# PANODE: WORLD EXPLORATION VIA INCREMENTAL PANORAMIC NODE GENERATION

**Anonymous authors**
Paper under double-blind review

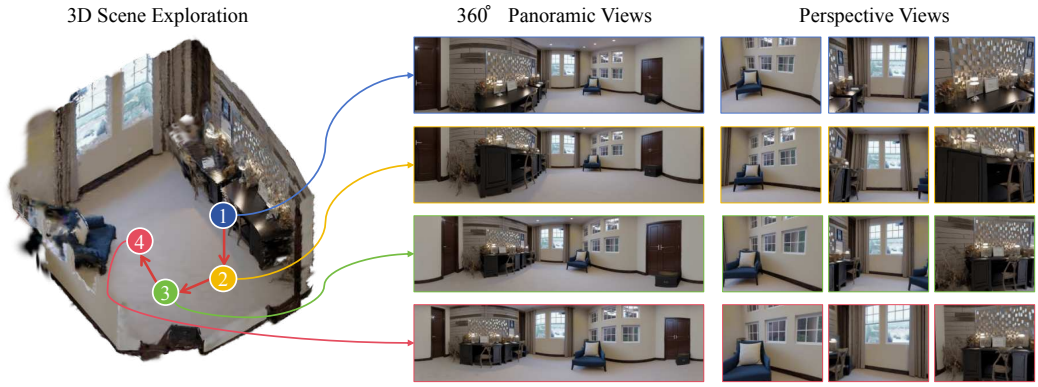



Figure 1: **Panode teaser.** Starting from a single RGB panorama, Panode incrementally explores a world through panoramic node generation, where each node produces a full 360° RGB-D observation and is merged into a shared Gaussian representation.

## ABSTRACT

Generating explorable worlds from a single image is a foundational capability for interactive content creation. Existing methods mostly expand the world through perspective video generation, leading to error accumulation over long-range exploration. To address this, we present Panode, which recasts world exploration as sparse panoramic node generation. Instead of densely sampling frames, we build up the world from a sequence of panoramic nodes, each producing a full 360° observation. At each node, a generative model jointly completes the panoramic RGB and depth, yielding a geometry-consistent omnidirectional observation. The new RGB-D panorama is then lifted into a 3D Gaussian scene and refined via sliding-window optimization over recent nodes. For efficient exploration, we further design an active trajectory planner that chooses the next node by balancing exploration value against motion feasibility. By tightly coupling generation, reconstruction, and planning, Panode progressively expands a single image into a coherent, explorable world. Experiments demonstrate that Panode achieves strong generation quality and exploration efficiency.

## 1 INTRODUCTION

Turning a single image into an explorable world has attracted increasing interest in visual content creation and immersive applications. Such a system must progressively expand the world as the viewpoint moves while keeping appearance and geometry consistent across observations.

Existing methods extend a single image through perspective video generation (Liang et al., 2025; Sun et al., 2025; Schneider et al., 2025). Constrained by a narrow field of view, these methods must synthesize several views at each location to complete the scene, which inflates generation cost and introduces cross-view inconsistencies. Panoramas offer full angular coverage from a single position, and recent work leverages panoramic videos for long-horizon world generation (Yang et al., 2025; Liu et al., 2026; HY-World et al., 2026; Team et al., 2025a). Adjacent panoramic frames in



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.

such panoramic videos, however, largely overlap in prior context and thus raise concerns about information efficiency and computational cost. Their wide field of view further complicates video spatial consistency. Such pipelines also treat dense video synthesis and 3D reconstruction as separate sequential stages, where reconstruction struggles to handle inconsistencies introduced by generation.

Our key insight is that world exploration can be recast as sparse panoramic node generation: instead of densely sampling frames along a trajectory, we build up the world from a sparse sequence of panoramic nodes, each producing a full 360° observation at a viewpoint. This avoids both the redundancy of panoramic video and the cross-view inconsistency of perspective synthesis, as shown in Figure 2.




Figure 2: **Panoramic nodes vs. perspective video.** A panoramic node covers 360° from one observation, avoiding the dense sampling and frame-wise error accumulation of perspective video.

To this end, we propose Panode, a framework for world exploration via panoramic node generation. Given a single input panorama, Panode initializes a 3D Gaussian scene as its persistent world state. At each node, a generative model jointly completes the panoramic RGB and depth, and the reconstruction module lifts them into 3D Gaussians (Kerbl et al., 2023) and refines them via sliding-window optimization, with an adjacency graph to keep the world consistent across nodes. Given the updated world state, the planner selects the next node from the current surface geometry, closing the loop for on-demand world expansion driven by interaction.

Our main contributions are:

- **A world exploration framework via panoramic node generation.** We propose Panode, a framework for world exploration that recasts the task as generating a sparse sequence of panoramic nodes, enabling efficient scene expansion from a single image.
- **Joint panoramic RGB-D view completion.** We train a completion model that jointly recovers panoramic RGB and depth at each node.
- **Adjacency-aware incremental update.** An incremental Gaussian update that couples neighboring Gaussians through an adjacency graph, stabilizing local surface structure and mitigating error accumulation across successive nodes.

## 2 Related Work

### 2.1 Panorama Generation

Panorama generation aims to synthesize a 360° view from text or limited-field-of-view inputs. PanoDiffusion (Wu et al., 2023) adapts the generative prior of a pretrained diffusion model to panorama completion, and the recent HY-Pano 2.0 (HY-World et al., 2026) further supports generating high-fidelity ERP panoramas from text or a single perspective image. Panoramic video generation further requires consistency across frames over the full spherical field of view (Yang et al., 2025; Liu et al., 2026; Wang et al., 2024; Liu et al., 2025). These methods focus on producing 2D panoramic images or videos, and do not directly yield a world that can be freely explored.

### 2.2 World Generation

Early works build an explorable world by generating a panorama and reconstructing it into 3D. PanoDreamer (Paliwal et al., 2025) and One2Scene (Wang et al., 2026) generate panoramas from image or text and use them for 3D reconstruction; in particular, One2Scene reprojects panoramic content into multiple perspective views or sparse anchors before handing them to a multi-view reconstruction module. Constrained to a single panoramic observation, these methods cover only the immediate surroundings and cannot support long-range exploration.

