Multi-view Pyramid Transformer: Look Coarser to See Broader

Supplementary Material

A. Additional Details.

Datasets. We adopt the official DL3DV [16] split for benchmark dataset. In total, we use 9,995 scenes for training and 140 scenes for benchmarking, with no overlap between the two splits. For zero-shot inference, we use the `train` and `truck` scenes in Tanks&Temples [13], following [12, 32], and 9 scenes in Mip-NeRF360 [3] dataset. Considering our evaluation resolution of 960×540, we use downsampled images (from the original images) whose resolution is closest to, but not smaller than, this target resolution (Tab. 1).

scene name	downsample	scene name	downsample
`bicycle`	4	`room`	2
`bonsai`	2	`stump`	4
`counter`	2	`flower`	4
`garden`	4	`treehill`	4
`kitchen`	2		

Table 1. Resolution of Mip-NeRF360 evaluation dataset.

During evaluation, we select every eighth frame in each sequence as target views:

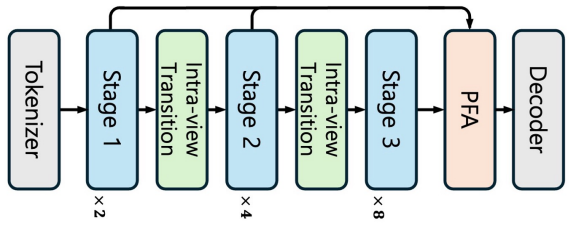
- **DL3DV**: For input views, we follow Long-LRM [32] and adopt their released indices for 16 to 128 input images. For additional settings with 192 and 256 input images, we uniformly sample input views from the remaining frames after excluding all target indices.
- **Tanks&Temples**: For the 32-view setting, we use the input-view indices provided by Long-LRM, while for the 64- and 128-view settings, input views are uniformly sampled from non-target frames.
- **Mip-NeRF360**: For all view configurations (32, 64, 128), input views are uniformly sampled from the frames excluding the target indices.

Importantly, target view indices are kept fixed across all input-view configurations for each scene.

Implementation details. The proposed MVP architecture consists of 14 transformer blocks. Within each attention [21] block, we first apply LayerNorm [2] to the input, and then perform QK-Norm [9] on the query and key projections using an RMSNorm [29] layer. Each block comprises a multi head attention module with head dimension 64, followed by a two layer feed forward network with GELU [8] activation. For the spherical harmonics representation, we set the degree to 1 for color and 2 for opacity.

For the post-activation parameterization of 3D Gaussians [12], we follow the implementations from Long-LRM and iLRM [11]. For rendering, we employ the gsplat [27] library for efficient differentiable rasterization.

We employ PRoPE [14], which represents camera poses as relative positional signals within the attention mechanism. Unlike the original implementation, we use a "Plücker (extrinsics and intrinsics) rays + PRoPE" formulation instead of "Cam (intrinsics only) rays + PRoPE", as we found this variant to yield better empirical performance.



Method	Stage 1	Stage 2	Stage 3	Seq. len.	Hidden dim.	# Register tokens
Final model	Frame(2)	Group(4)	Global(8)	$(L, \frac{1}{4}L, \frac{1}{16}L)$	(256, 512, 1024)	4 per frame

Figure 1. Architecture overview for our final MVP model. MVP employs a multi-stage hierarchy performing self-attention across increasing frame coverage: frame-wise, group-wise, and global. Each stage is linked by an Intra-view Transition layer using convolution for spatial downsampling and channel up-projection. Finally, a Pyramidal Feature Aggregation module fuses multi-level representations into a unified feature map.

Training details. We train MVP transformer using a three-stage training schedule:

- **First stage**: We train at 480×256 resolution for 100k iterations with a learning rate of $2e^{-4}$. Each training sample uses 32 input views and 12 target views, with a batch size of 8 per GPU, resulting in a total batch size of 256. The frame interval is uniformly sampled between 64 and 128.
- **Second stage**: We increase the resolution to 960×540 and train for 50k iterations with a learning rate of $2e^{-5}$. In this stage, we use 32 input views and 6 target views, with a batch size of 2 per GPU for a total batch size of 64. We sample input views from the full frame range and apply the intrinsic augmentation strategy proposed in [32].
- **Third stage**: We keep the resolution at 960×540 and further train for 30k iterations with a learning rate of $2e^{-5}$, using various numbers of input and target views to improve robustness across different view configurations. We again draw input views from the entire frame range, while continuing to use the intrinsic augmentation strategy.

