Re-Nerfing: Improving Novel View Synthesis through Novel View Synthesis

Felix Tristram[1,2*] Stefano Gasperini[1,2*] Nassir Navab[1,2] Federico Tombari[1,2,3]
[1] Technical University of Munich [2] Munich Center for Machine Learning (MCML) [3] Google

Abstract

Recent NeRF and Gaussian Splatting methods have shown remarkable reconstruction and novel view synthesis (NVS) capabilities, but require a substantial number of images of the scene from diverse viewpoints to render high-quality novel views. With fewer images, they struggle with correctly triangulating the underlying 3D geometry and converging to a sub-optimal solution (e.g., with floaters or blurry renderings). In this paper, we propose Re-Nerfing, a general approach that leverages NVS itself to tackle this convergence problem. Using an already optimized scene representation model, we generate novel views derived from existing perspectives and use these to augment the training data of a second model. We add the augmented views to improve the scene coverage and mask out their uncertain areas to enhance the quality of the training signal. This introduces additional multi-view constraints and allows the second model to converge to a better solution. With Re-Nerfing, we introduce an iterative paradigm that achieves significant improvements upon multiple pipelines based on NeRF and 3D Gaussian Splatting in sparse and highly-sparse view settings of the mip-NeRF 360, Tanks and Temples, and LLFF datasets. Notably, Re-Nerfing does not require prior knowledge or extra supervision signals, making it a flexible and practical enhancement to any learnable NVS pipeline.

1. Introduction

Novel view synthesis (NVS) methods based on Neural Radiance Field (NeRF) [39] and 3D Gaussian Splatting (3DGS) [29] have recently revolutionized 3D scene representation and rendering, enabling unprecedented quality in synthesizing novel views from a set of images. These techniques have been applied across various tasks, simplifying content creation, rendering, and reconstruction workflows.

Despite its remarkable success, NeRF [39] has several limitations, such as slow training [40], failures when noisy or no camera poses are available [5, 35], and computationally intense rendering mostly incompatible with mobile applications [13]. Many works have been proposed to address these issues, such as 3DGS [29], changing from an implicit to an explicit representation, thus shortening training times and simplifying inference. However, collecting enough high-quality images with sufficient overlap to ensure successful model convergence remains a challenging task. Towards this end, researchers have explored the application of NVS in sparse view settings with only a handful of images available [42, 61, 69].

Complex geometries and large-scale environments often lead to artifacts due to the inability to correctly triangulate the scenes' 3D structure [2, 16, 51]. This is particularly severe in sparse-view settings, where the limited views lead to hallucinations, e.g., floaters in free space due to the ambiguity of shape and radiance [16, 36, 61]. To mitigate these

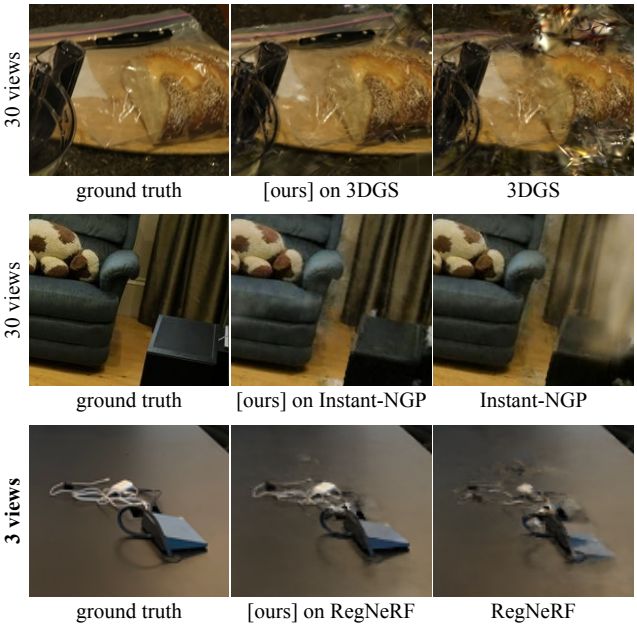

Figure 1. Examples of synthesized views by 3DGS [29], Instant-NGP [40], RegNeRF [42], with and without our proposed Re-Nerfing. Ours improves NVS by enhancing the optimization and reaching better global solutions, thanks to novel views synthesized by the baseline model added to the training data. Crops from the mip-NeRF 360 [3] dataset for 3DGS and Instant-NGP trained on 30 views and from LLFF [38] for RegNeRF trained on 3 views demonstrate the effectiveness of ours. White arrows indicate significant rendering errors.

*Equal contribution.

issues, depth and structural priors have been introduced to the optimization, leveraging additional geometric information [16, 51, 61]. However, reliable dense depth estimates are also hard to obtain, and using wrongly estimated depth can further decrease performance [16].

In this work, we mitigate these open issues with a simple and effective add-on solution exploiting the inherent view synthesis capability of NVS methods. We achieve this with Re-Nerfing, a multi-stage, general pipeline that introduces structural constraints from synthesized views generated by a previously trained model on the same scene. As shown in Figure 1, thanks to its ability to improve the view coverage while preserving quality by discarding uncertain regions, Re-Nerfing significantly improves the synthesis of new views. The main contributions of this paper can be summarized as follows:

- We show how NeRF's and 3DGS' novel view synthesis is beneficial to enhance their own rendering quality.
- We propose Re-Nerfing, a simple and effective iterative augmentation technique to enhance any NVS outputs by optimizing a new NVS model with the addition of synthesized views. We synthesize such views to improve the scene coverage and mask them to discard uncertain areas.
- We analyze the origin of Re-Nerfing's improvements with respect to the optimization process of NVS pipelines and show the wide applicability and effectiveness of Re-Nerfing upon various NVS methods [29, 40, 42, 62] and datasets [3, 30, 38].

