OptSplat: Recurrent Optimization for Generalizable Reconstruction and Novel View Renderings

BMVC 2024 Submission # ??

Abstract

We propose an efficient feed-forward model for novel view synthesis and 3D reconstruction based on Gaussian Splatting, featuring a scalable architecture that reliably predicts multi-view depth maps and 3D Gaussian primitives from as few as two input views. Existing multi-view depth estimation techniques typically depend on processing plane-swept cost volumes, which generate probability distributions over a discrete set of candidate depths. This approach limits scalability, especially when finer depth sampling or higher spatial resolution is required. To address this, we design an optimization-inspired architecture *OptSplat*, that employs recurrent iterative updates to refine depth maps and pixel-aligned Gaussian primitives based on previous predictions. Our model leverages a unified update operator that iteratively indexes global cost volumes, progressively improving predictions in the joint space of depth and Gaussian parameters. Comprehensive evaluations across the real world datasets of *RealEstate10K*, *ACID* and *DL3DV* shows that our model demonstrates strong cross-dataset generalization and competitive rendering quality for novel views compared to the existing works with plane swept cost volumes, while at the same time offering upto 5x reduction in the GPU memory requirements, especially for reconstruction with high-resolution inputs.

1 Introduction

Novel View Synthesis (NVS) involves generating photorealistic images of a scene from novel, unseen viewpoints, given one or more input views with known camera poses [5, 14, 34, 35, 46]. As a core problem in computer vision and graphics, NVS underpins a wide range of applications, including free-viewpoint video, scene relighting, virtual teleportation, and immersive content creation. The central challenge lies in accurately modeling both the 3D scene geometry and complex, view-dependent appearance, particularly from sparse or unstructured observations.

Neural rendering methods—especially NeRF-based models [3, 23]—have recently advanced the quality of novel view synthesis significantly. However, these methods typically rely on per-scene optimization, making them unsuitable for real-time or interactive applications. In contrast, generalizable NVS models aim to synthesize novel views for previously unseen scenes in a zero-shot or few-shot setting, offering significant gains in scalability, speed, and deployability. This property makes them particularly suitable for robotics, AR/VR, autonomous navigation, and real-time scene capture or editing, where fast, one-shot inference is critical.

While recent transformer-based approaches such as LVSM [13] achieve high-quality synthesis without requiring explicit geometry, they often lack the geometric precision necessary for downstream tasks like scene editing or mixed-reality integration. In this work, we argue for the importance of explicit geometry-aware scene representations in generalizable models—not only for accurate rendering, but also to support tasks requiring rich geometric and semantic understanding. Building on the success of 3D Gaussian Splatting, which models the scene as a continuous volumetric representation composed of Gaussian primitives, we present a generalizable approach that leverages this representation to achieve both photorealism and geometric fidelity.

A central requirement for accurate 3D Gaussian reconstruction is reliable scene geometry, typically obtained via Multi-View Stereo (MVS). MVS estimates dense, 3D-consistent depth maps from posed input views, enabling accurate placement of Gaussian primitives. However, existing generalizable Gaussian Splatting methods—such as pixelSplat [5], MVSplat [8], and LS-GRM [42]—rely on full cost volume construction via plane sweep stereo [44]. These methods scale poorly with image resolution, number of viewpoints, and depth hypothesis density, often resulting in high memory usage and frequent out-of-memory (OOM) failures on resource-limited hardware.

To address these limitations, we propose **OptSplat**, a memory-efficient and scalable MVS architecture based on iterative refinement of local cost volumes. Instead of building a global cost volume, our network computes lightweight local volumes on-the-fly at each refinement step, significantly reducing memory consumption while maintaining high performance. We frame depth prediction as an optimization problem, where our model progressively refines depth estimates through a series of update blocks, leading to improved convergence and robustness across varied scenes.

In addition to geometry, capturing accurate view-dependent radiance is critical for photorealistic rendering, especially in sparse-view and zero-shot settings. Our method introduces an iterative spherical harmonics refinement module that progressively improves the radiance field over inference steps. This approach stabilizes early predictions from limited inputs and allows the model to adaptively refine appearance features, improving generalization to novel scenes.

