StructGS: Adaptive Spherical Harmonics and Rendering Enhancements for Superior 3D Gaussian Splatting

Zexu Huang[a], Min Xu, IEEE Member[a], Stuart Perry[a]

[a]*School of Electrical and Data Engineering, The University of Technology Sydney*

Abstract

Recent advancements in 3D reconstruction coupled with neural rendering techniques have greatly improved the creation of photo-realistic 3D scenes, influencing both academic research and industry applications. The technique of 3D Gaussian Splatting and its variants incorporate the strengths of both primitive-based and volumetric representations, achieving superior rendering quality. While 3D Geometric Scattering (3DGS) and its variants have advanced the field of 3D representation, they fall short in capturing the stochastic properties of non-local structural information during the training process. Additionally, the initialisation of spherical functions in 3DGS-based methods often fails to engage higher-order terms in early training rounds, leading to unnecessary computational overhead as training progresses. Furthermore, current 3DGS-based approaches require training on higher resolution images to render higher resolution outputs, significantly increasing memory demands and prolonging training durations. We introduce StructGS, a framework that enhances 3D Gaussian Splatting (3DGS) for improved novel-view synthesis in 3D reconstruction. StructGS innovatively incorporates a patch-based SSIM loss, dynamic spherical harmonics initialisation and a Multi-scale Residual Network (MSRN) to address the above-mentioned limitations, respectively. Our framework significantly reduces computational redundancy, enhances detail capture and supports high-resolution rendering from low-resolution inputs. Experimentally, StructGS demonstrates superior performance over state-of-the-art (SOTA) models, achieving higher quality and more detailed renderings with fewer artifacts. (The link to the code will be made available after publication)

Keywords: 3D Gaussian Splatting, Neural Rendering, 3D Reconstruction, Novel View Synthesis

1. Introduction

Recent advancements in 3D reconstruction have enhanced novel-view synthesis, allowing for the creation of photorealistic representations of volumetric scenes. Neural Radiance Fields (NeRF) [1] represents a significant milestone in 3D rendering which employs a multi-layer perceptron (MLP) to produce high-quality, geometrically coherent images from novel viewpoints. However, this method incurs the drawback of time-consuming stochastic sampling, which can lead to slower performance and potential noise artefacts. Its variants primarily enhance rendering quality [2, 3] and accelerate the convergence speed [4, 5].

Due to limitations in rendering quality and training speeds of NeRF-based models, recent breakthroughs in 3D Gaussian Splatting (3DGS) [6] have significantly improved these aspects. This method fine-tunes a set of 3D Gaussians, initially shaped using Structure-from-Motion (SfM) [7] to effectively model scenes with volumetric continuity. This enables faster rasterization through projection onto 2D planes. Nonetheless, when camera angles deviate from the original training configurations or during close-up views, 3DGS often produces visual distortions due to inadequate detail resolution. To

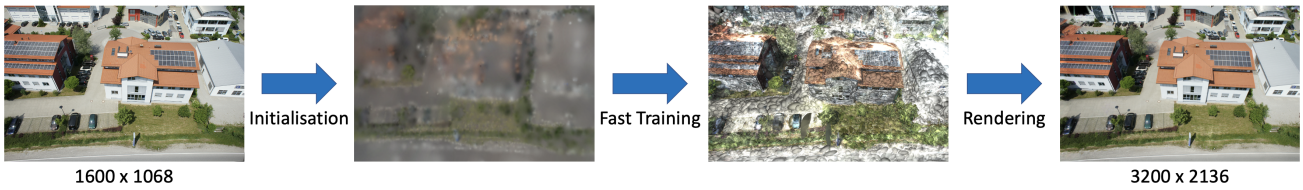

Figure 1: StructGS: We present a framework that produces high-quality renderings from a set of single-view camera images. Following scene reconstruction, our approach allows for rapid rendering at resolutions higher than that of the input image. To achieve this, we leveraged a patch SSIM loss and a total variation (TV) loss regularizer, which effectively capture nonlocal structural information and enhance image smoothness. Additionally, we proposed a dynamic adjustment strategy for spherical harmonics based on the opacity and distance of Gaussian spheres. We also integrated a pre-trained Multi-scale Residual Network to facilitate super-resolution rendering.