2. Related Work

Traditional interpolation techniques based on light field sampling can synthesize photorealistic novel views from dense image sets [33]. In sparser settings, neural networks exploit visual appearance correlation across views [20, 67], while 3DGS [29] optimizes scenes as 3D Gaussians with spherical harmonics.

Neural implicit representations encode 3D shapes in network weights rather than explicit representations like point clouds or meshes [48, 68]. Pioneering works include Occupancy Networks [37] and DeepSDF [44], which encode complex shapes in continuous function spaces without space discretization, enabling finer details [56].

NeRF NeRF [39] combines neural implicit representations with differentiable rendering, encoding volumetric properties and view-dependent appearance for high-quality novel views. Subsequent works address various limitations: faster training [9, 21, 23, 40, 57], imperfect camera poses [5, 35, 61], unbounded scenes [3], and dynamic scenes [8, 22, 45, 46]. Additional extensions include consistent geometry [51], output modification [28], scene understanding [31], city-scale rendering [58], and camera optimization [47]. PyNeRF [62] combines mip-NeRF [2] and grid-based models for fast anti-aliased rendering.

Sparse-View Settings Highly sparse settings are challenging as standard multi-view constraints are insufficient [14, 61]. Researchers use additional data and external models for geometric supervision [51, 61] or semantic knowledge [49, 65]. PixelNeRF [65] learns priors from multiple scenes, while Gaussian Splatting extensions handle sparse views [18, 34, 43, 69]. ReconFusion [64] uses diffusion models for additional view constraints. However, these methods require external models and large training datasets, eliminating NeRF and 3DGS's advantage of needing only scene-specific data.

Geometric Constraints Geometric consistency is crucial for NVS quality [51]. DS-NeRF [16] uses sparse COLMAP depth [53] for supervision, while [51] employs depth completion networks with uncertainty estimation. Urban Radiance Fields [50] utilizes LiDAR data for outdoor scenes. FSGS [69] and related methods [34, 43] rely on monocular depth prediction. SPARF [61] exploits multi-view geometry and pixel correspondences, minimizing reprojection error with jointly optimized depth and poses.

Data Augmentation for NVS Data augmentation increases training data with sample variations to regularize models [25, 32, 41]. For NVS, PANeRF [1], GeoAug [10], and VM-NeRF [6] generate views by warping existing ones via homography [1] or depth estimates [6, 10]. Aug-NeRF [12] trains robust NeRFs with worst-case perturbations of input coordinates, features, and outputs.

Distillation and Semi-Supervised Learning Knowledge distillation transfers knowledge from complex teacher models to efficient student models [27], extending to semi-supervised learning in segmentation [11] and depth estimation [24]. Naive-student [11] predicts pseudo-labels for unlabeled data. NeRF-based approaches include training stereo depth models [60] and monocular depth estimators [19] using NeRF-synthesized data.

We enhance NVS in sparse settings by exploiting the method's inherent synthesis capabilities through data augmentation using only available views without external data or models. We train a baseline NVS method on available data, then use it to generate novel views for training a subsequent model. We sample views to improve scene coverage while preserving quality by masking uncertain regions. Unlike [19, 60] who train depth estimators with NeRF data, we train NVS models with NVS data, and unlike other augmentation methods [1, 6, 10], our views are fully synthesized from NVS models.

3. Preliminaries

In the following, we provide an overview of the common optimization strategies of NeRF- and 3DGS-based approaches for NVS.

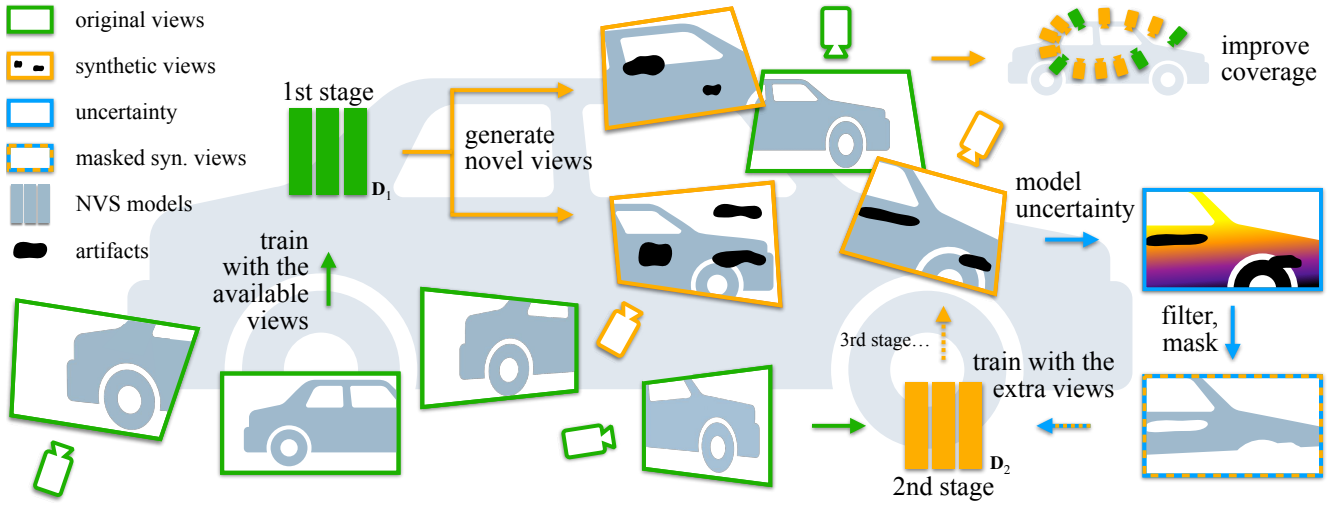

Figure 2. Re-Nerfing is an iterative framework. Compatible with any learned NVS pipeline, it first optimizes a model with the available views (green, 1st). This model then generates novel views from poses selected to improve the scene coverage (orange). Then, it computes the model's uncertainty on such novel views (blue) to keep only the more certain regions (orange-blue). Finally, masked and original views are used to optimize a second NVS model (orange, 2nd). This process can be repeated iteratively by generating novel views with the second model (3rd stage and following stages).