Overall, our framework can be viewed as learning to optimize for generalizable reconstruction and novel view synthesis. Our network comprises a sequence of update operators that emulate a first-order optimization process—not by explicitly computing gradients, but by retrieving features from cost volumes to inform each update. Unlike recurrent methods such as R-MVSNet [45], which focus solely on multi-view depth estimation, our update mechanism jointly refines both multi-view depth and 3D Gaussian parameters, enabling cohesive and efficient optimization across geometry and appearance.

We summarize our key contributions as follows:

1. We propose a scalable architecture with iterative optimization layers based on GRU units, enabling sequential estimation and refinement of 3D Gaussian representations.

2. Our method achieves competitive performance compared to state-of-the-art approaches, demonstrating strong cross-dataset generalization, while reducing GPU memory consumption by approximately 5×—making it suitable for deployment on resource-constrained hardware without compromising rendering quality.

3. The design of our model inherently supports scalability with respect to the number of input views, image resolution, and depth candidates, enabling efficient large-scale generalizable reconstruction and novel view synthesis.

2 Related Work

Optimizing 3D Representations: Neural Radiance Fields (NeRF) [23] pioneered differentiable volumetric scene representations for novel view synthesis from posed multi-view images. Subsequent efforts have advanced rendering quality [3, 4], robustness to pose uncertainty [16, 26], and real-time rendering speed [12, 25]. Extensions incorporating explicit point-based primitives [41, 47] improved efficiency but still rely on costly volumetric ray marching. Recently, Gaussian Splatting [7, 14] offers a continuous, explicit, and differentiable representation that supports real-time rasterization-based rendering with high fidelity, presenting a compelling alternative for scalable view synthesis.

Sparse View Reconstruction: Early NeRF and Gaussian Splatting methods required dense multi-view inputs (often exceeding 100 views) for per-scene optimization. Recent approaches target sparse-view reconstruction and synthesis [5, 6, 8, 46], typically involving hand-crafted depth priors [10, 24] or diffusion-based generative regularization [37] to handle underconstrained regions. Despite improved quality, these methods rely on costly test-time optimization. In contrast, our work addresses zero-shot reconstruction and synthesis, enabling direct feed-forward prediction from sparse inputs without per-scene fine-tuning.

Feed-Forward 3D Gaussian Splatting: Feed-forward models leveraging 3D Gaussians [5, 8, 28, 36] demonstrate advantages in real-time rendering compared to implicit NeRF representations [21, 39]. Single-view methods such as Splatter Image [29] and Flash3D [28] regress pixel-aligned Gaussians using monocular depth priors but remain limited by inherent monocular ambiguities and dependence on learned spatial priors. Multi-view approaches like pixelSplat [5], MVSplat [8], and DepthSplat [40] improve reconstruction by estimating consistent depth and Gaussian parameters from multiple views. These methods rely on plane-sweep stereo to construct global cost volumes for depth inference, with DepthSplat leveraging pretrained monocular depth features [43] for enhanced accuracy. However, processing full global cost volumes hinders scalability with input resolution and view count due to high memory demands. Our approach circumvents this bottleneck by computing local cost volumes and performing iterative depth refinement, enabling efficient large-scale reconstruction without sacrificing accuracy.

Recurrent Optimization for Scene Reconstruction: Optimization-inspired architectures have improved generalization across vision tasks by mimicking iterative solvers [1, 2]. RAFT [30] introduced GRU-based recurrent refinement of 4D correlation volumes for optical flow, a concept extended to multi-view stereo depth estimation [18, 22]. DROID-SLAM [31] and DPVO [32] further applied recurrent optimization to joint depth and pose estimation, typically trained with ground-truth supervision minimizing reprojection error. Our method relates to RAFT-MVS [22] by recurrently updating depth predictions from cost volumes but differs critically: we do not rely on ground-truth depth or pose for training, instead optimizing a photometric reconstruction loss. Furthermore, our recurrent update operator jointly refines multi-view depth and Gaussian radiance parameters, enabling accurate, scalable estimation of geometry and appearance within a fully feed-forward pipeline. This design supports zero-shot generalization and efficient rendering from sparse multi-view inputs, addressing key challenges in practical novel view synthesis.