For all stages, we use a cosine learning rate scheduler (with a 3k-step warmup in the first stage only) and the AdamW [17] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, and a weight decay of 0.05. Weight decay is not applied to normalization and bias parameters. We also employ EMA [7] and do not use gradient clipping, following

Method	16 views				32 views				64 views				128 views			
	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓
3D-GS$_{30k}$ [12]	21.96	0.766	0.237	8min	25.09	0.838	0.175	8min	28.02	0.890	0.134	8min	29.88	0.917	0.115	8min
Long-LRM [32]	20.65	0.707	0.328	0.50	23.54	0.776	0.270	0.84	23.15	0.787	0.263	2.08	20.78	0.741	0.307	6.39
iLRM [11]	21.62	0.746	0.316	0.19	23.93	0.800	0.259	0.53	24.11	0.816	0.243	1.66	22.72	0.804	0.251	5.61
Ours	**23.49**	**0.799**	**0.238**	**0.09**	**25.75**	**0.848**	**0.186**	**0.17**	**27.62**	**0.883**	**0.154**	**0.36**	**28.80**	**0.903**	**0.136**	**0.77**

Table 2. Quantitative comparisons on the DL3DV evaluation dataset with varying numbers of input views. For all metrics, we report the results by re-evaluating the models from their 32-view checkpoints.

Method	192 views				256 views			
	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s) ↓
3D-GS$_{30k}$ [12]	30.53	0.926	0.109	8min	30.75	0.929	0.107	8min
Long-LRM [32]	OOM (GPU memory limit exceeded, 80 GB)							
iLRM [11]	21.57	0.787	0.267	11.91	20.62	0.768	0.283	20.92
Ours	**29.32**	**0.911**	**0.130**	**1.23**	**29.42**	**0.913**	**0.128**	**1.84**

Table 3. Quantitative comparisons on the DL3DV evaluation dataset with varying numbers of input views.

[18]. To improve training efficiency, we employ FlashAttention2 [5] and gradient checkpointing [4]. We also use mixed-precision training with bfloat16 to speed up optimization while preserving numerical stability.

Tab. 4 provides our training configurations with different numbers of input views. We adjust the batch size and the number of target renderings to balance iteration time and memory consumption. As described in the main manuscript, we freeze the frame- and group-wise attention blocks and train only the global attention blocks, which enables efficient fine-tuning while preserving the learned local and group representations.

# Input views	16	32	64	128
Batch size (per GPU)	4	2	1	1
# Target views	6	6	6	1

Table 4. Training configurations for varying input views.

B. Additional Results

We additionally evaluate our method on the recently released DL3DV [16] evaluation split, which comprises 51 scenes in our experiments. Tab. 2 and 3 present quantitative results comparing our approach with 3D Gaussian Splatting [12] (3D-GS), Long-LRM [32], and iLRM [11]. Our method surpasses existing feed-forward approaches by a large margin in both reconstruction quality and inference efficiency. We evaluate 16-view metrics of Long-LRM and iLRM using checkpoints trained with 32 input views.

Additional zero-shot comparisons We also evaluate our method on the real-world ScanNet++ [28] dataset in Tab. 5 under sparse input-view settings, which naturally reflects robustness to larger viewpoint variations (Fig. 2).

Additional attention visualization. Using the first frame as the reference view, we begin by selecting three query patches. For each query patch, we visualize its top-3 attended tokens from other viewpoints across different stages. At stage 2, we display the attended tokens restricted to the same group, whereas at stage 3, we additionally include tokens attended from views outside the group (Fig. 4).

Method	16 views			32 views			64 views		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Long-LRM	24.42	0.873	0.224	28.31	0.909	0.195	27.20	0.905	0.207
iLRM	26.78	0.896	0.240	28.45	0.913	0.214	28.07	0.914	0.213
Ours	**27.41**	**0.900**	**0.198**	**28.63**	**0.910**	**0.187**	**29.11**	**0.916**	**0.185**

Table 5. Quantitative comparisons on the ScanNet++ dataset under sparse input view setting.

Figure 2. Qualitative performance on the ScanNet++ dataset.