3.1. Optimization of NeRF and 3DGS

To better understand the shortcomings of NeRF [39] and 3DGS [29] optimizations in sparse-view settings, we review how they optimize their scene representation and render images from it. Both methods are optimized through photometric consistency, where the color of each ray or pixel is compared to the ground truth color with an L2 loss or a mixture of L1 and SSIM losses [29].

Mathematically, this is represented as:

$$\mathcal{L} = \sum_{r \in R} \|C(r) - \hat{C}(r)\|^2, \tag{1}$$

where $C(r)$ is the color of ray r in the ground truth image, $\hat{C}(r)$ is the model's prediction for r, and R is the set of rays across all training views.

This color $C(r)$ is obtained through volumetric rendering via the following equations, where density o, transmittance T, and color $\mathbf{c}$ are taken along a ray or z-value at intervals δ_i:

$$C = \sum_{i}^{N} T_i \alpha_i \mathbf{c}_i, \text{ where} \tag{2}$$

$$\alpha_i = (1 - \exp(-o_i \delta_i)) \text{ and } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j). \tag{3}$$

This formulation holds for both NeRF-based architectures and neural point-based rendering approaches, such as 3DGS [29], where the image formation process is the same, but images are rendered through ray-tracing or rasterization.

4. Method

As shown in Figure 2, our proposed Re-Nerfing operates in a multi-stage fashion. First, a baseline method is trained with the available views (Section 3.1). It is then used to generate novel views to improve the scene coverage (Section 4.1), and lastly, a new model is trained on the original and the synthesized views (Section 4.2), discarding uncertain regions of the rendered views to improve the signal's quality. Re-Nerfing is a general framework compatible with any learnable pipeline for NVS. Furthermore, Re-Nerfing follows an iterative process, so consecutive iterations can be executed using the model trained at the previous iteration as a baseline for the next one.

4.1. Re-Nerfing: Augmenting via Synthesized Views

To represent the geometry of the given scene, only the density o_i at the sampling or point locations can be optimized. In scenarios with only a few views available, e.g., due to crowd-sourced or previously captured images, NVS is particularly difficult as this density can no longer be correctly triangulated since there are not enough multi-view constraints and excessive ambiguity in the color supervision.

This optimization depends critically on multi-view constraints, where views from different angles provide intersecting rays that enable triangulation of points in 3D space. However, when the number of views is sparse, the constraints from these rays become insufficient to uniquely determine the geometry, leading to multiple plausible 3D configurations that satisfy the observed data, a phenomenon known as "local minima" or "degenerate solutions."

This can be seen in Figure 3, where green cameras rep-

resent original views and the shaded green or orange regions represent valid 3D solutions where density could be distributed given the available green cameras. The overall optimization process would converge to one of the many valid 3D solutions within the green or orange region (e.g., a local minimum), leading to dissatisfying geometry reconstruction and impaired NVS.

Various methods tackled this with prior knowledge [16, 61] or regularization [42]. Instead, we propose to leverage the inherent reconstruction and rendering capabilities of NVS methods themselves. Our Re-Nerfing addresses the sparse-view challenges by augmenting the original views with synthesized views generated by the model itself. So, we *render novel views as additional training data* in subsequent optimizations (Section 4.2), thereby adding multi-view constraints *without* any prior knowledge or external models. From a theoretical perspective, our iterative view generation process can be understood as progressively increasing the density of multi-view constraints, which improves the conditioning of the optimization problem.

4.1.1. Enhanced Scene Coverage and Constraints

We seek to improve the view coverage of the captured scene or object through our augmentations. So, we generate multiple novel views from camera poses interpolated across existing neighboring views. Let $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$ represent the set of available camera poses in a sparse-view setting. Given a novel camera pose $\mathbf{P}_i^{\text{new}}$ interpolated from pairs in $\mathbf{P}$, we can synthesize a new view by rendering an image from this pose. The addition of $\mathbf{P}_i^{\text{new}}$ creates extra constraints for 3D points that may not have been observable from the original views alone (e.g., in Figure 3).

Mathematically, for a 3D point X, the reprojection error across all views, including our synthesized ones, becomes:

$$\mathcal{L}_{\text{reproj}} = \sum_{\mathbf{P}_i \in \mathbf{P} \cup \mathbf{P}^{\text{new}}} \|\pi(\mathbf{P}_i, X) - x_i\|^2, \tag{4}$$

where $\pi(\mathbf{P}_i, X)$ denotes the projection of point X onto the image plane of view $\mathbf{P}_i$, and x_i is the observed (or synthesized) pixel location in view $\mathbf{P}_i$. Adding new views $\mathbf{P}^{\text{new}}$ (orange cameras in Figure 3) increases the number of reprojection terms, effectively regularizing the solution for X and reducing the probability of degenerate reconstructions. In the figure, possible solutions are reduced from the green-shaded region to the orange-shaded region, aiding the triangulation of the scene's geometry/density.

