GenRec: Knowing Where to Reconstruct and Where to Generate

Ata Çelen[1] **Jaewoo Jung**[1,4] **Federico Tombari**[2] **Marc Pollefeys**[1,3]
Sunghwan Hong[1] **Michael Niemeyer**[2] **Daniel Barath**[1,2]

[1]ETH Zürich [2]Google [3]Microsoft [4]KAIST

https://atcelen.github.io/GenRec/



Figure 1: **Knowing where to reconstruct and where to generate.** Observed pixels of a novel view have a single correct value. With a generative objective alone (left), they lose fine detail to latent compression and to a distribution-matching loss that spreads mass around that value. GenRec (right) adds a mask-gated reconstructive objective on observed pixels, restoring the detail while keeping the generative prior elsewhere.

Abstract

Generative novel view synthesis from sparse input images is rarely all reconstruction or all generation: pixels visible in some source view have a unique correct value modulated only by view-dependent shading, while pixels in disocclusions or beyond the captured volume admit a distribution of plausible completions. Existing generative novel-view-synthesis methods conflate these regimes under a single uniform loss, blurring the line between geometric fidelity and creative hallucinations even when scene geometry is injected through warped point clouds or projected depth. We introduce GenRec, a multi-view flow matching model that builds the reconstruction–generation split directly into its architecture, supervision, and gradient flow. Guided by an observation mask derived from the source cameras and a monocular depth estimator, a flow matching backbone jointly denoises RGB and scene-coordinate maps across all target views, while a pixel-space refinement stage restores high-frequency detail on observed pixels; the same mask gates supervision so regression signals do not contaminate the generative prior. Across RealEstate10K, DL3DV-10K, and Mip-NeRF 360, in both single-view extrapolation and two-view interpolation, GenRec attains the best reconstruction fidelity in observed regions while also surpassing purely generative baselines on perceptual quality in unobserved ones, showing the effectiveness of our approach.

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

1 Introduction

Consider two scenarios: a robot navigates an indoor space from a handful of casually captured photographs, while a headset user wanders freely through the same scene in mixed reality. The robot must plan collision-free trajectories around furniture seen from a single angle, judge whether a gap between two chairs is wide enough to traverse, and anticipate what lies beyond a half-open door before committing to a turn. The headset user expects every captured chair and painting to remain exactly in its place from any angle, and any corridor they step into beyond what was captured to fill in seamlessly around them. All of these capabilities reduce to a single problem: generative novel-view synthesis with sparse input views. Yet within a single synthesized view, different pixels carry very different burdens. Where the new view overlaps with what the camera already observed, the system must *reconstruct* the scene faithfully: the same wall, the same texture, modulated only by view-dependent effects (e.g., specular highlights) arising from the angular shift. Where the new view ventures into unobserved regions (around a corner, behind an occluder, or beyond the captured volume), the inputs alone do not pin down a single answer, and the system must *infer* the most likely completion consistent with the visible evidence and a learned prior over the world.

Today's 3D scene representations, such as 3D Gaussian Splatting (3DGS) [20] and NeRFs [29], only handle the first half. They are interpolators: dense observations yield photorealistic in-between views, but they have nothing to say about the parts of the world the camera never looked at, and degrade sharply as input becomes sparse [46, 30]. To bridge this gap, a recent line of *generative reconstruction* methods leverages image and video diffusion priors [49, 12, 56, 34, 59, 21, 42, 1, 25] to hallucinate plausible novel views from one or a few input views, which can then be fed back as pseudo-observations to a 3DGS or NeRF [48]. These methods differ mainly in how they tell the generator *where* to look: pose-only conditioning via Plücker rays [13, 12, 59, 18] suffers from metric-scale ambiguity, while warp-based conditioning that injects forward-projected depth or point-cloud renders [56, 34, 11] grounds the model in the scene but inherits warp artifacts.

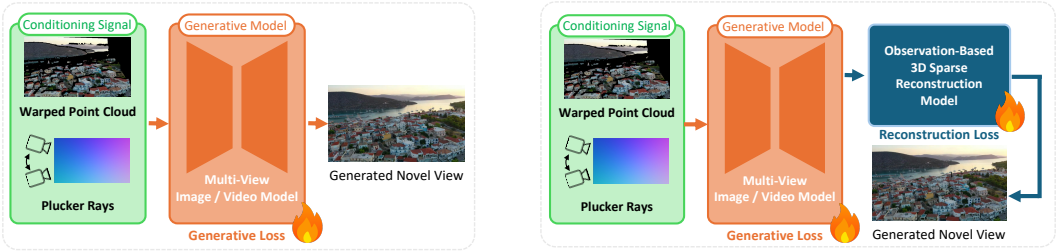
For all these methods, the generator must produce *every* pixel under a single distribution-matching loss. Yet, this overlooks a basic structural property of the task. A novel view is rarely all generation or all reconstruction; it is a combination of both. Pixels on a surface visible in the captured scene have a unique correct answer, dictated by geometry and appearance. Pixels in disocclusions or beyond the captured volume have many plausible answers, and the model must commit to a coherent one. Asking a single network to satisfy both regimes with the same supervisory signal forces a compromise: a loss that fits a distribution where many answers are valid penalizes fidelity where only one is.

We propose **GenRec**, which incorporates this separation directly into its architecture: the network *reconstructs* observed regions and *infers* unobserved ones. Given the input cameras and an off-the-shelf depth estimator, we forward-project source pixels into each target view to obtain a binary observed/unobserved mask. The generative backbone produces a full target view; pixel-space refinement then operates on the observed pixels, leveraging warped input evidence and learned 3D correspondences to recover the high-frequency detail suppressed by latent flow matching. These modules do not share parameters with the backbone and their gradients do not propagate into it. Thus, the distribution-matching objective is no longer required to anchor pixels whose answer is uniquely determined by geometry; the refinement modules supply this fidelity signal on observed pixels alone.

Our contributions are:

- A *geometry-grounded decomposition* of generative NVS that processes observed/unobserved pixels with separate modules and losses, governed by a per-pixel observation mask.
- A *multi-view flow matching backbone* that jointly denoises RGB and scene-coordinate maps, yielding cross-view consistency and a 3D correspondence signal in a single pass.
- A *pixel-space reconstruction branch* with sparse 3D cross-attention that recovers detail lost to latent compression without disturbing the generative prior in unobserved regions.

We validate **GenRec** on RealEstate10K, DL3DV-10K, and Mip-NeRF 360 under both single-view extrapolation and two-view interpolation. Across these settings, **GenRec** achieves the best reconstruction fidelity in observed regions while surpassing purely generative baselines on perceptual quality in unobserved ones, demonstrating that the separation of these two tasks yields both better reconstruction and generation quality. At the same time, our model is two orders of magnitude faster at inference than the strongest baseline.



(a) Prior generative reconstruction works (b) **GenRec**

Figure 2: **Overview of GenRec (right) vs. prior generative reconstruction methods (left).** Prior methods condition multi-view image or video models on warped point clouds or Plücker rays, supervising output pixels with one generative loss regardless of source coverage. **GenRec** factors the task by an observation mask: a generative backbone predicts every pixel, and a pixel-space reconstruction module refines the observed ones under a dedicated loss. The two pathways share neither parameters nor gradients, preserving the generative prior in unobserved regions.

2 Related Work

Novel view synthesis from sparse inputs. Neural Radiance Fields [29] and 3D Gaussian Splatting [20], together with their follow-ups for unbounded scenes [2] and faster rendering [3], set the modern bar for in-distribution view synthesis. They are, however, fundamentally interpolators: they require dense captures, are optimized per scene, and degrade once the camera leaves the convex hull of the inputs [46]. Geometric priors such as depth smoothness [30] or monocular cues [57] partially mitigate the sparse-view regime by regularizing the optimization. Feed-forward alternatives sidestep per-scene fitting altogether: PixelNeRF [55] conditions a radiance field on image features in a single forward pass, and recent work predicts 3D Gaussians directly from sparse views [4, 5, 53, 43, 52, 54], while large transformer synthesizers such as LVSM [18] and projection-conditioned variants [51] dispense with explicit 3D representations entirely. Across this entire family of methods, the output is deterministic, which is sufficient when the target view is well-constrained but breaks down when large regions must be imagined; the gap that the next family fills.

Pose-only multi-view diffusion. These methods condition a multi-view diffusion model purely on the target camera, typically through Plücker ray embeddings [40, 13]. Beginning with score distillation [33] and image-conditioned single-object generators [26, 38, 39], it has scaled to scene-level diffusion that produces tens of consistent novel views in a single denoising pass [49, 12, 59], and to joint RGB-plus-pointmap sampling for fast 3D scene generation [42]. Pose-only conditioning is flexible but struggles with the metric-scale ambiguity of ray embeddings. Crucially, the content of the scene actually visible in the input is not given any privileges: it is regenerated from scratch like the rest of the frame.

Warp- and geometry-conditioned generation. Another line of work explicitly injects scene geometry into the diffusion model. The most common recipe takes a depth-warped point cloud or RGB-D rendering of the source views and feeds it as a frame-aligned signal to a video diffusion backbone [56, 21, 1, 25]. GEN3C [34] formalizes this with a persistent 3D cache that is rendered into the model for precise camera control, and Geometry Forcing [47] pushes the video backbone toward 3D-consistent latents via auxiliary feature alignment. FlowR [11] replaces video diffusion with multi-view flow matching that maps renderings of a sparse 3DGS to renderings of a dense one. A complementary thread bridges geometric foundation models with diffusion latents [45, 22, 16, 17], and a third uses generative priors to clean or harmonize existing reconstructions [48]. These methods inject geometry where pose-only models cannot, but the supervision they apply remains uniform across the output: regions where the warped condition is reliable and regions where it is empty are pulled toward the same target by the same loss.

Region-aware supervision. Binary masks do appear in the geometry-coupled literature, but always at the periphery of the generative loop: dropping parts of the 3D representation during training [50], marking unwarped pixels in the input conditioning [56], or reweighting rendering losses in the downstream 3DGS optimization. None of these methods uses a per-pixel observation signal to split reconstruction from generation inside the generator itself. A related observation, recently sharpened

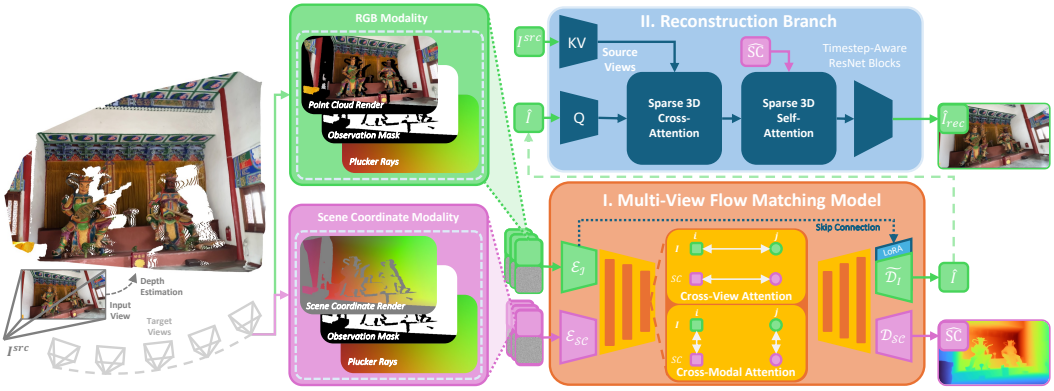


Figure 3: **Overview of GenRec**. Given posed input views and target poses, monocular depth and forward warping produce per-target observation masks and warped renders. These condition a *(I) multi-view flow matching backbone*, which jointly denoises RGB and scene-coordinate latents through cross-view and cross-modal attention. A *(II) reconstruction branch* then refines only the observed pixels via sparse 3D attention guided by the predicted scene coordinates, while the observation mask gates gradients so the generative prior is preserved in unobserved regions.

by pixel-space NVS diffusion [8], is that VAE latents impose a photometric fidelity ceiling on the detail that observed regions can carry through. **GenRec** addresses both: a shared observation mask binds architecture and loss, and a dedicated pixel-space branch restores detail where the inputs constrain it while leaving the generative prior untouched elsewhere.

