o rotation)	1.313	1.398	1.459	1.526
Ours (scene-adaptive quadrants)	**1.334**	**1.411**	**1.464**	**1.527**

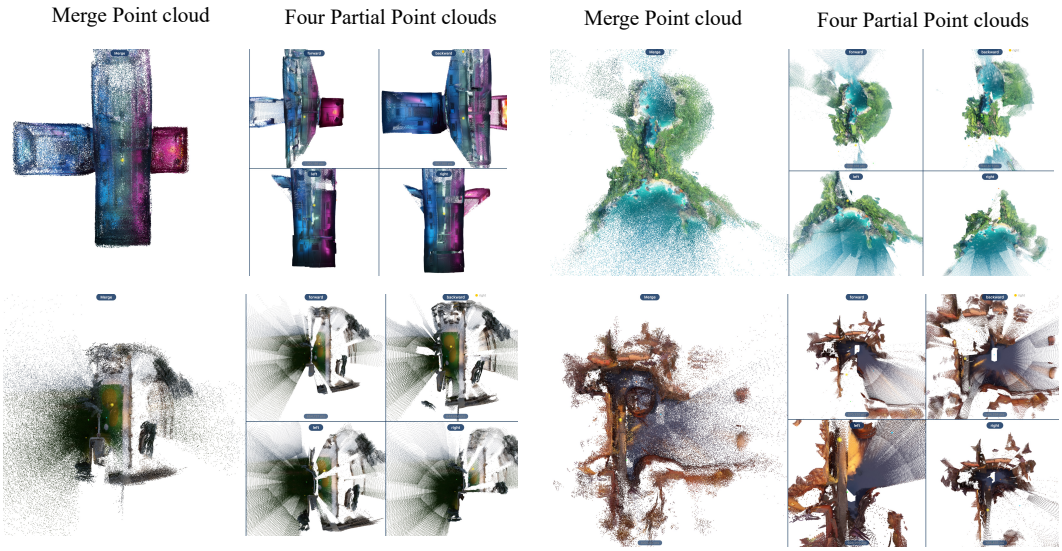

Figure 11: **Qualitative 3D scene expansion with WorldExpand.** Each generated target panorama recovers a different partial point cloud, revealing geometry that is occluded from the initial viewpoint. Merging the four complementary observations expands the reconstructable scene extent while preserving coherent geometric alignment, producing a broader and more complete point cloud.

pected outcome. Given the quadrants $Q_i(\theta)$, the per-quadrant choice in Equation (2) is an $\arg\max$ of c_v over a discrete candidate set and is therefore piecewise constant in θ: the four selected targets change only when the rotation moves a candidate across a quadrant boundary. Whenever the scene boundary is already roughly aligned with the source frame, or the open directions lie well inside the axis-aligned quadrants, $\theta^\star$ leaves the assignment of candidates to quadrants unchanged and the adaptive variant returns exactly the same four targets as Fixed quadrant (w/o rotation)—it degenerates to the fixed rule and cannot be worse than it. The rotation only takes effect in the opposite configuration, where the axis-aligned cross cuts through an open direction such as a doorway or a corner while leaving a nearby wall in the interior of a quadrant. That single open region is then split between two adjacent quadrants, each dominated by the wall it contains, so both targets are placed against an occluder and reveal little unseen content. Aligning the cross arms with the nearest boundary instead puts the low-information wall direction on a quadrant boundary, keeps each open region inside one quadrant, and lets that quadrant spend its target on the space occluded from the source. Since Table 4 averages over both regimes, the reported gap is diluted by the many scenes in which the two variants produce identical targets, while the scenes it does affect are exactly those where the fixed partition wastes a view on a blocked direction. We therefore keep the adaptive rule as the default: it is a no-cost correction that recovers these cases and reduces to the fixed partition everywhere else.

E Why N=4 Target Views Are Enough

The view planner in the main paper (Section 3.1) commits to N=4 target views. This choice reflects a trade-off at two levels. *First, cost:* each target panorama must be generated by the diffusion

Table 5: **Why $N{=}4$ target views are enough.** Scene-expansion ratio at four occupancy thresholds mc as the number of aux target views n varies (last columns: the four-threshold average and its marginal gain over $n{-}1$). Point clouds are subsampled to a common 1.5M budget so the ratio is not inflated by the larger raw point count at higher n. Expansion jumps sharply at $n{=}4$ and then saturates, confirming $N{=}4$ as the elbow. Column maxima are in **bold**; the bold $n{=}4$ marks our selected operating point.

n	$mc10$	$mc20$	$mc30$	$mc50$	avg	Δ
3	1.371	1.408	1.435	1.488	1.426	—
4	1.427	1.485	1.490	1.563	1.491	**+0.066**
5	1.436	1.475	1.514	1.596	1.505	+0.014
6	**1.447**	1.485	**1.518**	**1.608**	**1.515**	+0.009
7	1.441	**1.486**	1.511	1.597	1.509	−0.006
8	1.434	1.478	1.496	1.584	1.498	−0.011

model and then reconstructed, so the view count is bounded by GPU memory and inference/training compute—joint multi-pano generation scales its activation footprint with the number of views, and a larger N quickly becomes impractical. *Second, limited benefit from extra views:* beyond a handful of well-placed views, a new target either re-observes an already-covered region or is pushed toward the boundary where its warp is unreliable, adding little useful coverage. We verify that $N{=}4$ sits at the elbow of this trade-off by varying the number of aux target views n (the source panorama is always kept as the ground-truth-scale anchor and is not counted in n). For each n we back-project the per-view metric depths into a single point cloud and measure *scene expansion*—the ratio of the reconstructed floor footprint (top-down XZ occupancy after IoU scale-alignment to the source) to the source-only footprint. A view count is beneficial only if it enlarges this footprint. Crucially, the raw point count grows monotonically with n (from ∼20M to ∼45M points); to prevent "more views → more points" from inflating the ratio, every cloud is **randomly subsampled to the same point budget (1.5M) before the footprint is computed**, so the expansion ratio is compared fairly across n. We report the ratio at four occupancy thresholds (min-count $mc \in \{10, 20, 30, 50\}$ points per grid cell) to check threshold robustness.

