# G3R: Gradient Guided Generalizable Reconstruction

Yun Chen[1,2⋆] Jingkang Wang[1,2⋆]
Ze Yang[1,2] Sivabalan Manivasagam[1,2] Raquel Urtasun[1,2]

Waabi[1] University of Toronto[2]
{ychen, jwang, zyang, smanivasagam, urtasun}@waabi.ai

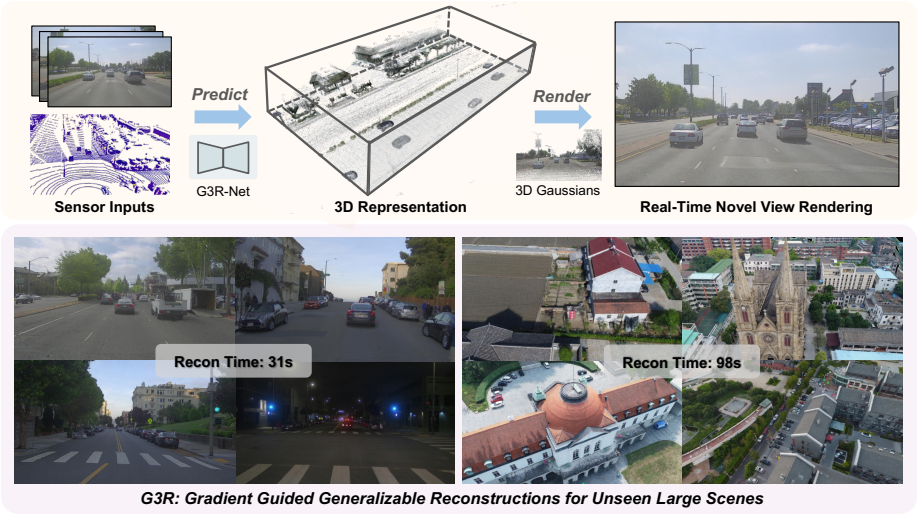


**Fig. 1: Gradient Guided Generalizable Reconstruction (G3R)**: Our method learns a single reconstruction network that takes multi-view camera images and an initial point set to predict the 3D representation for large scenes ($> 10,000m^2$) in two minutes or less, enabling realistic and real-time camera simulation.

**Abstract.** Large scale 3D scene reconstruction is important for applications such as virtual reality and simulation. Existing neural rendering approaches (*e.g.*, NeRF, 3DGS) have achieved realistic reconstructions on large scenes, but optimize per scene, which is expensive and slow, and exhibit noticeable artifacts under large view changes due to overfitting. Generalizable approaches, or large reconstruction models, are fast, but primarily work for small scenes/objects and often produce lower quality rendering results. In this work, we introduce G3R, a generalizable reconstruction approach that can efficiently predict high-quality 3D scene representations for large scenes. We propose to learn a reconstruction network that takes the gradient feedback signals from differentiable rendering to iteratively update a 3D scene representation, combining the benefits of high photorealism from per-scene optimization with data-driven priors from fast feed-forward prediction methods. Experiments

⋆ Equal contributions.

on urban-driving and drone datasets show that G3R generalizes across diverse large scenes and accelerates the reconstruction process by at least 10× while achieving comparable or better realism compared to 3DGS, and also being more robust to large view changes. Please visit our project page for more results: `https://waabi.ai/g3r`.

**Keywords:** Generalizable Reconstruction · Neural Rendering · Learned Optimization · 3DGS · Large Reconstruction Models

## 1 Introduction

Reconstruction of large real world scenes from sensor data, such as urban traffic scenarios, is a long-standing problem in computer vision and computer graphics. Scene reconstruction enables applications such as virtual reality and high-fidelity camera simulation, where robots such as autonomous vehicles can learn and be evaluated safely at scale [32,37,52,68,78]. To be effective, the 3D reconstructions must have high photorealism at novel views, be efficient to generate, enable scene manipulation, and enable real-time image rendering.

Recently, neural rendering approaches such as NeRF [38] and 3D Gaussian Splatting (3DGS) [18] have achieved realistic reconstructions for large scenes using camera and optionally LiDAR data. However, they require a costly per-scene optimization process to reconstruct the scene by recreating the input sensor data via differentiable rendering, which may take several hours to achieve high-quality. Moreover, they typically focus on the novel view synthesis (NVS) setting where the target view is close to the source views and often exhibit artifacts when the viewpoint changes are large (*e.g.*, meter-scale shifts), as it can overfit to the input images while not learning the true underlying 3D representation.

To enable faster reconstruction and better performance at novel views, recent works aim to synthesize a *generalizable* representation with a single pre-trained network, which can be used for NVS on unseen scenes in a zero-shot manner. These methods utilize an encoder to predict the intermediate scene representation by aggregating image features extracted from multiple source views according to camera and geometry priors, and then decode the representation for NVS via volume rendering or a transformer [7,26,69,70,87]. The encoder and decoder networks are trained across many scenes to reconstruction priors. Most recently, large reconstruction models (LRMs) are proposed to learn reconstruction priors by training on large-scale synthetic datasets for generalizable single-step 2D to 3D reconstruction [14,22,30,72,88]. However, both generalizable NVS and LRMs are primarily applied to objects or small scenes due to the complexity of large scenes, which are difficult to predict accurately from a single step network prediction. Furthermore, the computation resources and memory needed to utilize many input scene images ($> 100$) with existing techniques that aggregate ray features [69], build cost volumes [7] or perform image-based rendering [70] are prohibitive.

In this paper, we present Gradient Guided Generalizable Reconstruction (G3R), the first method that enables fast and generalizable reconstruction of

large scenes. Given a sequence of images and an approximate geometry scaffold (*e.g.*, points from LiDAR or multi-view stereo), G3R can produce a modifiable digital twin as a set of 3D Gaussian primitives in two minutes or less for large scenes ($> 10,000m^2$). This representation can be directly used for high-fidelity novel-view rendering at interactive frame rates ($> 90$ FPS). Our key idea is to learn a single reconstruction network that iteratively updates the 3D scene representation, combining the benefits of data-driven priors from fast prediction methods with the iterative gradient feedback signal from per-scene optimization methods. G3R can be viewed as a "learned optimizer" [4,73] for scene reconstruction. Towards this goal, we first initialize a neural scene representation which we call *3D Neural Gaussians* from the geometry scaffold that can be differentiably rendered. Rather than select a few close-by source views for unprojection like existing generalizable works, we propose a novel way of lifting 2D images to 3D space by rendering and backpropagating to obtain gradients w.r.t the current 3D representation. These 3D gradients can be seen as 2D images unprojected to 3D with the current representation as the 3D proxy, which takes the rendering procedure into account, and is thus naturally occlusion aware and contains a useful feedback signal. Moreover, it provides a unified representation that can efficiently aggregate as many 2D images as needed by just aggregating the gradients. Then, our reconstruction network (G3R-Net) takes the 3D gradients and current 3D representation as inputs and iteratively predicts updates to refine the representation. Since the G3R-Net incorporates the rendering feedback signal at each step and is trained across multiple scenes, it can significantly accelerate the convergence compared to standard gradient descent algorithms (*i.e*, 24 iterations v.s. 1000s of iterations). G3R-Net is trained across multiple scenes, enabling high quality reconstruction and improving robustness for NVS.