3 Method

Problem statement. We are given N_{src} posed source views $\{(I_j^{\text{src}}, \pi_j^{\text{src}})\}_{j=1}^{N_{\text{src}}}$ with images $I_j^{\text{src}} \in [0,1]^{3\times H\times W}$, and N target poses $\{\pi_i\}_{i=1}^{N}$. The task is to synthesize the corresponding target images $\{\hat{I}_i^{\text{rec}}\}_{i=1}^{N}$, with $\hat{I}_i^{\text{rec}} \in [0,1]^{3\times H\times W}$. As preprocessing, we run an off-the-shelf monocular depth prediction network on each source view and forward-warp the source RGB and the back-projected source 3D points into every target frame. This yields, for each target view i, three precomputed signals: a soft observation mask $O_i \in [0,1]^{1\times H\times W}$ recording which target pixels receive evidence, and per-modality warped renderings $W_i^{I} \in [0,1]^{3\times H\times W}$ of the source RGB and $W_i^{SC} \in [-1,1]^{3\times H\times W}$ of the source scene coordinates.

Generation–reconstruction split. Not all pixels in a target view are equally constrained by the source views. Pixels with source coverage have a near-deterministic target: the source observation collapses the predictive distribution to a tight mode varying only with viewing direction. Pixels without coverage (in disocclusions or beyond the captured frustum) have a stochastic target: only a posterior over completions consistent with the visible scene is identifiable. These regimes demand structurally different supervision (regression in the first, distribution matching in the second), and the latent generative path discards the high-frequency content the deterministic regime must preserve. **GenRec** therefore factors the model along O_i at every level (Fig. 3). A multi-view flow-matching backbone (Sec. 3.1) generates every pixel of every target view, handling the distributional regime. A lightweight pixel-space reconstruction stage (Sec. 3.2) then corrects the observed pixels, supplying the photometric signal the latent path cannot provide. The mask further isolates gradient flow: reconstruction gradients reach only the reconstruction parameters, never the backbone (Sec. 3.3).

3.1 Multi-View Flow Matching Backbone

The backbone is a denoising network v_θ trained under the Conditional Flow Matching framework. It transports Gaussian noise $x_0 \sim \mathcal{N}(0, I)$ to a clean latent x_1 along the path $x_t = (1-t)\,x_0 + t\,x_1$. A single denoising trajectory operates jointly on all N target views and two co-registered modalities, producing per-view, per-modality clean-end estimates $\hat{x}_{1,i}^{m} \in \mathbb{R}^{C\times h\times w}$ for $i \in \{1,\dots,N\}$ and $m \in \{I, SC\}$. The refinement stage of Sec. 3.2 consumes these latents.

Joint RGB and scene-coordinate denoising. Each target view comprises two co-registered modalities: the RGB image and a scene-coordinate map $SC \in [-1,1]^{3\times H\times W}$ encoding the normalized world-space (x,y,z) position of every pixel. Joint denoising enables the geometry-aware refinement of Sec. 3.2: the decoded scene coordinates $\widehat{SC}_i = \mathcal{D}_{SC}(\hat{x}^{SC}_{1,i})$ supply the 3D positions used by its cross-attention. Both modalities use pretrained VAEs $(\mathcal{E}_I, \mathcal{D}_I)$ and $(\mathcal{E}_{SC}, \mathcal{D}_{SC})$ with shared channel count C and spatial resolution $h \times w$. Following Fang et al. [10], we extend each transformer block of v_θ along two axes: (i) self-attention is rewired into *cross-view* attention over all N target tokens, yielding multi-view consistency in one forward pass; (ii) a zero-initialized *cross-modal* attention layer couples the RGB and scene-coordinate tokens of the same view. The zero initialization preserves the pretrained prior at training start; the modalities are paired as their joint distribution is learned.

Conditioning on geometric evidence. At each flow time $t \in [0,1]$, the network input is a tensor $\mathbf{Z}_{\text{in}}(t) \in \mathbb{R}^{N\times 2\times C_{\text{in}}\times h\times w}$ where the second axis indexes the two modalities and C_{in} is the per-modality concatenated channel count. For every view i and modality $m \in \{I, SC\}$,

$$\mathbf{Z}^{i,m}_{\text{in}}(t) \;=\; \big[\, x^m_{t,i} \,\big\|\, \mathbf{r}_i \,\big\|\, \tilde{O}_i \,\big\|\, \mathcal{E}_m\big(W_i{}^m\big) \,\big], \tag{1}$$

where $\|$ denotes channel-wise concatenation; $x^m_{t,i}$ is the noisy latent at time t; $\mathbf{r}_i$ is a Plücker ray map of the target, embedded by a small patch ViT [7] whose tokens are folded back to the latent grid; $\tilde{O}_i$ is the observation mask area-pooled to $h \times w$; and $\mathcal{E}_m(W_i{}^m)$ encodes the modality-specific warp. This tells the backbone *where* evidence is available (via $\tilde{O}_i$), *what* that evidence is (via the encoded warps), and *which* 3D ray each pixel corresponds to (via $\mathbf{r}_i$), even in regions it must hallucinate.

Flow-matching parameterization. We initialize v_θ from a pretrained $\mathbf{v}$-prediction latent diffusion model [35] and fine-tune it under Conditional Flow Matching with the Diff2Flow reparameterization of Schusterbauer et al. [37], which converts the $\mathbf{v}$-prediction into a flow velocity in closed form. This preserves the pretrained weights (cross-modal layer is zero-initialized) while letting us train straight flows that admit fast Euler sampling at inference [27]. The clean estimate is recovered in closed form as $\hat{x}^m_{1,i} = x^m_{t,i} + (1-t)\, v_\theta(\mathbf{Z}_{\text{in}}(t), t)_{i,m}$, where $(\cdot)_{i,m}$ selects the per-view, per-modality slice.

3.2 Pixel-Space Refinement of Observed Regions

The refinement stage takes the backbone's clean latents $\{\hat{x}^I_{1,i}, \hat{x}^{SC}_{1,i}\}_{i=1}^N$ together with the precomputed mask O_i and warps $W_i{}^I$, and produces the final RGB outputs $\{\hat{I}^{\text{rec}}_i\}_{i=1}^N$. Latent generative models impose a photometric ceiling on what can be recovered through the VAE decoder (acceptable for hallucinated content, but limiting where evidence already constrains the answer), so we restrict refinement to observed pixels. The stage has two cooperating components: a geometry-aware decoder adapter that injects warped RGB at the latent→pixel boundary and outputs an intermediate image $\hat{I}_i$, and a sparse 3D cross-attention branch that adds an explicitly geometry-aware residual Δ_i on top.

Geometry-Aware Decoder Adapter. We adapt the RGB decoder $\mathcal{D}_I$, which maps the latent $\hat{x}^I_{1,i}$ to pixels; both VAEs are otherwise frozen. The adaptation has two parts. Following Parmar et al. [31], Wu et al. [48], zero-initialized 1×1 skip convolutions additively inject features from $\mathcal{E}_I(W_i{}^I)$ before each up-block of $\mathcal{D}_I$. A LoRA adapter [15] on the decoder's convolutions and self-attention projections is jointly trained, allowing the decoder to consume these injected features. We denote the resulting decoder $\widetilde{\mathcal{D}}_I$, with output:

$$\hat{I}_i \;=\; \widetilde{\mathcal{D}}_I\big(\hat{x}^I_{1,i},\, W_i{}^I\big) \;\in\; [0,1]^{3\times H\times W}.$$

The two adaptations are complementary. *Locally*, the skip connections copy fine-grained photometric detail from the warped conditioning, recovering high-frequency content the latent representation cannot preserve. *Globally*, the LoRA weights affect every pixel and shift the decoder's output distribution toward sharper samples, so perceptual benefits extend beyond the observation mask. Supervision is gated by O_i (Sec. 3.3) to prevent drift in disoccluded regions.

Sparse 3D Reconstruction Branch. The reconstruction branch consumes the decoder-adapter output $\hat{I}_i$ together with the warped RGB $W_i{}^I$, the mask O_i, the source images $\{I^{\text{src}}_j\}_{j=1}^{N_{\text{src}}}$, and the decoded scene coordinates $\widehat{SC}_i = \mathcal{D}_{SC}(\hat{x}^{SC}_{1,i})$. It predicts a pixel-space residual $\Delta_i \in \mathbb{R}^{3\times H\times W}$ that is added back only on observed pixels, producing the final image

$$\hat{I}^{\text{rec}}_i \;=\; \hat{I}_i \;+\; O_i \odot \Delta_i.$$

Whereas the decoder adapter copies coarse appearance through skip connections, this branch reasons explicitly about 3D correspondences: each target pixel attends to a small set of geometrically nearest neighbours (first, in the source views, then across target views), with $\widehat{SC}_i$ supplying the 3D positions used to define the neighbour sets. The branch is composed of two sparse cross-attention blocks followed by FiLM-conditioned residual blocks [32] whose scale and shift are produced from the detached temporal embedding of the backbone, so the branch modulates its behaviour with the flow time t. The output projection of every cross-attention block and final convolution are zero-initialized, so $\Delta_i = \mathbf{0}$ at the start of training and the branch contributes only as it acquires useful signal.

Sparse source-to-target attention. The first attention block operates at full pixel resolution. For each target view i, queries and values are built from the per-view tensor $[\hat{I}_i \,\|\, W_i^I \,\|\, O_i]$, and keys are produced by encoding the source images $\{I_j^{\mathrm{src}}\}$ with a VAE-initialized encoder. To avoid quadratic cost across all source pixels, we precompute, for each target pixel n, the indices of its K geometrically nearest source pixels using $\widehat{SC}_i$ and the source point cloud, and restrict attention to that fixed neighbour set. Indexing those neighbours by $k \in \{1, \dots, K\}$, we form the attention logits:

$$\mathrm{attn}_{n,k} \;=\; \frac{Q_n^\top K_{n,k}}{\sqrt{d}} \;+\; \log w_{n,k}, \qquad w_{n,k} \;=\; \frac{(d_{n,k}+\varepsilon)^{-1}}{\sum_{k'}(d_{n,k'}+\varepsilon)^{-1}}, \tag{2}$$

where d is the head dimension, $\mathbf{p}_n^{\mathrm{tgt}} \in \mathbb{R}^3$ is the target-pixel 3D position read from $\widehat{SC}_i$, $\mathbf{p}_k^{\mathrm{src}} \in \mathbb{R}^3$ is the 3D position of its k-th source neighbour, and $d_{n,k} = \|\mathbf{p}_n^{\mathrm{tgt}} - \mathbf{p}_k^{\mathrm{src}}\|_2$. Since $\sum_k w_{n,k} = 1$, the additive $\log w_{n,k}$ acts as a soft logit prior favouring geometrically close correspondences while still allowing evidence to override the prior; invalid slots are masked to $-\infty$ before the softmax.

Sparse cross-target attention. Per-view source-to-target attention may produce refinements that disagree where different target frames observe the same surface. A second attention block re-imposes consistency by reusing the same operator, but with the neighbour set now constructed across target views: for each target pixel n in view i, we use $\widehat{SC}_i$ and $\widehat{SC}_{i' \neq i}$ to retrieve the K geometrically nearest pixels in the *other* target views. Queries come from the residual stream of the source-to-target stage, and keys and values are produced by re-encoding the per-view tensor $[\hat{I}_{i'} \,\|\, W_{i'}^I \,\|\, O_{i'}]$ from each other view. Eq. (2) is reused unchanged, with $d_{n,k}$ now measuring inter-target distance.

3.3 Training and Inference

Decoupled optimization. We train the model in two phases. In the first phase, only the multi-view flow-matching backbone is optimized, using the standard Conditional Flow Matching loss of Eq. (9), denoted $\mathcal{L}_{\mathrm{CFM}}$. In the second phase, the backbone is frozen and we train the decoder adapter and the reconstruction branch on its detached outputs, so no gradient from this phase can reach the backbone parameters. To prevent the adapters from drifting in disoccluded regions, gradients through the global LPIPS loss [58] are routed only through observed pixels using the masked stop-gradient construction:

$$\tilde{Y}_i \;=\; O_i \odot Y_i \;+\; (1 - O_i) \odot \mathrm{sg}(Y_i), \tag{3}$$

applied to each output of interest $Y_i \in \{\hat{I}_i, \hat{I}_i^{\mathrm{rec}}\}$.

