From Rays to Projections: Better Inputs for Feed-Forward View Synthesis

Supplementary Material

1. Additional Experiments

Ablation Studies on Positional Embedding. Tab. 1 highlights the importance of Rotary Position Embedding (RoPE). While adding RoPE to the LVSM baseline ('LVSM + RoPE') yields only a slight PSNR improvement, omitting positional encoding in our architecture causes a substantial performance drop. To illustrate this failure mode, Fig. 1 presents a toy experiment where we overfit a single-object scene: without RoPE, the model collapses to predicting completely identical patches over the empty regions, regressing to the mean background color of the ground-truth image.

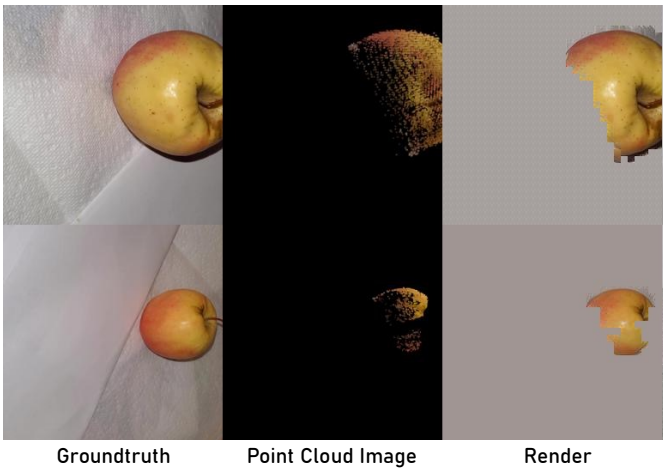

Figure 1. Without RoPE, the model produces degraded results on the identical patches.

	LVSM	LVSM + RoPE	Ours w/o RoPE	Ours
PSNR ↑	25.39	25.88	21.18	30.03

Table 1. **Ablation studies** on the use of RoPE [5].

Additional Comparisons with LVSM [3]. We show more qualitative comparisons with LVSM [3] on the RealEstate10K dataset [8] in Fig. 2. Without direct geometric cue from the projected point cloud, LVSM often produces wrong prediction on geometry.

Results on the pretraining and the finetuning stage. We also show results on the MAE pretraining stage and the finetuning stage in Fig. 3 and Fig. 4 respectively.

Robustness to alternative geometry priors. To test whether our gains are tied to a single geometry estimator, we add two additional evaluations in Tab. 2. First, we perturb the camera positions used to construct the projection cue with Gaussian translation noise (mean 10 cm, standard deviation 5 cm), which simulates imperfect upstream geometry. Second, we replace the depth backbone used to construct the projection cue with DepthAnything v3. In both cases, the performance degrades only modestly. We also compare against a stronger Gaussian baseline built from WorldMirror by providing ground-truth cameras, and then both ground-truth cameras and MapAnything depth. While these stronger priors improve the Gaussian baseline, our method still performs better overall, indicating that the gains are not solely explained by a particular depth or camera estimator.

	NVS Benchmark			Consistency Benchmark (PSNR-M ↑)			
	PSNR ↑	SSIM ↑	LPIPS ↓	Aniso-pixel	World Scale	FOV	Roll
WM+GT-Cam	22.29	0.758	0.242	21.55	22.39	20.33	18.15
WM+GT-Cam+MA	22.12	0.752	0.251	21.40	22.22	20.25	17.94
Ours	**26.90**	**0.851**	**0.133**	20.33	**25.78**	**21.55**	**20.04**
Ours+DAv3	26.55	0.848	0.134	-	-	-	-
Ours+Noise	25.57	0.796	0.148	19.85	25.58	20.76	19.96

Table 2. **Additional quantitative evaluation results.** "WM" denotes WorldMirror, "GT-Cam" uses ground-truth cameras, "MA" uses MapAnything depth, "DAv3" uses DepthAnything v3 depth, and "Noise" adds Gaussian translation noise to the cameras used for projection construction.