Experiments on two outdoor datasets with large-scale scenes demonstrate the generalizability of G3R. With as little as 24 iterations, G3R reconstructs large scenes with comparable or better realism at novel views than the per-scene optimization approaches while being at least $10\times$ faster. To the best of our knowledge, this is the first generalizable reconstruction approach that can reconstruct a faithful 3D representation for such large-scale scenes ($> 10,000m^2$) in high-resolution ($> 100$ source images at $1080 \times 1920$), showing the potential to build digital twins for the metaverse and simulation at large scale.

## 2 Related Work

*Optimization-based scene reconstruction:* The current state-of-the-art in scene reconstruction is optimizing differentiable radiance fields, such as NeRF [38] or 3DGS [18], which model the 3D scene either as neural networks or as Gaussian primitives, and then alpha-composite along the ray via either ray-marching or rasterization, respectively. To extend to city-scale scenes, some works decompose the scene into sub-components and represent each with a network to increase model capacity [27, 62, 66, 90]. To enable realistic and controllable sensor simulation, another line of work decomposes dynamic scenes (*e.g.*, urban driving

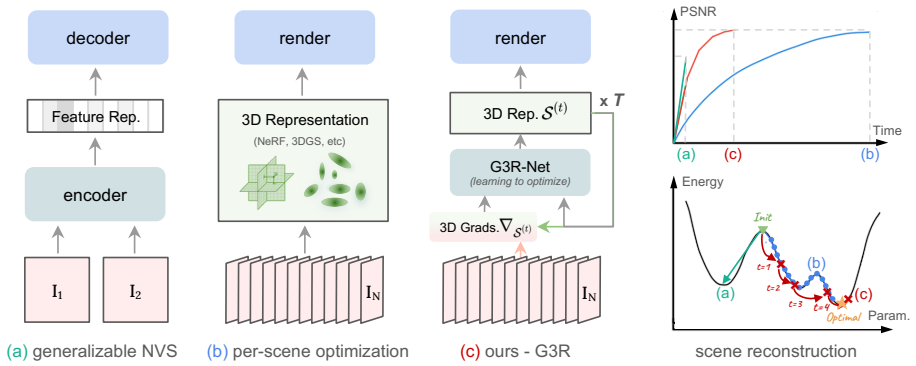


**Fig. 2: Three paradigms for scene reconstruction and novel view synthesis (NVS).** (a) Existing generalizable approaches select a few reference images (usually $\leq 5$) for feed-forward prediction of intermediate representation and then decode/render the feature representation to produce the rendered images. (b) Per-scene optimization approaches take all source images (*e.g.*, $> 100$ for large scenes) and reconstructs a 3D representation via energy minimization and differentiable rendering. (c) G3R conducts iterative prediction to refine the 3D representation with the 3D gradient guidance (*i.e.*, learned optimization) taking all source images. Compared to the other two paradigms, G3R leverages the benefits of both worlds (data-driven priors, gradient feedback) and achieves the best trade-off between the reconstruction quality and time (rightmost).

scenes) into static background and moving objects [16,29,42,64,76,79,81–84,92] or conduct inverse rendering for geometry, material, lighting and semantics decomposition [28,43,67,71]. These works require time-consuming (hours or days) per-scene optimization for large scenes and often exhibit artifacts at large view changes due to overfitting. In contrast, G3R predicts a high-quality and robust 3D representation for large scenes in a few minutes or less.

*Generalizable reconstruction:* To generalize to novel scenes, researchers train neural networks across diverse scenes and incorporate proxy geometry like depth maps for image-based rendering [3,20,46,47,74]. However, it is usually challenging or expensive to obtain high-quality geometry for real-world large scenes. To address this issue, recent works adopt transformers to either directly map the source images and camera embedding to the target view without any physical constraints [21,49–51,53] or aggregate points from source images along the epipolar lines for rendering [9,11,41,44,54,59,60,65,69,70,80]. Another popular approach is to lift 2D images to 3D cost volumes with geometry priors [7,9,17,26,31] but struggles with large camera movement. These methods do not produce a unified 3D representation, suffer from noticeable artifacts under large view changes, and are slow to render. On the other hand, some works that directly predict 3D representations such as multi-plane images (MPI) [12,58,91] or implicit representations [7,40,41,55,85] only work well on objects or small scenes. Concurrent work [6] predicts 3D Gaussians for generalizable reconstruction, but is limited to low-resoluation image pairs. In contrast, G3R take all available source images and predicts a unified representation for large-scale scenes including dynamics,

enabling scalable and realistic simulation. Most recently, large reconstruction models [14, 22, 30, 72, 88] (LRMs) achieve strong generalizability across small objects by training on large synthetic dataset such as Objaverse. To our best knowledge, G3R is the first LRM that generalizes across diverse large scenes and handles large view changes by training on large-scale real-world datasets.

*Iterative networks for 3D:* Our method falls under the "iterative network" framework, which conduct iterative updates to gradually refine the output. Prior works have studied iterative approaches on low-dimensional inverse problems [2, 5, 24, 35, 36] such 6-DOF pose and illumination estimation. In contrast, G3R solves a challenging high-dimensional inverse problem (*i.e.*, scene reconstruction) using a learned optimizer [4, 23, 73]. Specifically, we train a neural network that exploits spatial correlation to expedite the reconstruction process. Similar to G3R, DeepView [12] also employs an iterative network with gradient guidance to reconstruct a 3D representation (MPI), but for small baselines only. Moreover, it unfolds the optimization through a series of distinct CNN networks and loss-agnostic gradient components at each stage for each source image, limiting the number of input images, and leading to large memory usage and slow speed.

## 3 Gradient Guided Generalizable Reconstruction (G3R)

Given a set of source camera images $\mathbf{I}^{\text{src}} = \{\mathbf{I}_i\}_{1 \leq i \leq N}$ and an approximate geometry scaffold $\mathcal{M}$ (*e.g.*, obtained from either LiDAR or points from multi-view stereo) captured in-the-wild by a sensor platform moving through a large dynamic scene, our goal is to efficiently reconstruct a realistic and editable 3D representation $\mathcal{S}$ for accurate real-time camera simulation. In this paper, we introduce Gradient Guided Generalizable Reconstruction (G3R), the first method that can create modifiable digital clones of large real world scenes ($> 10,000m^2$) in two minutes or less, and that renders novel views with high photorealism at $>$90 FPS. Our method overview is shown in Fig. 3. G3R combines data-driven priors from fast prediction methods with the iterative gradient feedback signal from per-scene optimization methods by learning to optimize for large scene reconstruction (Fig. 2-left). G3R iteratively updates a representation we call 3D neural Gaussians, initialized from the scaffold $\mathcal{M}$, with a single neural network. The network takes the gradient feedback signals from differentiably rendering the representation to reconstruct the source images $\mathbf{I}^{\text{src}}$. G3R achieves the best trade-off between realism and reconstruction speed, achieving performance and scalability (see Fig. 2-right).