4.1.2. View Coverage Details

We define the hyperparameters N and Ω as the augmentation factor and the set of $N-1$ interpolation factors β_j, respectively. Specifically, for the case of equally spaced novel views:

$$\Omega = \left\{ \beta_j \mid \beta_j = \frac{j}{N},\ j = 1, 2, \dots, N-1 \right\} \tag{5}$$

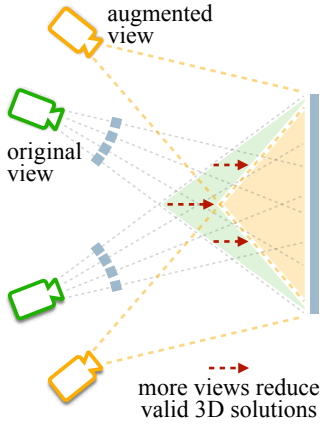

Figure 3. Graphical explanation of how our augmented views (orange) improve the scene geometry and the effectiveness of the multi-view constraints by reducing the valid 3D solutions to a smaller space (shaded orange) and ultimately enabling the convergence of the optimization to a better solution.

However, equally spaced synthesized views are not ideal as the quality would degrade around $\beta = 0.5$. This is particularly relevant in sparser settings where the larger camera displacement between existing poses would lead to worse synthesized views (Section 4.1.3).

Considering the existing camera poses $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ with $\mathbf{P}_i = [\mathbf{p}_i, \mathbf{q}_i]$, such that $\mathbf{p}_i$ is the position vector in 3D space and $\mathbf{q}_i$ is the quaternion representation of the rotation,

$$\bar{\mathbf{p}}_{i,j} = (1 - \beta_j)\mathbf{p}_i + \beta_j \mathbf{p}_{i+1} \tag{6}$$

is the interpolation of the position component given the interpolation factor β_j. For the rotation, we compute the interpolated $\bar{\mathbf{q}}_{i,j}$ from the original pose quaternions $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ using SLERP [54] as follows:

$$\bar{\mathbf{q}}_{i,j} = \frac{\sin((1-\beta_j)\theta)}{\sin(\theta)}\mathbf{q}_i + \frac{\sin(\beta_j \theta)}{\sin(\theta)}\mathbf{q}_{i+1} \tag{7}$$

Instead of interpolating across existing views, other strategies could generate novel views around a half dome centered at the scene or object center, or following a specific pattern. However, in real-world scenes, such techniques may raise issues with occlusions, as the sampled poses might collide with parts of the scene (e.g., a wall) or end up inside an object. Interpolating is more conservative and does not require extra information about the scene, such as the scene occupancy, to avoid collisions and unrealistic viewpoints. Nevertheless, multiple pose sampling strategies could be combined to further enhance the view synthesis.

The formulation above is based on two consecutive poses $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$. In settings where sequential information is not available, a sequence can be simulated by considering consecutive poses from neighboring positions in 3D space [15]. This can be achieved with a nearest neighbor

on the vectors $\mathbf{p}$ using the Euclidean distance or the angular distance between the quaternions $\mathbf{q}$.

4.1.3. Image Quality

Our proposed method has to balance the synthesized image quality with achieving as wide a baseline as possible with regard to the original views. This is because novel views close to the training views will be of higher quality and exhibit fewer artifacts compared to ones that are further away, but add fewer multi-view constraints. In practice, we achieve this trade-off through generating many interpolated views between the original training views $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ and achieve a good balance between both factors. We illustrate the trade-off in the Supplementary Material.

4.2. Re-Nerfing: Re-Optimizing NeRF or 3DGS

After optimizing the first model $\mathbf{D}_1$ and using it to generate novel views (Section 4.1), we train another NVS model $\mathbf{D}_2$ from scratch with the addition of the synthesized views as data augmentation. To improve the signal's quality, we mask out uncertain regions of the generated views (Section 4.2.2).

4.2.1. Reducing Local Minima via Iterative Optimization

The proposed Re-Nerfing uses a base model $\mathbf{D}_1$ to augment the training data for a later model $\mathbf{D}_2$. This introduces a general iterative paradigm, where each iteration n refines the model $\mathbf{D}_n$ by generating synthesized views $\mathbf{P}_j^{\text{new}}$ from the previous iteration's model $\mathbf{D}_{n-1}$. This iterative process can be seen as minimizing an energy function that converges toward a better minimum as the iteration number n increases:

$$\mathcal{L}_{\text{reproj},n} = \sum_{\mathbf{P}_i \in \mathbf{P} \cup \mathbf{P}_1^{\text{new}} \cup \cdots \cup \mathbf{P}_j^{\text{new}}} \|\pi(\mathbf{P}_i, X) - x_i\|^2, \quad (8)$$

where $\mathcal{L}_n$ is minimized over the parameters of $\mathbf{D}_n$. By augmenting via synthesized views with improved image quality at each iteration, we effectively increase the conditioning of the multi-view geometry problem, reducing the prevalence of local minima that the optimization could otherwise converge to. We avoid degeneration and forgetting [55] by optimizing $\mathbf{D}_n$ only on the views generated by $\mathbf{D}_{n-1}$ plus the original ones and not those synthesized from $\mathbf{D}_{n-2}$.

4.2.2. Masking Uncertain Regions

We seek to add a high-quality training signal to the new model $\mathbf{D}_n$ by means of augmentation with our synthesized views. Since the quality of such synthesized views is not on par with the originals, they may display artifacts that can impact the renderings. Therefore, we aim to remove such artifacts and improve the training signal. So, we estimate the reconstruction uncertainty and use it to mask out the uncertain regions from our synthesized images such that the optimization focuses on the high-confidence areas.