overcome these limitations, recent 3DGS variants [8, 9] have introduced a 3D smoothing filter to control extreme frequency variations and employ anchor points to set up 3D Gaussians. These enhancements significantly improve visual accuracy and versatility in a range of applications. Despite significant advancements in 3DGS, current models typically assess training performance by simply comparing rendered images against ground truth using individual-pixel Structural Similarity (SSIM). However, this lacks a comprehensive assessment of structural similarities, potentially overlooking subtle yet critical topological discrepancies. Additionally, the initialisation of spherical harmonics in 3DGS-based models typically involves setting the zeroth-degree coefficient (dc term) to the RGB colour of the ground truth while all remaining terms are initialised to zero. As shown in Tab. 1, such a method may lead to higher dimensions uninitialised in the early iterations and calculation redundancies and biases in higher-order spherical harmonics which compromises the model's efficiency. Furthermore, current 3DGS-based methods that render higher resolution images necessitate training with higher resolution images, which not only increases memory demands but also significantly increases the computation complexity.

To address these challenges, we introduce StructGS, a framework designed to enhance rendering quality, reduce redundancy and bias in higher-order spherical harmonics and achieve higher-resolution images. Following the NeRF method [10], StructGS uses a patch SSIM loss to capture nonlocal structural similarity from stochastically sampled pixels, combined with a TV loss regulariser for smoothness. To refine spherical harmonics initialisation and optimisation, we design a dynamic adjustment strategy that considers the transparency of Gaussian spheres. The first three RGB dimensions of the spherical harmonics are initialised and optimised based on the transparency levels of each sphere. Additionally, the remaining harmonics are adjusted dynamically using distance information: higher-order harmonics are used for points further from neighbours to capture finer details, while closer points use lower-order harmonics for broader features. During the rendering process, StructGS incorporates a pre-trained Multi-scale Residual Network [11] to perform super-resolution on the rendered images, which enables the model to train on low-resolution input images and then produce high-resolution, high-quality rendered images.

Through rigorous experiments, StructGS has demonstrated superior performance over current state-of-the-art (SOTA) 3DGS-based and previous 3D reconstruction models. It effectively reduces the calculation redundancy of higher-order spherical harmonics, achieving better quality with fewer training iterations. Moreover, the designed framework enhanced by loss functions and a pre-trained Multi-Scale Residual Network (MSRN) [11] enables StructGS to render enhanced quality and higher resolution images. (see Fig. 1)

Variance of Spherical Harmonics Terms				
Scene	degree 0	degree 1	degree 2	degree 3
bicycle (1000 iteration)	1.321	0.0	0.0	0.0
bicycle (2000 iteration)	1.624	1.472×10^{-4}	0.0	0.0
bicycle (3000 iteration)	1.501	3.592×10^{-4}	9.451×10^{-5}	0.0
bonsai (1000 iteration)	0.712	0.0	0.0	0.0
bonsai (2000 iteration)	0.860	1.217×10^{-4}	0.0	0.0
bonsai (3000 iteration)	0.894	2.998×10^{-4}	1.020×10^{-4}	0.0
villa (1000 iteration)	1.442	0.0	0.0	0.0
villa (2000 iteration)	1.477	4.895×10^{-4}	0.0	0.0
villa (3000 iteration)	1.440	0.001	3.958×10^{-4}	0.0

Table 1: Variance of spherical harmonics terms of different degrees for two scenes (bicycle, bonsai) from Mip-NeRF 360 dataset [3] and one villa scene at various training iterations of the original 3D Gaussian Splatting [6]. The results show that many higher-degree spherical harmonics terms are not initialised at early training iterations. Addressing this issue can improve the model's performance during the early stages of training.

Contributions

In summary, the contributions of this work are as follows:

1. Different from existing methods only considering individual pixel-based structural similarity, We leveraged a patch SSIM loss and a TV loss regularizer to effectively capture nonlocal structural information and enhance image smoothness. This approach allows for a more refined representation of structural details in 3D reconstructions.
2. We proposed a dynamic adjustment strategy for spherical harmonics based on the opacity and distance of Gaussian spheres from each other. This method optimises the initialisation and adjustment of spherical harmonics, reducing redundancy and bias in higher-order components, and improving the quality of training within fewer iterations.
3. We incorporated a pre-trained Multi-scale Residual Network for super-resolution, StructGS is capable of producing high-resolution, high-quality images from low-resolution inputs. This advancement enhances the framework's ability to handle detailed textures and complex geometries.