Additional design choice. We ablate the number of views per group (2, 4, and 8) in Tab. 6. In addition to our primary objective of novel view synthesis, we also evaluate the model on a spatial cognition task [14]. In this task, one camera pose in a set of image–camera pairs is corrupted and the model must identify the mismatched pair, which probes its multi-view awareness. We replace the 3D-GS linear decoder with a head that produces a single scalar per token. For each input view, these scalars are averaged across all tokens to yield a single score, and a softmax over all views then produces a probability vector indicating which image–camera pair is most likely to be corrupted. A group size of 4 (baseline) offers consistently strong accuracy across different numbers of views, while groups of 2 provide too little context and groups of 8 bring only marginal gains with higher computational cost. Therefore, we adopt a group size of four as our default setting.

We also ablate the usage of register tokens in Tab. 7. Register token [6] was introduced to mitigate abnormally high token norm values and artifacts in the attention maps of large Vision Transformer models, which often lead to attention artifacts and degraded dense prediction performance. Consistent with this observation, we find that the variant

Method	Novel view synthesis			Spatial cognition		
	PSNR ↑	SSIM ↑	LPIPS ↓	8 views	16 views	32 views
Group 2	22.94	0.711	0.299	51.3%	75.7%	83.6%
Group 4 (baseline)	23.18	0.716	0.300	77.1%	91.4%	96.4%
Group 8	23.18	0.717	0.297	84.3%	90.7%	97.1%

Table 6. Ablation studies on the number of views per group.

without register tokens exhibits higher $L2$ norms across frames, while incorporating register tokens yields modest performance gains. The configuration for this ablation is the same as that of the ablation studies in the main manuscript.

Method	PSNR	SSIM	LPIPS	Avg. intra-frame feature norm		
				Stage 1	Stage 2	Stage 3
Baseline	22.79	0.733	0.235	14.01	84.49	113.36
w/o Register	22.52	0.723	0.242	15.98	286.61	229.82

Table 7. Ablation study on register tokens.

Point map estimation. In addition to novel view synthesis, we also evaluate our method on point map estimation, which quantifies the geometric accuracy of the reconstructed 3D scenes represented with explicit primitives. We evaluate point map accuracy on NRGBD [1] and ETH3D [19], and report the Chamfer distance and the F1-score. Specifically, we first back-project the ground-truth depth maps into 3D point clouds using the camera poses. We then rigidly align the predicted point clouds to these reference points with the Umeyama algorithm [20], and finally apply the masks to discard points in invalid regions. We use an image resolution of 960×540, which matches the training resolution for both our method and the baselines. It is important to note that both our approach and the baselines are trained using only a photometric rendering loss from 3D-GS, in contrast to geometry-supervised methods that exploit ground-truth depth or point clouds [22–25]. Consequently, the numbers in Tab. 8 should be read primarily as evidence of relative improvements within this setting, rather than as a direct comparison to fully geometry-supervised models. Overall, our method outperforms the baseline, even though the baseline is additionally regularized using pretrained DepthAnything [26] model during training.

Method	Views	NRGBD [1]		ETH3D [19]	
		CD ↓	F1-score ↑	CD ↓	F1-score ↑
Long-LRM [32]	16	0.53	0.52	2.75	0.32
Ours		**0.18**	**0.54**	**1.74**	**0.34**
Long-LRM	32	0.43	**0.59**	2.69	0.39
Ours		**0.14**	0.56	**2.22**	**0.42**

Table 8. Quantitative comparison of point map estimation. We set the threshold value for f1-score to 0.1. We exclude iLRM [11] from this evaluation, as it predicts low-resolution Gaussians that would require additional upsampling, potentially degrading its performance and hurt a fair comparison.

Inference time comparison. Fig. 3 reports inference time as a function of the number of input views. We measure all timings at an input resolution of 960×540 using the official implementations released by the respective authors. Our method exhibits substantially better scalability than existing feed-forward baselines, achieving consistently lower latency across all view counts. Long-LRM encounters an out-of-memory issue on a 80GB GPU once the number of input views exceeds 192. Note that the reported inference time only accounts for the generation of 3D Gaussians. For novel view rendering, Long-LRM encounters a memory error when using more than 128 input views.