Refinement losses. The decoder-adapter output is supervised with a masked ℓ_1 photometric term and a full-image LPIPS term gated by Eq. (3),

$$\mathcal{L}_{\mathrm{dec}} \;=\; \sum_{i=1}^{N} \Big[\; \lambda_1 \left\| O_i \odot (\hat{I}_i - I_i) \right\|_1 \;+\; \lambda_{\mathrm{p}}\, \mathcal{L}_{\mathrm{LPIPS}}\big(\tilde{I}_i,\, I_i\big) \;\Big], \tag{4}$$

and the refined output by an analogous loss with an additional ℓ_1 regularizer on Δ_i that keeps the refinement small by default,

$$\mathcal{L}_{\mathrm{rec}} \;=\; \sum_{i=1}^{N} \Big[\; \lambda_1 \left\| O_i \odot (\hat{I}_i^{\mathrm{rec}} - I_i) \right\|_1 \;+\; \lambda_{\mathrm{p}}\, \mathcal{L}_{\mathrm{LPIPS}}\big(\tilde{I}_i^{\mathrm{rec}},\, I_i\big) \;+\; \lambda_\Delta \left\| \Delta_i \right\|_1 \;\Big]. \tag{5}$$

The second phase optimizes $\mathcal{L}_{\mathrm{dec}} + \mathcal{L}_{\mathrm{rec}}$ over the decoder adapter and reconstruction branch with the backbone kept fixed. Combined with the O_i-gating in Eq. (3), this two-phase schedule implements the generation–reconstruction split at the level of every gradient that touches the network.

Table 1: **Quantitative results on RealEstate10K** in the single-view extrapolation setting. *Global* metrics are computed over the full target frame; *Observed* restricts them to pixels that receive source evidence under the shared observation mask, and *Unobserved* to pixels that do not. Inference time is reported per scene. Best results in **bold**.

	Global				Observed		Unobserved		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	Inf. Time
CameraCtrl [13]	13.40	0.4424	0.4761	57.42	15.53	0.4568	12.40	0.3935	∼12s
ViewCrafter [56]	13.29	0.4792	0.4293	46.27	15.93	0.5196	11.83	0.3937	∼200s
SEVA [59]	13.50	0.4839	0.4674	41.64	15.97	0.4999	12.40	0.4189	∼20s
GF [47]	13.11	0.3806	0.4945	54.58	15.56	0.3986	11.96	0.3125	∼40s
LVSM [18]	13.81	0.4094	0.5186	57.89	16.32	0.4224	12.61	0.3435	∼1s
GLD [17]	13.73	0.4232	0.4699	44.29	16.07	0.4533	12.82	0.3765	∼42s
Gen3R [16]	13.29	0.4747	0.5107	49.46	15.82	0.4862	12.14	0.4044	∼90s
GEN3C [34]	15.24	0.5289	0.3624	35.68	18.13	0.5689	13.63	0.4529	∼1200s
Ours	**17.05**	**0.6071**	**0.3220**	**33.09**	**20.85**	**0.6738**	**14.18**	**0.4743**	∼11s

Iterative-denoising supervision for the refinement stage. A subtle but important design choice concerns *which* clean-end estimate $\hat{x}^I_{1,i}$ is used to train the refinement stage. The naive recipe samples $t \in [0,1]$, forms the linear interpolant $x^I_{t,i} = (1-t)\, x^I_{0,i} + t\, x^I_{1,i}$, and feeds the corresponding one-step estimate $\hat{x}^I_{1,i} = x^I_{t,i} + (1-t)\, v_\theta(\mathbf{Z}_{\text{in}}(t), t)_{i,I}$ to the refinement stage. This is misaligned with test-time behaviour: at inference, the refinement stage operates on a $\hat{x}^I_{1,i}$ produced from a multi-step Euler trajectory that has accumulated errors, whereas $x^I_{t,i}$ lies exactly on the noise–data line. We instead sample $t \in [0,1]$, integrate the flow from pure noise up to time t to obtain a trajectory state $\hat{x}^I_{t,i}$, and supervise the refinement stage on $\hat{x}^I_{1,i} = \hat{x}^I_{t,i} + (1-t)\, v_\theta(\mathbf{Z}_{\text{in}}(t), t)_{i,I}$.

Inference. Given source views and target poses, we precompute O_i, W_i^I, W_i^{SC}, and the geometric neighbour indices. Starting from $x^m_{0,i} \sim \mathcal{N}(0, I)$, we run T Euler steps of the joint RGB+SC flow in a single multi-view denoising trajectory, yielding $\hat{x}^I_{1,i}$ and $\hat{x}^{SC}_{1,i}$. Decoding $\hat{x}^I_{1,i}$ through $\widetilde{\mathcal{D}}_I$ produces $\hat{I}_i$; a single forward pass through the reconstruction branch then yields Δ_i and the final $\hat{I}^{\text{rec}}_i$. Both stages share the observation mask O_i, so the generation–reconstruction split holds at test time as well.

4 Experiments

We organize our experiments into four groups. First, we evaluate **GenRec**'s extrapolation capabilities in the single-view conditioned setting. Second, we assess its interpolation abilities in the two-view conditioned setting, where target views lie between the two source views. Third, we measure the multi-view consistency of the generated frames using 3D-based metrics. Finally, we conduct ablation studies that validate our design choices and quantify the impact of our observation-based formulation on overall performance.

Experimental Setup. We evaluate in two settings. *Single-view extrapolation* uses the first frame of each test sequence as the source view and samples target views from subsequent frames at a fixed stride. *Two-view interpolation* additionally uses the last frame as a second source. The stride is 4

Table 2: **Quantitative Results on DL3DV-10K.** We compare our method with previous works on single-view conditioned generation. We follow the same evaluation protocol as in Table 1.

	Global				Observed		Unobserved		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	Inf. Time
CameraCtrl [13]	13.26	0.3346	0.4893	69.49	13.45	0.3418	12.76	0.2957	∼12s
ViewCrafter [56]	17.57	0.5302	0.3092	46.36	18.75	0.5659	15.36	0.4282	∼200s
SEVA [59]	14.58	0.4300	0.4359	43.51	14.86	0.4401	13.92	0.3819	∼20s
GLD [17]	15.54	0.4001	0.3979	51.85	16.10	0.4184	14.33	0.3368	∼42s
Gen3R [16]	13.74	0.3754	0.5173	130.10	14.02	0.3886	12.95	0.3048	∼90s
GEN3C [34]	18.47	0.5717	0.2860	44.54	19.66	0.6026	16.24	0.4802	∼1200s
Ours	**19.80**	**0.6195**	**0.2188**	**32.36**	**21.35**	**0.6605**	**16.66**	**0.4983**	∼11s

Table 3: **Quantitative results on Mip-NeRF 360.** Out-of-distribution comparison with prior methods in the single-view extrapolation setting. We follow the same evaluation protocol as in Table 1.

	Global				Observed		Unobserved		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	Inf. Time
CameraCtrl [13]	10.89	0.1870	0.6534	214.2	11.31	0.1869	10.67	0.1821	∼12s
ViewCrafter [56]	11.75	0.2197	0.5694	172.5	14.46	0.2510	10.74	0.1913	∼200s
SEVA [59]	11.52	0.2231	0.5857	139.8	12.09	0.2315	11.30	0.2171	∼20s
GLD [17]	12.03	0.1986	0.5631	162.1	13.32	0.2044	11.45	0.1900	∼42s
Gen3R [16]	11.07	0.2056	0.6564	223.6	11.21	0.2072	10.96	0.2015	∼90s
GEN3C [34]	12.86	0.2343	0.5796	207.4	14.43	0.2608	12.12	0.2117	∼1200s
Ours	**13.10**	**0.2478**	**0.4895**	**139.4**	**15.37**	**0.2858**	**12.16**	**0.22**	∼11s

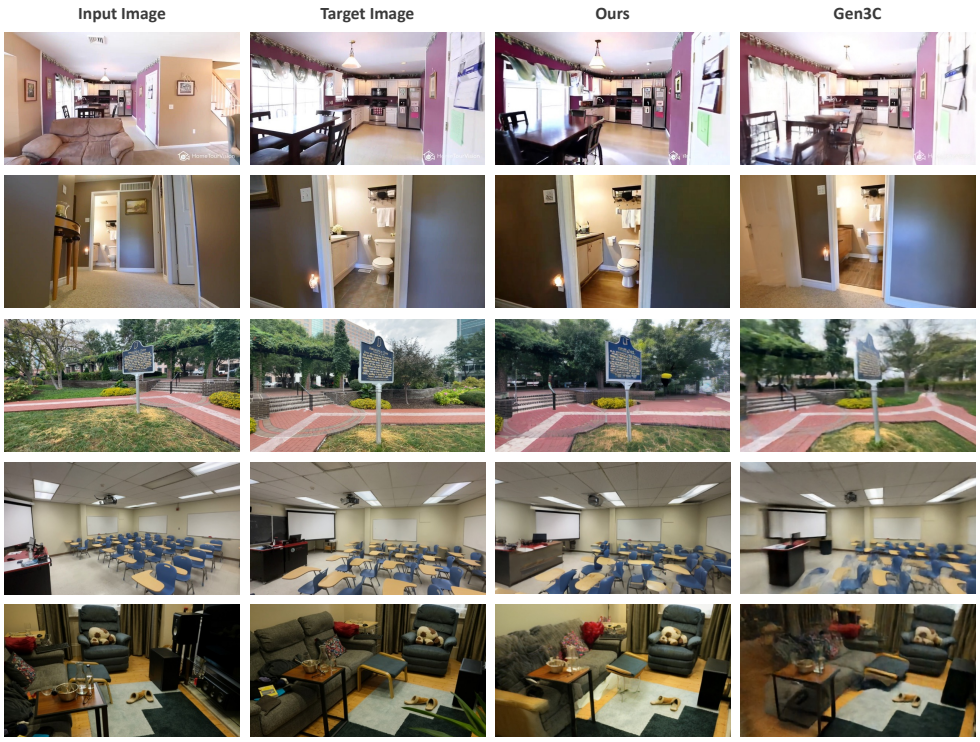


Figure 4: **Comparative qualitative results.** Single-view extrapolation against our strongest baseline, Gen3C [34]. Each row shows the input view, ground-truth target, **GenRec**, and Gen3C.

on DL3DV-10K and Mip-NeRF 360, and 25 on RE10K to account for its slower camera motion. Video-architecture baselines receive the full pose sequence and produce outputs at the strided frames. Baselines conditioned on forward-warped RGB renders use DA3 for depth estimation and pose-depth alignment, matching **GenRec**. All baselines are open-source; we follow the evaluation protocols from their official repositories.

Single-View Conditioned Extrapolation. We evaluate on two in-distribution benchmarks, RealEstate10K [60] (RE10K) and DL3DV-10K [23], and on Mip-NeRF 360 [2] for out-of-distribution generalization. We sample 100 scenes from the official test splits of RE10K and DL3DV-10K and use all 9 Mip-NeRF 360 scenes. We report PSNR, SSIM, and LPIPS for reconstruction quality and FID for generation quality. Unlike prior work, we report reconstruction metrics globally and separately for observed and unobserved regions. For fair comparison, observed and unobserved regions are scored against a shared binary observation mask from Gen3C's forward-warping, applied to all methods. We compare against baselines spanning different conditioning strategies and architectures: CameraCtrl [13], ViewCrafter [56], Stable Virtual Camera (SEVA) [59], GLD [17], Gen3R [16], and Gen3C [34]. On RE10K, we additionally compare against Geometry Forcing (GF) [47] and LVSM [18], both RE10K-only methods.

Table 4: **Quantitative results on RE10K and DL3DV-10K.** Comparison with prior methods in the two-view interpolation setting.