Let $U(\mathbf{P}_i^{\text{new}})$ be the uncertainty at view $\mathbf{P}_i^{\text{new}}$. The masked loss is then:

$$\mathcal{L}_{\text{masked}} = \sum_{r \in R, \mathbf{P}_i \in \mathbf{P} \cup \mathbf{P}^{\text{new}}} \|C(r) - \hat{C}(r)\|^2 \cdot \mathbf{1}_{\{U(\mathbf{P}_i) < \tau\}} i, \quad (9)$$

where $\mathbf{1}_{\{U(\mathbf{P}_i)<\tau\}}$ is an indicator function that includes rays from our synthesized views only if their uncertainty is below a threshold τ.

The uncertainty obtained at each sampling location $\mathbf{x}$ can then be rendered via volumetric rendering. Specifically, for each ray $\mathbf{r}$ of a novel pose $\mathbf{P}_i^{\text{new}}$ with ray samples $\mathbf{x}$, we synthesize the color $C_{\mathbf{D}_1}(\mathbf{r})$ and render the corresponding uncertainty $U(\mathbf{r}(\mathbf{x}))$. Then, we create pixel masks where rays with $U(\mathbf{r}(\mathbf{x}))$ higher than the threshold τ are masked out. To optimize the second model $\mathbf{D}_2$, we use the generated views as part of the training images and query their masked rays during the course of optimization. This masking step reduces the influence of unreliable regions on the optimization, enhancing the robustness of Re-Nerfing.

In this work, we estimate the uncertainty using Bayes' Rays [26], but the method is compatible with other techniques, too. Bayes' Rays [26] outputs a volumetric uncertainty field for every input coordinate $\mathbf{x}$:

$$U(\mathbf{x}) = \text{Trilinear}(\mathbf{x}, \sigma), \quad \text{where} \quad \sigma = ||\boldsymbol{\sigma}||_2 \quad (10)$$

Here $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ represents the diagonal entries of the variance $\boldsymbol{\Sigma}$ of the deformation field, which intuitively encodes how much one can permute the underlying NeRF geometry (or Gaussian positions) without affecting the reconstruction quality. $\boldsymbol{\Sigma}$ is approximated via the diagonal of the Hessian $\boldsymbol{H}(\boldsymbol{\theta})$, approximated for all rays $\boldsymbol{r}$ of a sampling pool R on the parameters $\boldsymbol{\theta}$ of the deformation field as:

$$\boldsymbol{H}(\boldsymbol{\theta}) \approx \frac{2}{R} \sum_{\boldsymbol{r}} \boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{r})^T \boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{r}) + 2\lambda \boldsymbol{I} \quad (11)$$

$$\boldsymbol{\Sigma} \approx \text{diag}(\boldsymbol{H}(\boldsymbol{\theta}))^{-1} \quad (12)$$

For the full derivation of this formulation, we refer to [26]. Originally proposed for NeRF-based methods, we extend Bayes' Rays to 3DGS by replacing the sampling locations $\mathbf{x}$ with the Gaussian positions.

4.2.3. Training Schedule

For both NeRF and 3DGS pipelines, optimization first focuses on scene geometry, then appearance. Since synthesized views have lower quality than real images, especially distant from training views (Section 4.1.3), we sample from them only during early optimization stages where they help disambiguate geometry, then remove them at iteration γ as they become detrimental to appearance.

Third and later stages Since synthesized data augmentation regularizes later models $\mathbf{D}_n$, these are less prone

to overfitting, potentially allowing increased model size and higher quality. However, this is beyond our scope as we use identical architectures across iterations n. By iteratively augmenting training views, Re-Nerfing improves NVS problem conditioning, increasing multi-view constraint density and reducing local minima prevalence (Figure 3). Uncertainty masking refines this by ensuring only high-confidence synthesized data contributes to optimization, resulting in more globally optimal 3D scene reconstruction and rendering despite limited initial data.

5. Experiments and Results

Since having a more diverse set of views during the start of optimization benefits reconstruction, we observe faster convergence when using Re-Nerfing. Interestingly, not only does using Re-Nerfing improve convergence speed and test-view quality, but we also observe that it boosts PSNR values on the training views, further reinforcing our intuition.

5.1. Experimental Setup

Datasets We evaluated Re-Nerfing on three public datasets. We used the 7 scenes from **mip-NeRF 360** [3] (bounded indoor and unbounded outdoor scenes), the 8 scenes from **LLFF** [38] (smaller forward-facing scenes), and the 7 training scenes from **Tanks and Temples** (T2) [30] (free-form captures with varied camera trajectories and lighting). For mip-NeRF 360 and Tanks and Temples, we created sparse settings by sampling training views while preserving scene coverage, then selecting test views from the remaining original test set. We used **30 views** out of 232 available on average for mip-NeRF 360 (50 for *stump* and *bicycle* due to baseline convergence issues), and 64 out of 470 images (13.6%) for Tanks and Temples. For LLFF, we used highly sparse settings with as few as **3 views**, following RegNeRF [42]. Exact data splits will be made available.

Metrics We evaluate test views using standard metrics: PSNR, SSIM [63], and LPIPS [66]. For 3D reconstruction quality, we compare depth images from sparse-view models against a baseline model optimized with all available views as pseudo ground truth. This evaluation is provided on Tanks and Temples. Due to unbounded scenes with sky regions causing large depth variance, we report both L1-Error and L1-Error@0.9 (discarding the largest 10% of each scene's depth distribution). While Tanks and Temples provides laser scans, coordinate system misalignment between camera poses and scans prevents their use, as reconstruction noise makes initial alignment and ICP [4] impossible.

Prior Works and Baselines We showcase the effectiveness of our method by applying it to a diverse set of NVS methods, namely PyNeRF [62], Instant-NGP [40], RegNeRF [42], and 3DGS [29]. RegNeRF is implemented in JAX [7], for which no implementation of Bayes'Rays [26] exists. Therefore, with RegNeRF, we show only the impact of our augmentations without uncertainty masks.