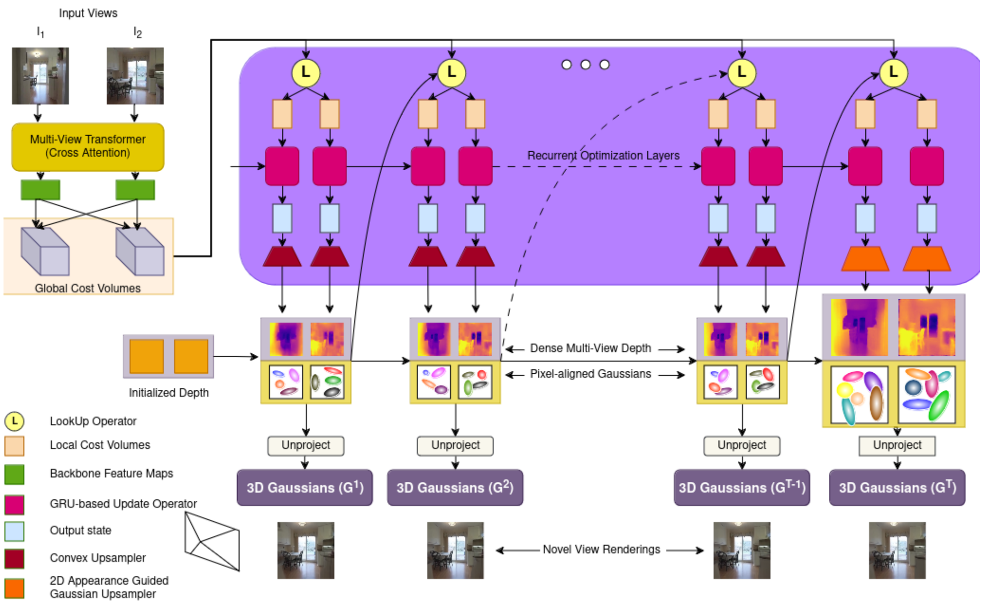

Figure 1: **Overview of the OptSplat architecture**. Given posed multi-view RGB inputs, OptSplat constructs local cost volumes via plane sweep stereo and iteratively refines 3D-consistent depth maps and 3D Gaussians using a GRU-based update operator. The entire pipeline operates in a fully feed-forward, zero-shot manner, enabling efficient and scalable novel view synthesis with high geometric and visual fidelity

3 Method

Given N sparse input images $\mathcal{I} = \{I_i\}_{i=1}^N$, $(I_i \in \mathbb{R}^{H\times W\times 3})$, their corresponding intrinsics $\mathcal{K} = \{K_i\}_{i=1}^N$ and known camera poses $\mathcal{E} = \{E_i\}_{i=1}^N$, $(E_i = [R_i|t_i]$, with R_i and t_i being the rotation matrices and translation vectors respectively), we aim to (1) reconstruct the scene using a representation $\mathcal{G}$ comprised of Gaussian primitives, and (2) synthesize novel views I_t given target camera intrinsics K_t and extrinsics E_t.

Scene Representation: The scene representation $\mathcal{G}$ consists of a set of 3D Gaussians [14] $\mathcal{G} = \{(\mu_i, \sigma_i, \Sigma_i, c_i)\}_{i=1}^M$, where $\mu_i \in \mathbb{R}^3$ is the mean (position) of the Gaussian in 3D space, $\sigma_i \in \mathbb{R}$ is the opacity, $\Sigma_i \in \mathbb{R}^{3\times 3}$ is the covariance matrix, $s_i \in \mathbb{R}^3$ is the coefficient vector for spherical harmonics [14], and M is the total number of Gaussians. Following prior works [5, 8], we assign a 3D Gaussian for every $p \times p$ patch in the input image. The task thus reduces to learning the parameters θ of a neural network f_θ, that maps the inputs $(\mathcal{I}, \mathcal{K}, \mathcal{E})$ to the scene representation $\mathcal{G}$:

$$f_\theta : \{I_i, K_i, E_i\}_{i=1}^N \longrightarrow \{(\mu_i, \sigma_i, \Sigma_i, s_i)\}_{i=1}^{\frac{H}{p}\times\frac{W}{p}\times N} \tag{1}$$

In the following sections, we give an overview of the key components of our model architecture (refer Fig 1).