Recent methods further combine video generation with 3D reconstruction to enlarge the accessible region. Perspective-video-based methods synthesize multi-view sequences along a given camera trajectory and then recover a 3D representation through per-scene iterative Gaussian optimization (Zhang et al., 2025; Sun et al., 2025). With advances in feed-forward reconstruction (Charatan

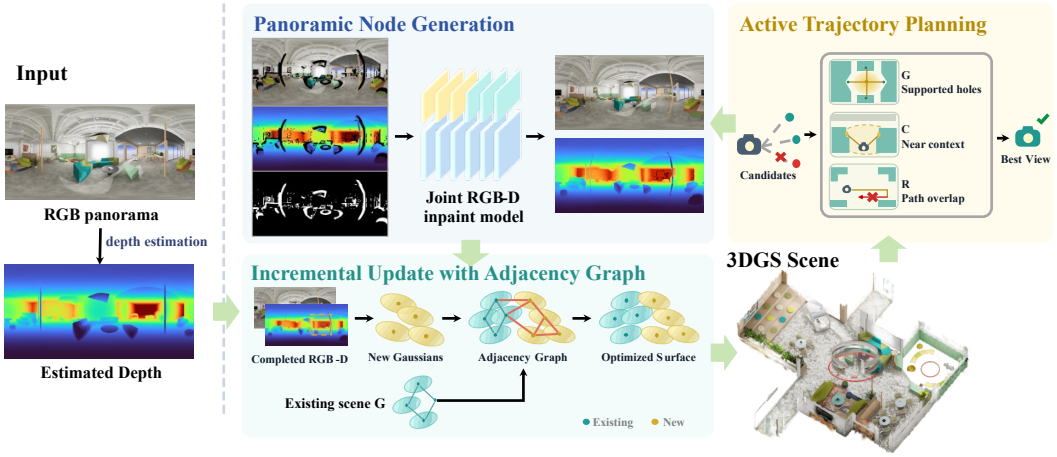



Figure 3: Overview of Panode. At each node, a joint RGB-D completion model turns the rendered partial view into a full panoramic observation, which is then merged into the Gaussian scene by an adjacency-aware incremental update. An active planner selects the next node, closing the loop of panoramic node generation.

et al., 2024; Chen et al., 2024), Wonderland (Liang et al., 2025) further predicts 3DGS directly from the latent representations of a video diffusion model, and VideoScene (Wang et al., 2025) combines video generation with sparse-view reconstruction to reduce construction cost. Constrained by the perspective field of view, the spatial coverage of such methods typically depends on the chosen trajectory and video length. Panoramic-video-based methods expand the spatial coverage: Hunyuan-World (Team et al., 2025a), Matrix-3D (Yang et al., 2025), and OmniRoam (Liu et al., 2026) couple trajectory-controlled panoramic video generation with panoramic 3D reconstruction. These methods typically treat video generation and 3D reconstruction as sequential stages over dense frames, leading to redundant computation and error accumulation.

## 3 METHOD

Given a single RGB panorama $I_0 \in [0,1]^{H\times W\times 3}$, our goal is to synthesize panoramic observations at different locations for efficient and consistent world exploration. We denote the target scene by a Gaussian representation $\mathcal{G}$, viewpoints by $\mathbf{c}_i \in \mathbb{R}^3$, and their panoramic RGB-D observations by $\mathcal{O}_i = (I_i, D_i)$, where $I_i$ is the RGB image and $D_i$ the radial depth map. Observations across viewpoints must remain consistent to jointly support the reconstruction of a renderable scene $\mathcal{G}$. An overview of Panode is illustrated in Figure 3.

### 3.1 PRELIMINARIES

**Panoramic image generation.** Panoramas are typically represented in equirectangular projection (ERP), which maps the full spherical field of view onto a rectangular image plane. Recent panoramic generators (HY-World et al., 2026) adopt flow matching in a VAE latent space. Let $z_0$ be the clean latent of a panorama and $\epsilon \sim \mathcal{N}(0, I)$ a Gaussian noise sample; along a linear schedule $\sigma_\tau \in [0,1]$, the noisy latent and the target velocity are defined as

$$z_\tau = (1-\sigma_\tau)z_0 + \sigma_\tau \epsilon, \qquad v^*(z_0,\epsilon) = \epsilon - z_0. \tag{1}$$

A network $v_\theta(z_\tau, \tau; c)$ is trained to regress this velocity field under a condition $c$, such as text or a reference image, and the clean latent is recovered by ODE integration from noise at inference.

**ERP-based 3D reconstruction.** ERP maps longitude $\theta \in [-\pi, \pi)$ and latitude $\phi \in [-\pi/2, \pi/2]$ to image coordinates, so each pixel corresponds to a ray on the unit sphere $\mathbf{r}(\theta,\phi) = (\cos\phi\sin\theta, \sin\phi, \cos\phi\cos\theta)^\top$. Given a camera center $\mathbf{c}$ and the radial depth $D(\theta,\phi)$ along the ray, the pixel is lifted to a 3D point $\mathbf{x} = \mathbf{c} + D(\theta,\phi)\mathbf{r}(\theta,\phi)$. From these points a 3DGS representation (Kerbl et al., 2023) can be optimized to recover details. To render pixel $(\theta,\phi)$, Gaussians hit by

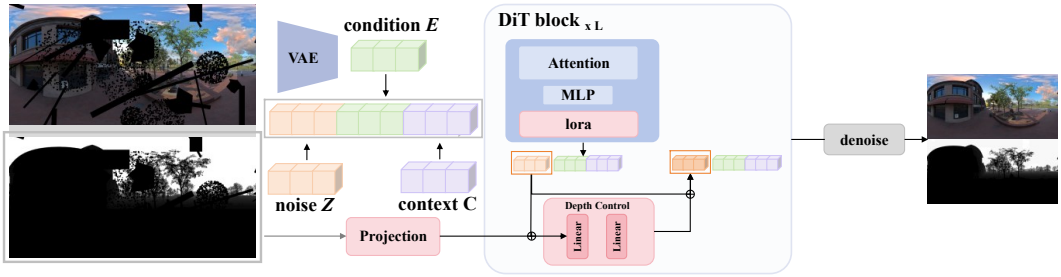



Figure 4: Joint panoramic RGB-D completion model. Noisy, condition, and context latents are encoded and concatenated as input to the DiT. The depth of the edit condition fed into the depth-control branch, injecting residuals into the generation tokens layer by layer. At inference, denoising recovers the complete RGB and depth.

a sampled spherical ray are sorted in depth order and their colors are alpha-blended as

$$C(\theta, \phi) = \sum_{i=1}^{N} c_i \, \alpha_i \prod_{j<i} (1 - \alpha_j), \tag{2}$$

where $c_i$ and $\alpha_i$ denote the color and opacity of the $i$-th Gaussian along the ray.