	RE10K			DL3DV-10K		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
CameraCtrl [13]	12.87	0.4271	0.5022	15.60	0.3456	0.4705
ViewCrafter [56]	12.61	0.4550	0.4573	17.38	0.5214	0.3087
SEVA [59]	16.15	0.5833	0.3477	21.38	0.6906	0.1845
LVSM [18]	16.88	0.5493	0.3983	21.31	0.6801	0.2223
GLD [17]	17.64	0.5707	0.2900	19.49	0.5575	0.2376
Gen3R [16]	12.17	0.4385	0.5697	13.99	0.3813	0.5030
GEN3C [34]	15.70	0.5632	0.3282	21.16	0.6869	0.2137
Ours	**19.80**	**0.7043**	**0.2205**	**21.99**	**0.7009**	**0.1558**

Table 5: **Ablation study on the effectiveness of individual components on RE10K and DL3DV.** Starting from a vanilla multi-view flow matching baseline, we progressively add the VAE skip connections + LoRA weights, and the reconstruction branch.

		Global				Observed	
	Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑
RE10K	Generative Backbone	15.86	0.5483	0.3402	37.82	19.20	0.6139
	+ VAE Skip & LoRA	15.86	0.5591	0.3394	35.27	19.23	0.6152
	+ Reconstruction Branch	**17.05**	**0.6071**	**0.3220**	**33.09**	**20.85**	**0.6738**
DL3DV	Generative Backbone	18.62	0.5796	0.2556	39.94	19.96	0.6163
	+ VAE Skip & LoRA	18.64	0.5828	0.2512	36.90	20.02	0.6215
	+ Reconstruction Branch	**19.80**	**0.6195**	**0.2188**	**32.36**	**21.35**	**0.6605**

Quantitative Results. Tables 1, 2, and 3 report quantitative results across the benchmarks. **GenRec** outperforms the baselines on every metric, demonstrating the effectiveness of our approach. In single-view extrapolation, baselines conditioned solely on Plücker rays are limited by pose-only scale ambiguity, while those using forward-warped RGB renders exploit metric-scale geometry for more consistent generations. Figure 4 compares qualitatively against our strongest baseline, Gen3C [34]. Gen3C uses the forward-warped RGB render directly as the denoising target, yielding highly 3D-consistent outputs when warps are reliable; as camera motion grows, warp quality degrades and so do the generated images. This matches Gen3C's higher FID on DL3DV-10K and Mip-NeRF 360, which exhibit larger viewpoint changes than RE10K. We strongly encourage the readers to consult the supplementary for more examples. App. B.1 ablates the generation–reconstruction split directly, and App. B.7 reports depth-robustness studies.

Two-View Conditioned Interpolation. Next we evaluate **GenRec** in a predominantly reconstructive regime, where most of the information required to synthesize the target frames is already present in the two source views. We conduct this evaluation on RE10K and DL3DV-10K datasets. Table 4 reports reconstruction quality in the interpolation setting. **GenRec** outperforms all baselines on every reconstruction metric, confirming that our approach remains effective even when the task reduces to a nearly reconstruction-only (i.e., no generation) problem. App. B.3 extends this comparison to dedicated reconstruction methods, and App. B.2 evaluates interpolation from four and six input views.

Ablating Individual Components. Table 5 ablates the contribution of each architectural component on RE10K and DL3DV-10K. The VAE skip connections together with the decoder LoRA and the reconstruction branch play complementary roles. Adding the skip connections and LoRA weights primarily improves generative quality, as reflected in the lower FID scores on both datasets. The reconstruction branch, in contrast, sharpens reconstruction fidelity (PSNR, SSIM, LPIPS) without sacrificing generative quality.

5 Conclusion

We presented **GenRec**, a multi-view flow matching model that treats generative novel-view synthesis as a structured composition of reconstruction and generation rather than a single distribution-matching

problem. A per-pixel observation mask, derived from monocular depth and forward warping, governs the architecture, the loss, and the gradient flow: a flow matching backbone jointly denoises RGB and scene-coordinate latents to handle the distributional regime, while a pixel-space refinement stage with a skip- and LoRA-adapted decoder and a sparse 3D cross-attention branch restores high-frequency detail on observed pixels alone, with masked stop-gradients keeping its regression signal away from the generative prior. Across RealEstate10K, DL3DV-10K, and Mip-NeRF 360, in both single-view extrapolation and two-view interpolation, GenRec attains the best reconstruction fidelity in observed regions and matches or surpasses the strongest generative baselines on perceptual quality in unobserved ones, while running an order of magnitude faster than warp-as-target video baselines.

Limitations. GenRec inherits the limitations of its monocular depth front-end, whose accuracy bounds the quality of the observation mask and warped conditioning, though monocular metric depth estimators continue to improve rapidly. Furthermore, computational constraints, compounded by our architecture's bi-modal nature, currently limit training to a limited number of views per scene. Apps. B.8 and B.7 state our failure modes explicitly and quantify the depth dependence.

Acknowledgments and Disclosure of Funding

This work is part of the ETHAR project "Fast and Scalable Multi-View 3D Reconstruction with Semantic Shape Priors". This research was also supported by the Swiss AI Initiative through a grant from the Swiss National Supercomputing Centre (CSCS) under project IDs ab036 and a0144 on the Alps infrastructure; the Swiss National Science Foundation (Advanced Grant 216260: "Beyond Frozen Worlds: Capturing Functional 3D Digital Twins from the Real World"); the European Union's Horizon Europe research and innovation programme under grant agreement No. 101214398 (ELLIOT); the Horizon Europe programme under grant agreement No. 101298421 (GRAIL); and the ETH AI Center through an ETH AI Center postdoc fellowship and an ETH Zurich Career Seed Award awarded to Sunghwan Hong.

References

[1] Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B Lindell, Zan Gojcic, Sanja Fidler, et al. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. *arXiv preprint arXiv:2509.19296*, 2025.

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[4] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[5] 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*, pages 370–386. Springer, 2024.

[6] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Noam Elata, Bahjat Kawar, Yaron Ostrovsky-Berman, Miriam Farber, and Ron Sokolovsky. Novel view synthesis with pixel-space diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26756–26766, 2025.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[10] Chuan Fang, Heng Li, Yixun Liang, Jia Zheng, Yongsen Mao, Yuan Liu, Rui Tang, Zihan Zhou, and Ping Tan. Spatialgen: Layout-guided 3d indoor scene generation. *arXiv preprint arXiv:2509.14981*, 3, 2025.

[11] Tobias Fischer, Samuel Rota Bulò, Yung-Hsu Yang, Nikhil Keetha, Lorenzo Porzi, Norman Müller, Katja Schwarz, Jonathon Luiten, Marc Pollefeys, and Peter Kontschieder. Flowr: Flowing from sparse to dense 3d reconstructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27702–27712, 2025.

[12] 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.

[13] 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.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[16] Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3r: 3d scene generation meets feed-forward reconstruction. *arXiv preprint arXiv:2601.04090*, 2026.

[17] Wooseok Jang, Seonghu Jeon, Jisang Han, Jinhyeok Choi, Minkyung Kwon, Seungryong Kim, Saining Xie, and Sainan Liu. Repurposing geometric foundation models for multi-view diffusion. *arXiv preprint arXiv:2603.22275*, 2026.

[18] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024.

[19] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*, 2026.

[20] 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.

[21] 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 Computer Vision and Pattern Recognition Conference*, pages 798–810, 2025.

[22] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[23] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[25] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *IEEE Transactions on Image Processing*, 2026.

[26] 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*, pages 9298–9309, 2023.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[28] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2021.

[30] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5480–5490, 2022.

[31] Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024.

[32] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[34] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6121–6132, 2025.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[36] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[37] Johannes Schusterbauer, Ming Gui, Frank Fundel, and Björn Ommer. Diff2flow: Training flow matching models via diffusion model alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28347–28357, 2025.

[38] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[39] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[40] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34:19313–19325, 2021.

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[42] Stanislaw Szymanowicz, Jason Y Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T Barron, and Philipp Henzler. Bolt3d: Generating 3d scenes in seconds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24846–24857, 2025.

[43] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[44] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025.

[45] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[46] Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18120–18130, 2023.

[47] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025.

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26024–26035, 2025.

[49] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21551–21561, 2024.

[50] Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. Genfusion: Closing the loop between reconstruction and generation via videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6078–6088. IEEE, 2025.

[51] Zirui Wu, Zeren Jiang, Martin R Oswald, and Jie Song. From rays to projections: Better inputs for feed-forward view synthesis. *arXiv preprint arXiv:2601.05116*, 2026.

[52] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025.

[53] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024.

[54] Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. *Iclr*, 2026.

[55] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.

[56] 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.

[57] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[59] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12405–12414, 2025.

[60] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Technical appendices and supplementary material

A Qualitative Results

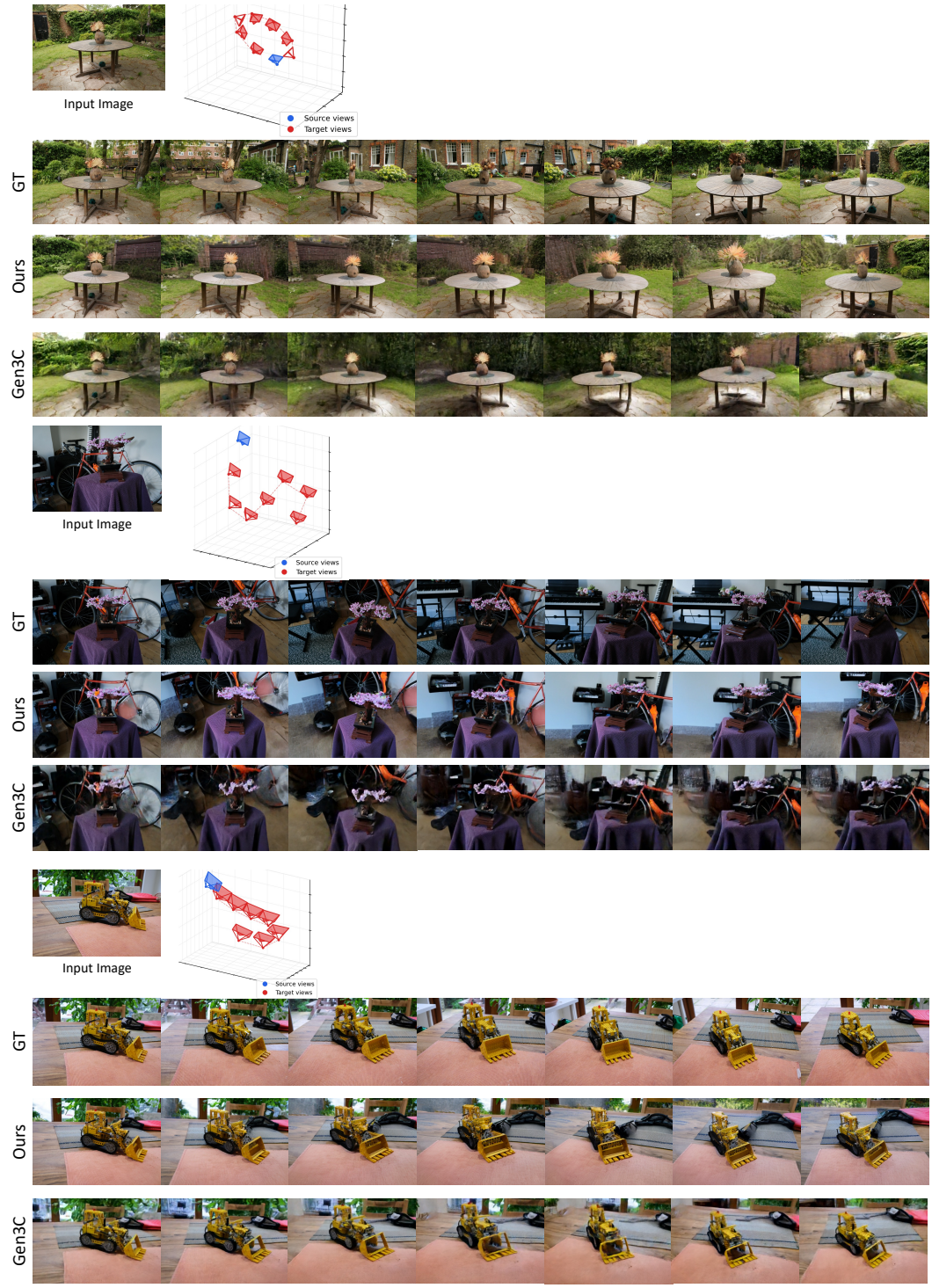


Figure 5: **Qualitative comparison on Mip-NeRF 360** [2]. Each row shows a held-out target view rendered from a single reference, compared against GEN3C. Mip-NeRF 360's unbounded outdoor scenes and wide-baseline trajectories stress geometric consistency in the unobserved background; our model preserves thin structures and produces sharper far-field detail than the baselines.