The organization of this paper is as follows: Section 2 provides a review of prior research pertinent to our study. Section 3 describes the methods utilized in our research. Section 4 details our experimental setup, describes our model implementations, compares the performance of our model with other advanced 3D reconstruction methods and discusses the results of ablation studies as well as the limitations of our approach. The paper is concluded in Section 5, where we summarize the principal contributions of our work.

2. Related Work

Novel View Synthesis Image-based rendering methodologies have traditionally been employed in the field of novel view synthesis for real-world scenes. Techniques such as Structure-from-Motion (SfM) [7] play a pivotal role by facilitating the estimation of camera parameters using a series of images. These parameters are crucial for accurately projecting the colours of input images onto a new viewpoint [12]. This method fundamentally depends on the construction of approximate geometry which typically involves creating mesh models or point clouds. To enhance these models, refinements are often

introduced by employing Multi-View Stereo (MVS) processes [13, 14]. Although these methods are effective, real-world data frequently exhibit issues such as inaccuracies in camera calibration and geometric errors [15]. These problems result in undesirable outcomes during the rendering process, including artifacts at object borders and diminished sharpness of details. To address these challenges, recent advancements have incorporated neural rendering techniques [16], which significantly mitigate such artifacts and improve the fidelity of the synthesized views. Some approaches use rendering techniques like viewpoint interpolation [17] or depth sensors to capture scene geometry directly [18]. However, these methods are typically limited to indoor environments. This restriction is due to the sensor's range constraints and their primary focus on adapting 2D content.

MLP-based Radiance Fields Initial approaches in neural fields have employed multi-layer perceptrons (MLPs) as the primary mechanism for approximating the geometry and visual characteristics of 3D scenes. In these systems, spatial coordinates and viewing directions are fed into the MLP. The MLP then computes attributes such as the signed distance from the scene surface (SDF) [19, 20, 21] or determines the colour and density at a specific coordinate [1, 2, 3]. While MLP-based methods leverage their volumetric capabilities to achieve state-of-the-art results in novel view synthesis, they face significant challenges. The primary issue is the necessity for the MLP to process a vast number of point samples along each camera ray, which drastically slows down rendering and limits the ability to handle large and complex scenes effectively. Recent research efforts have aimed to enhance the efficiency and broaden the scalability of Neural Radiance Fields (NeRF). These advancements [4, 22, 5, 23, 24] have involved the adoption of discrete or sparse volumetric structures, such as voxel grids and hash tables. These structures are essential because they incorporate learnable features that function as positional encodings for three-dimensional coordinates. Moreover, these approaches [3, 22, 25] apply hierarchical sampling strategies and they utilize techniques for low-rank approximations. However, the ongoing dependence on volumetric ray marching poses compatibility issues with standard graphics platforms, which are traditionally optimized for polygonal rendering.

Point-based Radiance Fields In point-based radiance field rendering, using point clouds as explicit proxy representations offers significant advantages. These point clouds are efficiently captured using technologies like LiDAR [26] and Structure-from-Motion/Multi-View Stereo methods [14]. Comprising an unstructured assortment of spatial samples with varied neighbour distances, point clouds faithfully represent the original data captured. The process of rendering these point clouds is generally rapid [27]. Enhancements using neural descriptors [28] or specially optimized attributes [29] allow for high-quality visual outputs through differentiable point renderers [30, 31]. Nevertheless, the discrete rasterization process can lead to visual artifacts like aliasing or overdrawing, particularly when multiple points converge on the same pixel. Recently, the 3DGS framework [6] innovatively applies directionally dependent 3D Gaussians for rendering three-dimensional scenes. This approach leverages Structure from Motion (SfM) [7] to set up the initial 3D Gaussian structures which are then refined and optimized as volumetric models. This approach combines anisotropic Gaussians with a high-speed tiled renderer, optimizing the sizes of splats through gradient descent. However, it is crucial to limit the number of Gaussians to prevent performance degradation. This constraint may result in the over-blurring of small, detailed elements.