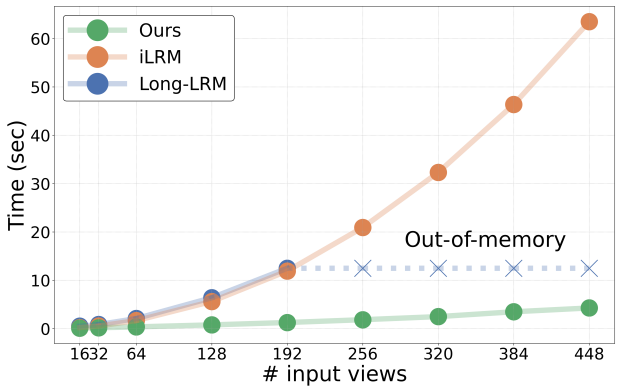


Figure 3. Inference time comparison.

FLOPs comparison. We report theoretical FLOPs comparisons or our model with global- and alternating-attention variants in Tab. 9, assuming 14 transformer layers and identical input tokenizers. Group size for our model is set to 4. V, L, and D denote the number of views, the number of tokens per input view, and the hidden dimension, respectively.

Method	Theoretical FLOPs	PFLOPs with (V, L, D) input		
		(32, 1920, 1024)	(32, 8160, 1024)	(128, 8160, 1024)
Global. Att.	$VLD(336D + 56VL)$	0.24	4.0	62.93
Alter. Att.	$VLD(336D + 28(1 + V)L)$	0.13	2.11	31.89
Ours	$VLD(21D + (3.3125 + \frac{V}{16})L + 1.5)$	0.01	0.1	0.81

Table 9. Theoretical FLOPs comparison.

Robustness to camera pose. We evaluate the sensitivity of our method to inaccurate camera poses on DL3DV benchmark dataset (32-view, 960×540) by adding random Gaussian noise with varying standard deviations to the rotation and translation components. As shown in Tab. 10, performance degrades noticeably as noise increases, particularly for translation perturbations. This indicates that our method relies on reasonably accurate camera poses, which remains a limitation to be addressed in future work.

Remarks on posed setting. We agree that known camera poses may limit applicability in some cases. Nevertheless, posed settings still often remain highly relevant

std.	0			0.001			0.005		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Rotation	25.96	0.847	0.187	24.65	0.839	0.190	23.34	0.746	0.237
Translation	-	-	-	24.75	0.804	0.204	20.85	0.645	0.297

Table 10. Robustness evaluations on camera pose.

in practice, including production-ready volumetric multi-view video systems, autonomous driving with calibrated cameras, and robotics applications where reasonably accurate poses can be estimated from inertial sensors. We also view posed multi-view modeling as an important step toward unposed settings, as evidenced by recent transformer-based approaches (e.g., GS-LRM [30], LVSM [10]) and their follow-ups (e.g., VGGT [22], Depth Anything 3 [15]), which adopt multi-view transformers as a core component. We hope our work aligns with this evolution and provides a solid foundation for future pose-free extensions.

Additional qualitative results. We provide further qualitative comparisons on the RE10K [31], DL3DV, Tanks&Temples [13], and Mip-NeRF360 [3] datasets in the remainder of this manuscript (Fig. 5, 6, 7, 8, 9 and 10).

Figure 4. Attention visualization. For colored query patches (red, yellow, green) in the reference view, we highlight top-3 attended tokens: on the left, tokens attended within the group (blue overlay), and on the right, tokens attended within and outside the group (green overlay).