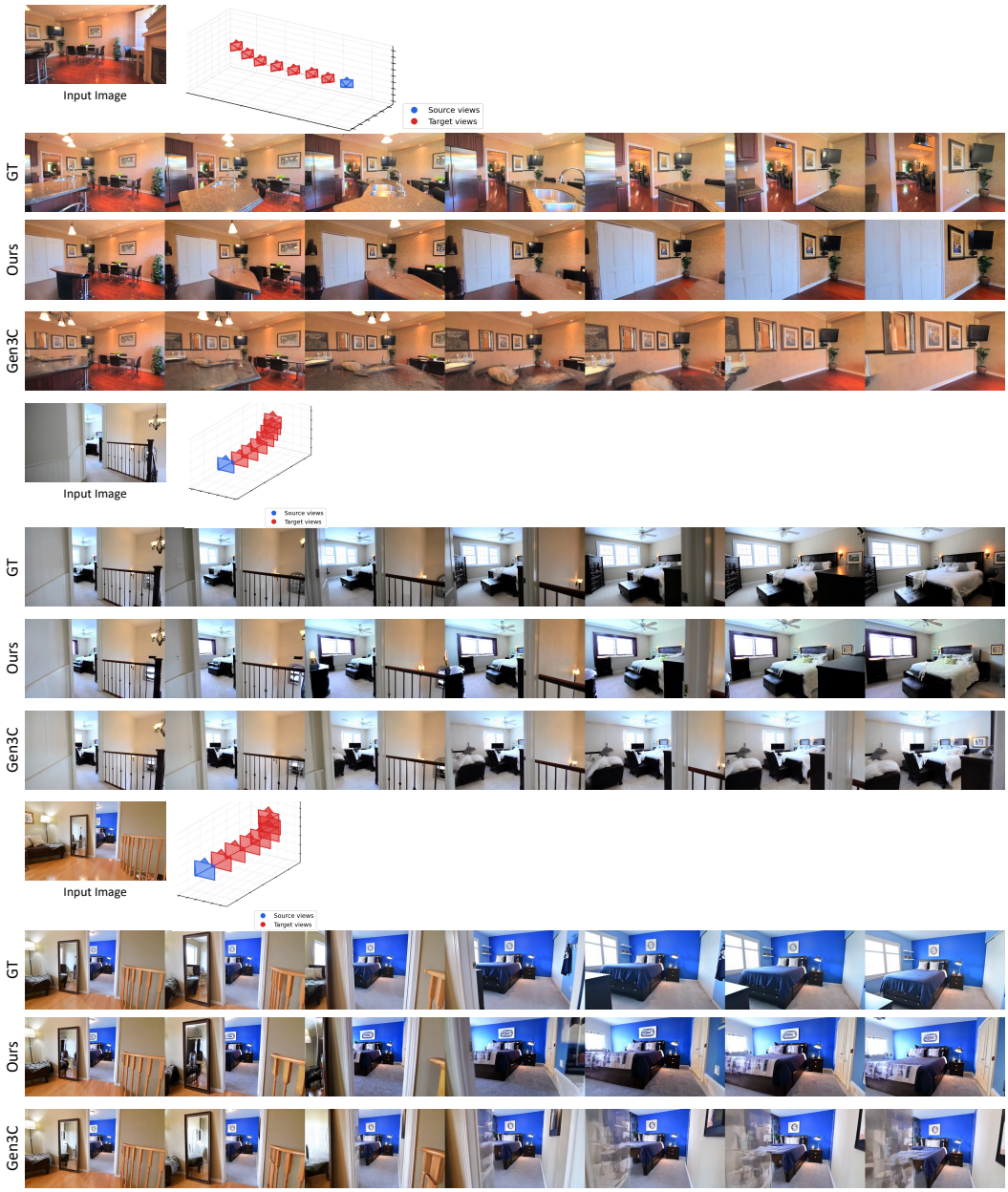


Figure 6: **Qualitative comparison on RealEstate10K** [60]. Indoor real-estate walkthroughs feature predominantly forward camera motion and narrow baselines. We highlight cases with disocclusions of structured surfaces (walls, doorways, furniture edges); our method recovers plausible texture in unobserved regions while baselines exhibit blur or ghosting along occlusion boundaries.

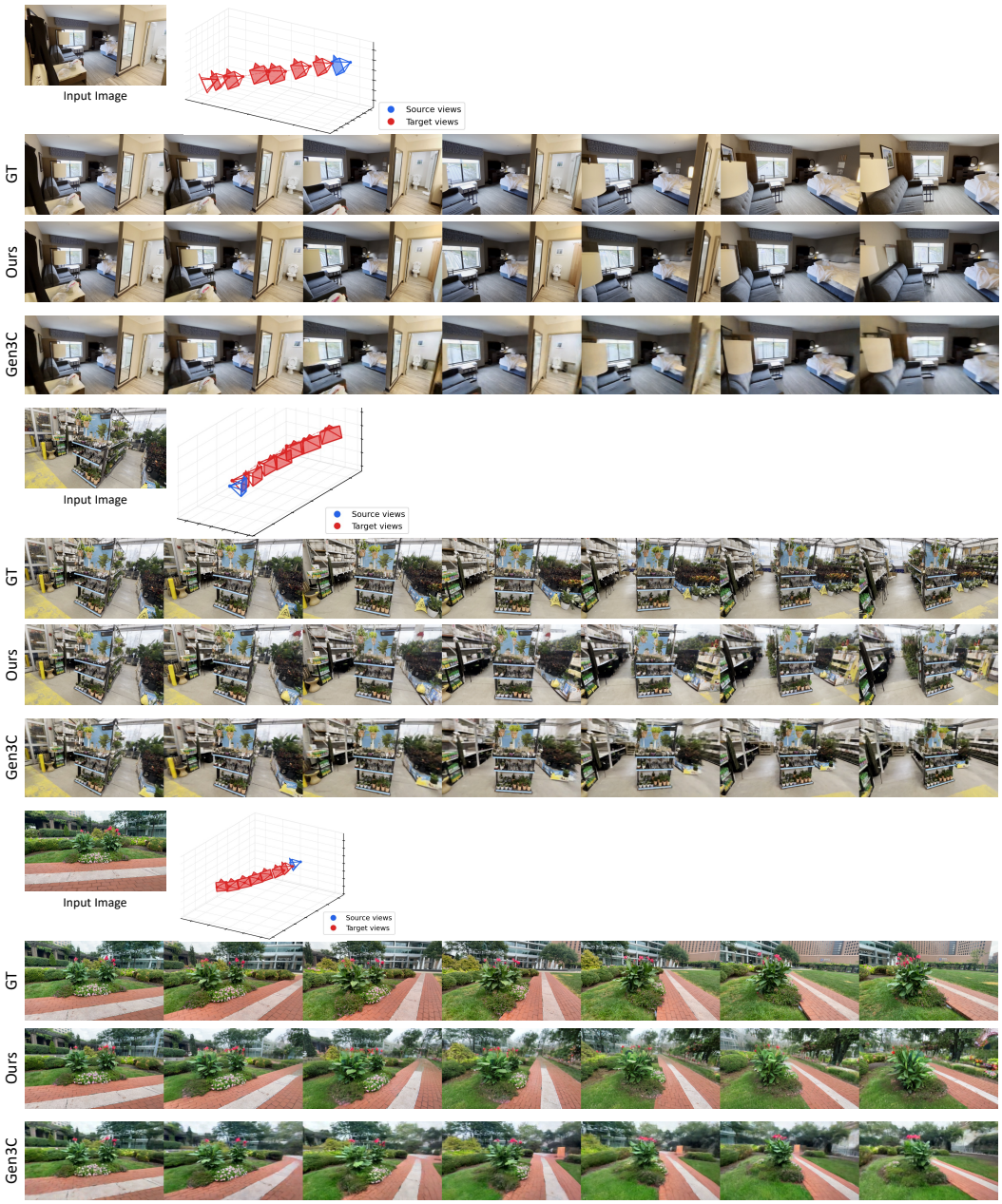


Figure 7: **Qualitative comparison on DL3DV-10K** [23]. DL3DV-10K provides diverse, wide-baseline captures of indoor and outdoor scenes. The larger camera motions amplify the gap between methods that only inpaint warped pixels and methods that synthesize globally consistent geometry; our outputs remain sharp and view-consistent under aggressive target poses.

B Additional Experiments and Ablation Studies

B.1 Effectiveness of the Generation–Reconstruction Split

Table 6: **Ablation of the generation–reconstruction split** on RealEstate10K, single-view extrapolation. *Rec.*: dedicated pixel-space reconstruction branch; in (a) the branch is removed and the mask-gated regression loss is applied to the backbone directly. *Gate*: mask-gated supervision, i.e., flow-matching supervision on all pixels with an additional regression supervision on observed ones. *BP*: reconstruction gradients back-propagated into the backbone through the last $K{=}1$ sampling steps, following Clark et al. [6]. Rows (b)–(d) share the same architecture, parameter count, and training procedure. Best results in **bold**.

				Global				Observed		Unobserved	
	Rec.	Gate	BP	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
(a)	×	✓	–	16.51	0.5779	0.3459	36.84	19.88	0.6535	13.96	0.4667
(b)	✓	×	×	16.97	0.6020	0.3433	40.43	20.55	0.6724	**14.28**	**0.4940**
(c)	✓	✓	✓	16.99	0.6067	0.3429	34.92	20.77	0.6732	14.13	0.4738
(d)	✓	✓	×	**17.05**	**0.6071**	**0.3220**	**33.09**	**20.85**	**0.6738**	14.18	0.4743

Table 6 isolates the effect of the generation–reconstruction split on RE10K in the single-view extrapolation setting. In row (a), the reconstruction branch is removed entirely, and the mask-gated supervision is instead applied to the generative backbone directly through a pixel-wise loss. In row (b), the branch is added but the gating is removed, so the branch is supervised on all pixels. In row (c), both the branch and the gating are present, but reconstruction gradients are back-propagated into the backbone through the last $K{=}1$ sampling steps, following Clark et al. [6]. Row (d) is our full model: the branch is added, the gating is applied, and no reconstruction gradient reaches the backbone. Rows (b)–(d) share the same architecture, parameter count, and training procedure, differing only in whether supervision is gated by the observation mask and whether reconstruction gradients reach the backbone; row (a) removes the branch entirely as a reference point.

Supervision — rows (b) vs. (d). These rows differ only in the mask gating. Removing the gating significantly degrades performance, with FID increasing by 7.34. Although row (b) has more capacity than row (a), the lack of gating significantly damages the generative quality of the model.

Gradient flow — rows (c) vs. (d). These rows differ only in whether reconstruction gradients reach the backbone. Allowing them through leaves row (c) behind the decoupled row (d) on every metric we measure, even though row (c) effectively has more trainable parameters through the additional finetuning of the backbone. The extra gradient flow from the reconstruction objective improves neither reconstruction fidelity nor generative quality.

Architecture — rows (a) vs. (d). With gating applied in both, routing the regression signal through a separate, gradient-isolated pixel-space branch rather than applying it to the backbone directly yields significant improvements in both reconstruction fidelity (PSNR, SSIM) and generative quality (FID, LPIPS), indicating that the reconstruction-branch architecture is itself an important factor in the performance of the model.