Result. Table 5 shows the same trend at all four thresholds: expansion rises steeply from $n{=}3$ to $n{=}4$ (a $+0.05$ to $+0.08$ jump, $+0.066$ on the four-threshold average), and then flattens—the marginal gain collapses to ≤ 0.014 for $n{=}5, 6$ and turns *negative* at $n{=}7, 8$, as extra views add redundant overlap or drag less-reliable boundary content into the footprint. **The fourth view captures essentially all the newly expandable scene coverage, while further views only pay compute for redundancy**, so $N{=}4$ is the natural operating point.

F Discussion on Condition Modalities

Our generator is grounded by an explicit geometric condition introduced in the main paper (Section 3.2): the source RGB-D panorama is unprojected into a colored point cloud $\mathcal{P}_0$ and rendered from each target pose to obtain the warp W_i and disocclusion mask M_i. We motivate this choice against two alternatives—mesh warping and depth-as-color—and explain why we implement $\mathrm{Render}_{\mathrm{ERP}}$ with Gaussian splatting.

Point-cloud Warp via Gaussian Splatting. Rendering the colored point cloud is perspective-correct (near surfaces cover more pixels than far ones) and resolves occlusion by depth ordering, so foreground correctly hides background without building an explicit surface. We rasterize $\mathcal{P}_0$ with Gaussian splatting rather than plain point rendering: it is essentially independent of point count and much faster (Table 6), and it stays robust when the target camera moves close to the scene—there $\mathcal{P}_0$ becomes locally sparse, yet the anti-aliased splats still produce a clean warp while genuine disocclusions remain marked in M_i. This cleaner warp also translates into better synthesis quality: under an identical setup, conditioning on the splatting-rendered warp outperforms the plain point-rendered warp by $+0.76$ dB PSNR and $+0.065$ SSIM (Table 7).

Mesh Warping. Reconstructing a mesh from the panorama depth and rasterizing it produces stretched "rubber-sheet" triangles across depth discontinuities (Figure 12), which both smear texture and *fill* regions that should be disoccluded, hiding the background and corrupting M_i. Meshing is also the runtime bottleneck (tens of seconds per batch, Table 6). Crucially, a connected mesh is hard

to augment: our warp-side degradations (main paper Section 3.2: hole expansion, low-frequency remapping, edge blur) act naturally on points and pixels but have no clean analogue on triangles, so the mesh path cannot reproduce the train–test degradations that make the model robust to imperfect inference-time geometry.

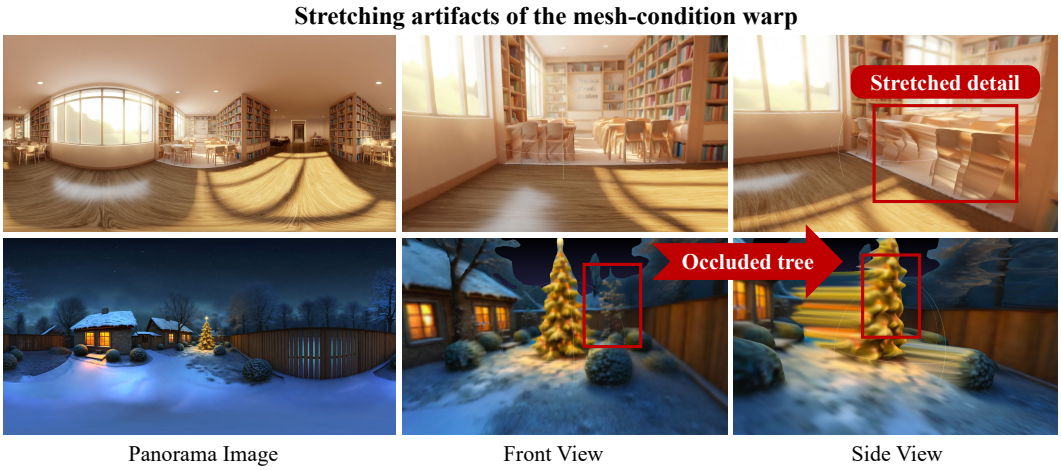

Figure 12: **Stretching artifacts of the mesh-condition warp.** Panoramic input, front view, and side view for two scenes. The front view looks clean, but at side angles the mesh warp reveals severe stretching. The artifact originates in the triangulation: at depth discontinuities the mesh connects a foreground edge vertex to a far-away background vertex, spawning a highly elongated "rubber-sheet" triangle. Because this triangle is a *solid* rasterized surface, it occludes the true background behind it (e.g., the hidden tree) and smears texture across the gap, degrading fine detail. Point-cloud warping rendered with Gaussian splatting has no such connectivity and thus avoids these artifacts, which is why we adopt it as the sole geometric condition.