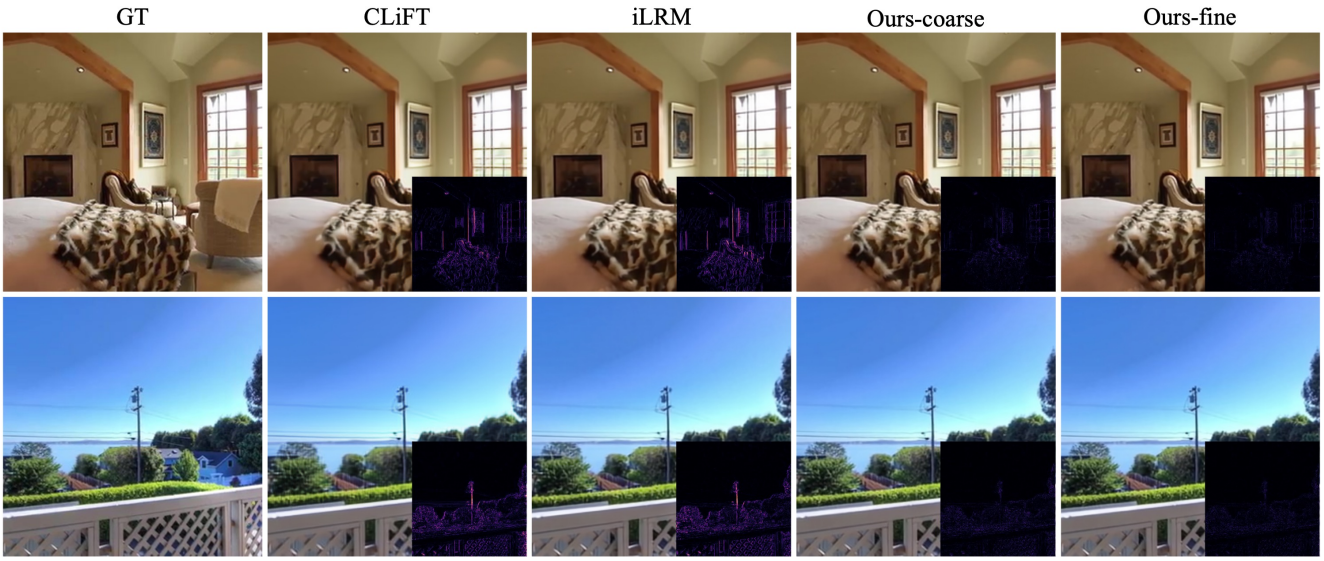


Figure 5. Qualitative results on the 4-view RE10K dataset. Per-pixel error maps are shown in the bottom-right corner of each image.