B.2 Multi-View Interpolation Beyond the Training Regime

Table 7: **Multi-view interpolation on RE10K. GenRec** is trained with at most two input views, so the 4- and 6-view results are zero-shot (marked *OOD*), whereas Gen3C's 3D cache ingests additional views natively. Best result per view count in **bold**.

#Views	Method	PSNR ↑	SSIM ↑	LPIPS ↓
2	Gen3C [34]	15.70	0.5632	0.3282
	GenRec	**19.80**	**0.7043**	**0.2205**
4	Gen3C [34]	20.80	0.7596	0.1860
	GenRec (OOD)	**23.68**	**0.7988**	**0.1387**
6	Gen3C [34]	20.86	0.7656	0.1793
	GenRec (OOD)	**24.07**	**0.8061**	**0.1342**

The architecture places no cap on the number of source views: sources enter only through the warp and mask precomputation and through the key set of the sparse source-to-target attention, both linear in the number of source views. The two restrictions in our benchmarks are budgetary rather than architectural: the number of

input views ($T_{\text{in}} \in \{1, 2\}$) and the total number of views per training sample (App. C.4) bound the training sequence length, and neither is enforced at inference. Table 7 evaluates multi-view interpolation on RE10K against Gen3C, the strongest baseline in our comparisons and the closest to **GenRec** in conditioning. Since we train on at most two input views, the 4- and 6-view results are zero-shot, whereas Gen3C's 3D cache ingests additional views natively.

GenRec leads Gen3C on every metric at every view count, by 2.9 to 4.1 dB PSNR, despite never seeing more than two input views during training. The trend is more informative than the margins: between four and six views, **GenRec** gains 0.39 dB while Gen3C gains 0.06 dB and has effectively saturated. We read this through the paper's own lens. More source views raise the observed fraction of each target frame, exactly the regime the reconstruction branch exists to exploit, whereas a warp-as-target formulation stops benefiting once its cache already covers the target. The decomposition therefore scales with the available evidence rather than saturating with it, and does so zero-shot at four and six views.

B.3 Comparison with Reconstruction Methods

Table 8: **Comparison with reconstruction methods** in the two-view interpolation setting on RE10K and DL3DV-10K. The baselines are specialised, deterministic reconstruction models; **GenRec** is generative. Best results in **bold**.

	RE10K			DL3DV-10K		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS [20]	14.95	0.543	0.421	19.76	0.681	0.248
Difix3D+ [48]	15.06	0.543	0.387	19.27	0.655	0.250
pixelSplat [4]	16.96	0.614	0.392	19.29	0.625	0.328
MVSplat [5]	16.64	0.592	0.383	17.53	0.535	0.366
DepthSplat [52]	**19.92**	**0.723**	0.277	19.79	0.673	0.253
GenRec	19.80	0.704	**0.221**	**21.99**	**0.701**	**0.156**

The two-view interpolation setting is predominantly reconstructive, so we additionally compare against dedicated reconstruction methods: 3DGS [20], Difix3D+ [48], pixelSplat [4], MVSplat [5], and DepthSplat [52], in Table 8. **GenRec** leads on all three metrics on DL3DV-10K (+2.20 dB PSNR over DepthSplat, with the best SSIM and LPIPS) and on LPIPS on both datasets. DepthSplat remains ahead on RE10K PSNR and SSIM, though **GenRec** trails by only 0.12 dB. We attribute the remaining gap to a systematic effect: deterministic regressors trained under squared error tend toward the conditional mean, which PSNR and SSIM reward and LPIPS penalizes. These baselines are specialised reconstruction models, whereas **GenRec** is generative; it is therefore on par with them in the regime they are built for, while additionally extrapolating beyond the input views, which they cannot do.

B.4 3D Consistency

Table 9: **Camera pose accuracy and multi-view consistency on RE10K and DL3DV-10K.** ATE and RPE_t are reported in the scale of the dataset camera poses; RPE_r in degrees. Inference time is reported per scene. Lower is better for all metrics.

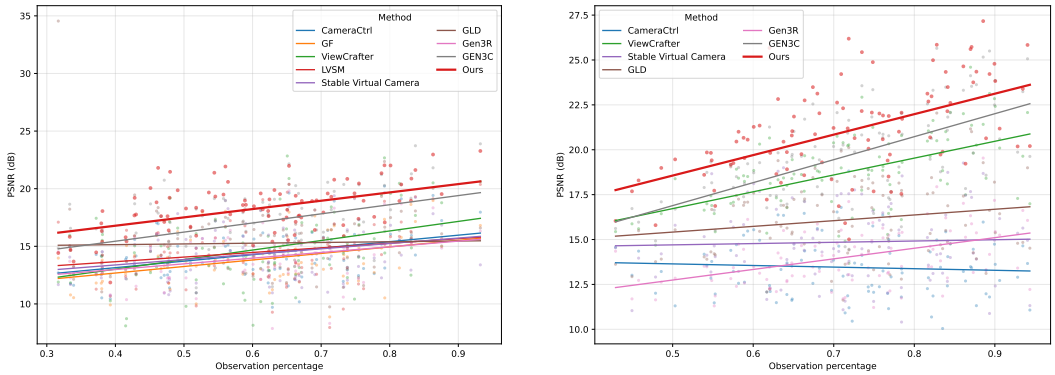
	RE10K			DL3DV			
Method	ATE ↓	RPE_t ↓	RPE_r ↓	ATE ↓	RPE_t ↓	RPE_r ↓	Inf. Time
GLD [17]	0.131	0.158	0.837	0.030	0.041	0.395	∼42s
ViewCrafter [56]	0.134	0.148	1.446	0.045	0.060	0.535	∼200s
Gen3R [16]	0.134	0.162	1.124	0.370	0.526	4.900	∼90s
Gen3C [34]	**0.085**	**0.083**	**0.809**	0.037	0.052	0.430	∼1200s
Ours	0.087	0.106	1.048	**0.021**	**0.029**	**0.252**	∼11s

Beyond per-frame image quality, a generative novel view synthesis model should produce frames that are mutually consistent in 3D. We evaluate this property for **GenRec** in the single-view conditioned extrapolation setting on the two in-distribution datasets, RE10K and DL3DV-10K. We pass the generated frames through an off-the-shelf feed-forward 3D reconstruction model [45] to estimate their camera poses. Because VGGT recovers poses only up to a global similarity, we Sim(3)-align the estimates to the ground-truth target poses before computing Absolute Trajectory Error (ATE) and Relative Pose Error in rotation (RPE_r) and translation

(RPE_t). The resulting values are therefore reported up to scale, but remain directly comparable across methods evaluated under the same protocol. Together, these metrics quantify how faithfully the generated frames adhere to the target pose conditioning.

Table 9 reveals a clear dataset-dependent pattern. On DL3DV-10K, **GenRec** achieves the best performance across all three metrics, with reductions of 29–36% relative to the strongest baseline, GLD. On RE10K, Gen3C is the strongest method on every metric; **GenRec** performs comparably on ATE but underperforms on the relative-pose metrics, where it is also outperformed by GLD on rotation. We attribute this dataset-dependent ordering to a warp-quality dynamic: RE10K's narrow camera baselines yield reliable forward warps, which directly benefit Gen3C's warp-as-target formulation, whereas DL3DV-10K's larger viewpoint changes degrade warp quality and surface the advantages of **GenRec**'s observation-based design.

B.5 Reconstruction Quality Across Observation Coverage



(a) RealEstate10K. (b) DL3DV-10K.

Figure 8: **PSNR as a function of per-scene observation percentage** on (a) RealEstate10K and (b) DL3DV-10K in the single-view conditioned setting. Each point is one test scene; lines are per-method linear fits. **GenRec** (red) sits above every baseline across the full range, from generation-heavy scenes on the left to near-reconstruction scenes on the right.

Figure 8 stratifies single-view PSNR by per-scene observation percentage on RealEstate10K and DL3DV-10K, sweeping from generation-heavy scenes ($\sim$30% observed) to near-reconstruction scenes ($>$90% observed). **GenRec** dominates across the full range on both datasets: the regression line sits above every baseline at low observation rates, where most of the target view must be hallucinated, and remains the steepest at high observation rates, where the forward-warped point cloud is an accurate prior. This second regime is the more telling one. Methods such as GEN3C use the warped render directly as the denoising target and therefore inherit a strong reconstruction signal essentially for free in well-observed scenes. **GenRec** matches or exceeds them despite never tying its denoising target to the warped pixels, which we attribute to the reconstruction branch supplying the photometric fidelity that the latent generative path cannot. The result confirms that the generation–reconstruction split adapts gracefully to the actual observation budget of each scene.

B.6 Effectiveness of Multi-Modal Generation

Table 10: **Ablation on multi-modal generation.** We compare our joint RGB and scene-coordinate (SC) backbone against an RGB-only variant trained from the same initialization for 20K steps, evaluated in the single-view conditioned extrapolation setting.

	RE10K				DL3DV-10K			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
RGB	15.09	0.5384	0.3602	38.22	18.05	0.5681	0.2631	38.53
RGB + SC (Ours)	**15.39**	**0.5457**	**0.3528**	**37.13**	**18.31**	**0.5749**	**0.2575**	**37.41**

In Table 10 we isolate the effect of the joint RGB and scene-coordinate denoising introduced in Section 3.1. We train an RGB-only variant of the backbone that removes the scene-coordinate input channels, the scene-coordinate output head, and the zero-initialized cross-modal attention layers, leaving the cross-view attention and all other components unchanged. Both checkpoints are trained from the same Stable Diffusion 2.1 [35] initialization for

20K steps (∼1/3 of full convergence) and evaluated under the single-view conditioned extrapolation protocol on RE10K and DL3DV-10K.

Across both datasets, the multi-modal variant improves every metric, including FID and LPIPS, indicating that the joint denoising not only sharpens reconstruction in observed regions but also tightens the perceptual quality of the generated image. We attribute this to the cross-modal attention forcing the RGB branch to commit to a 3D-consistent interpretation of the scene at every denoising step, a regularizer that pure RGB diffusion lacks.

B.7 Robustness to the Depth Front-End

Table 11: **Depth source at inference** on RealEstate10K, single-view extrapolation. **GenRec** is trained on DA3 renders (†); the estimator is swapped at test time with no retraining. Evaluated on the same RealEstate10K subset as Table 12, so values are not comparable to Table 1. Best results in **bold**.

	Global				Observed		Unobserved	
Depth	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DA3† [22]	**17.35**	**0.6169**	**0.3206**	63.69	**22.34**	**0.6736**	**15.26**	**0.5242**
MapAnything [19]	17.23	0.6151	0.3236	**63.66**	22.08	0.6717	15.15	0.5192
VGGT [45]	16.43	0.5734	0.3478	67.48	20.54	0.6092	14.79	0.4952

GenRec relies on a monocular depth front-end to build its observation mask and warped conditioning. We quantify this dependence with two studies.

Swapping the depth estimator at inference. Table 11 replaces the depth front-end at inference only, measuring what is lost when depth no longer comes from the estimator the model was trained with (DA3 [22]). Swapping in MapAnything [19] costs 0.12 dB PSNR and marginally improves FID, so the model is not tied to the estimator it was trained with; VGGT [45], whose depth differs more, costs 0.92 dB. Neither swap requires retraining, so **GenRec** also inherits future improvements in monocular depth for free: the front-end is a replaceable component, not a fixed dependency.

Table 12: **Depth-robustness sweep** on RealEstate10K (25-scene subset, so clean values are not comparable to Tables 1–3). At inference, the source depth is corrupted after the Umeyama alignment that fixes scale; every method builds its own conditioning from the corrupted depth, and all runs are scored against a single frozen observation mask. Each entry averages over the severity levels of the corresponding family: three levels for the affine corruptions, one for boundary, and three for noise. Best result per block in **bold**.