Recent 3DGS variants [8, 9] have implemented a 3D smoothing filter to manage abrupt frequency changes and utilize

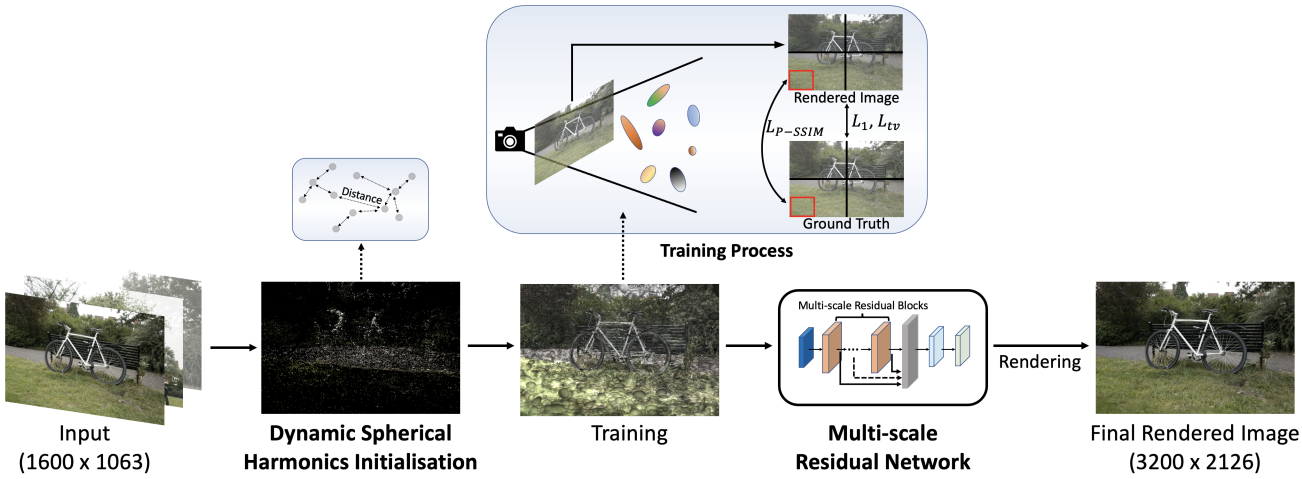

Figure 2: **Overview of StructGS:** In the initialisation phase, our model employs a **dynamic adjustment of spherical harmonics** based on opacity weighting to optimise the first three RGB dimensions for each Gaussian sphere. Distance information further refines initialisation, with higher-order harmonics capturing more details for distant points and lower-order for nearer points. During the training phase, the rendered and ground truth images are divided into several patches. Within these patches, the SSIM loss (L_{SSIM}) for small areas is calculated using a kernel. The results are then summed and averaged. The rendered and ground truth images are also assessed for total variation loss (L_{tv}) and L1 loss (L_1). After training, our model incorporates a pre-trained **Multi-scale Residual Network** to render high-quality and high-resolution images.

anchor points for establishing 3D Gaussians. They [8, 9] successfully eliminated some artifacts at high frequencies in 3DGS and also enhanced the rendering quality. Despite significant progress in 3DGS, current models [6, 8, 9] often evaluate training performance by simply comparing rendered images with ground truth using SSIM at the pixel level, potentially missing subtle yet crucial topological differences and often omitting important structural information. Moreover, initializing spherical harmonics in 3DGS models typically sets the DC term to the ground truth's RGB color, with all other terms set to zero, leading to redundancies and biases in higher-order harmonics that reduce model efficiency. Furthermore, 3DGS methods rendering higher resolution images require training on higher resolution images, which increases memory demands and significantly prolongs training time. As shown in Fig. 2, our method addresses these issues by incorporating non-local structural information and a total variation (TV) regulariser term during the loss calculation. Additionally, we dynamically adjust the initialisation of spherical harmonics to reduce computational redundancies. Furthermore, we integrate a pre-trained Multi-scale Residual Network to enhance the rendering quality and resolution, thus avoiding the increased training time and computational costs previously required for training on high-resolution images.