Method	PSNR	SSIM	LPIPS
Instant-NGP	20.76	0.659	0.301
+ Re-Nerfing [ours]	**21.35**	**0.680**	**0.273**
PyNeRF	22.65	0.746	0.182
+ Re-Nerfing [ours]	**23.51**	**0.772**	**0.160**
3DGS	20.49	0.600	0.259
+ Re-Nerfing [ours]	**21.51**	**0.647**	**0.231**

Table 1. Rendering performance on the **mip-NeRF 360** [3] dataset with **sparse views**. The proposed Re-Nerfing is applied on Instant-NGP [40], PyNeRF [62] and 3DGS [29].

Implementation Details We train our method on images down-sampled by 8 from the original image sizes for the available outdoor scenes and by 4 for the indoor scenes of mip-NeRF 360, bringing both to a comparable image size. For LLFF, we down-sample all images by 8, following RegNeRF [42]. For PyNeRF [62], 3DGS [29] and Instant-NGP [40], we used their implementation available in Nerfstudio [59]. For RegNeRF, we used the original implementation in JAX [7]. We trained all Nerfstudio models for 30k iterations from scratch with a batch size of 8192 rays for PyNerf and 1024 rays for Instant-NGP and inherited all other losses and hyperparameters of the baselines. For RegNeRF, we followed their training paradigm and hyperparameters. We set our hyperparameter γ to 8k for PyNerf and Gaussian-Splatting, 5K for RegNeRF, and 200 for Instant-NGP. We train all models on a 24GB NVIDIA RTX 3090 or 4090 GPU.

5.2. Quantitative Results

5.2.1. Re-Nerfing as Flexible Enhancer

In Table 1, we report the results on the mip-NeRF 360 dataset [3] applying the proposed Re-Nerfing on Instant-NGP [40], PyNeRF [62], and 3D Gaussian Splatting [29] in sparse view settings. In Table 2, we report the results on the LLFF dataset [38] applying Re-Nerfing on RegNeRF [42] in highly-sparse 3, 6, and 9 view settings.

Our method brings notable improvements across various settings, increasing PSNR with a single round by **1.02 dB over 3DGS**, 0.86 dB over PyNeRF, 0.64 dB over Instant-NGP (Table 1), and 0.50 dB over RegNeRF (6 views, Table 2), while improving SSIM and LPIPS, too. With more rounds (Appendix Table 5), ours reaches **3.53 dB PSNR over 3DGS** [29]. Despite their synthesized nature, adding our augmented views mitigates overfitting and acts as a regularizer, supporting the improvements.

In Table 2, in particular, we combine Re-Nerfing with

3 Views	PSNR	SSIM	LPIPS
baseline: mip-NeRF	14.62	0.351	0.495
+ RegNeRF	19.08	0.685	0.148
+ Re-Nerfing, 1 round [ours]	19.35	0.713	0.133
+ extra round (2) [ours]	19.47	0.724	**0.126**
+ extra round (3) [ours]	**19.52**	**0.725**	0.127
6 Views	PSNR	SSIM	LPIPS
baseline: mip-NeRF	20.87	0.692	0.255
+ RegNeRF	23.10	0.822	0.078
+ Re-Nerfing [ours]	**23.60**	**0.839**	**0.069**
9 Views	PSNR	SSIM	LPIPS
baseline: mip-NeRF	24.26	0.805	0.172
+ RegNeRF	24.86	0.870	0.055
+ Re-Nerfing [ours]	**25.14**	**0.877**	**0.052**

Table 2. **Highly sparse** settings with 3, 6, and 9 views on the **LLFF** dataset [38]. We apply ours on RegNeRF [42] (which builds on mip-NeRF [2]), and then iteratively via extra rounds. Results are averaged over all scenes.

the regularization of RegNeRF, showing how Re-Nerfing can adapt to any optimized NVS method. This is because Re-Nerfing is an add-on enhancer and not an alternative to them. Paired with RegNeRF, Re-Nerfing further improves the quality of the novel views.

We also perform comparisons on more in-the-wild captures of the Tanks & Temples dataset [30] in Table 3 with 3DGS [29], where our method brings an improvement in both image metrics (0.49dB) and L1 depth errors (ca. 17%).

Overall, the efficacy of our method on both implicit (NeRF-based) and explicit (3DGS) scene representations is particularly notable as it is achieved without using extra data, additional models, or supervision, but only with information already available to the baselines.

5.2.2. How it Works

As described in Section 3.1, NeRF's and 3DGS' optimization is similar to SfM approaches, as matching pixel features are triangulated in 3D. Especially in sparse-view settings, this is highly under-constrained with many degenerate solutions. This can lead to local minima, which fit well with the training views but not the test views due to the inconsistent geometry (Section 4.1).

With Re-Nerfing, we build upon this (potentially sub-optimal) initial optimization and render synthesized views as extra constraints for consecutive optimization rounds. It is widely known in the community that more viewpoints help NVS. This same concept drives Re-Nerfing, too, albeit a step further, with synthesized views. As described in Section 4.1, thanks to our extra constraints from the augmented views, some degenerate solutions are removed and the optimization converges to a better 3D structure. This is supported by the improved L1-Error in Table 3 and 7 in the Supplementary Material and can also be observed in the supplementary video, where the rendered depth maps are notably smoother and more complete.

Metrics	3DGS [29]	+ Re-Nerfing
PSNR	16.64	**17.13**
SSIM	0.509	**0.530**
LPIPS	0.365	**0.352**
L1-Error	2.941	**2.438**
L1-Error@0.9	2.069	**1.704**

Table 3. Comparison of Re-Nerfing paired with 3DGS [29] on a sparse version of the training samples from the Tanks & Temples dataset [30].