### 3.2 Panoramic Node Generation

Given the input panorama $I_0$ and its estimated depth, we initialize a 3D Gaussian scene $\mathcal{G}$ by lifting pixels to points and briefly optimizing the Gaussians against $I_0$. At a new viewpoint, rendering $\mathcal{G}$ produces a partial RGB panorama $\tilde{I}$, depth $\tilde{D}$, and a mask $M$ marking pixels to be completed. Completing RGB alone leaves the geometry of new content underdetermined, while completing appearance and depth in separate branches easily introduces cross-modal inconsistency. We therefore finetune a panoramic generator to jointly complete RGB and depth in a unified representation, augmented with a dedicated depth-control branch, as illustrated in Figure 4.

**Joint RGB-D representation.** We form a joint RGB-D image from the partial observation $(\tilde{I}, \tilde{D})$ by normalizing the partial depth and replicating it across three channels as $g(\tilde{D})$, then concatenating with $\tilde{I}$ along the height axis as $J = \mathrm{Concat}_H(\tilde{I},\, S,\, g(\tilde{D}))$, where $S$ is a separator band. $J$ is VAE-encoded into the edit condition $E$, and previously observed RGB-D panoramas are independently encoded as context $C$. Together with a noise latent $Z$, they are concatenated along the sequence dimension as $[Z; E; C]$, so that joint attention in each DiT layer lets the completion leverage both the currently visible regions and the history. We then integrate $v_\theta$ from $Z$ via an ODE solver to recover the clean latent, decode it and split into RGB and depth halves; the depth channels are averaged into a single-channel normalized map, and a robust scale-and-shift alignment against the known-region depth yields the final RGB-D observation.

**Layer-wise depth control.** Since the depth signal in the joint image is easily dominated by the visually richer RGB stream, we add a dedicated depth-control branch to strengthen depth generation. The depth part of the edit condition $E$ passes through a projection layer to form the depth condition $q_D$, and after every DiT block the branch predicts a residual that updates only the generation stream:

$$h_{\mathrm{gen}} \leftarrow h_{\mathrm{gen}} + C_\ell(h_{\mathrm{gen}}, q_D), \tag{3}$$

where $C_\ell$ is the control branch of layer $\ell$ with a zero-initialized output projection, and $h_{\mathrm{gen}}$ denotes the hidden features of the noise latent $Z$, so that the condition tokens $C$ and $E$ remain untouched.

**Training.** We construct partial-full pairs by applying various masks with different distributions to full RGB-D panoramas; the masks cover boundary gaps, stripes, cracks, and speckle holes to simulate rendering deficiencies. The full RGB-D image is VAE-encoded into the clean latent $Z_0$, from which we sample the noisy latent under the flow-matching schedule of Section 3.1, $Z_\tau = (1 - \sigma_\tau) Z_0 + \sigma_\tau \epsilon$, and take it as $Z$ in place of the pure noise used at inference. We freeze the VAE, apply LoRA finetuning to the DiT, and fully train the depth-control branch.

The velocity regressor is conditioned on the edit condition $E$ and the depth control signal $q_D$. The training loss couples a global flow-matching term with a mask-focused term over the missing region:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda \mathcal{L}_{\text{hole}}. \tag{4}$$

$\mathcal{L}^k_{\text{RGB}}$ is the L1 error between rendered and target RGB, augmented with a D-SSIM term to preserve local structure. $\mathcal{L}^k_D$ measures the L1 discrepancy between rendered radial depth and the completed depth $D_k$, so that the Gaussians remain grounded on the observed surfaces. $\mathcal{L}^k_A$ penalizes $|\bar{\alpha}_k - 1|$ over the rendered accumulated alpha $\bar{\alpha}_k$, discouraging translucent gaps and ensuring the surface is fully covered. $\mathcal{L}^k_S$ uses the cosine dissimilarity $1 - \widehat{\mathbf{n}}_G \cdot \widehat{\mathbf{n}}_D$ to align the rendered normal $\widehat{\mathbf{n}}_G$ with the normal $\widehat{\mathbf{n}}_D$ derived by finite differences on the rendered depth, which keeps neighboring Gaussians flat on the same surface and suppresses spiky outliers. Together with $\mathcal{L}_{\text{adj}}$, these four terms couple photometric fidelity, geometric grounding, opacity closure, and local flatness into a single joint objective. Since we operate on ERP panoramas, all pixel-level losses are further weighted by the spherical area factor $\cos\phi$ to remove the pole-oversampling bias.

### 3.3 Incremental Update with Adjacency Graph

Given a new RGB-D panorama, this module merges it into the current Gaussian scene $\mathcal{G}$. Since pixel-level losses supervise each Gaussian's projected appearance independently and impose no constraint on their geometric relations, few-step optimization accumulates error across incremental updates. We address this by imposing an adjacency graph that ties neighboring Gaussians together during optimization.

**Gaussian initialization.** We initialize $\mathcal{G}$ from the input RGB-D as a set of thin, surface-aligned Gaussians. We set each Gaussian's tangential scale to the spherical arc length subtended by its pixel on the ERP panorama, and its normal scale to a small fraction of the tangential scale, so that the Gaussian conforms to the underlying surface.

**Gaussian adjacency graph.** We build an adjacency graph over Gaussians, initialized by the four-neighborhood on the ERP grid. At each subsequent node, we append edges for newly added Gaussians and further retrieve existing Gaussians within distance and normal-angle thresholds in 3D space using a KD-tree, yielding an adjacency graph $E$ that covers both intra-frame and cross-frame neighbors. When one endpoint is removed by subsequent artifact pruning, all edges attached to it are removed together.