		Global				Observed		Unobserved	
Corruption	Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Clean	**GenRec**	**17.35**	**0.6196**	**0.3201**	**63.12**	**22.36**	**0.6759**	**15.25**	**0.5263**
	Gen3C [34]	15.44	0.5613	0.3627	67.04	20.04	0.6037	13.81	0.4893
	ViewCrafter [56]	13.01	0.4749	0.4505	85.04	17.27	0.5222	11.43	0.3875
Affine	**GenRec**	**15.08**	0.5122	0.4101	**67.08**	**18.55**	0.5232	**14.09**	0.4691
	Gen3C [34]	14.64	**0.5240**	**0.4071**	70.91	18.30	**0.5404**	13.56	**0.4736**
	ViewCrafter [56]	12.04	0.4217	0.4992	91.53	15.76	0.4491	10.92	0.3584
Boundary	**GenRec**	**17.23**	**0.6130**	**0.3251**	**65.87**	**22.09**	**0.6653**	**15.13**	**0.5169**
	Gen3C [34]	15.53	0.5690	0.3620	66.95	19.73	0.5961	14.06	0.5042
	ViewCrafter [56]	12.83	0.4674	0.4589	89.49	17.02	0.5118	11.21	0.3776
Noise	**GenRec**	**16.53**	**0.5765**	**0.3666**	76.48	**20.55**	**0.5986**	**14.86**	**0.5077**
	Gen3C [34]	15.25	0.5524	0.3793	**67.96**	19.32	0.5761	13.88	0.4902
	ViewCrafter [56]	11.66	0.3944	0.5202	103.82	15.12	0.4129	10.69	0.3403

Stress-testing beyond real estimator error. The second study corrupts the source depth at inference time, after the Umeyama alignment that fixes scale, using three families of corruption: an affine scale-and-shift in disparity, which models the residual ambiguity that survives alignment; boundary artifacts at depth discontinuities, which attack the observed/unobserved partition most directly; and spatially varying noise. Each family is evaluated at several severity levels, and Table 12 reports the average over levels. Since every method builds its own conditioning from the corrupted depth, the corruption propagates to both the reconstruction branch and the backbone; all runs are scored against a single frozen observation mask, so the pixel sets remain fixed across

comparisons. One caveat concerns Gen3C: its clean FID comes from the filter-disabled clean run, which is also the correct baseline for its noise rows, since those require the filter disabled (at the default setting, the renderer returns an empty mask at the strongest noise level). Measured against that run, Gen3C's noise loss is 0.53 dB rather than the 0.20 dB implied by the clean row.

Across the sweep, **GenRec** retains the PSNR lead in every corruption family and in every region, and the FID lead in all families except noise. The pattern of where ground is lost is more informative than the margins themselves. Boundary corruption, which attacks the observed/unobserved partition most directly, is the setting our design is built for, and it is also the one **GenRec** survives best. The affine family costs the most, exactly as the design predicts: affine rescaling deforms the warped evidence that the reconstruction branch consumes, so the path that benefits most from accurate depth is also the one most exposed to inaccurate depth. The same logic explains where the damage lands: in every family, the observed-region loss is two to five times the unobserved-region loss, i.e., the reconstruction path absorbs the depth error while the generative path remains comparatively insulated from it. Finally, the sweep perturbs the clean DA3 prediction rather than sensor depth, so it quantifies sensitivity to depth *error* and should not be read as bounding absolute accuracy.

B.8 Failure Modes

The studies above isolate concrete failure modes of the unified formulation and of **GenRec** itself, which we state explicitly here.

Uniform supervision. Row (b) of Table 6 is the unified-supervision configuration, and its failure is specific: fidelity on observed pixels improves while realism degrades (FID 40.43, the weakest in the table), appearing as over-smoothed or mean-collapsed content in disocclusions.

Depth-induced mask errors. When depth is substantially wrong, the mask can label a genuinely unobserved pixel as observed, and the reconstruction branch then confidently reconstructs a mis-warped region rather than deferring to the generative prior. The depth-corruption sweep of Table 12 confirms this mechanism: in every corruption family, the damage is two to five times larger in observed than in unobserved regions, exactly as the split predicts, while **GenRec** remains the strongest method on PSNR throughout.

Very low coverage. When a target view is almost entirely unobserved, the mask approaches zero and **GenRec** reduces to its backbone, so the split contributes little.

Strong view-dependent effects. On observed surfaces with strong view-dependent appearance, the source evidence available to the reconstruction branch is not the target appearance.

C Preliminaries

C.1 Flow Matching

Flow Matching [24] has emerged as a simpler and more efficient alternative to diffusion-based approaches for generative modeling. It is built around a time-dependent vector field $v : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, whose trajectories define a time-dependent diffeomorphism $\phi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ called a *flow*. For each initial point x, the curve $t \mapsto \phi_t(x)$ is the integral curve of v_t starting at x, characterized by the ordinary differential equation:

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)), \quad \phi_0(x) = x. \tag{6}$$

The flow ϕ_t induces a *probability path* p_t by pushing forward the initial distribution p_0 through the flow: $p_t = (\phi_t)_{\#}p_0$, where $p_0 = \mathcal{N}(0, I)$ is a standard Gaussian prior and $p_1 \approx q(x)$ is the target data distribution. The goal of Flow Matching is to learn a parametric vector field v_θ whose induced probability path transforms p_0 into q.

Following the optimal-transport conditional path of Lipman et al. [24], a sample at any intermediate time $t \in [0, 1]$ is obtained by linearly interpolating between a noise sample $x_0 \sim p_0$ and a data sample $x_1 \sim q$:

$$x_t = (1 - t)\, x_0 + t\, x_1, \quad t \in [0, 1]. \tag{7}$$

Differentiating the interpolant with respect to t yields the conditional velocity field

$$u_t(x_t \mid x_0, x_1) = \frac{d}{dt}\big[(1 - t)\, x_0 + t\, x_1\big] = x_1 - x_0,$$

which is constant along each interpolation trajectory. Because x_t depends linearly on the endpoints, x_t together with the velocity at time t uniquely determines x_1. Rearranging the interpolant yields a closed-form estimate of the clean sample at any $t \in [0, 1)$:

$$\hat{x}_1 = x_t + (1 - t)\, v_\theta(x_t, t), \tag{8}$$

so the learned velocity network v_θ implicitly acts as an x_1-predictor at every timestep. A single forward pass at arbitrary t therefore yields both a denoising direction and an estimate of the target, without the noise-level-dependent reweighting required by ϵ-prediction in diffusion models [14].

The network v_θ is trained with the Conditional Flow Matching (CFM) objective [24], which regresses onto the per-sample target velocity:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\,x_0,\,x_1}\left\|v_\theta(x_t, t) - (x_1 - x_0)\right\|^2, \tag{9}$$

where $x_0 \sim \mathcal{N}(0, I)$, $x_1 \sim q(x)$, and $x_t = (1-t)\,x_0 + t\,x_1$. We follow Esser et al. [9] and draw t from a logit-normal distribution, $t = \sigma(u)$ for training. Because both the interpolated sample and its target velocity are available in closed form, this loss admits unbiased stochastic estimates and is *simulation-free*, meaning training never requires integrating the ODE. At inference, a data sample is generated by drawing $x_0 \sim \mathcal{N}(0, I)$ and integrating the learned vector field from $t = 0$ to $t = 1$. Discretizing the interval into N steps with $\Delta t = 1/N$, the simplest Euler update advances the state from t_k to $t_{k+1} = t_k + \Delta t$ as

$$x_{t_{k+1}} = x_{t_k} + \Delta t \cdot v_\theta(x_{t_k}, t_k), \tag{10}$$

which can be replaced with any higher-order ODE solver (e.g. Heun or Runge–Kutta) for improved accuracy at the same number of function evaluations. We utilize the Euler update in our work as shown in Equation 10.

C.2 Diff2Flow [37]

While Flow Matching offers attractive properties such as straighter sampling trajectories and simulation-free training, the most widely available early foundation generative models (e.g., Stable Diffusion [35]) are trained as denoising diffusion models [14, 41] rather than flow models. Diff2Flow bridges this gap by aligning the two paradigms through coupled reparameterizations.

A diffusion model corrupts a data sample $x_0^{\mathrm{DM}} \sim q(x)$ with Gaussian noise $x_T^{\mathrm{DM}} \sim \mathcal{N}(0, I)$ along discrete timesteps $t_{\mathrm{DM}} \in \mathbb{Z}_{\geq 0} \cap [0, T]$. Under a variance-preserving schedule, the noisy interpolant is

$$x_{t_{\mathrm{DM}}}^{\mathrm{DM}} = \alpha_{t_{\mathrm{DM}}}\, x_0^{\mathrm{DM}} + \sigma_{t_{\mathrm{DM}}}\, x_T^{\mathrm{DM}}, \qquad \alpha_{t_{\mathrm{DM}}}^2 + \sigma_{t_{\mathrm{DM}}}^2 = 1, \tag{11}$$

with $\alpha_{t_{\mathrm{DM}}}$ monotonically decreasing and $\sigma_{t_{\mathrm{DM}}}$ monotonically increasing in t_{DM}. The network is trained either to regress the noise ϵ or, in the $\boldsymbol{v}$-parameterization [36], the target

$$v_{t_{\mathrm{DM}}}^{\mathrm{DM}} = \alpha_{t_{\mathrm{DM}}}\, x_T^{\mathrm{DM}} - \sigma_{t_{\mathrm{DM}}}\, x_0^{\mathrm{DM}}, \tag{12}$$

the parameterization used by Stable Diffusion 2.1, on which we focus throughout. Note the convention mismatch with Flow Matching: the diffusion timestep treats x_0^{DM} as data and x_T^{DM} as noise, opposite to the FM convention introduced above. The two are reconciled by the boundary identification $x_0^{\mathrm{DM}} \equiv x_1$ and $x_T^{\mathrm{DM}} \equiv x_0$.

To align the trajectories, Diff2Flow constructs invertible reparameterizations $f_t : [0, T] \to [0, 1]$ on time and $f_x : \mathbb{R}^d \to \mathbb{R}^d$ on the interpolant, chosen such that the diffusion trajectory in Eq. (11) coincides with the FM trajectory $x_t = (1-t)\,x_0 + t\,x_1$ under the above boundary identification. Equating the two expressions and solving for t in terms of the diffusion coefficients yields

$$f_t(t_{\mathrm{DM}}) = \frac{\alpha_{t_{\mathrm{DM}}}}{\alpha_{t_{\mathrm{DM}}} + \sigma_{t_{\mathrm{DM}}}}, \qquad f_x\big(x_{t_{\mathrm{DM}}}^{\mathrm{DM}}\big) = \frac{1}{\alpha_{t_{\mathrm{DM}}} + \sigma_{t_{\mathrm{DM}}}}\, x_{t_{\mathrm{DM}}}^{\mathrm{DM}}. \tag{13}$$

Under both variance-preserving and variance-exploding schedules f_t is strictly monotonic and hence invertible. The discrete coefficients $\{\alpha_{t_{\mathrm{DM}}}, \sigma_{t_{\mathrm{DM}}}\}$ are linearly interpolated between their nearest integer neighbors so that f_t and f_t^{-1} are well-defined on the entire continuous interval; this extension is justified empirically by the observation that diffusion networks trained on integer timesteps produce coherent outputs at fractional ones [37]. Given an FM timestep $t \in [0, 1]$ and interpolant x_t, the diffusion-side quantities are recovered as

$$t_{\mathrm{DM}} = f_t^{-1}(t), \qquad x_{t_{\mathrm{DM}}}^{\mathrm{DM}} = f_x^{-1}(x_t) = \big(\alpha_{t_{\mathrm{DM}}} + \sigma_{t_{\mathrm{DM}}}\big)\, x_t. \tag{14}$$

Rather than fine-tuning the diffusion network to directly emit FM velocities, which forces it to abandon its existing parameterization and degrades performance under parameter-efficient adaptation [37], Diff2Flow keeps the network in its native parameterization and analytically converts its output. Solving the linear system formed by Eq. (11) and Eq. (12) for the endpoints gives

$$\hat{x}_0^{\mathrm{DM}} = \alpha_{t_{\mathrm{DM}}}\, x_{t_{\mathrm{DM}}}^{\mathrm{DM}} - \sigma_{t_{\mathrm{DM}}}\, v_\theta^{\mathrm{DM}}\big(x_{t_{\mathrm{DM}}}^{\mathrm{DM}}, t_{\mathrm{DM}}\big), \qquad \hat{x}_T^{\mathrm{DM}} = \sigma_{t_{\mathrm{DM}}}\, x_{t_{\mathrm{DM}}}^{\mathrm{DM}} + \alpha_{t_{\mathrm{DM}}}\, v_\theta^{\mathrm{DM}}\big(x_{t_{\mathrm{DM}}}^{\mathrm{DM}}, t_{\mathrm{DM}}\big). \tag{15}$$

Substituting the boundary identification into the FM target $x_1 - x_0$ then yields a closed-form FM velocity expressed entirely through the diffusion network's $\boldsymbol{v}$-prediction:

$$v_\theta(x_t, t) \;=\; \hat{x}_0^{\mathrm{DM}} - \hat{x}_T^{\mathrm{DM}} \;=\; (\alpha_{t_{\mathrm{DM}}} - \sigma_{t_{\mathrm{DM}}})\, x_{t_{\mathrm{DM}}}^{\mathrm{DM}} \;-\; (\alpha_{t_{\mathrm{DM}}} + \sigma_{t_{\mathrm{DM}}})\, v_\theta^{\mathrm{DM}}\big(x_{t_{\mathrm{DM}}}^{\mathrm{DM}}, t_{\mathrm{DM}}\big), \tag{16}$$

with $t_{\mathrm{DM}} = f_t^{-1}(t)$ and $x_{t_{\mathrm{DM}}}^{\mathrm{DM}} = f_x^{-1}(x_t)$ given by Eqs. (13)–(14). An analogous derivation holds for ϵ-parameterized priors.