Depth-as-color. Encoding metric depth as an image is compact but fragile for outdoor panoramas (Figure 13). The standard recipe clips depth to its 2–98 percentile range and normalizes to $[0, 1]$, but the sky and far background span an enormous depth range, so the foreground is compressed into a tiny sub-interval and its color contrast is nearly lost (Figure 13b). Using inverse depth $1/d$ (disparity) instead spreads out the near field but crushes the far field toward zero, discarding background structure (Figure 13c). A principled fix is to first segment and exclude the sky, then normalize only over foreground depths; however, even a strong zero-shot matting model (Kim et al., 2025) does not segment the sky cleanly on our diverse scenes (Figure 13d), so the residual sky pixels (with huge depth) still dominate the normalization range and the foreground contrast remains as poor as the naive version (Figure 13e). Depth-as-color is therefore not a reliable geometric condition.

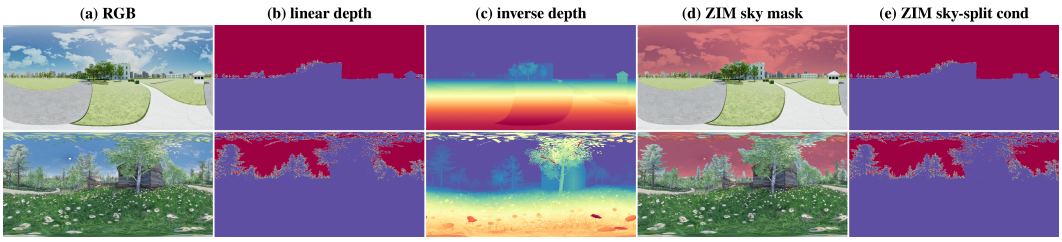

Figure 13: **Why depth-as-color fails as a condition.** Two scenes (rows). **(a)** RGB panorama. **(b)** Linear depth normalized to $[0, 1]$: the huge sky depth compresses the foreground into a narrow band, so it is almost uniform in color. **(c)** Inverse depth $1/d$: the near field is spread out but the far field collapses to near-zero, losing background detail. **(d)** Sky mask from a zero-shot matting model (Kim et al., 2025): the segmentation is not clean (rough edges, misses, false positives). **(e)** Normalizing over the (imperfectly) desky-ed depth: residual sky with large depth still dominates the range, so the foreground contrast is nearly identical to (b). No variant preserves foreground detail, motivating our point-cloud warp condition.

Table 6: **Render cost of the warp condition** (total time for the $N{=}4$ target views, ERP output at 1K/2K per-view resolution). Gaussian splatting is $\sim$5–6$\times$ faster than direct point rendering and orders of magnitude faster than meshing, with cost essentially independent of point count.

Render engine	1K	2K	Notes
Gaussian splatting (ours)	**0.72 s**	**0.89 s**	fastest, stable, point-count independent
Point cloud (1M pts)	4.21 s	4.51 s	usable, but $\sim$5–6$\times$ slower
Mesh	60 s	$\sim$73 s	dominated by surface reconstruction, not rasterization; too slow for the pipeline

Table 7: **Point-cloud rendering ablation: Gaussian splatting vs. plain point rendering.** Both variants condition on the *same* colored point cloud $\mathcal{P}_0$ and share an identical training/evaluation protocol; metrics are the full-image average over the $N{=}4$ target views. Gaussian splatting produces a cleaner warp—most notably where $\mathcal{P}_0$ is locally sparse—and yields clearly better PSNR/SSIM at a fraction of the render cost (Table 6), with LPIPS essentially tied.

Point-cloud renderer	PSNR ↑	SSIM ↑	LPIPS ↓
Plain point rendering	19.40	0.593	**0.252**
Gaussian splatting (ours)	**20.16**	**0.658**	0.256

Summary. Point-cloud warping rendered by Gaussian splatting gives the best trade-off: perspective-correct and occlusion-aware like a mesh but far cheaper and free of stretching artifacts, robust to the large outdoor depth range unlike depth-as-color, and—uniquely—compatible with the per-view warp augmentation central to our robustness. We therefore adopt it as the sole geometric condition.

G Warp-Side Augmentation: In-Domain Accuracy vs. Robustness

Training warps use clean synthetic depth, whereas inference-time warps rely on estimated geometry and inevitably contain mask errors, geometric distortions, and boundary blur. A model trained only on clean warps can therefore over-trust and directly copy the warp stream, amplifying these errors and bypassing source- and cross-view reasoning. Our warp-side augmentation (main paper Section 3.2) closes this gap by independently composing hole expansion, low-frequency geometric remapping, and edge-local RGB blur for each target view. Because the perturbations differ across views, the model cannot rely on perfect warp–target alignment and must instead learn to correct imperfect geometric guidance using the source panorama and cross-view context. This is particularly important in the wild, where warps are produced from off-the-shelf depth estimates (Wang et al., 2026c;b) rather than ground-truth depth. Since the perturbations deliberately move training conditions away from the clean data manifold, we expect a small cost on clean, in-domain evaluation in exchange for robustness to noisy real inputs.

Setup. We compare two full-global 1K$\rightarrow$2K models that are identical except for whether warp-side (depth) augmentation is enabled, evaluated on a clean-depth in-domain set (each at its best checkpoint).

Result. As summarized in Table 8, on this clean-depth set the augmented model is uniformly and slightly lower (-0.53 dB PSNR, -0.022 SSIM, $+0.012$ LPIPS) and also converges more slowly (best checkpoint at step 8200 vs. 5500). This is expected and, crucially, does *not* measure the intended benefit: the evaluation itself uses clean synthetic depth, which structurally favors the non-augmented model. The payoff of the augmentation is robustness to *estimated* depth on in-the-wild inputs, where warps are noisy; since such wild panoramas have no ground-truth target, we assess this regime qualitatively.