For each edge $(i, j)$, we quantify endpoint alignment by the geometric mean of the alpha each Gaussian contributes at the other's center, with $\alpha_{i\to j} = o_j \exp(-\frac{1}{2}\Delta_{ij}^\top \Sigma_j^{-1} \Delta_{ij})$ and $\alpha_{j\to i}$ defined symmetrically. The adjacency affinity is

$$\rho_{ij} = -\log \sqrt{\alpha_{i\to j}\, \alpha_{j\to i}}, \tag{5}$$

where $\boldsymbol{\mu}_i, \Sigma_i, o_i$ denote the center, covariance, and opacity of Gaussian $i$, and $\Delta_{ij} = \boldsymbol{\mu}_j - \boldsymbol{\mu}_i$ (see Appendix A.3 for the full derivation). A smaller $\rho_{ij}$ indicates that the two Gaussians lie within each other's response range with sufficient opacity.

Using the value $\rho^0_{ij}$ at edge creation as a soft anchor, we regularize Gaussian optimization by an adjacency-consistency loss:

$$\mathcal{L}_{\text{adj}} = \frac{\sum_{(i,j)\in E} \lambda_{ij} \left[\log(\rho_{ij}/\rho^0_{ij})\right]^2}{\sum_{(i,j)\in E} \lambda_{ij}}, \tag{6}$$

where $\lambda_{ij}$ normalizes over Gaussian area and node degree. Unlike hard constraints that lock Gaussians onto a plane, $\mathcal{L}_{\text{adj}}$ leaves each Gaussian free to fit the reconstruction while requiring neighbors to move together, thereby stabilizing the seam structure between Gaussians.

**Joint optimization.** At each node, old Gaussians whose projected centers fall in the artifact region $B_t$ (details in Appendix A.4) are pruned by visibility, and new Gaussians are instantiated by per-pixel back-projection within $M_t$ with scales set as above. We then jointly optimize all Gaussians with a small number of iterations over a sliding window of frames $\{\mathcal{O}_k\}^t_{k=t-w}$:

$$\mathcal{L}_t = \sum_{k=t-w}^{t} \left(\mathcal{L}^k_{\text{RGB}} + \mathcal{L}^k_D + \mathcal{L}^k_A + \mathcal{L}^k_S\right) + \mathcal{L}_{\text{adj}}. \tag{7}$$

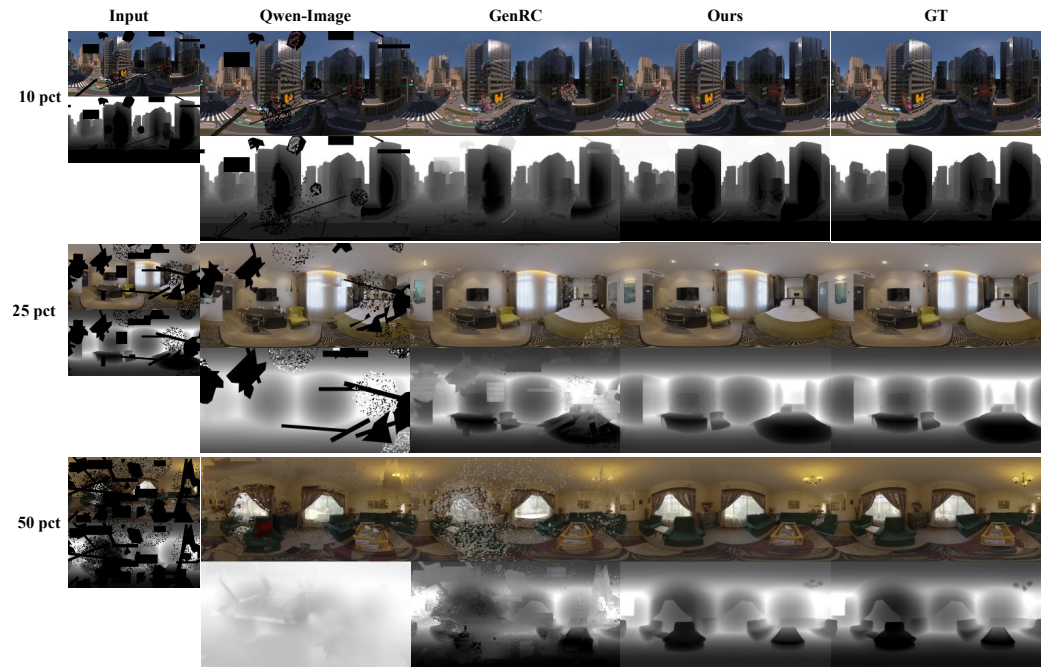



Figure 5: Qualitative completion results at $10\%/25\%/50\%$ missing ratios. Our joint model keeps RGB and depth consistent even at high missing ratios.

$\mathcal{L}^k_{\mathrm{RGB}}$ and $\mathcal{L}^k_D$ are the RGB and radial-depth reconstruction errors. $\mathcal{L}^k_A$ encourages the accumulated alpha to approach 1 to ensure surface coverage. $\mathcal{L}^k_S$ aligns the rendered normal $\widehat{\mathbf{n}}_G$ with the normal $\widehat{\mathbf{n}}_D$ derived from the rendered depth, keeping Gaussians flat on the surface. Since we operate on ERP panoramas, all pixel-level losses are weighted by the spherical area factor $\cos\phi$ to remove the pole-oversampling bias.

### 3.4 Active Trajectory Planning

We select the next node to maximize an exploration reward under a geometric feasibility constraint. Given the current Gaussian state $\mathcal{S}_t$ and a candidate move direction $a$, the planner scores each candidate by a completion-benefit gain $\mathcal{R}_{\mathrm{gain}}(a)$ gated by a feasibility indicator $\mathcal{C}(a \mid \mathcal{S}_t) \in \{0, 1\}$, and picks the action of the highest reward:

$$a_t^\star = \arg\max_a \; \mathcal{R}_{\mathrm{gain}}(a) \cdot \mathbb{1}[\mathcal{C}(a \mid \mathcal{S}_t) = 1]\,. \tag{8}$$

**Scale-adaptive candidates.** From the sky-excluded ERP-weighted median depth $D$ of the current viewpoint and a fixed angular shift $\alpha$, we set a scene-adaptive base step $s = D \tan\alpha$ and retain $\{0.5, 1.0, 1.5\}\, s$ as candidate poses along each action $a$.