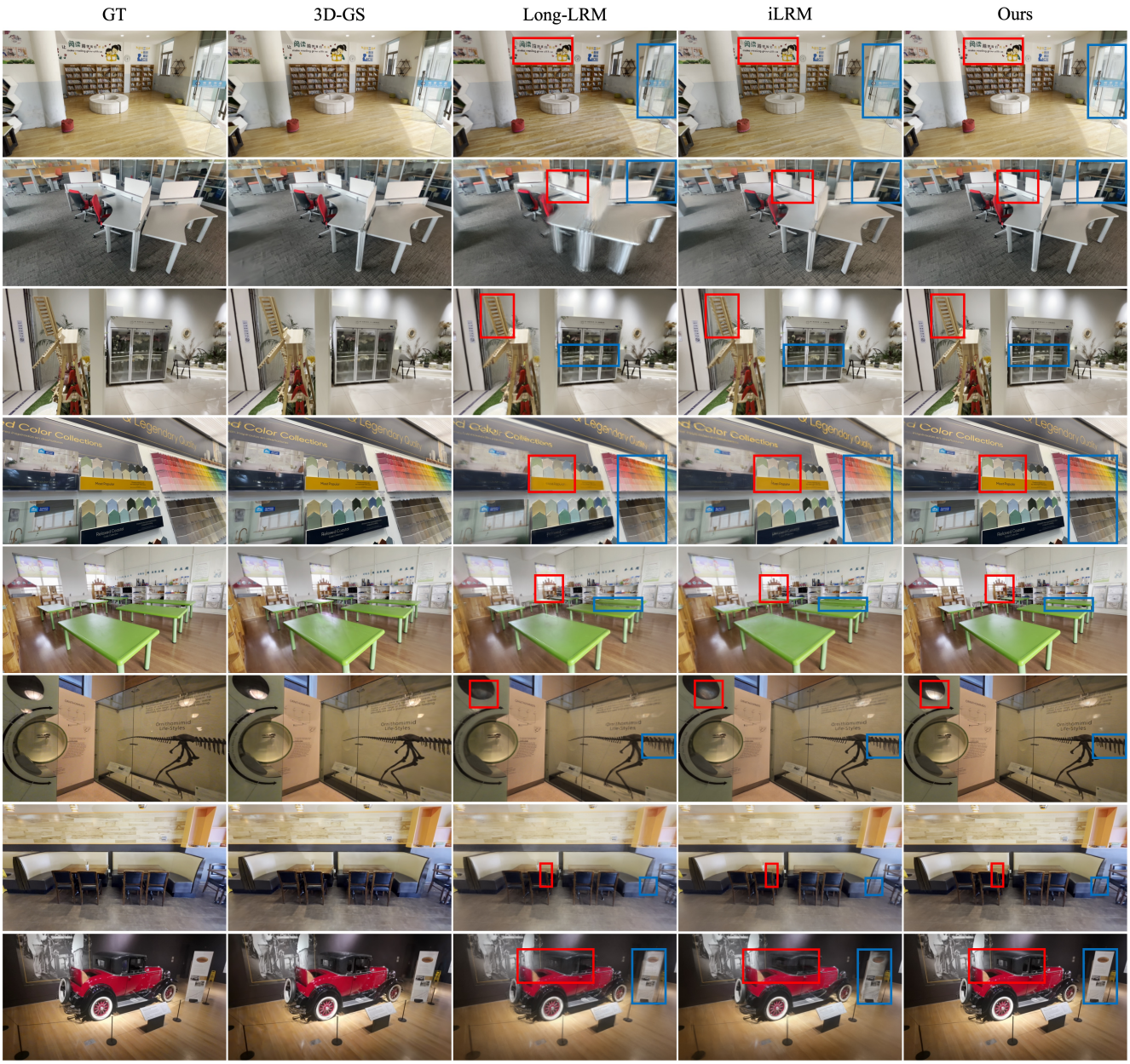


Figure 6. Qualitative results on the DL3DV-Benchmark across varying input view counts (128, 64, 32, and 16). The rows are arranged in descending order of view count, with two rows displayed for each setting.

Figure 7. Qualitative results on the DL3DV-Evaluation across varying input view counts (128, 64, 32, and 16). The rows are arranged in descending order of view count, with two rows displayed for each setting.