Conclusion. Clean-depth in-domain metrics structurally reward a model that trusts the warp, so the small drop from warp-side augmentation is a favorable trade: it prevents the model from over-copying an imperfect warp and yields more reliable synthesis when the input depth is noisy—the setting that matters for real single-panorama inputs.

Table 8: **In-domain (clean-depth) cost of warp-side augmentation**, best checkpoint per model. Numbers are expected to slightly favor the non-augmented model because the evaluation uses clean depth; the augmentation instead targets robustness to noisy estimated depth at inference.

Metric	w/o warp augmentation	w/ warp augmentation (ours)	Δ
PSNR ↑	**20.92**	20.39	−0.53
SSIM ↑	**0.674**	0.652	−0.022
LPIPS ↓	**0.234**	0.246	+0.012

H Ablation on Joint Four-View Generation

Table 9 and Figure 14 provide the quantitative and qualitative ablations of joint four-view generation, respectively. When the target views are generated independently (single-view generation) instead of jointly, all per-view metrics degrade and the model produces inconsistent content across views over multiple inference runs. Together, these results confirm that joint generation is essential for both per-view fidelity and cross-view consistency.

Table 9: **Ablation on joint four-view generation.** *w/o four-view* trains a single-view variant whose conditioning inputs support only one target warp at a time, so the four target panoramas are denoised independently at inference. The global–local attention schedule and warp-side augmentation are separate efficiency/robustness trade-offs, summarized in Figure 6 and Section G, respectively. Best results per metric are in **bold**.

	w/o four-view			Full		
View	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
V1	19.16	0.589	0.251	**20.61**	**0.674**	**0.243**
V2	18.20	0.558	0.290	**20.05**	**0.663**	**0.256**
V3	18.51	0.570	0.281	**19.91**	**0.647**	**0.262**
V4	17.23	0.523	0.315	**20.06**	**0.649**	**0.261**
Avg.	18.27	0.560	0.284	**20.16**	**0.658**	**0.256**

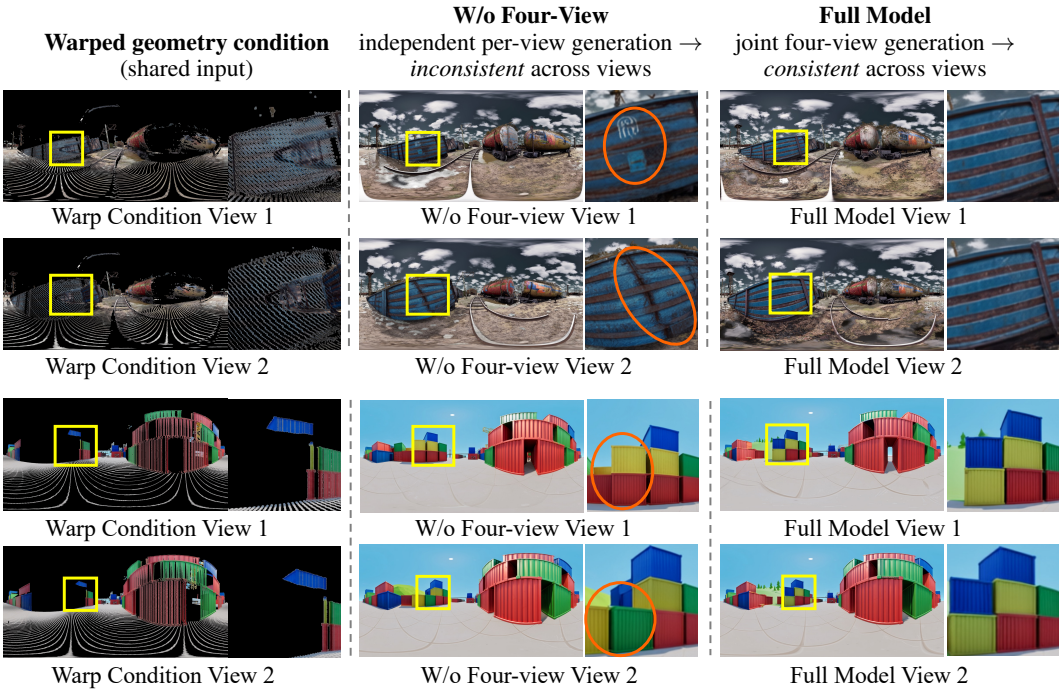

Figure 14: **Joint four-view generation improves consistency.** Independently generated views disagree in overlapping regions (orange ellipses), whereas joint generation produces coherent content.

Table 10: **Local-attention token scopes during 2K training** on one node with 8×H20 GPUs (96 GB each). Time is measured per training step; OOM denotes per-GPU memory exhaustion.

Variant	Local tokens $\mathbf{H}_i^{\text{local}}$	#Local	VRAM	Train step	Speed-up
v1	$[\mathbf{P}_{txt}; \mathbf{Z}_i]$	20	85.7 GB	213 s	1.40×
		30	82.7 GB	169 s	1.77×
		40	80.9 GB	127 s	2.35×
v2	$[\mathbf{P}_{txt}; \mathbf{Z}_i; \mathbf{W}_i]$	20	89.7 GB	229 s	1.31×
		30	88.4 GB	196 s	1.53×
		40	87.8 GB	162 s	1.85×
v3	$[\mathbf{P}_{txt}; \mathbf{S}_0; \mathbf{Z}_i; \mathbf{W}_i]$	20	OOM	–	–
		30	OOM	–	–
		40	OOM	–	–

I Trade-off Between Number and Cost of Attention Layers

The global–local schedule has two design choices: the token scope visible to each local layer and the number of layers made local. We first select the most efficient local token scope, then fix that design and study the layer-count trade-off.