Exact definitions of benchmark transformations. Let the target camera have intrinsics $K = \mathrm{diag}(f_x, f_y, 1)$ with principal point (c_x, c_y), rotation R, and camera center $\mathbf{C}$. Unless otherwise noted, transformations are applied only to the target view while the context views remain unchanged.

For **Anisotropic Pixel**, we apply independent scaling factors s_x and s_y to the horizontal and vertical focal lengths:

$$f'_x = s_x\, f_x, \quad f'_y = s_y\, f_y, \quad s_x \neq s_y, \tag{1}$$

which changes the effective pixel aspect ratio while preserving the target image resolution. The target RGB image is unchanged.

For **World Scale**, we multiply all camera centers (both context and target) and depths by a common scalar $s > 0$:

$$\mathbf{C}'_i = s\,\mathbf{C}_i, \quad Z'_i = s\,Z_i, \quad \forall\, i, \tag{2}$$

while keeping intrinsics, rotations, and the RGB target unchanged. This is the gauge transformation discussed in the main text.

For **FOV**, we scale both target focal lengths by a common factor $\alpha > 0$:

$$f'_x = \alpha\, f_x, \quad f'_y = \alpha\, f_y, \tag{3}$$

which changes the vertical field of view from θ to $\theta' = 2\arctan\big(\tan(\theta/2)\,/\,\alpha\big)$. The target image is resampled accordingly, producing zoom-in ($\alpha > 1$) or zoom-out ($\alpha < 1$) viewpoints under a standard pinhole camera.

Figure 2. More qualitative comparisons with LVSM [3] on the RealEstate10K dataset [8].

For **Roll**, we rotate the target camera around its optical axis by an angle ϕ:

$$R' = R_z(\phi)\, R, \quad R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

which corresponds to an in-plane rotation of the target image by ϕ. The camera center and intrinsics remain unchanged.

Object-centric Experiments. We further evaluate our method on an object-centric dataset G-Objaverse [1, 4, 9], which contains the rendered G-buffers of the objects in the Objaverse dataset [1] in around 30 view directions. We follow the same pre-training then finetuning strategy as in the main paper, and the results are shown in Fig. 5. For object-level scenes, we additionally add a layer of random colored Gaussians behind the depth surface to address ambiguities in the projected point cloud image.

2. Limitations

We discuss the limitations of our method in this section. First, similar to prior regression-based methods [2, 3], our method only interpolates between the context views, so its ability to hallucinate unseen regions remains limited. Although we show better performance on unseen regions (Tab. 3 in the main paper), it is still biased toward regressing the "average" completion supported by the training dataset (e.g., walls, floors, or sky). This limitation is particularly visible under large disocclusions, where the projection cue explicitly marks missing content as holes but does not resolve the underlying ambiguity. Future work could consider combining our conditioning interface with generative models [7] to generate more diverse and realistic completions.

Second, our method depends on the quality of the upstream depth and camera estimates used to construct the projection cue. Although the additional experiments above show that the method is not tied to a single estimator, severe errors can still create holes, double edges, or misplaced structures that the decoder must correct. Finally, our method is restricted to static scenes: when presented with dynamic objects, it can produce ghosting, blurriness, or inconsistent results across frames. Though the pretraining stage does not impose a static-scene assumption, more diverse training data and a dedicated fine-tuning strategy are necessary to handle dynamic scenes in our pipeline.

3. Dolly Zoom Camera Motion

The dolly zoom (also known as the Hitchcock shot [6]) is a camera motion where the camera is translated along its viewing direction while the focal length is adjusted so that a chosen object keeps a constant image size. This creates the characteristic effect that the foreground object stays fixed in scale while the background appears to expand or contract.