Using the surface map maintained in Section 3.3, we densely sample points along each candidate path and set $\mathcal{C}(a \mid \mathcal{S}_t) = 1$ only if every sample keeps a clearance of at least $\delta$ from the nearest observed surface. If no candidate is feasible at the current scale, we fall back to a step-shrinking strategy: the base step $s$ is halved, and this halving repeats until at least one candidate becomes feasible or the step falls below a minimum length.

**Exploration reward.** For each candidate $a$ that passes the feasibility check, we render a low-resolution preview at that pose to obtain a partial panorama and a hole mask. We define the exploration reward as

$$\mathcal{R}_{\mathrm{gain}}(a) = G(a) \cdot C(a) \cdot (1 - 0.1\, R(a)), \tag{9}$$

where $G$, $C$, and $R$ denote the supported-hole gain, the near-context coverage, and the redundancy penalty, respectively. It favors viewpoints that reveal new content while preserving sufficient geometric context.

Table 1: Quantitative results of panoramic RGB completion. Our method consistently outperforms prior approaches across all mask ratios in PSNR, LPIPS, AbsRel and $\delta_1$.

| Method | 10% miss. | | | | 25% miss. | | | | 50% miss. | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | LPIPS ↓ | AbsRel ↓ | $\delta_1$ ↑ | PSNR ↑ | LPIPS ↓ | AbsRel ↓ | $\delta_1$ ↑ | PSNR ↑ | LPIPS ↓ | AbsRel ↓ | $\delta_1$ ↑ |
| Qwen-Image | 17.36 | 0.216 | 1.883 | 0.129 | 13.03 | 0.434 | 2.610 | 0.098 | 10.05 | 0.651 | 2.898 | 0.096 |
| GenRC | 25.98 | 0.089 | 0.349 | 0.811 | 22.31 | 0.204 | 0.530 | 0.792 | 18.85 | 0.383 | 0.515 | 0.744 |
| **Ours** | **39.70** | **0.008** | **0.100** | **0.914** | **35.75** | **0.018** | **0.103** | **0.918** | **32.77** | **0.034** | **0.101** | **0.916** |

$G$ quantifies the new content revealed by the candidate. For each pixel in the hole mask, we locate the nearest valid foreground neighbors on both sides along the horizontal and vertical directions, and denote their 3D distance by $d$. The support strength is defined as $w = \max(1 - d/(2s_0), 0)$, where $s_0 = D_0 \tan\alpha$ is the reference step scale estimated at the first frame. We take the larger of the horizontal and vertical support strengths as the per-pixel value $w(p)$, and aggregate it under latitude weighting:

$$G(a) = \frac{\sum_{p\in\mathcal{H}(a)} w(p)\cos\phi_p}{\sum_p \cos\phi_p}, \tag{10}$$

where $\mathcal{H}(a)$ is the set of hole pixels in the candidate preview. This prevents wide holes or distant gaps from earning high rewards purely by mask area. The near-context coverage $C$ complements $G$ by rewarding nearby, reusable geometry: we collect foreground pixels $\mathcal{F}(a)$ whose depth lies below the threshold $d_c = s_0/\tan\alpha$, and take their latitude-weighted area fraction similarly. Finally, $R$ discourages retraversal along previous paths: it is the fraction of the current step overlapping with prior collinear steps, pushing the planner toward unexplored directions.

# 4 Experiments

## 4.1 Experimental Setup

**Implementation details.** We fine-tune the joint RGB-D completion model from Qwen-Image-Edit-2509 (Wu et al., 2025a). The VAE and the multimodal condition encoders are frozen; we apply LoRA fine-tuning to the DiT with rank/alpha both set to 64, and fully train the depth-control branch. Panoramas are represented in ERP at a resolution of $640\times1280$. We use AdamW with a learning rate of $10^{-5}$ and weight decay $0.01$. Standard inference performs $40$ ODE steps and can be switched to a $4$-step Lightning schedule (Lin et al., 2024). Training is carried out on $8\times$RTX PRO 6000 GPUs. The Gaussian scene is optimized for $800$ steps at initialization and $100$ steps per incremental update, with a keyframe window of length $4$.

**Datasets.** For the joint RGB-D completion model, we construct a dataset from three panoramic datasets: DAP (Lin et al., 2026), Matterport3D360 (Chang et al., 2017), and Omni360-Scene (Ge et al., 2026), covering both indoor and outdoor environments. The collection is split into a training set used to finetune the completion model and a validation set of 60 panoramas (20 per dataset).

For world exploration, we build the benchmark on the InteriorGS dataset (Team et al., 2025b) following the protocol of OmniRoam (Liu et al., 2026). InteriorGS provides high-quality indoor 3D scenes, each equipped with a renderable 3DGS representation that can produce RGB panoramas along reference trajectories as well as independent probe views. We select $20$ scenes from it and sample two trajectories per scene, yielding $40$ trajectories for evaluation.

**Baselines.** For the completion model, we compare against GenRC (Li et al., 2024) and Qwen-Image-Edit (Wu et al., 2025a). GenRC obtains complete RGB-D observations by performing RGB completion on perspective views coupled with depth estimation; we adapt it to panoramic input by splitting the panorama into perspective tiles. Qwen-Image-Edit is a general large-scale image-editing diffusion model, and we complete RGB and depth separately with the mask as input. All three methods use identical images, masks, prompts, and evaluation resolution, and no context is provided to any method.




Figure 6: Qualitative Comparison of World Exploration. Our method produces more consistent results along the trajectory on both indoor and outdoor scenes.

Table 2: Quantitative Comparison of World Exploration. Our method outperforms video-based baselines in PSNR, SSIM, and LPIPS while achieving the shortest runtime.

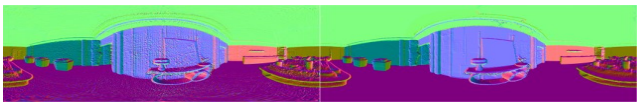
| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Time / trajectory(s) ↓ |
|---|---|---|---|---|
| Matrix-3D | 19.85 | 0.707 | 0.267 | 1636.85 |
| OmniRoam | 16.78 | 0.608 | 0.319 | 657.77 |
| **Ours** | **20.68** | **0.735** | **0.186** | **153.85** |

For world exploration, we compare against two panoramic-video-based baselines, Matrix-3D (Yang et al., 2025) and OmniRoam (Liu et al., 2026). Matrix-3D couples trajectory-controlled panoramic video generation with panoramic 3D reconstruction; OmniRoam further improves long-horizon panoramic roaming video generation. All methods start from the same RGB panorama and follow the same InteriorGS reference trajectory. Since our method observes at sparse locations, for a fair comparison we parameterize each trajectory by a constant velocity, treat the input as position $s = 0$, and take the normalized arc lengths $s_j = j/6$ ($j = 1, \ldots, 6$) as six scoring positions. Panode incrementally generates panoramic nodes at each position, while Matrix-3D and OmniRoam generate full videos from which only the frames aligned with the six target poses are evaluated. All outputs are converted to the same ERP orientation and resolution.