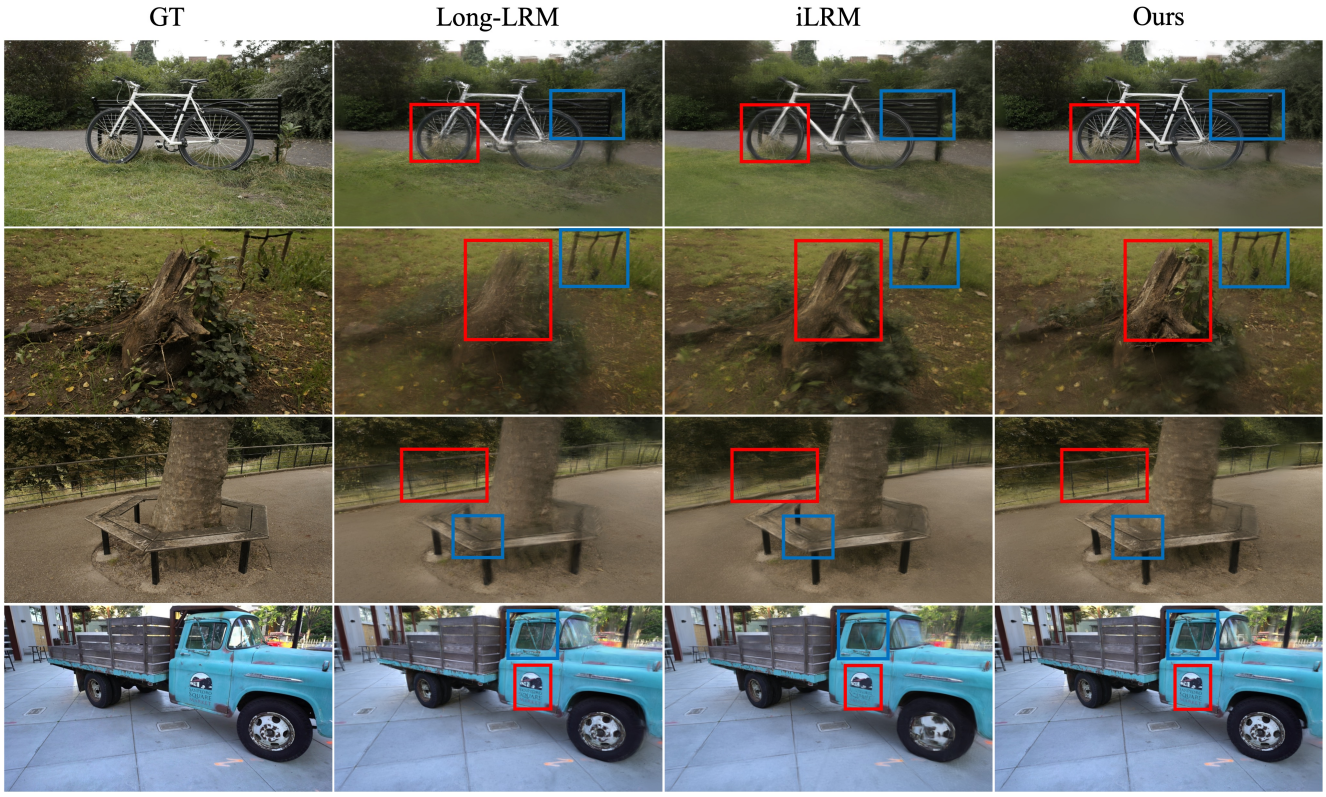


Figure 8. Qualitative results on the Mip-NeRF360 (top three rows), and `truck` scene from Tanks&Temples (bottom row). We visualize our rendering results with 32 input views, showing that our method demonstrates clear improvements on generalization and multi-view consistency under sparse inputs.

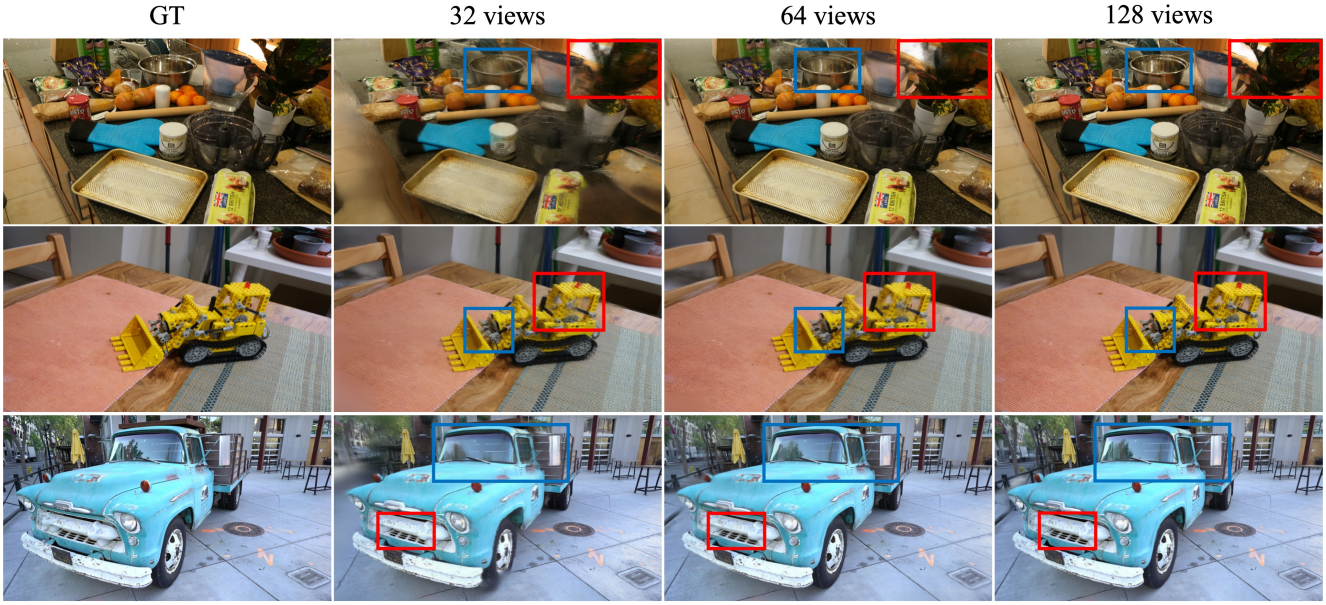


Figure 9. Qualitative results on the Mip-NeRF360 (top two rows), and `truck` scene from Tanks&Temples (bottom row). We visualize our rendering results as the number of input views increases, revealing progressively improved image quality and demonstrating that our method scales effectively with the number of views.