We model the camera with intrinsics $K = \mathrm{diag}(f_x, f_y, 1)$ and principal point $\mathbf{c} = (c_x, c_y)$. For a 3D point $\mathbf{X} = (X, Y, Z)^\top$ in camera coordinates, the pinhole projection is

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad u = f_x \frac{X}{Z} + c_x,\ v = f_y \frac{Y}{Z} + c_y. \quad (5)$$

Thus the apparent size of an object at depth Z scales proportionally to f_y/Z.

Let $\mathbf{C}_0$ be the initial camera center and let $\mathbf{n}_0 \in \mathbb{R}^3$ denote the unit forward direction (the third column of the

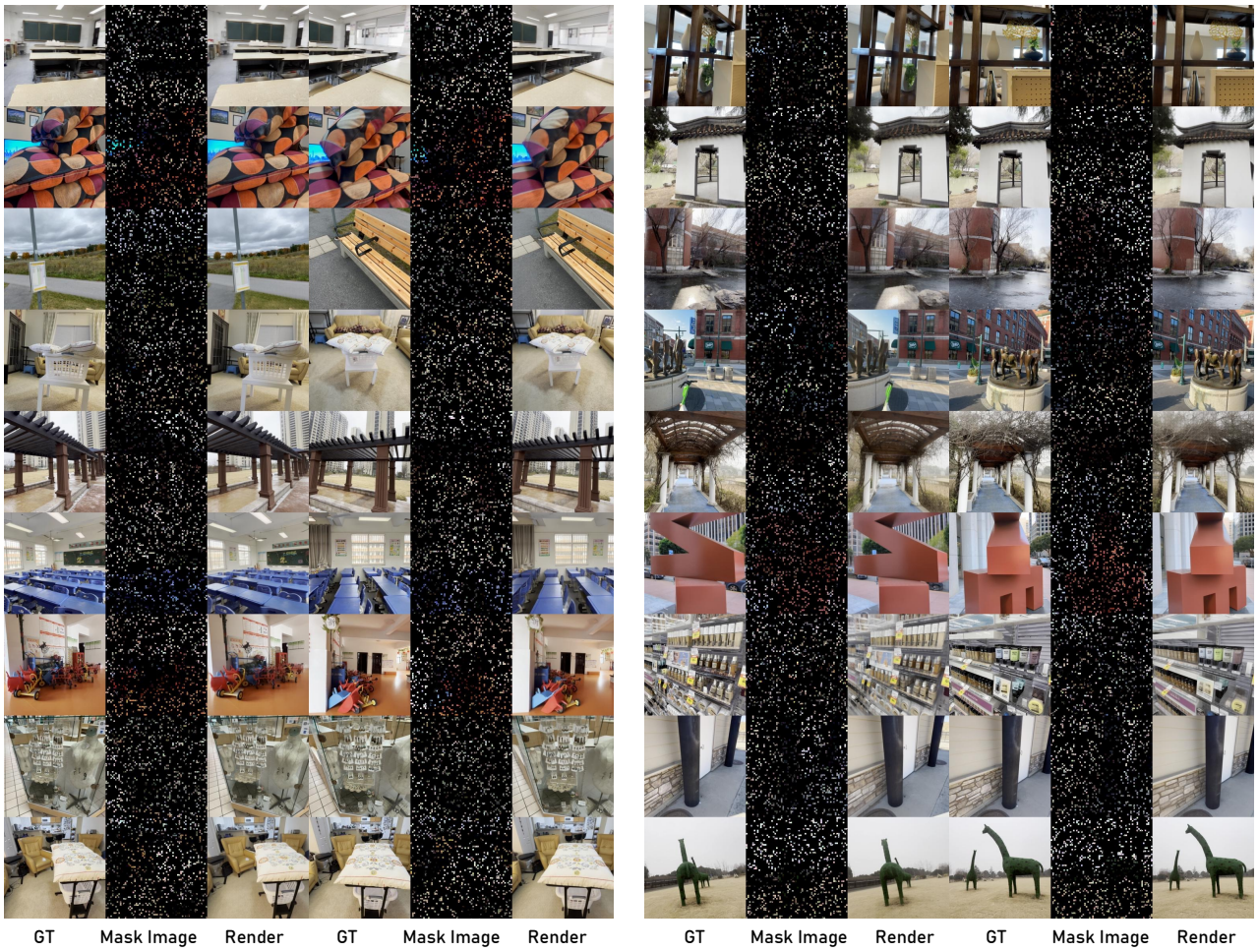

Figure 3. Additional results on the MAE pretraining stage.