(a) (b)

Figure 2: **Overview of the proposed modules in OptSplat**. (a) Iterative local cost volume extraction: At each refinement step, we dynamically index a global cost volume using the current depth estimate to construct localized cost volumes for recurrent depth refinement. (b) Appearance-Guided Upsampler: A UNet-style architecture upsamples the final hidden state o^T to the input image resolution, producing dense 3D Gaussian parameters guided by image appearance features.

3.1 Multi-View Feature Extraction and Cost Volumes

Given a set of K images $\{I_i\}_{i=1}^{K}, (I_i \in \mathbb{R}^{H\times W\times 3})$, we use a UniMatch backbone [38], similar to [8], to extract cross-view context-aware feature maps $\{F_i\}_{i=1}^{K}, (F_i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times d})$, where s is the downsampling factor and d the feature dimension. The backbone consists of a shallow ResNet [11] followed by a Swin-style transformer [20] with self- and cross-attention modules [33] to encode cross-view contextual information.

To enable geometry-aware scene reconstruction, we estimate dense depth maps for each input view via plane sweep stereo [9]. We uniformly sample D depth values between $d_{\min}$ and $d_{\max}$ in inverse depth space. Then, for each input view i, we construct a cost volume $C_i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times D}$ by measuring the similarity between the reference feature map F_i and the features of other views warped onto the depth planes of view i.

This cost volume serves as the geometric basis for depth prediction. For a comprehensive overview of feature warping, projection, and view synthesis, we direct the readers to the supplementary material.

3.2 Iterative Local Cost Volume Extractor

Computing and processing full cost volumes at high resolution is computationally expensive and memory-intensive, especially when scaling to high-resolution images, large depth sampling rates, or many input views. Prior works either restrict input resolution to 256×256 [5, 8], or downsample the cost volumes significantly (e.g., by a factor of 8) [40], limiting reconstruction fidelity.

We propose a memory-efficient, iterative depth refinement strategy based on local cost volume indexing. Rather than estimating depths in a single forward pass from the entire cost volume, we model the problem as an iterative search over a 1D depth space. At each step t in T total iterations, the network predicts depth values d^t per pixel by locally querying the global cost volume C_i around the previous estimate d^{t-1} (refer Fig 2a).

We extract a local cost vector $LC_i^{t-1}(u_{p.q}) \in \mathbb{R}^{2R+1}$ for each pixel by performing a differentiable lookup over the global cost volume in a small window of radius R, centered at the normalized previous estimate:

$$LC_i^{t-1}(u_{p.q}) = \texttt{LookUp}\left(C_i(u_{p,q}), \mathcal{N}(\tilde{d}_{p,q}^{t-1})\right) \quad (2)$$

This local feature vector captures fine-grained matching scores near the current estimate and enables the network to infer more accurate depth updates over time. Full lookup logic and normalization details are provided in the supplementary material.

3.3 Recurrent Updates for Depth and Gaussian Predictions

We propose a unified, recurrent framework that jointly estimates multi-view depths and pixel-aligned 3D Gaussian primitives. Unlike prior MVS approaches that refine depth alone [18, 44, 45], our model performs iterative updates over both geometry and appearance parameters, enabling efficient and accurate scene representation.

At each iteration t, the model refines a depth map d^t and predicts a set of 3D Gaussians $\mathcal{G}^t$. The update module receives the previous depth estimate d^{t-1}, a local cost volume LC^{t-1} (indexed from the global cost volume as described in Sec. 3.2), and context features F_c derived from backbone features via two convolutional layers. These are concatenated and passed through a GRU-based update operator implemented using convolutional gates:

$$z^t = \sigma\left(\text{Conv}_{3\times3}([h^{t-1}, x^t])\right), \quad r^t = \sigma\left(\text{Conv}_{3\times3}([h^{t-1}, x^t])\right) \tag{3}$$

$$\tilde{h}^t = \tanh\left(\text{Conv}_{3\times3}([r^t \odot h^{t-1}, x^t])\right), \quad h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \tilde{h}^t \tag{4}$$