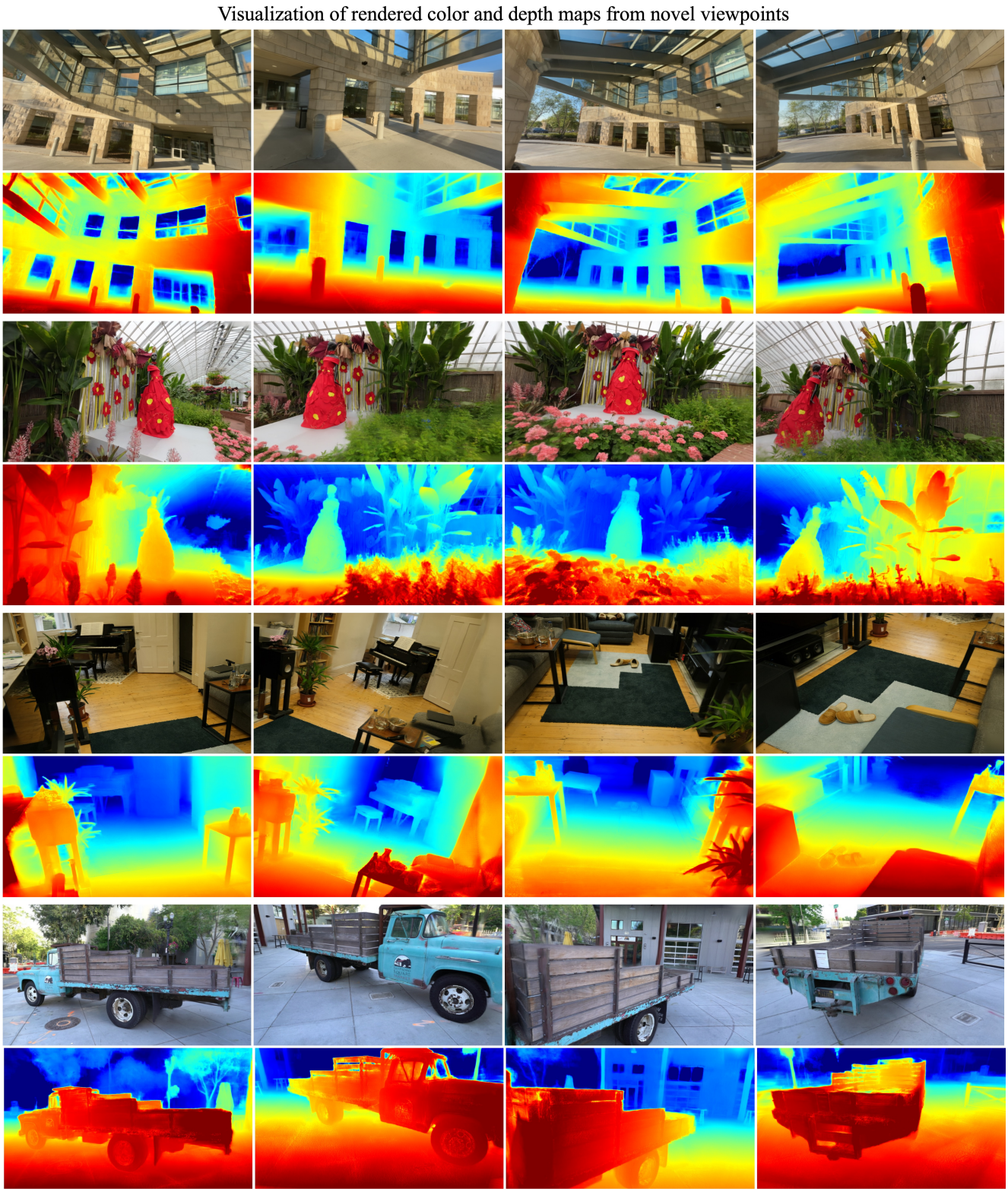


Figure 10. Qualitative visualization of rendered color and depth maps from novel viewpoints using 32 input images. Scenes from DL3DV (top four rows), Mip-NeRF360 (fifth and sixth row), and Tanks&Temples (bottom two rows) are shown.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

[4] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[5] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. *arXiv preprint arXiv:2307.08691*, 2023.

[6] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[7] David Haynes, Steven Corns, and Ganesh Kumar Venayagamoorthy. An exponential moving average algorithm. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[8] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[9] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.

[10] 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 *ICLR*, 2025.

[11] Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. ilrm: An iterative large 3d reconstruction model. *arXiv preprint arXiv:2507.23277*, 2025.

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

[13] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[14] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *Advances in Neural Information Processing Systems*, 2025.

[15] 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.

[16] 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.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[18] Nithin Gopalakrishnan Nair, Srinivas Kaza, Xuan Luo, Vishal M Patel, Stephen Lombardi, and Jungyeon Park. Scaling transformer-based novel view synthesis with models token disentanglement and synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28567–28576, 2025.

[19] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2538–2547, 2017.

[20] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[22] 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.

[23] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[24] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning, 2025.

[25] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[26] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

[28] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[29] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 2019.

[30] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2024.

[31] 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.

[32] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4349–4359, 2025.