rotation matrix R_0). We pick an anchor point $\mathbf{X}_\star$ on the object whose size we wish to keep fixed. Its initial depth is:

$$Z_0 = \mathbf{n}_0^\top (\mathbf{X}_\star - \mathbf{C}_0). \tag{6}$$

During a dolly zoom, the camera is translated along $\mathbf{n}_0$ to:

$$\mathbf{C}(t) = \mathbf{C}_0 + \Delta(t)\, \mathbf{n}_0, \tag{7}$$

while the orientation is kept constant, $R(t) = R_0$. The depth of the anchor point in the new camera is then:

$$Z(t) = \mathbf{n}_0^\top (\mathbf{X}_\star - \mathbf{C}(t)) = Z_0 - \Delta(t). \tag{8}$$

To keep the anchor's image size constant, we require that its scale factor $f_y(t)/Z(t)$ remains equal to the initial value f_{y_0}/Z_0:

$$\frac{f_y(t)}{Z(t)} = \frac{f_{y_0}}{Z_0} \;\rightarrow\; f_y(t) = f_{y_0}\, \frac{Z_0 - \Delta(t)}{Z_0}. \tag{9}$$

Equation (9) is the core constraint of the dolly zoom: as the camera moves closer to the object ($\Delta(t) > 0$), the focal length must decrease to preserve the ratio f_y/Z; moving the camera away requires increasing the focal length.

If we parameterize the camera by its vertical field of view $\theta(t)$ instead of $f_y(t)$, for an image of height H pixels:

$$f_y(t) = \frac{H}{2\tan(\theta(t)/2)}. \tag{10}$$

Combining this with (9) yields

$$\tan\left(\frac{\theta(t)}{2}\right) = \tan\left(\frac{\theta_0}{2}\right) \frac{Z_0}{Z_0 - \Delta(t)}, \tag{11}$$

where θ_0 is the initial field of view. In practice, we pick an anchor frame, estimate Z_0 for a reference pixel (e.g., the image center), and then, for each target field of view $\theta(t)$, translate the camera center by $\Delta(t)\, \mathbf{n}_0$ and adjust the intrinsics according to the relations above. This realizes a physically consistent dolly zoom trajectory in a standard pinhole camera model.

Figure 4. Additional results on the finetuning stage.

References

[1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 2, 5

[2] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A Self-supervised Large View Synthesis Model, arXiv, 2505.00702, 2025. 2

[3] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2

[4] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9914–9925, 2024. 2, 5

[5] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, arXiv, 2104.09864, 2023. 1

[6] F. Truffaut. *Hitchcock: A Definitive Study of Alfred Hitchcock*. Paladin/Grafton, 1986. 2

[7] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders, arXiv, 2510.11690, 2025. 2

[8] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018. 1, 2

[9] Qi Zuo, Xiaodong Gu, Yuan Dong, Zhengyi Zhao, Weihao Yuan, Lingteng Qiu, Liefeng Bo, and Zilong Dong. High-fidelity 3d textured shapes generation by sparse encoding and adversarial decoding. In *European Conference on Computer Vision*, 2024. 2, 5