The hidden state h^t is used to generate an output state o^t and an upsampling mask m^t:

$$[o^t, m^t] = \texttt{OutputHead}(h^t), \quad o^t_s = \texttt{ConvexUpsampling}(o^t, m^t) \tag{5}$$

This upsampled output o^t_s is then decoded into three key prediction heads:

$$\Delta\tilde{d}^t = \texttt{DisparityHead}(o^t_s), \quad \sigma^t = \texttt{DensityHead}(o^t_s), \quad [\Sigma^t, s^t] = \texttt{GaussianHead}(o^t_s) \tag{6}$$

These heads predict residual disparity updates $\Delta\tilde{d}^t$, Gaussian opacity σ^t, and both covariance Σ^t and SH color coefficients s^t. The predicted disparity is then refined as $\tilde{d}^t = \tilde{d}^{t-1} + \Delta\tilde{d}^t$.

To enforce a consistent optimization trajectory, the update module is weight-tied across all iterations, analogous to a learned first-order optimizer.

3.4 2D Appearance-Guided Upsampler for 3D Gaussians

While the update operator predicts depths and pixel-aligned Gaussian parameters at the input image resolution, the GRU itself operates at a lower spatial scale defined by the cost volume resolution (downsampling factor s). The intermediate upsampling via convex masks is context-aware but limited in resolving fine object boundaries, often leading to depth blurring and inaccurate Gaussian placements when unprojected to 3D.

To address this, we introduce a final refinement module based on a UNet-style architecture [38], designed to progressively upsample the final output state o^T using multi-scale image features (refer Fig 2b). The encoder processes the input image $\mathcal{I}$ into feature maps at varying resolutions (down to s), while the decoder upsamples o^T to the full resolution. Skip connections inject appearance cues from the encoder into the decoder to guide high-fidelity upsampling.

Method	Time (s)	GPU Memory (MB)	RealEstate10K			ACID		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
pixelSplat	-	-	25.89	0.858	0.142	28.14	0.839	0.150
latentSplat	-	-	23.93	0.812	0.164	-	-	-
GS-LRM	-	-	**28.10**	**0.892**	**0.114**	-	-	-
MVSplat	**0.061**	1217	26.39	0.869	0.128	28.25	0.843	0.144
DepthSplat	0.089	2638	27.47	0.889	**0.114**	-	-	-
OptSplat (Ours)	0.091	**658**	25.74	0.866	<u>0.124</u>	28.17	**0.849**	**0.136**

Table 1: In-domain novel view synthesis: Comaprison on RealEstate10K and ACID datasets

The refined output state o_f is passed through parallel prediction heads to obtain the final estimates for disparity, density, covariance, and color:

$$o_f = \texttt{UNet}(\mathcal{I}, o^T) \tag{7}$$

$$[d_f, \sigma_f, \Sigma_f, s_f] = \texttt{PredictionHeads}(o_f) \tag{8}$$

These final predictions define the full-resolution multi-view depths and 3D Gaussian parameters, which are then rendered from novel views using a tile-based rasterizer [14].

4 Experiments

4.1 Settings

Datasets: We use large-scale scene level datasets of RealEstate10K [48], ACID [19] and DL3DV-10K [17] to train a generalizable view synthesis model. RealEstate10K primarily comprises of real estate scenes with indoor layouts downloaded from YouTube, which are split into 67,477 training scenes and 7,289 testing scenes, while ACID contains nature scenes captured by aerial drones, which are split into 11,075 training scenes and 1,972 testing scenes. DL3DV-10K is a comprehensive large-scale dataset of real-world scenes captured from different points of interest like restaurants, shopping malls, tourist spots, etc, and with diverse transparency, lighting and reflectance conditions. It consists of 51.2 million frames in 4K resolution from 10,510 videos, and following DepthSplat [40], we split the dataset into 9076 training scenes and 95 test scenes.
Comparison to Baselines: To demonstrate the effectiveness of our method, we consider several recent methods tackling the problem of generliazable novel view synthesis. We only consider works that has 3D reconstruction as an intermediate step to novel view synthesis [5, 8, 36, 40, 42], especially the ones with an explicit differentiable scene representation parametrized by 3D Gaussians. We do not provide any comparison with models that treats novel view synthesis as an end-to-end view prediction problem from input views without the need for any geometry-aware scene representation [13, 15, 27], as this deviates from the spirit of our proposed approach.