Selecting the Local Token Scope. We compare three per-view token scopes while leaving the global layers unchanged:

$$\begin{aligned} \text{v1:} \quad & \mathbf{H}_i^{\text{local,v1}} = [\mathbf{P}_{txt}; \mathbf{Z}_i], \\ \text{v2:} \quad & \mathbf{H}_i^{\text{local,v2}} = [\mathbf{P}_{txt}; \mathbf{Z}_i; \mathbf{W}_i], \\ \text{v3:} \quad & \mathbf{H}_i^{\text{local,v3}} = [\mathbf{P}_{txt}; \mathbf{S}_0; \mathbf{Z}_i; \mathbf{W}_i]. \end{aligned} \tag{10}$$

Here $\mathbf{P}_{txt}$ denotes text tokens, $\mathbf{S}_0$ the source-panorama tokens, $\mathbf{Z}_i$ the noisy tokens of target view i, and $\mathbf{W}_i$ its geometric-warp tokens. The three scopes differ in how much conditioning each local layer still sees: v1 lets a local layer refine only its own target view, v2 additionally keeps that view's geometric warp inside the local layer, and v3 further keeps the shared source panorama. Every stream that is excluded locally is still available in the global layers, so the variants change *where* conditioning is attended to, not whether it is used. Since each added stream lengthens the per-view local sequence, cost grows from v1 to v3, as measured at 20, 30, and 40 local layers in Table 10.

Across every tested layer count, v1 gives the lowest memory and training time. Variant v2 retains more conditioning tokens locally but consequently provides much smaller savings, while v3 exceeds device memory at all three layer counts. We therefore adopt v1: selected global layers retain source/warp-mediated cross-view reasoning, while local layers focus on per-view target refinement at the lowest computational cost.

Varying the Number of Local Layers. All subsequent layer-count experiments use the v1 token scope. Following the per-layer diagnostic, we localize the *least-sensitive* N layers (smallest ΔPSNR) rather than simply choosing the last N layers. Measurements use one node with 8×H20 GPUs (96 GB nominal, approximately 95 GiB usable per GPU) at 2K resolution (1024×2048 per view), DeepSpeed ZeRO-2, bf16, gradient checkpointing, LoRA (rank 64), text/VAE offloading, and batch size 1. We report rank-0 peak memory and training-step time (forward + backward); the full-global baseline costs 90.5 GB and 299 s per step.

Table 11: **Local-attention layer-count trade-off under real 2K training** (single node, 8×H20). Our backbone (Qwen-Image-Edit-2509) is a 60-layer DiT, so #Local layers is out of 60. Baseline (all-global): 90.5 GB / 299 s per step. ΔMem is memory saved versus the baseline; speedup is relative to the baseline step time.

#Local layers	Peak mem (GB)	ΔMem (GB)	Step time (s)	Speedup
0 (all global)	90.5	–	299	1.00×
20	85.7	−4.8	213	1.40×
30	82.7	−7.8	169	1.77×
40	**80.9**	**−9.6**	**127**	**2.35×**

Findings. Because each local layer keeps only the per-view noisy tokens, adding more local layers steadily saves both memory and time: at 40 local layers the model reaches 80.9 GB with a 2.35× speedup, leaving ∼14 GB of headroom, while 20/30 local layers offer more conservative savings. The accuracy cost of localizing more layers is modest (Table 12): even at 40 local layers, the most sensitive selected layer only drops −0.71 dB in the training-free diagnostic, and after finetuning the model re-adapts to local attention, so the practical degradation is smaller. **We therefore localize 30 of the 60 DiT layers by default—an empirical choice that balances efficiency and cross-view consistency—while 40 layers push efficiency further and 20 layers give a more conservative accuracy margin.**

Table 12: **Accuracy cost of the selected local layers** (training-free per-layer diagnostic, overall ΔPSNR; closer to 0 is safer). Larger N includes progressively more sensitive layers.

#Local layers	Most sensitive selected layer (ΔPSNR)	Mean ΔPSNR of the set
20	−0.19 dB	−0.07 dB
30	−0.34 dB	−0.14 dB
40	−0.71 dB	−0.23 dB

Inference Speed-Up. Local attention also shortens the per-view attention sequence at sampling time, so it accelerates inference as well (Table 13): at 40 local layers the profiling latency drops from 66.8 s to 27.5 s, a 2.42× speed-up, and 30 local layers (our default) already give 1.79×. Each reported measurement sums one conditional and one unconditional forward pass for a single denoising step. It is therefore a controlled per-step profiling proxy for comparing attention configurations, not the wall-clock runtime of a complete sampling run. Unlike training, the peak memory stays essentially flat (∼49.7 GB across all settings). This is expected: inference has no backward pass, so activations are freed layer-by-layer and are not the bottleneck; the peak is instead dominated by the resident model weights and the full set of image tokens (82,944 tokens for all views), which are unchanged by the attention scope. With a memory-efficient (Flash) attention kernel the attention matrix is never materialized, so even full-global attention does not inflate memory—local attention only reshapes *which* tokens attend (block-diagonal per view), reducing FLOPs and latency rather than stored tensors.

Table 13: **Per-step profiling speed-up from local attention at 2K** (1024×2048 per view). Each latency sums one conditional and one unconditional forward pass over all views for a single denoising step. These controlled measurements compare attention configurations and do not represent complete end-to-end inference time. Speed-up is relative to the all-global baseline. Peak memory is essentially constant because, without a backward pass, memory is dominated by weights and image tokens rather than attention activations.