### 4.2 Comparison Results

**Panoramic RGB-D completion.** We compare Panode, GenRC, and Qwen-Image-Edit on the 60-panorama test set constructed above at three missing ratios (10%, 25%, and 50%). PSNR, LPIPS, AbsRel, and $\delta_1$ are computed on the full ERP panorama. As reported in Table 1, Panode significantly outperforms the baselines under all ratios in both RGB and depth metrics; in particular, at the high missing ratio (50%), it still produces stable RGB appearance and structurally reasonable depth. Figure 5 gives qualitative comparisons under different missing ratios.

**World exploration.** We compare Panode against Matrix-3D and OmniRoam on the InteriorGS benchmark. All outputs are resampled to $1024 \times 512$ ERP for evaluation. We report full-panorama PSNR, SSIM, and LPIPS, together with the generation time per trajectory. Note that Panode yields a renderable Gaussian world directly, while Matrix-3D and OmniRoam only produce panoramic videos; the reported time therefore reflects generation only, and the two baselines would incur additional cost to obtain a renderable representation.



w/o Adjacency-Graph w Adjacency-Graph

Figure 7: Normal comparison of the adjacency ablation. The full module yields markedly smoother surfaces.

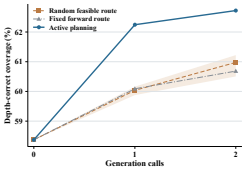



Figure 8: Ablation of active planning. The active planner achieves higher coverage.

Table 3: Ablation of the adjacency-graph module. The full module outperforms both ablated variants.

| Setting | Holes ↓ | AbsRel ↓ | PSNR ↑ |
|---|---|---|---|
| w/o graph module | 11.85% | 10.99% | 23.27 |
| w/o $\mathcal{L}_{\text{adj}}$ only | 9.10% | 10.96% | **23.30** |
| **Full** | **6.61%** | **10.96%** | 23.28 |

As reported in Table 2, Panode attains the highest PSNR, SSIM, and LPIPS while requiring the shortest runtime per trajectory, outperforming both panoramic-video baselines across all metrics. This indicates that generating panoramic nodes coupled with the adjacency-aware incremental update, is sufficient to preserve appearance and geometry consistency along the trajectory. Consistent with this observation, the qualitative results in Figure 6 show that Panode maintains stable structure and texture on frames far from the starting viewpoint. Together with the runtime comparison, these results demonstrate that sparse panoramic nodes provide a more favorable quality–cost trade-off for world exploration.

### 4.3 Ablation Study

**Adjacency graph.** On the same scene, we compare three settings: removing the entire adjacency-graph module, keeping the graph but disabling $\mathcal{L}_{\text{adj}}$, and the full module. We evaluate on held-out views and report the fraction of uncovered holes, the mean relative depth error, and spherically weighted PSNR. As shown in Table 3, the full module reduces uncovered holes by nearly a half against the graph-free variant, while retaining comparable AbsRel and PSNR, indicating that the adjacency constraint mainly acts as a structural regularizer rather than sacrificing photometric fidelity. Figure 7 further visualizes the rendered normals on the same views, where the adjacency loss visibly suppresses local surface fragmentation and produces smoother, better-oriented surfaces that pixel-level metrics alone do not fully expose.

**Active planning.** Fixing the completion model, the incremental update, and the total node budget, we swap only the trajectory selector: a random policy sampling uniformly from the feasibility-filtered candidates, a fixed-forward policy taking the frontal candidate whenever feasible, and our active planner. All three share the same candidate set, feasibility check, and step-shrinking fallback, so the comparison targets reward-driven direction selection alone. We plot depth-correct coverage against node count on the same held-out views. As shown in Figure 8, the active planner consistently attains higher coverage under the same budget, indicating stronger exploration efficiency.

## 5 Conclusion

We present Panode, a framework for world exploration through incremental panoramic node generation. Starting from a single RGB panorama, Panode builds up the world one panoramic node at a time by coupling joint RGB-D completion, adjacency-aware Gaussian updates, and active trajectory planning within a shared panoramic representation. Experiments on panoramic RGB-D completion and world exploration show that Panode generates renderable Gaussian worlds with consistent appearance and geometry along the trajectory, while remaining more efficient than panoramic-video-based baselines. We hope that the sparse-node view of world exploration and the panoramic node abstraction can serve as a compact foundation for future interactive and open-ended world generation systems.

### AI USE STATEMENT

In this work, we used generative AI tools to polish the manuscript text and equations, tidy up code, and generate parts of the visualization code. We did not use generative AI tools for research ideation, method design, or the generation or analysis of experimental data, and the remaining tasks with required disclosure are not applicable to this work. In addition, we used generative AI tools to assist with literature search. We have reviewed all AI-assisted work: all AI-suggested references were manually verified by the authors against the original sources to confirm their authenticity and relevance, and all AI-polished text and equations were checked sentence by sentence by the authors to ensure accuracy of exposition and truthfulness of claims. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

Generated environments may hallucinate spatial structure and should not be interpreted as verified reconstructions of real places. Use of real-world panoramas requires attention to identifiable people and private spaces.

### REPRODUCIBILITY STATEMENT

All datasets used in this work are open-source, and their processing pipelines are described in Section 4. The method description and Appendix A document the implementation and configuration in detail, from which the results can be reproduced. The code and pretrained weights will be partially released upon publication.

## REFERENCES

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386. Springer, 2024.

Xian Ge, Yuling Pan, Yuhang Zhang, Xiang Li, Weijun Zhang, Dizhe Zhang, Zhaoliang Wan, Xin Lin, Xiangkai Zhang, Juntao Liang, Xiangtai Li, WenJie Jiang, Bo Du, Ming-Hsuan Yang, and Lu Qi. AirSim360: A panoramic simulation platform within drone view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26931–26940, 2026.