In what follows, we first introduce our scene representation (3D neural Gaussians) designed for handling dynamic and unbounded large scenes (Sec. 3.1). Then we show how to lift 2D images to 3D space by propagating the gradients (Sec. 3.2), followed by iterative refinements in Sec. 3.3. We describe training the network across multiple scenes in Sec. 3.4.

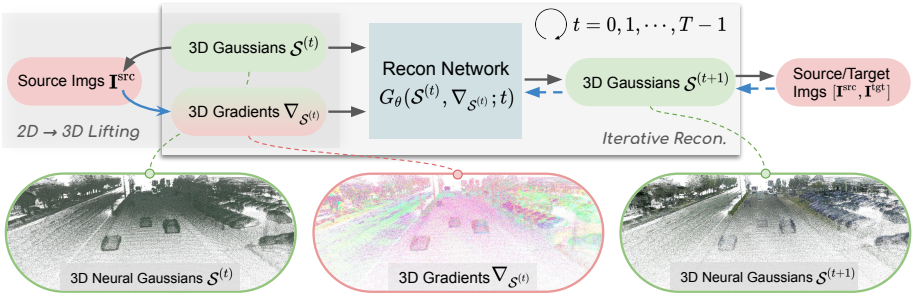


**Fig. 3: Method overview.** We model the generalizable reconstruction as an iterative process, where the 3D neural Gaussians $\mathcal{S}^{(t)}$ are iteratively refined with reconstruction network $G_\theta$. We first lift the source 2D images $\mathbf{I}^{\text{src}}$ to 3D space by backpropogating the rendering procedure to get the gradients w.r.t the representation $\nabla_{\mathcal{S}^{(t)}}$ (blue arrow). Then the reconstruction network $G_\theta$ takes the 3D representation $\mathcal{S}^{(t)}$, the gradient $\nabla_{\mathcal{S}^{(t)}}$ and the iteration step $t$ as input, and predicts an updated 3D representation $\mathcal{S}^{(t+1)}$. To train the network, we render $\mathcal{S}^{(t+1)}$ at source and novel views, and compute loss. The backward gradient flow for training $G_\theta$ is highlighted with dashed blue arrows.

### 3.1 G3R's Scene Representation

3D Gaussian Splatting [18] (3DGS) is a differentiable rasterization technique that allows real-time rendering of photorealistic scenes learned from posed images and an intitial set of points from SfM [56]. 3DGS represents the scene with a set of 3D Gaussians (*i.e.*, points) $\mathcal{G} = \{g_i\}_{1 \leq i \leq M}$, where $g_i \in \mathbb{R}^{14}$ consists of position ($\mathbb{R}^3$), scale ($\mathbb{R}^3$), orientation ($\mathbb{R}^4$), color ($\mathbb{R}^3$) and opacity ($\mathbb{R}^1$). These gaussian points $\mathcal{G}$ can be rendered to 2D images with camera poses $\Pi$ using a differentiable tile rasterizer $f_{\text{rast}}(\mathcal{G}, \Pi)$, where each point is projected and splatted to the image plane based on the scale and orientation, then the color is blended with other points based on the opacity and depth to camera. However, 3DGS's explicit representation lacks modelling capacity useful for learning-based optimization. Furthermore, 3DGS [18] focuses on small static scenes or individual objects, and has challenges modeling large-scale dynamic scenes, such as self-driving scenarios. In this paper, we make two enhancements to 3DGS's representation. First, we augment its representation with a latent feature vector, which we call *3D neural Gaussians*, providing additional capacity for generalizable reconstruction and learning-based optimization. Second, we decompose the scene into the nearby static scene, dynamic actors, and a distant region to enable modelling of large unbounded dynamic scenes. We now describe these two enhancements and then detail the rendering process.

*3D Neural Gaussians:* We define our scene representation $\mathcal{S}$ as a set of *3D Neural Gaussians,* $\mathcal{S} = \{h_i\}_{1 \leq i \leq M}$, where each point is represented by a feature vector $h_i \in \mathbb{R}^C$. This latent representation helps encode information about the scene during the iterative updates in the learning-based optimization described in Sec. 3.3. To render, we convert the 3D neural Gaussians to a set of explicit color 3D Gaussians $\mathcal{G} = \{g_i\}_{1 \leq i \leq M}$, using a Multi-Layer Perceptron (MLP)

network $g_i = f_{\text{mlp}}(h_i)$. To encode geometry and additional physical information about the scene into $h_i$ and ensure stable optimization, we designate the first 14 channels as the 3D Gaussian attributes and add a skip connection in $f_{\text{mlp}}$ such that it updates these channels to generate $g_i$.

*Representing rigid dynamic objects and unbounded scenes:* We decompose the dynamic scene and its set of 3D neural Gaussians $\mathcal{S}$ into a static background $\mathcal{S}^{\mathcal{B}}$, a set of dynamic actors $\mathcal{S}^{\mathcal{A}}$ and a distant region $\mathcal{S}^{\mathcal{Y}}$ (*e.g.*, far-away buildings and sky). We assume rigid motion $\mathcal{T}(\mathcal{S}^{\mathcal{A}}, \boldsymbol{\xi}^{\mathcal{A}})$ for dynamic actors, where $\mathcal{T}$ is the rigid transformation and $\boldsymbol{\xi}^{\mathcal{A}}$ are the actor extrinsics. The dynamic points $\mathcal{S}^{\mathcal{A}}$ are moved across different frames using 3D bounding boxes that specify each foreground actor's size and location. We initialize the 3D neural Gaussians for the static background and dynamic actors using the provided approximate geometry scaffolds $\mathcal{M}$ (*e.g.*, aggregated LiDAR points or multi-view stereo points). We further position a fixed number of points at a large distance to model the distant region. See Sec. 4 and supp. for details.

*Rendering:* Given $\mathcal{S}$ and camera poses $\Pi = \{\mathbf{K}_i, \boldsymbol{\xi}_i\}$, where $\mathbf{K}_i$ and $\boldsymbol{\xi}_i$ are the camera intrinsics and extrinsics for view $i$, we convert $\mathcal{S}$ to 3D Gaussians $\mathcal{G}$ and then leverage the differentiable tile rasterizer [18] to render the images $\hat{\mathbf{I}}$:

$$f_{\text{render}}(\mathcal{S}; \Pi) := f_{\text{rast}}(\mathcal{G}; \Pi) = f_{\text{rast}}(f_{\text{mlp}}(\mathcal{S}); \Pi) \tag{1}$$

$$= f_{\text{rast}}(f_{\text{mlp}}(\mathcal{S}^{\mathcal{B}}, \mathcal{S}^{\mathcal{Y}}, \mathcal{T}(\mathcal{S}^{\mathcal{A}}, \boldsymbol{\xi}^{\mathcal{A}})); \Pi) \tag{2}$$

### 3.2 Lift 2D Images to 3D as Gradients

Previous generalizable works [7,26,69] lift a few 2D images (*e.g.*, $\leq 5$) to 3D by aggregating image features extracted from source views according to camera and geometry priors (*e.g.*, epipolar geometry or multi-view stereo). Since each image is processed by a neural network separately, it cannot take many source images due to the high memory usage in both training and inference, limiting its applicability to small objects under small viewpoint changes. This is because large scenes usually have complex topology/geometry and cannot be reconstructed accurately with only a small set of source images. Moreover, it can be challenging to select and merge source views and also ensure spatial consistency.