#Local layers	Peak mem (GB)	Per-step profile (s)	Speedup
0 (all global)	49.65	66.76	1.00×
20	49.67	47.14	1.42×
30	49.67	37.34	1.79×
40	49.67	**27.53**	**2.42**×

Is Full-Global Attention Worth Its Cost? The trade-off above measures compute; here we check the matching quality side under real 2K training. We compare the full-global model (all 60 layers global) against our default global–local schedule (30 least-sensitive layers switched to local), both resumed from the *same* base checkpoint (Section J). As shown in Figure 15, the two curves track each other closely: after switching to local attention our schedule quickly rejoins the full-global curve and both fluctuate within a narrow band around the base, with best overall PSNR of 20.49 dB (full-global) versus 20.16 dB (global–local)—a gap of only ∼0.33 dB. **Full-global attention is thus marginally better in quality but, as quantified in Table 11, substantially more expensive (90.5 GB / 299 s per step vs. 82.7 GB / 169 s at 30 local layers). Since the quality drop is small while the compute saving is large, the global–local schedule is a favorable trade-off, which is why we adopt it by default for 2K synthesis.**

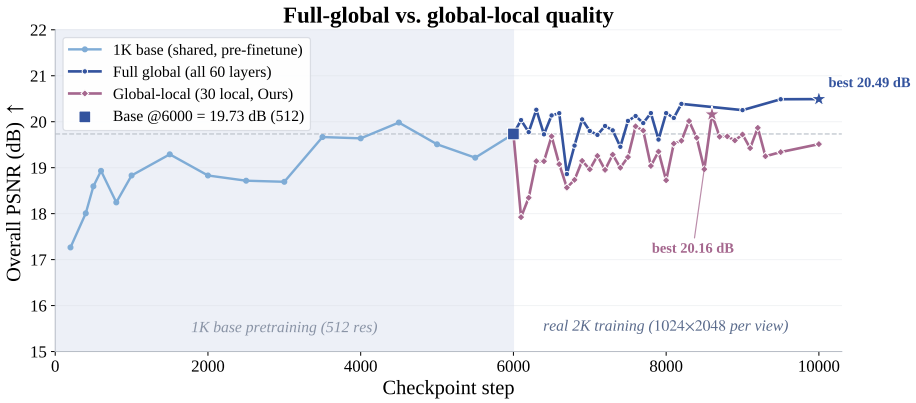

Figure 15: **Quality side of the global–local trade-off on real 2K.** Overall PSNR versus checkpoint step for full-global attention (all 60 layers) vs. our default global–local schedule (30 local layers). The shaded region is the shared 1K base pretraining (512 res); at step 6,000 both runs fork from the same base checkpoint (blue square, 19.73 dB) and are then trained for 2,000 steps at real 2K resolution (1024×2048 per view). Each run reports its own best checkpoint ($\star$: 20.49 vs. 20.16 dB, a $\sim$0.33 dB gap), so localizing 30 layers costs little quality while saving substantial memory and time (Table 11).

J Robustness of Layer Selection Across Training Checkpoints

Our global–local schedule in the main paper (Section 3.2) fixes the set of local layers from a per-layer diagnostic (Figure 3) measured at a *single* checkpoint. A natural concern is whether this selection is an artifact of the particular training stage. To probe this, we run the identical per-layer diagnostic on an early checkpoint (ckpt-2000, during 1K training) and the final checkpoint (ckpt-8000, after 6,000 steps at 1K and 2,000 steps at 2K) of the same model, and compare the per-layer ΔPSNR sensitivities.

As shown in Figure 16, the two checkpoints agree closely: the per-layer sensitivities are strongly correlated (Pearson $r = 0.98$ over all 60 layers, rank correlation Spearman $\rho = 0.86$), and 28 of the 30 layers chosen as local are identical across the two checkpoints. As training proceeds, the overall degradation shrinks (mean ΔPSNR $-0.76 \rightarrow -0.51$ dB), i.e. the model becomes more tolerant to localization, yet the *relative* ordering of which layers are globally versus locally oriented is preserved. **This indicates that the global/local role of each layer is largely a property of the pretrained backbone rather than of a specific training stage, so the one-shot layer selection is stable and can be reused without re-running the diagnostic per checkpoint.**

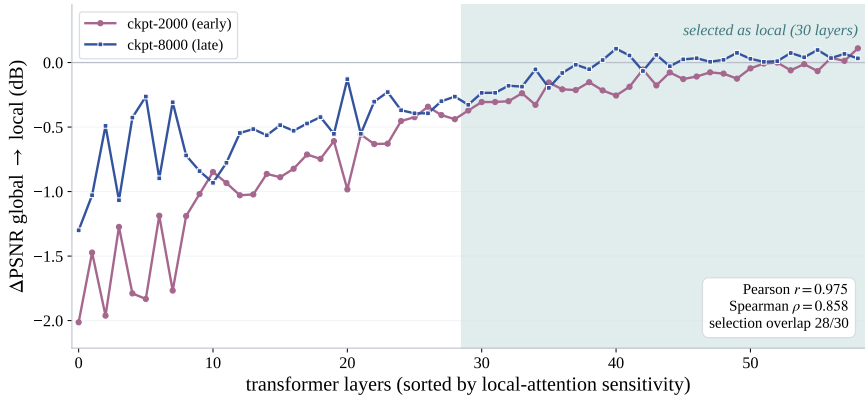

Figure 16: **Layer-selection sensitivity is stable across training checkpoints.** For each transformer layer we report its ΔPSNR when switched from global to local attention (training-free), measured at ckpt-2000 during 1K training and ckpt-8000 after 6,000 1K steps plus 2,000 2K steps; the 59 non-degenerate layers are sorted along the x-axis by their local-attention sensitivity. **The two curves nearly coincide (Pearson r=0.98, Spearman ρ=0.86), and $28/30$ of the layers selected as local (shaded region, the 30 least-degrading) are shared across the two checkpoints**, indicating the global/local role is a property of the pretrained backbone rather than a particular training stage. Layer 0 is an extreme outlier ($-12.8/-13.1$ dB), always kept global and omitted from this sorted view.