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.

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

Ming-Feng Li, Yueh-Feng Ku, Hong-Xuan Yen, Chi Liu, Yu-Lun Liu, Albert YC Chen, Cheng-Hao Kuo, and Min Sun. Genrc: Generative 3d room completion from sparse image collections. In *European Conference on Computer Vision*, pp. 146–163. Springer, 2024.

Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 798–810. IEEE, 2025.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26835–26844, 2026.

Jinxiu Liu, Shaoheng Lin, Yinxiao Li, and Ming-Hsuan Yang. Dynamicscaler: Seamless and scalable video generation for panoramic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6144–6153. IEEE, 2025.

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*, 2026.

Avinash Paliwal, Xilong Zhou, Andrii Tsarov, and Nima Kalantari. Panodreamer: Optimization-based single image to 360 3d scene with diffusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–10, 2025.

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.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhu, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with decoupled video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13695–13706. IEEE, 2025.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025a.

MTI SpatialVerse Research Team et al. Interiorgs: A 3d gaussian splatting dataset of semantically labeled indoor scenes, 2025b.

Hanyang Wang, Fangfu Liu, Jiawei Chi, and Yueqi Duan. Videoscene: Distilling video diffusion model to generate 3d scenes in one step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16475–16485. IEEE, 2025.

Pengfei Wang, Liyi Chen, Zhiyuan Ma, Yanjun Guo, Guowen Zhang, and Lei Zhang. One2scene: Geometric consistent explorable 3d scene generation from a single image. *arXiv preprint arXiv:2602.19766*, 2026.

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

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.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025b.

Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. Panodiffusion: 360-degree panorama outpainting via diffusion. *arXiv preprint arXiv:2307.03177*, 2023.

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.

Shengjun Zhang, Jinzhao Li, Xin Fei, Hao Liu, and Yueqi Duan. Scene splatter: Momentum 3d scene generation from single image with video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6089–6098. IEEE, 2025.

# A Implementation Details

## A.1 Hyperparameters

Table 4 lists the hyperparameters of Panode, grouped by the three modules of Section 3.

Table 4: Reference hyperparameters used in Panode.

| Setting | Value |
|---|---|
| *Panoramic RGB-D completion* | |
| ERP panorama resolution | $640 \times 1280$ |
| Separator band $S$ width | 16 pixels |
| DiT LoRA rank / alpha | 64 / 64 |
| Learning rate / weight decay | $10^{-5}$ / 0.01 |
| ODE steps (standard / Lightning) | 40 / 4 |
| Hole loss weight $\lambda$ | 1.0 |
| *Incremental update* | |
| Gaussian optimization steps (init / per update) | 800 / 100 |
| Sliding-window size $w$ | 4 |
| Normal-to-tangential scale ratio | 0.6 |
| Surface loss $\mathcal{L}_S$ weight | 0.05 |
| *Active trajectory planning* | |
| Preview render resolution | $160 \times 320$ |
| Candidate directions | 6 |
| Step-length multipliers of $s$ | $\{0.5, 1.0, 1.5\}$ |
| Angular shift $\alpha$ | $30^\circ$ |
| Clearance $\delta$ | $0.1\, s_0$ |

## A.2 Training Mask Construction

To cover missing regions of varying scales and shapes in panoramic rendering, we construct incomplete RGB-D samples using a mixed-shape mask. We first sample a target missing ratio $r \sim \mathcal{U}(0.10, 0.60)$, and then combine six types of binary primitives: boundary blocks, wavy lines, parallel stripes, wavy bands, scattered points, and boundary trails. This design reflects the observation that holes in panoramic novel views concentrate near the poles and often appear as elongated stripes. Based on our experiments, the sampling weights are set to $0.10$, $0.28$, $0.25$, $0.25$, $0.06$, and $0.06$. The mask is the union of the sampled primitives, and sampling stops once the coverage reaches the target ratio or the primitive count reaches the limit. To preserve panoramic seam continuity, polylines and polygons that cross the left and right boundaries are drawn with horizontal periodicity.

The same mask $M$ is applied to both RGB and depth: missing RGB pixels are filled with black and missing depth values are set to zero, producing a spatially consistent incomplete RGB-D editing condition. The training supervision retains the complete RGB-D. The joint mask used for loss computation replicates the same $M$ across the RGB and depth halves, zeros out the separator band, and is mapped to the latent space via nearest-neighbor interpolation. Representative examples are shown in Figure 9.

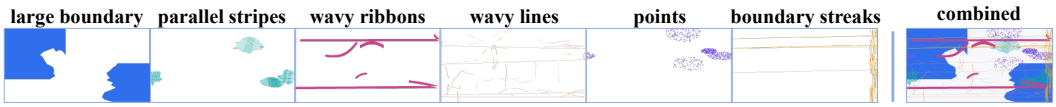



Figure 9: Examples of training masks.