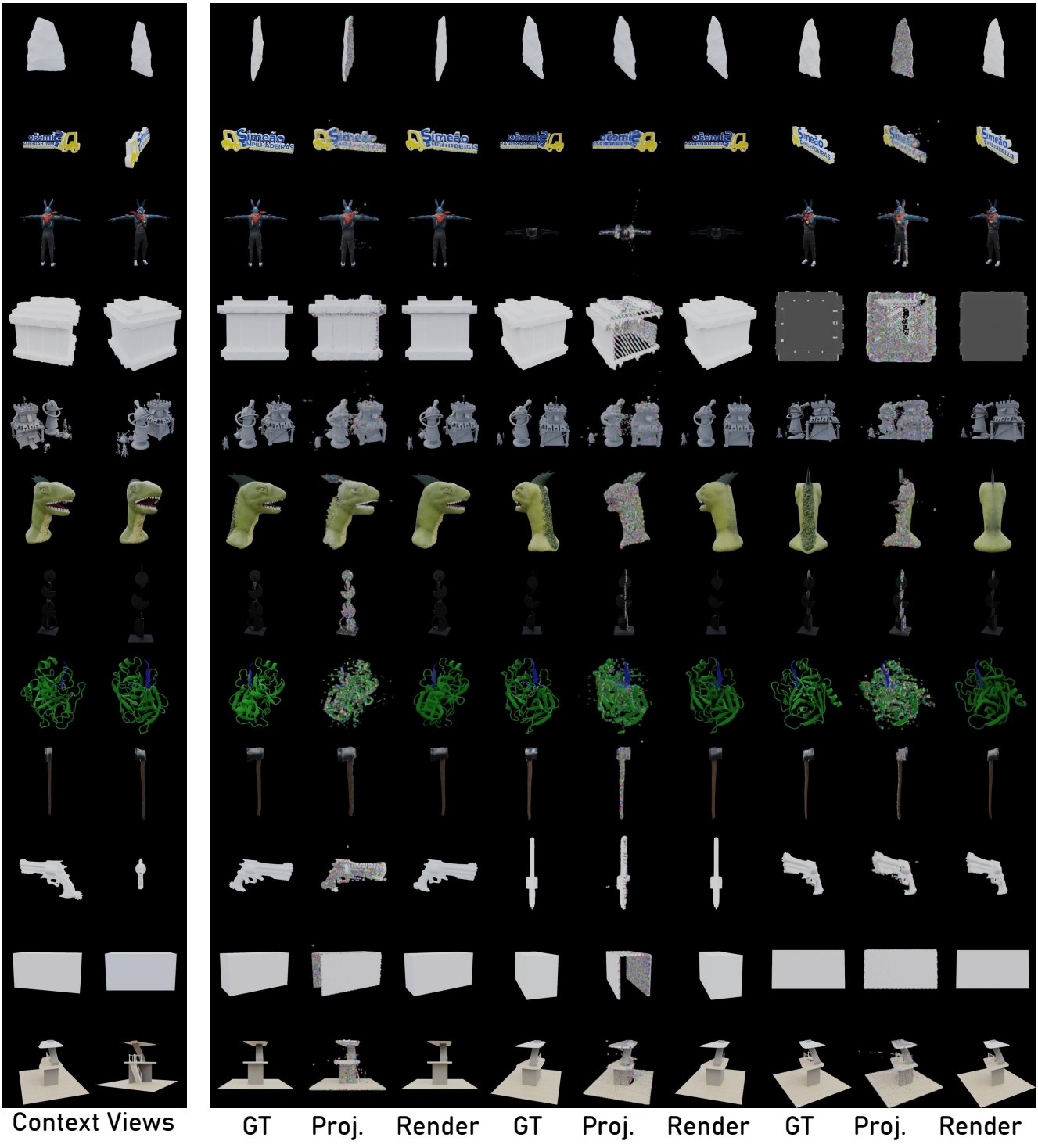

Figure 5. Results on G-Objaverse dataset [1, 4, 9]. Different from scene-level datasets, we additionally add a random colored layer behind the seen surfaces to prevent symmetry ambiguities.