Instead, we propose to lift 2D images to 3D space by "rendering and backpropagating" to obtain gradients w.r.t the 3D representation. Compared to leveraging networks to process images independently, 3D gradients provide a unified representation that can efficiently aggregate as many images as needed. Moreover, 3D gradients take the rendering procedure into account, naturally handling occlusions. It also enables adjustment of the 3D representation, which is not done in traditional depth rendering for view warping. Finally, the 3D gradients are fast to compute with modern differentiable rasterization engines.

Specifically, given the 3D representation $\mathcal{S}$, we first render the scene to source input views $\hat{\mathbf{I}}^{\text{src}} = f_{\text{render}}(\mathcal{S}; \Pi^{\text{src}})$ using Eqn. 2. Then, we compare the rendered images with the inputs $\mathbf{I}^{\text{src}}$, compute the reconstruction loss $L$, and back-

propagate the difference to 3D representation $\mathcal{S}$ to get accumulated gradients $\nabla_{\mathcal{S}} := \nabla_{\mathcal{S}} L(\mathcal{S}, \mathbf{I}^{\text{src}}; \Pi^{\text{src}})$ as shown in Fig. 3, with

$$L(\mathcal{S}, \mathbf{I}^{\text{src}}; \Pi^{\text{src}}) = \sum_i \left\| \mathbf{I}_i^{\text{src}} - \hat{\mathbf{I}}_i^{\text{src}} \right\|_2 = \sum_i \left\| \mathbf{I}_i^{\text{src}} - f_{\text{render}}(\mathcal{S}; \Pi_i^{\text{src}}) \right\|_2, \quad (3)$$

$$\nabla_{\mathcal{S}} L(\mathcal{S}, \mathbf{I}^{\text{src}}; \Pi^{\text{src}}) = \frac{\partial L(\mathcal{S}, \mathbf{I}^{\text{src}}; \Pi_i^{\text{src}})}{\partial \mathcal{S}} = \sum_i \frac{\partial \left\| \mathbf{I}_i^{\text{src}} - f_{\text{render}}(\mathcal{S}; \Pi_i^{\text{src}}) \right\|_2}{\partial \mathcal{S}}. \quad (4)$$

The differentiable function $f_{\text{render}}$ builds a connection between 2D and 3D, and the gradient $\nabla_{\mathcal{S}}$ encodes the 2D images in 3D using $\mathcal{S}$ as the proxy.

### 3.3 Iterative Reconstruction with a Neural Network

We now describe how we iteratively refine the scene representation $\mathcal{S}$ given the source images $\mathbf{I}^{\text{src}}$. At each step $t$, we take the current 3D representation $\mathcal{S}^{(t)}$ as a proxy to compute the gradient $\nabla_{\mathcal{S}^{(t)}}$ via differentiable rendering, thereby unprojecting 2D source images $\mathbf{I}^{\text{src}}$ to 3D, and then feed $\nabla_{\mathcal{S}^{(t)}}$ into the network $G_\theta$ to predict the updated 3D representation $\mathcal{S}^{(t+1)}$:

$$\mathcal{S}^{(t+1)} = \mathcal{S}^{(t)} + \gamma(t) \cdot G_\theta(\mathcal{S}^{(t)}, \nabla_{\mathcal{S}^{(t)}} L(\mathcal{S}^{(t)}, \mathbf{I}^{\text{src}}; \Pi^{\text{src}}); t), \quad t = 0, 1, \ldots, T-1. \quad (5)$$

$\gamma(t)$ defines the update scale at different step $t$. Intuitively, similar to gradient descent, we desire a decaying schedule $\gamma(t)$ and a small $T$ so that the network can predict an initial coarse representation and then quickly refine it. We use the cosine scheduler from DDIM [57] for $\gamma(t)$. We use a 3D UNet [10] with sparse convolution [63] as $G_\theta$ to process the neural Gaussians $\mathcal{S}$. The iterative process allows us to refine the 3D representation to achieve better quality and use a smaller network that is more efficient and easier to learn.

### 3.4 Training & Inference

We now describe the training process to train the learned optimizer $G_\theta$ and neural decoding MLP $f_{\text{mlp}}$. For each scene, we initialize the scene representation $\mathcal{S}^{(0)}$ from the geometry scaffold $\mathcal{M}$. We iteratively refine $\mathcal{S}$ with the network prediction for $T$ steps. To enhance the generalizability of reconstruction network, we render the updated representation to both source views $\mathbf{I}^{\text{src}}$ and novel views $\mathbf{I}^{\text{tgt}}$ during training ($\mathbf{I} = [\mathbf{I}^{\text{src}}, \mathbf{I}^{\text{tgt}}]$), and backpropagate the gradients to the parameters of the reconstruction network $G_\theta$ and the $f_{\text{mlp}}$. Note that in Eqn. 3 only the gradients from source views are used as input to $G_\theta$ for the next iteration, as the target views will not be available at test time. $G_\theta$ is trained to minimize final rendering loss for every iteration step $t$. We train the networks across many large outdoor scenes. The total loss $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}_{\text{mse}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\text{lpips}} \mathcal{L}_{\text{lpips}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}(\mathcal{G}), \quad (6)$$

**Table 1: Comparison to reconstruction methods on PandaSet.** The methods with best photorealism are marked using gold🟡, silver⚪, and bronze🟤 medals. † denotes the method needs to reconstruct the scene again with different source images when rendering each new view.

| Models | | Novel View Synthesis | | | Inference Time | |
|---|---|---|---|---|---|---|
| | | PSNR↑ | SSIM↑ | LPIPS↓ | Recon Time | Render FPS |
| Generalizable | MVSNeRF$_{ft}$ [7] | 23.68 | 0.659 | 0.482 | 35min 31s | 0.0392 |
| | ENeRF [26] | 24.43 | 0.736🟤 | 0.306🟡 | 0.057s† | 6.93 |
| | GNT [69] | 23.99 | 0.693 | 0.408 | 0.32s† | 0.00498 |
| | PixelSplat [6] | 23.21 | 0.653 | 0.490 | 0.74s† | 147 |
| Per-scene Opt. | Instant-NGP [39] | 24.34 | 0.729 | 0.436 | 7min 16s | 3.24 |
| | 3DGS [18] | 25.14⚪ | 0.747🟡 | 0.372🟤 | 50min 14s | 121 |
| Ours | G3R (turbo) | 24.76🟤 | 0.720 | 0.438 | 31s | 121 |
| | G3R | 25.22🟡 | 0.742⚪ | 0.371⚪ | 123s | 121 |

where $\hat{\mathbf{I}}$ is the rendered images, $\mathcal{L}_{\text{mse}}$ is the photometric loss, $\mathcal{L}_{\text{lpips}}$ is the perceptual loss [89], and $\mathcal{L}_{\text{reg}}$ is the regularization term applied on the shape of the transformed Gaussians $\mathcal{G}$ to be flat for better alignment with the surface.

$$\mathcal{L}_{\text{reg}}(\mathcal{G}) = \sum_i \max(0, d_i^{\min} - \epsilon), \tag{7}$$

where $d_i^{\min}$ is the minimal value of the 3-channel scale for each Gaussian $g_i$. We encourage it to be smaller than a threshold $\epsilon$.