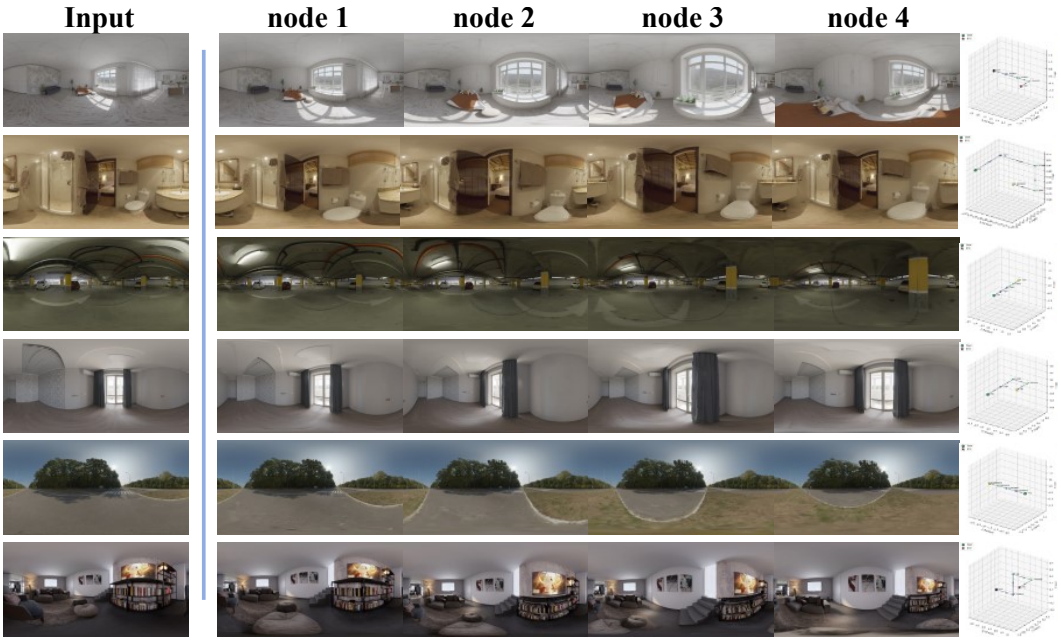



Figure 10: Additional incremental 3D generation results. Novel views at each planned node across indoor and outdoor scenes.

### A.3 Derivation of the Adjacency Affinity

In 3DGS, each Gaussian is parameterized by a center $\boldsymbol{\mu}$, a covariance $\Sigma$, and an opacity $o$, and its alpha response at an arbitrary point $\mathbf{x}$ is

$$\alpha(\mathbf{x}) = o\,\exp\!\big(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\big). \tag{11}$$

Evaluating this response at the center of a neighboring Gaussian gives the alpha contributed by Gaussian $j$ at the center of Gaussian $i$,

$$\alpha_{i\to j} = o_j\,\exp\!\big(-\tfrac{1}{2}\Delta_{ij}^{\top}\Sigma_j^{-1}\Delta_{ij}\big), \tag{12}$$

$$\Delta_{ij} = \boldsymbol{\mu}_j - \boldsymbol{\mu}_i, \tag{13}$$

and symmetrically $\alpha_{j\to i} = o_i \exp\!\big(-\tfrac{1}{2}\Delta_{ij}^{\top}\Sigma_i^{-1}\Delta_{ij}\big)$. Their geometric mean

$$\sqrt{\alpha_{i\to j}\cdot\alpha_{j\to i}} = \sqrt{o_i\,o_j}\cdot\exp\!\Big(-\tfrac{1}{4}\big(\Delta_{ij}^{\top}\Sigma_i^{-1}\Delta_{ij} + \Delta_{ij}^{\top}\Sigma_j^{-1}\Delta_{ij}\big)\Big) \tag{14}$$

provides a symmetric measure of mutual response between the two Gaussians. Taking its negative logarithm yields the adjacency affinity used in the main text (Section 3.3),

$$\rho_{ij} = -\log\sqrt{\alpha_{i\to j}\,\alpha_{j\to i}} = \tfrac{1}{4}\big(\Delta_{ij}^{\top}\Sigma_i^{-1}\Delta_{ij} + \Delta_{ij}^{\top}\Sigma_j^{-1}\Delta_{ij}\big) - \tfrac{1}{2}(\log o_i + \log o_j). \tag{15}$$

A smaller $\rho_{ij}$ corresponds to a larger geometric-mean alpha, indicating that the two centers lie within each other's response range with sufficient opacity.

### A.4 Artifact Detection

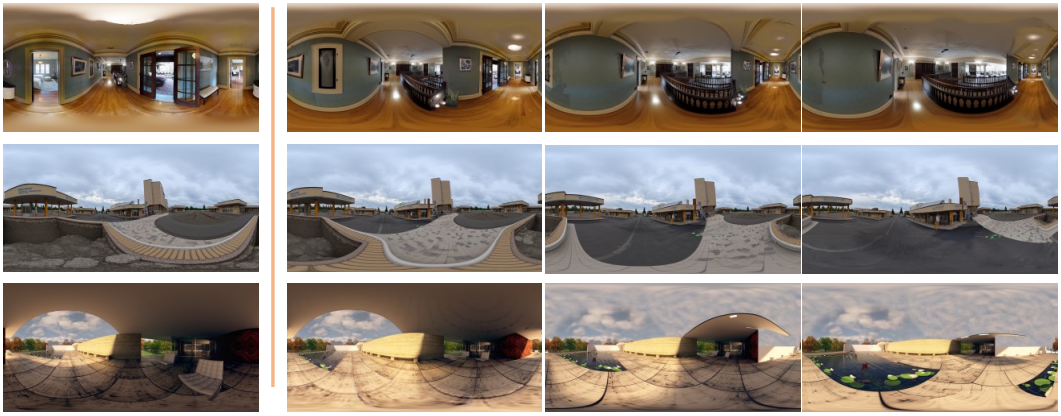



Figure 12: Results under long-range viewpoint offsets. Panode sustains consistent appearance and geometry as the viewpoint moves far from the input position.

Apart from the holes $H_t$ caused by insufficient alpha coverage, the rendered view also suffers from artifacts in poorly observed regions. Feeding these artifacts back as supervision would propagate the error into $\mathcal{G}$.




Figure 11: Artifact detection results. Detected regions are marked as red.

To address this, we employ a pretrained enhancement model (Wu et al., 2025b) to extract features from $\hat{I}_t$ and historical images $\{I_k\}_{k<t}$, and aggregate the historical features into a reference bank $\mathcal{M}$. For a patch feature of the current render at location $p$, we retrieve its $k$-NN cosine similarity $\kappa_p$ in $\mathcal{M}$ and take $S_t(p) = 1 - \kappa_p$ as the artifact score. An adaptive threshold binarizes it into the artifact mask $B_t = \mathbb{1}[S_t > \tau_a]$, which is merged with the hole mask $H_t$ to yield the regeneration mask of the current step, $M_t = B_t \cup H_t$. Figure 11 shows examples of the detected artifact regions.

# B Additional Results

## B.1 Incremental 3D Generation Results

We provide additional results across diverse indoor and outdoor scenes in Figure 10. To further probe robustness under large viewpoint changes, Figure 12 shows results under long-range viewpoint offsets, where Panode still produces geometrically consistent and visually plausible observations.

# C Limitations and Discussion

Panode depends on the quality of the generative model and the depth estimator, and geometric drift may still occur in challenging cases. Its performance on longer-range scenarios, such as driving-scale roaming, requires further validation and refinement, which we leave to future work.