Ablation study Through an ablation study based on 3DGS [29], Table 4 shows the importance of:

- early optimization stages for the scene geometry,
- the impact of the synthesized views on the scene geometry and early stages,
- removing the synthesized views as the scene geometry converges to not learn wrong details from them, and
- resetting the optimization at each of the Re-Nerfing rounds.

Specifically, adding the synthesized views without resetting the optimization (a2) does not bring any benefits and worsens LPIPS. Resetting and keeping the rendered views until the end (a3) improves, but removing them leads to better details (a5). On the contrary, adding the views after the scene geometry has already been captured (a4) leads to mixed results compared to the baselines. Finally, masking out uncertain regions of the synthesized views (a6) further improves across the board as it increases the quality of the supervision signal. While Re-Nerfing increases the training iterations, e.g., by 2x in the table, simply doubling the iterations performs even worse than the baseline (a1), as the optimization overfits on the training views and converges to a sub-optimal local minima.

Altogether, Re-Nerfing reaches a better global solution without adding extra information to the baseline. This aspect is similar to other iterative or multi-stage frameworks from different domains [11, 24], where resetting the optimization at each round leads to better solutions. However, what is notable with Re-Nerfing is that we use the optimized model itself to fully generate the augmented training data.

5.3. Qualitative Results

In Figure 4, we demonstrate Re-Nerfing's benefits on the Tanks and Temples dataset [30] using 3DGS. Our method

ID	Configuration	PSNR	SSIM	LPIPS
a0	baseline: 3DGS, optimized for 30k iterations	20.49	0.600	0.259
a1	a0 + 30k extra iterations (60k total)	20.40	0.598	0.260
a2	add to a0 our synthesized views after 30k iterations (no reset)	20.05	0.538	0.343
a3	reset the a0 optimization at 30k, then add our synthesized views	20.60	0.586	0.269
a4	reset a0 at 30k, then add our synthesized views after 15k	20.81	0.590	0.277
a5	reset a0 at 30k, add our syn.views, then remove them after 8k	21.30	0.627	0.235
a6	a5 + uncertainty masks = **Re-Nerfing**	**21.51**	**0.647**	**0.231**

Table 4. **Ablation study**. Sparse-view settings on the mip-NeRF 360 dataset [3] showing the impact of our method on 3DGS [29]. Configuration a6 represents our Re-Nerfing.

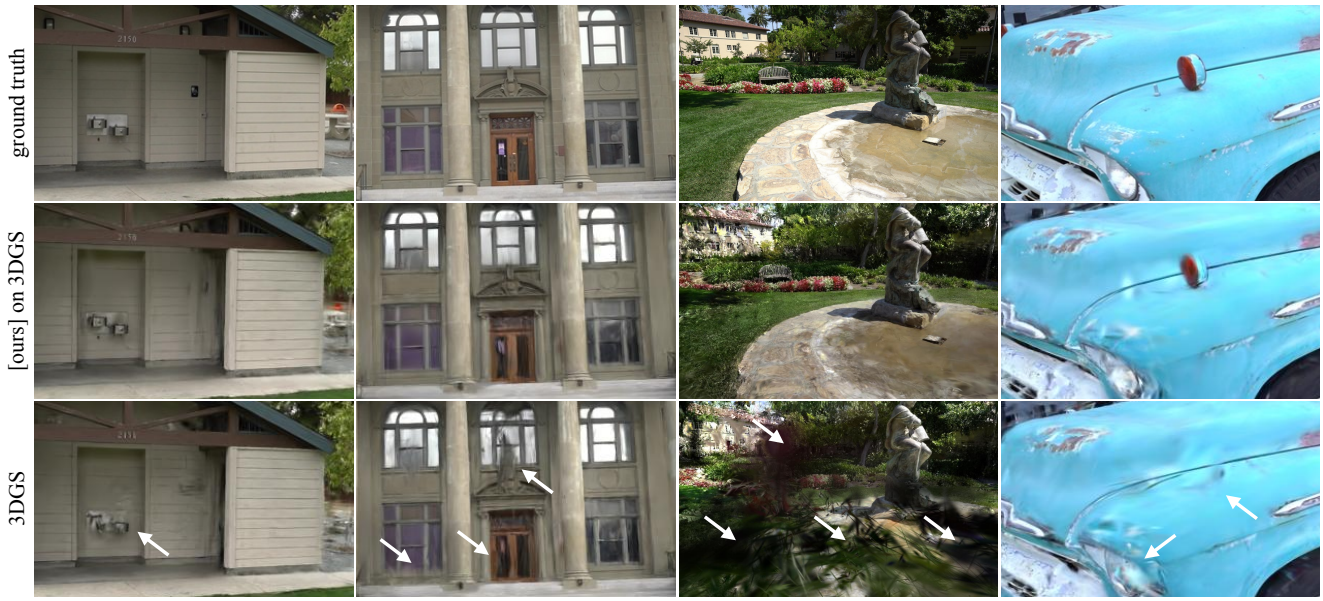

Figure 4. Qualitative results on images from the test set of the Tanks and Temples dataset [30]. 3DGS [29] novel views are shown with (second row) and without (third row) our proposed method. White arrows indicate significant rendering errors.

significantly improves rendering quality of challenging details like water fountains and windows, removes major floater artifacts, and better preserves geometric information such as the orange indicator light on the truck. These results demonstrate Re-Nerfing's effectiveness across diverse scenarios.

The **Supplementary Material** includes more details, quantitative and qualitative results, and a qualitative video, which showcases the effectiveness of Re-Nerfing with RGB, depth images, and rendered uncertainty.