*Inference:* Given the pre-trained reconstruction network $G_\theta$ and neural Gaussian decoder MLP $f_{\text{mlp}}$, we can now reconstruct novel scenes not seen during training. Specifically, we take all input images $\mathbf{I}^{\text{src}}$ for the novel scene and the 3D neural Gaussian initialization $\mathcal{S}^{(0)}$ to iteratively compute the gradients $\nabla_{\mathcal{S}}$ and refine the 3D representation. Finally, we export $\mathcal{S}^{(T)}$ to standard 3D Gaussians $\mathcal{G}^{(T)}$ for real-time rerasterization.

## 4 Experiments

We compare G3R against state-of-the-art (SoTA) generalizable and per-scene optimization approaches, ablate our design choices, and demonstrate the capability of generalization across datasets. Finally, we show that G3R-predicted representation is editable and we can generate realistic multi-camera videos.

### 4.1 Experimental Setup

*Datasets:* We conduct experiments on two public datasets with large real-world scenes: PandaSet [77], which contains dynamic actors in driving scenes and BlendedMVS [86], which contains large static infrastructure. We select 7 diverse

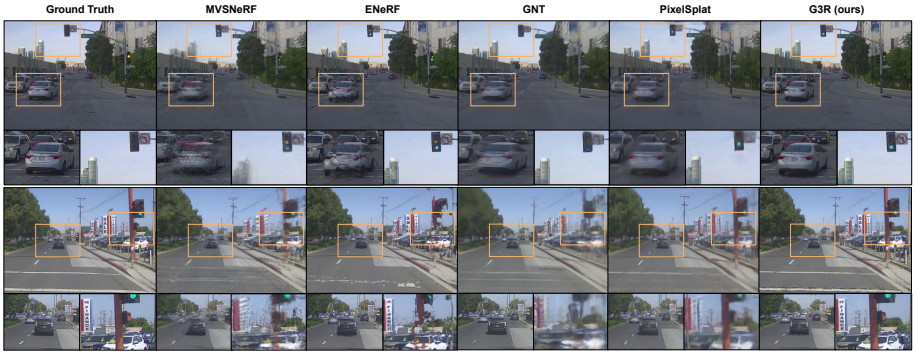


**Fig. 4: Qualitative comparison to generalizable approaches on PandaSet.**

scenes for testing with each covering around $200 \times 80m^2$, and the rest (96 scenes) for training. BlendedMVS-large is a collection of 29 real-world scenes captured by a drone, ranging in size from $10,000m^2$ to over $100,000m^2$, and also includes reconstructed meshes from multi-view stereo [1]. We select 25 scenes for training and 4 for testing. For both datasets, we use every other frame as source and the remaining for test. BlendedMVS has more challenging novel views, as the distance between two nearby views can be large (See supp. for more anlaysis).

*Implementation details:* We initialize the 3D neural Gaussians' $\mathcal{S}^{(0)}$ positions using downsampled 3D points from LiDAR points in PandaSet or mesh faces in BlendedMVS. To ensure geometry coverage, the scale for each Gaussian is initialized isotropically as the distance to its third nearest point. The rotation is set to identity and the opacity to 0.7. The other feature channels are randomly initialized. We disable view-dependent spherical harmonics from the original 3DGS [18] for simplicity and improved memory usage. We normalize the 3D gradients $\nabla_{\mathcal{S}^{(t)}} L(\mathcal{S}^{(t)})$ by channel across all the points before feeding to the network. For dynamic scenes, we adopt 3 separate networks for the background, actors, and the distant region. `tanh` activation is applied in the output layer. The per-scene reconstruction step $T$ is set as 24 during training. We train for 1000 scene iterations in total using Adam optimizer [19] with learning rate 1e-4. This takes roughly 30 hours on 2 RTX 3090 GPUs. We adopt a warm-up strategy during training that gradually increases the scene reconstruction steps in the first few scene iterations. The network is updated at each reconstruction step. We provide two variants during evaluation, where the faster model, G3R (turbo), uses fewer iterations and fewer 3D neural Gaussians. See supp. for more details.

*Baselines:* We compare G3R against both generalizable NVS (Fig. 2a) and per-scene optimization approaches (Fig. 2b). For generalizable NVS, we compare against MVSNeRF [7], ENeRF [26], GNT [69] and concurrent work PixelSplat [6]. MVSNeRF warps 2D image features onto a plane sweep and then applies a 3D CNN to reconstruct a NeRF which can be finetuned further. Similarly, ENeRF also warps multi-view source images and leverages depth-guided sampling for efficient reconstruction and rendering. GNT samples points along

each target ray and predicts the pixel color by learning the aggregation of view-wise features from the epipolar lines using transformers. PixelSplat predicts 3D Gaussians with a 2-view epipolar transformer to extract features and then predict the depth distribution and pixel-aligned Gaussians. Except for MVSNeRF, which finetunes the predicted representation on new scenes, all generalizable methods need to reconstruct the scene again with different nearest neighboring source images when rendering a new view. Unless stated otherwise, we train and evaluate all generalizable models using the same data as G3R. For per-scene optimization approaches, we compare against Instant-NGP [39] and 3DGS [18]. Instant-NGP is an efficient NeRF framework with multi-hash grid encoding and tiny MLP for fast reconstruction. We enhance Instant-NGP with depth supervision for better performance. 3DGS models the scene with 3D Gaussians and uses a differentiable rasterizer for fast scene reconstruction and real-time rendering. We enhance 3DGS to support dynamic actors and unbounded scenes with the same implementation as G3R. We optimize each test scene separately using all source frames. Please see supp. for additional details.