K Which Layers to Localize: Least- vs Most-Sensitive Selection

A plausible hypothesis is that the global–local schedule in the main paper (Section 3.2) could localize *any* set of layers and still, after enough finetuning, reach the same quality—i.e. that careful selection is unnecessary. To test whether the diagnostic score predicts which choices are harmful, our default localizes the 30 *least-sensitive* layers (smallest ΔPSNR in the training-free diagnostic), while a control (*INV*) localizes the 30 *most-sensitive* layers among layers 1–59. We deliberately exclude layer 0 from the control: localizing this single extreme outlier already causes an immediate ∼13 dB drop and severe visual collapse (Figure 16), which would make the comparison trivially degenerate. Layer 0 is therefore kept global in both variants, and the reported gap is not caused by including this known failure case.

Setup. Both variants start from the same 1K base checkpoint at step 6,000 (Section J) and continue 2K training under an identical configuration, differing only in the local-layer set. We align the x-axis to finetuning steps (= step − 6,000), so step 0 is the shared base right before switching to local attention.

Results. As shown in Figure 17, our selection dips briefly then re-adapts within ∼100 steps back to ≈ 20 dB (on par with full-global), whereas the control collapses to 11.8 dB and, despite recovering with more training, stays ∼3 dB below throughout (best 16.83 vs. 19.90 dB), with SSIM and LPIPS in agreement. Figure 18 shows the same gap qualitatively: the control hallucinates wrong structure and color early on and remains blurrier and lower-fidelity later. This refutes the hypothesis—localizing the wrong layers inflicts damage that longer training does not repair.

Conclusion. For the Qwen-Image-Edit backbone, a subset of layers carries essential global information and should remain global; forcing many of these layers into local attention breaks cross-view synthesis even after finetuning. This control shows that the diagnostic score is predictive of harmful layer choices, without relying on the degenerate layer 0 failure case.

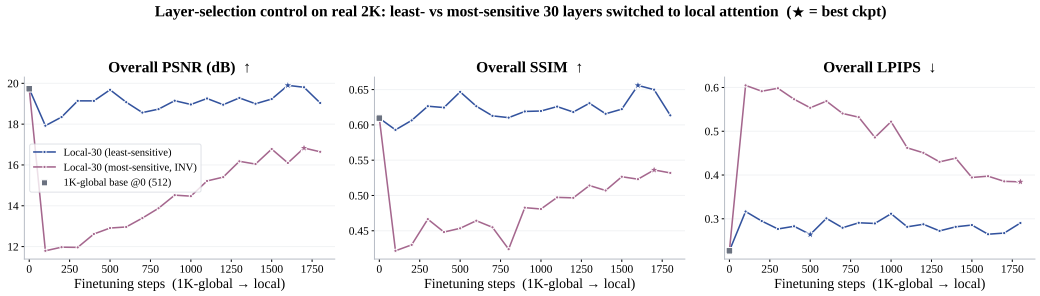

Figure 17: **Which layers to localize matters.** Overall PSNR/SSIM/LPIPS versus finetuning steps for localizing the 30 least-sensitive layers (ours, blue) vs. the 30 most-sensitive layers among layers 1–59 (INV control, plum). The degenerate layer 0 is kept global in both variants. Both start from the same 1K base (gray square, step 0). Our selection recovers to $\approx$ 20 dB, while the control stays $\sim$3 dB lower throughout ($\star$ = best checkpoint per run).



Figure 18: **Qualitative comparison of layer selection.** Each row is one scene; columns are the warped condition, our selection (least-sensitive local), the most-sensitive control over layers 1–59 (INV; layer 0 remains global), and the ground truth. Top two rows: finetuning 100 (the control produces wrong structure/color); bottom two rows: finetuning 1700 (the control recovers layout but stays blurrier and lower-fidelity than ours).

L Full Per-View NVS Results

The main paper reports NVS metrics averaged over the four target views for compactness. Table 14 preserves the complete per-view results used to compute those averages.

Table 14: **Full per-view NVS results.** PSNR/SSIM/LPIPS against ground truth at each target pose on UE synthetic and RealSee3D. Best in **bold**.

		UE synthetic					RealSee3D				
Method	Metric	V1	V2	V3	V4	Avg.	V1	V2	V3	V4	Avg.
GenEx	PSNR ↑	15.59	15.16	15.32	14.64	15.18	15.53	15.89	16.25	15.73	15.85
	SSIM ↑	0.477	0.465	0.467	0.437	0.461	0.638	0.643	0.654	0.642	0.644
	LPIPS ↓	0.440	0.475	0.456	0.500	0.468	0.475	0.447	0.427	0.472	0.455
Matrix-3D	PSNR ↑	17.76	17.17	16.89	17.39	17.30	18.97	16.56	16.59	17.62	17.44
	SSIM ↑	0.513	0.511	0.488	0.494	0.502	0.671	0.649	0.641	0.643	0.651
	LPIPS ↓	0.340	0.372	0.378	0.363	0.363	0.310	0.430	0.435	0.387	0.391
Ours	PSNR ↑	**20.61**	**20.05**	**19.91**	**20.06**	**20.16**	**19.43**	**18.14**	**16.86**	**17.82**	**18.06**
	SSIM ↑	**0.674**	**0.663**	**0.647**	**0.649**	**0.658**	**0.731**	**0.708**	**0.702**	**0.703**	**0.711**
	LPIPS ↓	**0.243**	**0.256**	**0.262**	**0.261**	**0.256**	**0.283**	**0.361**	**0.400**	**0.355**	**0.350**