Training proceeds with the standard CFM loss in Eq. (9), where the velocity prediction is evaluated through Eq. (16). Sampling uses the Euler update of Eq. (10) on the FM trajectory: at each step, the current state x_{t_k} and timestep t_k are mapped to the diffusion side via f_t^{-1} and f_x^{-1}, the network is queried in its native parameterization, and its $\boldsymbol{v}$-prediction is converted back to an FM velocity through Eq. (16).

Inference Protocol. At test time we sample with 50 solver steps using the EMA weights accumulated during training. We employ classifier-free guidance with a scale of 1.5.

C.3 Implementation Details

Generative Backbone. We initialize the multi-view generative backbone from the Stable Diffusion 2.1 (SD-2.1) checkpoint [35] and zero-initialize the cross-modal attention layers. The number of attention heads and head dimensions are reused from SD-2.1 and is symmetric for both cross-view and cross-modal attention. We reuse the pretrained SD-2.1 VAE for the image modality and adopt the scene-coordinate VAE from SpatialGen [10] for the scene-coordinate modality; both VAEs remain frozen throughout training. Plücker ray conditioning is embedded via the SpatialGen ray encoder.

VAE Skip Adapter and LoRA. The mask-conditioned skip adapter (Section 3.1) uses a bias-free 1×1 convolution per skip, initialized with standard deviation 10^{-5} so that an untrained adapter reproduces the pretrained decoder within numerical noise. We adapt the decoder with low-rank residuals via PEFT [28], targeting every convolution and attention projection together with the four skip convolutions: rank $r = 8$, scaling $\alpha = r$, Gaussian initialization, biases untouched.

Reconstruction Branch. The branch operates at an internal width of D=64 with four FiLM-conditioned residual blocks interleaved with sparse geometric attention layers (4 heads, head dimension 16). The attention layers between the The query and source streams are projected to D via 3×3 and 1×1 convolutions respectively. Sparsity in both the geometric cross-attention and self-attention is enforced through precomputed k-nearest-neighbour index maps with k=8.

C.4 Training Details

Our pipeline follows two stages: we first train the generative backbone together with the VAE skip connections and the decoder LoRA weights, then freeze it and train the reconstruction branch with iterative denoising supervision.

Data. We train our method on a mixture of RealEstate10K [60], DL3DV-10K [23] and SpatialVID [44] with sampling weights 0.5, 0.2 and 0.3 respectively, all loaded at a longest-side resolution of 616 pixels with dataset-specific frame strides. From SpatialVID we filter out dynamic scenes by thresholding the depth-projection error between frames computed with DA3 depth, retaining only scenes whose static-geometry assumption holds. These datasets together form a collection of around 160K indoor and outdoor multi-view consistent scenes. For every scene we sample $T = 8$ consecutive frames at the dataset-specific stride; the number of clean input views T_{in} is drawn from $\{1, 2\}$ with probabilities 0.8 and 0.2, placed at the first T_{in} positions of the sampled window, and the remaining $T_{\text{out}} = T - T_{\text{in}}$ views are noised. Metric depth is computed by Depth Anything 3 (DA3) [22] at 504 resolution, matching DA3's base training resolution. We use Umeyama alignment between the predicted camera poses from DA3 and the ground-truth poses to scale the depth estimation for training.

Multi-view Generation Backbone. We train the generative backbone for 20 epochs ($\sim$72K optimizer steps) with an effective batch size of 16 using 8-bit AdamW, learning rate 10^{-5}, no weight decay, and a global gradient-norm clip of 1.0. The learning rate is linearly warmed up over 500 steps and decayed with a cosine schedule to a minimum of 10^{-6}. To enable classifier-free guidance at inference, we drop the geometric conditioning (warped renders and observation mask) independently with probability 0.1 during training. Training is performed in `bfloat16` mixed precision with gradient checkpointing enabled throughout, distributed across 16 GH200 GPUs via PyTorch DDP for a total wall-clock time of approximately three days. We maintain an exponential-moving-average copy of the weights with decay 0.9999.

Reconstruction Branch. The branch is trained in the same optimization loop as the frozen generative backbone but with fully decoupled gradients: gradients influence only the residual predictor. To match the distribution of samples encountered at inference, the detached $\hat{x}_1^I$ is produced by an unrolled denoising rollout of 50 no-gradient solver steps, terminated at a randomly chosen step whose corresponding diffusion timestep, recovered from the flow time t via the Diff2Flow mapping f_t^{-1} on SD2.1's 1000-step schedule, falls below 200. This exposes the branch to realistic, near-final samples rather than the single-step estimates available during diffusion training. The branch parameters form a separate optimizer group with learning rate 5×10^{-5}, weight decay 10^{-4}, and a 1000-step linear warm-up; all other settings (8-bit AdamW, gradient-norm clip 1.0, `bfloat16` mixed precision) match those of the backbone. The branch converges within 10K steps.

Loss Weights. For the geometry-aware decoder adapter (Eq. 4), we set $\lambda_1 = 0.1$ and $\lambda_{\text{p}} = 0.3$. For the pixel-space reconstruction branch (Eq. 5), we set $\lambda_1 = 0.1$, $\lambda_{\text{p}} = 0.5$, and $\lambda_\Delta = 0.05$.

D Societal Impacts

GenRec synthesizes novel views of real-world scenes from one or two posed images, with applications in robotic navigation under sparse observation, mixed-reality and accessibility. The order-of-magnitude inference speedup over warp-as-target video baselines makes such uses practical on commodity hardware.

As with any photorealistic generative model, outputs could in principle be misused to fabricate imagery, and the learned prior reflects the distribution of its training data. We train on RealEstate10K, DL3DV-10K, and SpatialVID, which are publicly released, mostly widely adopted benchmarks in the 3D vision community; the resulting biases are those already characterized in the literature surrounding these datasets, and the privacy considerations are governed by the curation policies under which they were released. We do not introduce new data collection.

A property of GenRec's design that is directly relevant here is that the observation mask O_i separates pixels recovered by reconstruction from pixels produced by the generative prior, and this separation is available at inference at no additional cost. The mask functions as a built-in provenance signal: it identifies, per pixel, whether the model is reporting evidence or imagining a plausible completion. We encourage downstream systems to surface this signal (through overlays, metadata, or content-credential standards such as C2PA) rather than discarding it once the final image is composited.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three concrete contributions: (i) a geometry-grounded decomposition of generative novel-view synthesis into observed and unobserved regions via a per-pixel observation mask, (ii) a multi-view flow matching backbone that jointly denoises RGB and scene-coordinate maps, and (iii) a pixel-space reconstruction branch with sparse 3D cross-attention. Each is realised in the method (Section 3) and validated by the experiments on RealEstate10K, DL3DV-10K, and Mip-NeRF 360 in Section 4 and the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated *Limitations* paragraph at the end of Section 5 discusses two factors: (i) the dependence of the observation mask and warped conditioning on the accuracy of the off-the-shelf monocular metric depth estimator, and (ii) the computational cost of the bi-modal architecture, which currently restricts the number of views per scene during training.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not introduce new theoretical results. Appendix C recaps Flow Matching and the Diff2Flow [37] reparameterization from prior work for completeness, with all assumptions and derivations attributed to the original papers.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 fully specifies the architecture, conditioning inputs, loss functions, and the two-phase training schedule. Appendix C.3 (Implementation Details) and Appendix C.4 (Training Details) document the backbone initialisation (SD-2.1), VAE choices, LoRA configuration, reconstruction-branch hyperparameters, optimizer, learning-rate schedule, EMA, mixed precision, batch size, dataset mixing weights, image resolution, view sampling protocol, and inference-time solver settings. The evaluation protocol, datasets, baselines, and metrics are described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [N/A]

   Justification: We plan to release the GenRec code and trained checkpoints with the camera-ready version, accompanied by documentation covering training and inference instructions, dataset preparation, and evaluation scripts.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes data splits, view-sampling strides, the observation-mask protocol used for region-wise metrics, and the choice of baselines. Appendix C.4 provides the optimizer, learning rates, warm-up and cosine decay schedules, EMA decay, gradient clipping, mixed-precision settings, batch size, and loss weights. Inference settings are also stated in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars. Each evaluation is computed over a fixed test set shared across all baselines, with deterministic camera trajectories and source views, so the comparison across methods is controlled. Re-running training with multiple seeds is prohibitively expensive given the bi-modal flow matching backbone and the 16-GPU training budget required per run. We follow the same evaluation convention as the baselines we compare against.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.4 states that the generative backbone is trained for 20 epochs across 16 NVIDIA GH200 GPUs in bfloat16 mixed precision with gradient checkpointing, an effective batch size of 16, and EMA tracking. Inference timings for our method and all baselines are reported in Tables 1, 2, and 3 (Inf. Time column), allowing direct comparison of compute footprints.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research uses publicly available scene datasets that do not contain personally identifying or sensitive content beyond what is already public, builds on openly licensed pretrained models, and does not involve human subjects. The work conforms to the NeurIPS Code of Ethics in all respects.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section D discusses both positive impacts (e.g., robotic navigation under sparse observation, mixed-reality) and negative ones (e.g., potential misuse for fabricating imagery, dataset-inherited biases), along with the role of the observation mask as a built-in provenance signal for the generations in unobserved regions.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: GenRec is a scene-level novel-view synthesis model that requires posed source images as conditioning and operates on real-estate and indoor/outdoor scene datasets; it does not generate content involving identifiable people, faces, or text, and its capabilities are narrower than general-purpose image or video generators. We therefore consider the misuse risk to be modest relative to such models and do not implement specific gated-release safeguards beyond the standard practice of citing licences and crediting the upstream pretrained components.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (RealEstate10K [60], DL3DV-10K [23], SpatialVID [44], Mip-NeRF 360 [2]), pretrained models (Stable Diffusion 2.1 [35], SpatialGen scene-coordinate VAE and ray encoder [10], Depth Anything 3 [22]), and software components used in this work are publicly released and cited in the paper. We use them in accordance with their respective licences and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets or models are released with the submission. We plan to release the GenRec code and trained checkpoints with the camera-ready version, accompanied by documentation covering training and inference instructions, dataset preparation, and evaluation scripts.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All evaluations use automatic metrics (PSNR, SSIM, LPIPS, FID) on public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so no IRB review or equivalent approval was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as part of the core methodology. Any use of LLMs was limited to incidental writing assistance (grammar, phrasing) and does not affect the scientific content, methodology, or results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.