### 4.2 Generalizable Reconstruction on Large Scenes

*Scene Reconstruction on PandaSet:* We report scene reconstruction results on PandaSet in Tab. 1 and Fig. 4. Compared to SoTA generalizable approaches, G3R achieves significantly better photorealism and real-time rendering with an affordable reconstruction cost (2 min or less). In contrast, baselines conduct image-based rendering and result in noticeable artifacts for dynamic actors due to the lack of explicit 3D representation that can model dynamics. Moreover, they often produce blurry rendering results, especially in nearby regions where there are large view changes, due to flawed representation prediction and poor geometry estimation for view warping. We note that ENeRF achieves good LPIPS with image warping, but has severe visual artifacts and low PSNR. We also compare G3R with SoTA per-scene optimization approaches including Instant-NGP and 3DGS. Our approach achieves on par or better photorealism while shortening the reconstruciton time to 2 minutes. We note that PixelSplat leads to a higher FPS since it can only process low-resolution images and predicts a smaller number of 3D Gaussian points compared to G3R due to memory limitations.

*Scene Reconstruction on BlendedMVS:* We further consider BlendedMVS to evaluate the robustness of different methods to handle many source inputs and large view changes. As shown in Fig. 5 and Tab. 2, existing generalizable approaches including ENeRF, GNT and PixelSplat cannot handle large view changes and produce bad rendering results with significant visual artifacts due to bad geometry estimation (*e.g.*, blurry appearance, unnatural discontinuity, wrong color palette, etc). To address this issue, we adapt PixelSplat, named PixelSplat++, to leverage the 3D scaffold to reduce ambiguity and take all available source images for good coverage. Please see supp. for details. While achieving signficiant performance boost over existing generalizable methods, PixelSplat++ is still far from per-scene optimization approaches due to the challenge

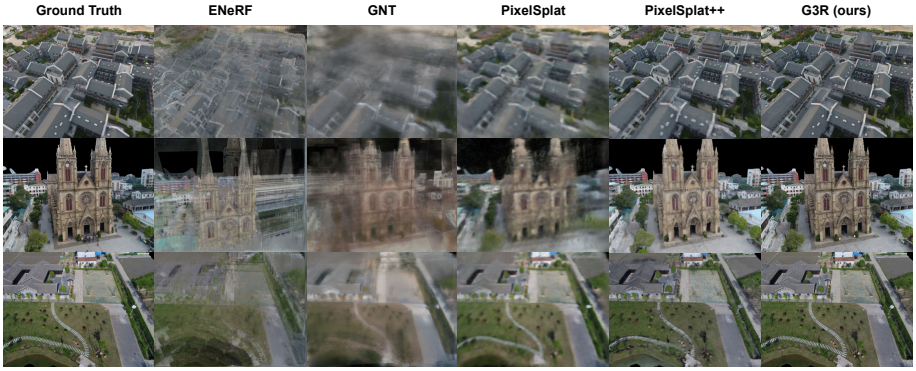


**Fig. 5: Qualitative comparison to generalizable approaches on BlendedMVS.**

**Table 2: Comparison on BlendedMVS.** The methods with best photorealism are marked using gold🟡, silver⚪, and bronze🟤 medals. † denotes the method needs to reconstruct the scene again with different source images when rendering each new view.

| Models | | Novel View Synthesis | | | Inference Time | |
|---|---|---|---|---|---|---|
| | | PSNR↑ | SSIM↑ | LPIPS↓ | Recon Time | Render FPS |
| Generalizable | ENeRF [26] | 15.21 | 0.270 | 0.660 | 0.11s† | 2.65 |
| | GNT [69] | 16.42 | 0.366 | 0.707 | 0.35s† | 0.00249 |
| | PixelSplat [6] | 16.24 | 0.344 | 0.781 | 1.14s† | 176 |
| | PixelSplat++ | 19.60 | 0.404 | 0.601 | 69s | 158 |
| Per-scene Opt. | Instant-NGP [39] | 24.86🟤 | 0.639 | 0.459🟤 | 26min 48s | 1.65 |
| | 3DGS [18] | 25.12⚪ | 0.668🟤 | 0.462 | 39.5min | 97.0 |
| Ours | G3R (turbo) | 24.56 | 0.674⚪ | 0.421⚪ | 98s | 97.0 |
| | G3R | 25.22🟡 | 0.707🟡 | 0.390🟡 | 210s | 97.0 |

of one-step prediction with limited network capacity. Our method results in the best photorealism, minimal reconstruction time and enables real-time rendering speed, which again verifies the effectiveness of our proposed paradigm. Moreover, G3R outperforms per-scene optimization methods especially in perceptual quality. We hypothesize this is because the learned data-driven prior helps handle large view changes better.

*Robust 3D Gaussian Prediction:* We compare with the rendering performance of 3DGS at novel views in Fig. 6. We observe that while 3DGS has sufficient capacity to memorize the source frames, it suffers a significant performance drop when rendering at novel views due to poor underlying geometry [8, 13]. In contrast, G3R predicts 3D gaussians in a more robust way because G3R is trained with novel view supervision across many scenes (Eq. 6) and this supervision helps regularize the 3D neural Gaussians to generalize rather than merely memorize the source views. We also consider a more challenging *extrapolation* setting where we select 20 consecutive frames as source views and simulate the future 3 frames (*e.g.*, 3 - 6 meters of shift) to evaluate the robustness when rendering at extrapolated views. As shown in Fig. 6, G3R results in more realistic rendering

**Fig. 6: Robustness of G3R vs. 3DGS.** 3DGS is sharper on interpolation views (*Interp.*), but has artifacts on extraopolation views (*Extrap.*).

**Table 3: Ablation study on PandaSet.**

| Models | PSNR | SSIM | LPIPS |
|---|---|---|---|
| **Ours** | **25.22** | **0.742** | **0.371** |
| − 3D neural Gaussian representation | 24.72 | 0.718 | 0.420 |
| − iterative reconstruction | 20.03 | 0.510 | 0.623 |
| − training with novel views | 24.59 | 0.715 | 0.419 |
| − update schedule $\gamma(t)$ | 25.03 | 0.732 | 0.400 |

**Table 4: Cross-dataset Generalization.** Pandaset-pretrained model outperforms baselines trained on BlendedMVS (see Tab 2).

| | PSNR | SSIM | LPIPS |
|---|---|---|---|
| Zero-short transfer | 24.11 | 0.653 | 0.448 |
| Finetune on 2 scenes | 24.99 | 0.676 | 0.428 |

performance. In contrast, 3DGS has severe visual artifacts highlighted by pink arrows (*e.g.*, black holes or wrong colors in road, sky and actor regions). Please refer to supp. for more analysis.

*Ablation study:* In Tab. 3, we ablate the key components proposed in G3R on PandaSet, including replacing the 3D neural Gaussians with the standard 3D Gaussian representation, conducting one-step prediction in both training and inference, training the network only with source view supervision, and switching decaying schedule $\gamma(t)$ to a constant update scale (0.3) at each step. As shown in Tab. 3, our proposed neural Gaussian representation is more expressive, thus easing the network prediction. The iterative refinement is critical in the proposed paradigm and single-step prediction fails to generate high-quality reconstruction results. We notice that single-step G3R is worse than PixelSplat as we enforce smaller updates per step for stable convergence. Moreover, we show training the network with novel views on many scenes is necessary to enhance the robustness of 3D representation for realistic novel view rendering. Finally, a proper update schedule further improves performance.