Please refer to the supplementary material for the training and implementation details.

4.2 Main Results

In this section, we demonstrate the effectiveness of our trained models in terms of the quality of novel view renderings. We use Peak-Signal-To-Noise-Ratio (PSNR, Structural-Similarity-Index-Measure (SSIM) and Learned-Perceptual-Image-Patch-Similarity (LPIPS) measures to compare the quality of rendered images with the ground truth.

Method	GPU Memory (MB)	DL3DV			ACID		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MVSplat	1217	25.55	0.833	0.119	28.15	0.841	0.147
DepthSplat	2638	**27.99**	**0.897**	**0.084**	**28.37**	**0.847**	**0.141**
OptSplat (Ours)	**658**	<u>26.69</u>	<u>0.875</u>	<u>0.093</u>	27.39	0.836	<u>0.144</u>

Table 2: Cross-domain novel view synthesis: Zero-shot generalization on DL3DV and ACID datasets for models trained on RealEstate10K

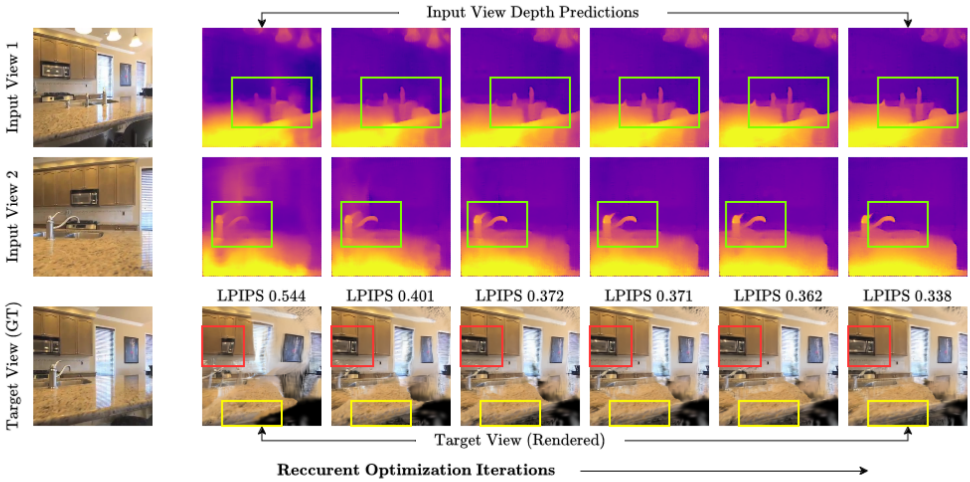

Figure 3: Input view depth predictions and novel view renderings from OptSplat over the iterations of recurrent optimization.

In-domain novel view synthesis: In Table 1, we compare the rendering quality and memory efficiency of our model against MVSplat and DepthSplat on the RealEstate10K and ACID datasets. Our approach achieves competitive rendering performance while requiring significantly less memory—approximately 50% and 25% of the memory used by MVSplat and DepthSplat, respectively. Specifically, our model operates within a memory footprint of under 700,MB for 256×256 resolution inputs, compared to over 2600,MB required by DepthSplat, which offers only marginal improvements in rendering fidelity.

Cross-domain novel view synthesis: Table 2 presents a cross-dataset generalization study, where models trained on RealEstate10K are evaluated on the ACID and DL3DV datasets. Despite the DL3DV scenes featuring more complex geometry and larger viewpoint variations, our model generalizes robustly across datasets, maintaining strong rendering quality while consuming only a fraction of the memory used by DepthSplat. For visual comparison of depth predictions and novel-view renderings of our model with MVSplat and DepthSplat, we refer the readers to the supplementary material.

4.3 Ablation Study and Analysis

Impact of number of refinement iterations: Fron Table 3, we could see that our Table 3 highlights the convergence behavior of our model as the number of recurrent refinement iterations increases. In this analysis, we only consider outputs refined using convex upsampling.

The results validate our design objective: the update operator effectively learns to perform optimization-like refinement in a feed-forward manner, progressively improving scene geometry and appearance reconstruction. Furthermore, as shown in Figure 3, both the predicted depth maps and the synthesized novel views exhibit consistent refinement across iterations, capturing increasingly fine-grained details and producing sharper reconstructions over time.