Limitations Our method relies on the first stage to provide reasonable renderings and a decent estimate of the scene geometry, so it is challenging to enhance poorly rendered scenes, e.g., the *stump* scene with Instant-NGP [40], which could not converge. The proposed method also relies on the uncertainty estimates to discard artifacts and low-quality areas. Therefore, better uncertainty estimates would be beneficial. Uncertainty calibration of the threshold τ per scene could help refine the estimates while preserving the signal in higher-quality areas.

6. Conclusion

This work enhances NeRFs and 3DGSs by leveraging their own novel view synthesis capabilities. The introduced Re-Nerfing is a general, iterative approach that synthesizes novel views to improve the optimization of a successive model by increasing the scene coverage and augmenting its training data. To provide a valuable training signal for the next model optimization, Re-Nerfing estimates the uncertainty for each augmented view and discards the uncertain regions. As shown with various NeRF- and 3DGS-based pipelines, our method leads to better optimization outcomes and yields improvements in both reconstruction and rendering quality in sparse scenes. This makes the proposed Re-Nerfing a solid and flexible complement to improve the output quality of any learned NVS pipeline.

References

[1] Young Chun Ahn, Seokhwan Jang, Sungheon Park, Ji-Yeon Kim, and Nahyup Kang. PANeRF: Pseudo-view augmentation for improved neural radiance fields based on few-shot inputs. *arXiv preprint arXiv:2211.12758*, 2022. 2

[2] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1, 2, 7

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 1, 2, 6, 8, 3, 5, 7, 10, 11, 12

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 6

[5] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-NeRF: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 1, 2

[6] Matteo Bortolon, Alessio Del Bue, and Fabio Poiesi. VM-NeRF: Tackling sparsity in NeRF with view morphing. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 63–74. Springer, 2023. 2

[7] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 6

[8] Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*. Springer, 2022. 2

[10] Di Chen, Yu Liu, Lianghua Huang, Bin Wang, and Pan Pan. GeoAug: Data augmentation for few-shot nerf with geometry constraints. In *Proceedings of the European Conference on Computer Vision*, pages 322–337. Springer, 2022. 2

[11] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 695–714. Springer, 2020. 2, 7

[12] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-NeRF: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15191–15202, 2022. 2

[13] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16569–16578, 2023. 1

[14] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7781–7790, 2019. 2

[15] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Transactions on Information Theory*, 13(1):21–27, 1967. 4

[16] 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. 1, 2, 4

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 1, 2

[18] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. InstantSplat: Unbounded sparse-view pose-free Gaussian Splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2024. 2

[19] Casimir Feldmann, Niall Siegenheim, Nikolas Hars, Lovro Rabuzin, Mert Ertugrul, Luca Wolfart, Marc Pollefeys, Zuria Bauer, and Martin R Oswald. NeRF-mentation: NeRF-based augmentation for monocular depth estimation. *arXiv preprint arXiv:2401.03771*, 2024. 2

[20] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 5515–5524, 2016. 2

[21] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[22] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2

[23] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2

[24] Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari. Robust monocular depth estimation under challenging conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8177–8186, 2023. 2, 7

[25] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021. 2

[26] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification for Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20061–20070, 2024. 5, 6, 2

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[28] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. CoNeRF: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022. 2

[29] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1, 2, 3, 6, 7, 8, 4, 5, 10, 11, 12

[30] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78, 2017. 2, 6, 7, 8, 1

[31] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2

[32] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Nassir Navab, Benjamin Busam, and Federico Tombari. 3D adversarial augmentations for robust out-of-domain predictions. *International Journal of Computer Vision*, pages 1–33, 2023. 2

[33] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, page 31–42. Association for Computing Machinery, 1996. 2

[34] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024. 2

[35] 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. 1, 2

[36] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 1

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[38] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38 (4):1–14, 2019. 1, 2, 6, 7, 5, 8, 9

[39] 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. In *Proceedings of the Euro-*

pean Conference on Computer Vision, pages 405–421. Springer, 2020. 1, 2, 3

[40] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 2, 6, 8, 3, 4, 5, 7, 10, 11, 12

[41] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-context data augmentation for 3D scenes. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 116–125. IEEE, 2021. 2

[42] 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. 1, 2, 4, 6, 7, 5, 8, 9

[43] Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. CoherentGS: Sparse novel view synthesis with coherent 3d gaussians. In *Proceedings of the European Conference on Computer Vision*, pages 19–37. Springer, 2024. 2

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[45] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[46] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (ToG)*, 40(6), 2021. 2

[47] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T Barron, and Ricardo Martin-Brualla. CamP: Camera preconditioning for neural radiance fields. *ACM Transactions on Graphics (TOG)*, 42(6): 1–11, 2023. 2

[48] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[49] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022. 2

[50] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 2

[51] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 1, 2

[52] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12716–12725, 2019. 1, 2

[53] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 2, 1

[54] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 245–254, 1985. 4

[55] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024. 5

[56] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[57] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2

[58] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2

[59] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen,

Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 6, 1

[60] Fabio Tosi, Alessio Tonioni, Daniele De Gregorio, and Matteo Poggi. Nerf-supervised deep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 855–866, 2023. 2

[61] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. SPARF: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023. 1, 2, 4

[62] Haithem Turki, Michael Zollhöfer, Christian Richardt, and Deva Ramanan. PyNeRF: Pyramidal neural radiance fields. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6, 1, 3, 4, 5, 7, 10, 11, 12

[63] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[64] 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. 2

[65] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[67] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Proceedings of the European Conference on Computer Vision*, pages 286–301. Springer, 2016. 2

[68] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2

[69] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In *Proceedings of the European Conference on Computer Vision*, pages 145–163. Springer, 2024. 1, 2