*Generalization study:* We further evaluate the PandaSet-trained G3R model (static background module) on BlendedMVS (self-driving → drone). The results in Tab. 4 show that G3R trained only on PandaSet achieves significantly better performance in BlendedMVS than generalizable baselines trained on BlendedMVS directly. We further finetune the G3R model with only 2 BlendedMVS scenes, achieving comparable results as directly training on full BlendedMVS. We also showcase applying a Pandaset-pretrained G3R model to Waymo Open Dataset (WOD) [61] scenes in Fig. 7, unveiling the potential for scalable real-world sensor simulation. See supp. for more analysis.

*Realistic and controllable camera simulation:* We now showcase applying G3R for high-fidelity multi-camera simulation in large-scale driving scenarios. Com-

**Fig. 7: PandaSet-pretrained model generalizes to Waymo Open Dataset.**

**Fig. 8: Realistic and controllable multi-camera simulation on PandaSet.** G3R reconstructs a manipulable 3D scene representation.

pared to previous generalizable approaches, our method can reconstruct a standalone representation, which allows us to control, edit and interactively render the scene for various applications. In Fig. 8, we show G3R-reconstructed scene can synthesize consistent and high-fidelity multi-camera videos from one single driving pass (top row). Moreover, we can manipulate the scene by freezing the sensors and changing the positions of dynamic actors, and render corresponding multi-camera (second row) or panorama images (bottom row).

*Limitations:* Our approach has artifacts in large extrapolations, which may require scene completion. Better surface regularization [8,13] and adversarial training [48,82] may mitigate these issues. G3R's performance suffers when initialized with sparse points, but can leverage LiDAR or fast MVS techniques [75] to mitigate this. We also do not model non-rigid deformations [34] and emissive lighting. See supp. for details.

## 5 Conclusion

In this paper, we introduce G3R, a novel approach for efficient generalizable large-scale 3D scene reconstruction. By leveraging gradient feedback signals from differentiable rendering, G3R achieves acceleration of at least 10× over state-of-the-art per-scene optimization methods, with comparable or superior photo-realism. Importantly, our method predicts a standalone 3D representation that exhibits robustness to large view changes and enables real-time rendering, making it well-suited for VR and simulation. Experiments on urban-driving and drone datasets showcase the efficacy of G3R for in-the-wild 3D scene reconstruction. Our learning-to-optimize paradigm with gradient signal can apply to other 3D representations such as triplanes with NeRF rendering, or other inverse problems such as generalizable surface reconstruction [15, 25, 33, 45].

## Acknowledgement

We sincerely thank the anonymous reviewers for their insightful comments and suggestions. We thank the Waabi team for their valuable assistance and support.

## References

1. Altizure: Mapping the world in 3d. https://www. altizure.com
2. Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. arXiv (2017)
3. Aliev, K.A., Sevastopolsky, A., Kolos, M., Ulyanov, D., Lempitsky, V.: Neural point-based graphics. In: ECCV (2020)
4. Andrychowicz, M., Denil, M., Colmenarejo, S.G., Hoffman, M.W., Pfau, D., Schaul, T., de Freitas, N.: Learning to learn by gradient descent by gradient descent. NeurIPS (2016)
5. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. CVPR (2015)
6. Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. arXiv (2023)
7. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV (2021)
8. Cheng, K., Long, X., Yang, K., Yao, Y., Yin, W., Ma, Y., Wang, W., Chen, X.: Gaussianpro: 3d gaussian splatting with progressive propagation. arXiv (2024)
9. Chibane, J., Bansal, A., Lazova, V., Pons-Moll, G.: Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. CVPR (2021)
10. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19 (2016)
11. Cong, W., Liang, H., Wang, P., Fan, Z., Chen, T., Varma, M., Wang, Y., Wang, Z.: Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In: ICCV (2023)
12. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R., Snavely, N., Tucker, R.: Deepview: View synthesis with learned gradient descent. In: CVPR (2019)
13. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. arXiv (2023)
14. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3d. In: The Twelfth International Conference on Learning Representations (2024), `https://openreview.net/forum?id=sllU8vvsFF`
15. Huang, J., Gojcic, Z., Atzmon, M., Litany, O., Fidler, S., Williams, F.: Neural kernel surface reconstruction. CVPR (2023)
16. Huang, S., Gojcic, Z., Wang, Z., Williams, F., Kasten, Y., Fidler, S., Schindler, K., Litany, O.: Neural lidar fields for novel view synthesis. arXiv (2023)
17. Johari, M.M., Lepoittevin, Y., Fleuret, F.: Geonerf: Generalizing nerf with geometry priors. In: CVPR (2022)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. TOG (2023)

19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2015)
20. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. Computer graphics forum (Print) (2021)
21. Kulh'anek, J., Derner, E., Sattler, T., Babuvska, R.: Viewformer: Nerf-free neural rendering from few images using transformers. ECCV (2022)
22. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=2lDQLiH1W4
23. Li, K., Malik, J.: Learning to optimize. ICLR (2016)
24. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: Deepim: Deep iterative matching for 6d pose estimation. IJCV (2018)
25. Liang, Y., He, H., Chen, Y.: Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. NeurIPS (2023)
26. Lin, H., Peng, S., Xu, Z., Yan, Y., Shuai, Q., Bao, H., Zhou, X.: Efficient neural radiance fields for interactive free-viewpoint video. In: SIGGRAPH Asia 2022 Conference Papers (2022)
27. Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., et al.: Vastgaussian: Vast 3d gaussians for large scene reconstruction. arXiv (2024)
28. Lin, Z.H., Liu, B., Chen, Y.T., Forsyth, D., Huang, J.B., Bhattad, A., Wang, S.: Urbanir: Large-scale urban scene inverse rendering from a single video. arXiv (2023)
29. Liu, J.Y., Chen, Y., Yang, Z., Wang, J., Manivasagam, S., Urtasun, R.: Real-time neural rasterization for large scenes. In: ICCV (2023)
30. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9298–9309 (2023)
31. Liu, Y., Peng, S., Liu, L., Wang, Q., Wang, P., Theobalt, C., Zhou, X., Wang, W.: Neural rays for occlusion-aware image-based rendering. In: CVPR (2022)
32. Ljungbergh, W., Tonderski, A., Johnander, J., Caesar, H., Åström, K., Felsberg, M., Petersson, C.: Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. arXiv preprint arXiv:2404.07762 (2024)
33. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: ECCV (2022)
34. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. arXiv (2023)
35. Ma, W.C., Wang, S., Gu, J., Manivasagam, S., Torralba, A., Urtasun, R.: Deep feedback inverse problem solver. ECCV (2021)
36. Manhardt, F., Kehl, W., Navab, N., Tombari, F.: Deep model-based 6d pose refinement in rgb. ECCV (2018)
37. Manivasagam, S., Bârsan, I.A., Wang, J., Yang, Z., Urtasun, R.: Towards zero domain gap: A comprehensive study of realistic lidar simulation for autonomy testing. In: ICCV (2023)
38. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. ECCV (2020)
39. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding (2022)
40. Müller, N., Simonelli, A., Porzi, L., Bulò, S.R., Nießner, M., Kontschieder, P.: Autorf: Learning 3d object radiance fields from single view observations. CVPR (2022)

41. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR (2020)
42. Ost, J., Mannan, F., Thuerey, N., Knodt, J., Heide, F.: Neural scene graphs for dynamic scenes. CVPR (2021)
43. Pun, A., Sun, G., Wang, J., Chen, Y., Yang, Z., Manivasagam, S., Ma, W.C., Urtasun, R.: Neural lighting simulation for urban scenes. In: NeurIPS (2023)
44. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotný, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. ICCV (2021)
45. Ren, Y., Wang, F., Zhang, T., Pollefeys, M., Susstrunk, S.E.: Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. CVPR (2022)
46. Riegler, G., Koltun, V.: Free view synthesis. ECCV (2020)
47. Riegler, G., Koltun, V.: Stable view synthesis. CVPR (2021)
48. Roessle, B., Müller, N., Porzi, L., Bulò, S.R., Kontschieder, P., Nießner, M.: Ganerf: Leveraging discriminators to optimize neural radiance fields. ACM Trans. Graph. (2023)
49. Rombach, R., Esser, P., Ommer, B.: Geometry-free view synthesis: Transformers and no 3d priors. ICCV (2021)
50. Sajjadi, M.S.M., Mahendran, A., Kipf, T., Pot, E., Duckworth, D., Lucic, M., Greff, K.: Rust: Latent neural scene representations from unposed imagery. CVPR (2022)
51. Sajjadi, M.S.M., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., Uszkoreit, J., Funkhouser, T., Tagliasacchi, A.: Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. CVPR (2022)
52. Sarva, J., Wang, J., Tu, J., Xiong, Y., Manivasagam, S., Urtasun, R.: Adv3d: Generating safety-critical 3d objects through closed-loop simulation. In: 7th Annual Conference on Robot Learning (2023), `https://openreview.net/forum?id=nyY6UgXYyfF`
53. Seitzer, M., van Steenkiste, S., Kipf, T., Greff, K., Sajjadi, M.S.M.: Dyst: Towards dynamic neural scene representations on real-world videos. arXiv (2023)
54. Sitzmann, V., Rezchikov, S., Freeman, W., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. NeurIPS (2021)
55. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019)
56. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. SIGGRAPH (2006)
57. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. ICLR (2020)
58. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. arXiv (2019)
59. Suhail, M., Esteves, C., Sigal, L., Makadia, A.: Light field neural rendering. CVPR (2021)
60. Suhail, M., Esteves, C., Sigal, L., Makadia, A.: Generalizable patch-based neural rendering. In: ECCV (2022)
61. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J.,

Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
62. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: CVPR (2022)
63. Tang, H., Yang, S., Liu, Z., Hong, K., Yu, Z., Li, X., Dai, G., Wang, Y., Han, S.: Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In: IEEE/ACM International Symposium on Microarchitecture (MICRO) (2023)
64. Tonderski, A., Lindström, C., Hess, G., Ljungbergh, W., Svensson, L., Petersson, C.: Neurad: Neural rendering for autonomous driving. arXiv (2023)
65. Trevithick, A., Yang, B.: Grf: Learning a general radiance field for 3d representation and rendering. In: ICCV (2021)
66. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: CVPR (2022)
67. Wang, J., Manivasagam, S., Chen, Y., Yang, Z., Bârsan, I.A., Yang, A.J., Ma, W.C., Urtasun, R.: CADSim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation. In: 6th Annual Conference on Robot Learning (2022)
68. Wang, J., Pun, A., Tu, J., Manivasagam, S., Sadat, A., Casas, S., Ren, M., Urtasun, R.: Advsim: Generating safety-critical scenarios for self-driving vehicles. In: CVPR (2021)
69. Wang, P., Chen, X., Chen, T., Venugopalan, S., Wang, Z., et al.: Is attention all nerf needs? arXiv (2022)
70. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021)
71. Wang, Z., Shen, T., Gao, J., Huang, S., Munkberg, J., Hasselgren, J., Gojcic, Z., Chen, W., Fidler, S.: Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In: CVPR (2023)
72. Wei, X., Zhang, K., Bi, S., Tan, H., Luan, F., Deschaintre, V., Sunkavalli, K., Su, H., Xu, Z.: Meshlrm: Large reconstruction model for high-quality mesh. arXiv preprint arXiv:2404.12385 (2024)
73. Wichrowska, O., Maheswaranathan, N., Hoffman, M.W., Colmenarejo, S.G., Denil, M., de Freitas, N., Sohl-Dickstein, J.N.: Learned optimizers that scale and generalize. ICML (2017)
74. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. arXiv (2019)
75. Wu, J., Li, R., Xu, H., Zhao, W., Zhu, Y., Sun, J., Zhang, Y.: Gomvs: Geometrically consistent cost aggregation for multi-view stereo. In: CVPR (2024)
76. Wu, Z., Liu, T., Luo, L., Zhong, Z., Chen, J., Xiao, H., Hou, C., Lou, H., Chen, Y., Yang, R., et al.: Mars: An instance-aware, modular and realistic simulator for autonomous driving. arXiv (2023)
77. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., et al.: Pandaset: Advanced sensor suite dataset for autonomous driving. In: ITSC (2021)
78. Xiong, Y., Ma, W.C., Wang, J., Urtasun, R.: Learning compact representations for lidar completion and generation. In: CVPR (2023)
79. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians for modeling dynamic urban scenes. arXiv (2024)

80. Yang, H., Hong, L., Li, A., Hu, T., Li, Z., Lee, G.H., Wang, L.: Contranerf: Generalizable neural radiance fields for synthetic-to-real novel view synthesis via contrastive learning. In: CVPR (2023)
81. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., et al.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. arXiv (2023)
82. Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.C., Yang, A.J., Urtasun, R.: Unisim: A neural closed-loop sensor simulator. In: CVPR (2023)
83. Yang, Z., Manivasagam, S., Chen, Y., Wang, J., Hu, R., Urtasun, R.: Reconstructing objects in-the-wild for realistic sensor simulation. In: ICRA (2023)
84. Yang, Z., Manivasagam, S., Liang, M., Yang, B., Ma, W.C., Urtasun, R.: Recovering and simulating pedestrians in the wild. In: Conference on Robot Learning. pp. 419–431. PMLR (2021)
85. Yang, Z., Wang, S., Manivasagam, S., Huang, Z., Ma, W.C., Yan, X., Yumer, E., Urtasun, R.: S3: Neural shape, skeleton, and skinning fields for 3d human modeling. In: CVPR. pp. 13284–13293 (2021)
86. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. CVPR (2020)
87. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021)
88. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. arXiv preprint arXiv:2404.19702 (2024)
89. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. CVPR (2018)
90. Zhenxing, M., Xu, D.: Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In: ICLR (2022)
91. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. SIGGRAPH (2018)
92. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. arXiv (2023)