M Qualitative Comparisons on UE Synthetic Data

We provide controlled qualitative comparisons on held-out UE synthetic scenes in Figures 19 and 20. **WorldExpand produces sharper structures and more faithful geometry in the target views than the baselines while maintaining consistency with the source panorama.**

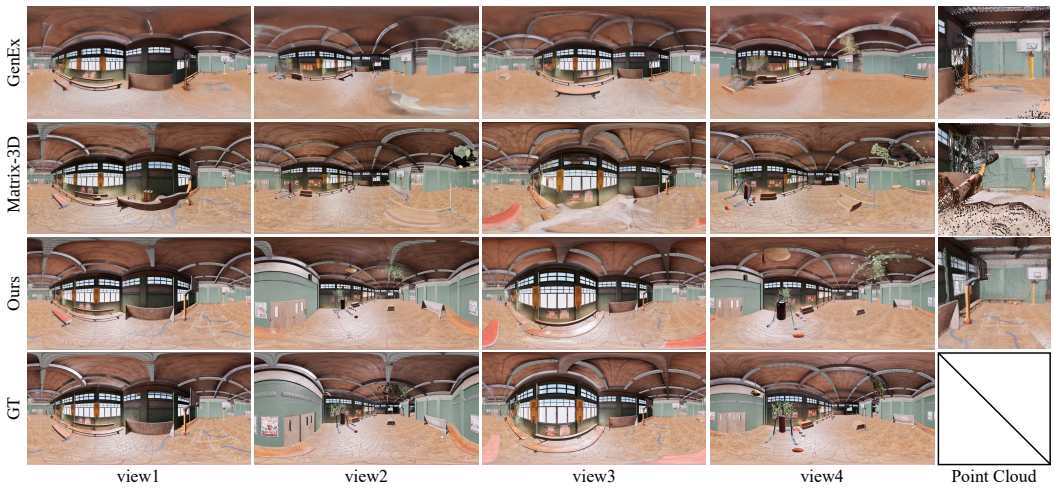

Figure 19: **Qualitative comparison on a held-out UE synthetic scene.** The four target views generated by WorldExpand preserve scene structure and produce a cleaner downstream point cloud than independently generated panoramic-video keyframes.

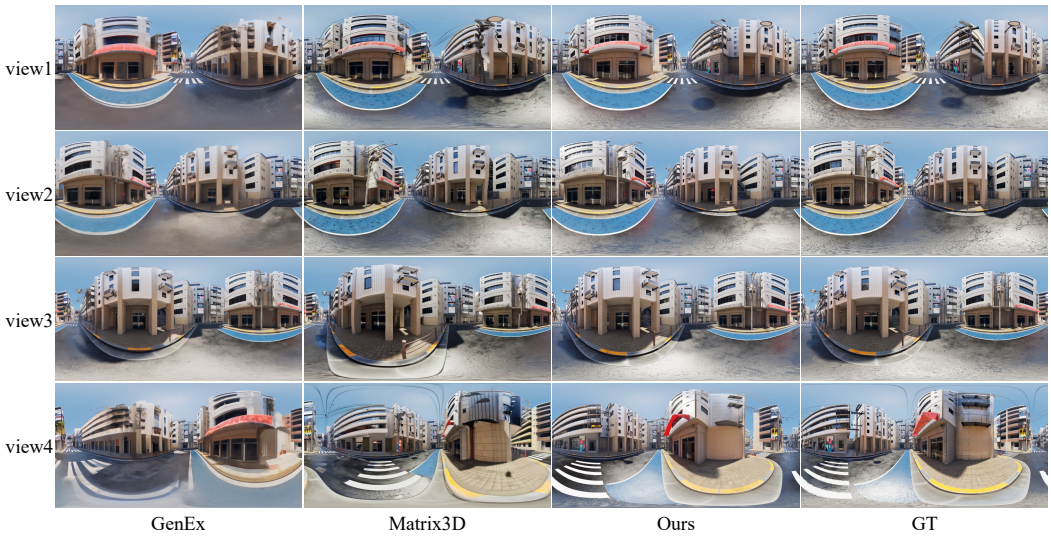

Figure 20: **Qualitative comparison on novel view synthesis from a single panorama.**

N QUALITATIVE COMPARISONS ON REALSEE3D

To complement the quantitative RealSee3D results in the main paper (Table 1), Figure 21 shows qualitative novel-view panorama synthesis on the synthetic subset of RealSee3D (Li et al., 2026b). Despite the domain shift from our UE training data, WorldExpand recovers sharper structures and more faithful geometry in disoccluded regions than the baselines, while remaining consistent with the shared warped condition and ground truth.



Figure 21: **Qualitative comparison on RealSee3D.** Four scenes (rows of blocks); within each block we show the four target views (VIEW1–VIEW4). Columns: the point-cloud warp condition (WARP), Matrix-3D, GenEx, our WorldExpand, and the ground-truth panorama (GT). **Compared with the baselines, WorldExpand synthesizes disoccluded content with sharper structure and more faithful geometry while remaining consistent with the condition and the ground truth.**