# Recurrent Updates	**RealEstate10K**			**DL3DV**			**Time**	**GPU Memory**
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	(s)	(MB)
1	22.793	0.771	0.178	21.754	0.690	0.2	0.061	334
2	23.361	0.791	0.166	23.019	0.751	0.17	0.064	338
3	23.458	0.794	0.164	23.125	0.755	0.168	0.071	338
4	23.493	0.795	0.163	23.154	0.756	0.168	0.078	338
5	23.509	0.796	0.163	23.163	0.756	0.168	0.080	338

Table 3: Model evaluation with different number of iterations of recurrent updates

Resolution of the Cost Volumes: We evaluate OptSplat with recurrent updates operating at downscale factors $s = 4$ and $s = 8$. As shown in Table 4, reducing the resolution of the local cost volume has minimal impact on rendering quality. Thanks to our update operator and appearance-guided upsampler, the model achieves up to 50% memory and 20% runtime savings with negligible performance drop.

Cost Volume Resolution	**RealEstate10K**			**DL3DV**			**Time**	**GPU Memory**
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	(s)	(MB)
4	25.76	0.867	0.124	26.75	0.876	0.092	0.156	1279
8	25.69	0.866	0.123	26.34	0.868	0.095	0.127	658

Table 4: Comparison with different cost volume resolutions

Impact of Appearance Guided Upsampling: As shown in Table 5, our appearance-guided upsampling module leads to consistent improvements across all evaluation metrics. This demonstrates its effectiveness in leveraging 2D image cues to refine and upsample the 3D Gaussian predictions. While this enhancement introduces a modest increase in memory consumption, the gains in reconstruction quality justify the trade-off.

Model	**RealEstate10K**			**DL3DV**			**Time**	**GPU Memory**
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	(s)	(MB)
w/o Appearance-Guided Gaussian Upsampler	23.51	0.796	0.163	23.16	0.756	0.168	0.080	338
Full Model	25.69	0.866	0.123	26.34	0.868	0.095	0.127	658

Table 5: Effect of Appearance-Guided Gaussian Upsampling

5 Conclusion

In this work, we have demonstrated that a generalizable model equipped with recurrent update blocks offers a scalable architecture suitable for deployment in resource-constrained environments, with minimal compromise in reconstruction and rendering quality. Furthermore, our framework opens promising directions for future research, particularly in integrating mechanisms inspired by scene-specific optimization techniques into the iterative refinement stage—potentially enhancing the model's generalization capabilities even further.

References

[1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

[3] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

[5] 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.

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo, 2021. URL https://arxiv.org/abs/2103.15595.

[7] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.

[8] 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, 2024. URL https://arxiv.org/abs/2403.14627.

[9] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE computer society conference on computer vision and pattern recognition*, pages 358–363. Ieee, 1996.

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12882–12891, 2022.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5875–5884, 2021.

[13] 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.

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[15] Marc Levoy and Pat Hanrahan. Light field rendering. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. URL https://api.semanticscholar.org/CorpusID:1363510.

[16] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021.

[17] 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.

[18] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021.

[19] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[21] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views, 2022. URL https://arxiv.org/abs/2206.05737.

[22] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview stereo with cascaded epipolar raft. In *European Conference on Computer Vision*, pages 734–750. Springer, 2022.

[23] 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, 2020. URL https://arxiv.org/abs/2003.08934.

[24] 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.

[25] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021.

[26] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. IEEE, 2023.

[27] 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.

[28] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arxiv*, 2024.

[29] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[30] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[31] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[32] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[34] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering, 2021. URL https://arxiv.org/abs/2102.13090.

[35] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[36] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European Conference on Computer Vision*, pages 456–473. Springer, 2024.

[37] 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.

[38] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023.

[39] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024.

[40] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024.

[41] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022.

[42] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation, 2024. URL https://arxiv.org/abs/2403.14621.

[43] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[44] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[45] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5525–5534, 2019.

[46] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. URL https://arxiv.org/abs/2012.02190.

[47] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–12, 2022.

[48] 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.