3. Methodology

3.1. Preliminary

Previous research [6] introduced a method to represent 3D scenes using a group of scaled 3D Gaussian $\{G_n | n = 1, \ldots, N\}$ and render images via splatting. Each Gaussian primitive G_n is characterized by an opacity $\alpha_n \in [0, 1]$, a center position $\mathbf{q}_n \in \mathbb{R}^{3\times1}$, and a covariance matrix $\Sigma_n \in \mathbb{R}^{3\times3}$ defined in world space. The form of each scaled Gaussian function is given by:

$$G_n(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{q}_n)^T \Sigma_n^{-1}(\mathbf{x}-\mathbf{q}_n)}. \tag{1}$$

To ensure that Σ_n remains a valid covariance matrix, it is expressed using a positive semi-definite parameterization: $\Sigma_n = \mathbf{U}_n \mathbf{d}_n \mathbf{d}_n^T \mathbf{U}_n^T$. In this formulation, $\mathbf{d} \in \mathbb{R}^3$ represents a scaling vector while $\mathbf{U} \in \mathbb{R}^{3\times3}$ is a rotation matrix defined by a quaternion for orientation [6]. Rendering an image from a given viewpoint requires transforming these 3D Gaussians $\{G_n\}$ into camera space, defined by a rotation matrix $\mathbf{C} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. The centre and covariance matrix are transformed as follows:

$$\mathbf{q}'_n = \mathbf{C}\mathbf{q}_n + \mathbf{t}, \quad \Sigma'_n = \mathbf{C}\Sigma_n\mathbf{C}^T. \tag{2}$$

Then, a local transformation projects these transformed Gaussians into ray space using a Jacobian matrix J_n, which approximates the projective transformation at the Gaussian centre $\mathbf{q}'_n$. This transformation results in an updated covariance matrix:

$$\Sigma''_n = J_n \Sigma'_n J_n^T. \tag{3}$$

By eliminating the third row and column of Σ''_n, we obtain a 2D covariance matrix Σ_n^{2D}, which enables the efficient modelling of 2D Gaussians, denoted as G_n^{2D}. Eventually, 3D Gaussian Splatting (3DGS) [6] uses spherical harmonics to model view-dependent colour c_n, and renders the image through alpha blending based on the Gaussian's depth order (1 to N):

$$c(\mathbf{x}) = \sum_{n=1}^{N} c_n \alpha_n G_n^{2D}(\mathbf{x}) \prod_{j=1}^{n-1} (1 - \alpha_j G_j^{2D}(\mathbf{x})), \tag{4}$$

where:

$$G_n^{2D}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{q}_n)^T (\Sigma_n^{2D} + h\mathbf{I})^{-1} (\mathbf{x}-\mathbf{q}_n)}. \tag{5}$$

The symbol $\mathbf{I}$ denotes a 2D identity matrix and h serves as a scalar hyperparameter for dilation.

3.2. Non-Local Structural Information in SSIM

As shown in Fig. 2, our model undergoes a specific iteration procedure during the training phase to compute the Structural Similarity Index (SSIM) over stochastic patch pairs derived from both the rendered image R_i and the Ground Truth image G_i. Each image is segmented into $p \times p$ patches after k iterations. For each patch, a rendered patch is formed from a random selection of pixels within the patch and SSIM is computed for these stochastic patch pairs using a $K \times K$ kernel according to [10]. The stride size for this operation is denoted as s. The SSIM is computed for each patch pair, and the process continues until partial SSIM values are calculated for the entire image. The summation of partial SSIM values is then averaged to obtain the final Patch-based SSIM (P-SSIM). Mathematically, the P-SSIM can be expressed as:

$$\text{P-SSIM}(r_i, g_i) = \frac{1}{P} \sum_{i=1}^{P} \text{SSIM}(r_i, g_i), \tag{6}$$

where P represents the number of stochastic patch pairs assessed, and r_i and g_i represent the individual patches of the rendered and ground truth images, respectively. Given that SSIM values range between $[-1, 1]$, we define our loss based on

P-SSIM as:

$$L_{\text{P-SSIM}} = 1 - \text{P-SSIM}(r_i, g_i) = 1 - \frac{1}{P}\sum_{i=1}^{P} \text{SSIM}(r_i, g_i). \tag{7}$$

Following the previous work [10], we set $K = 4$ and the stride $S = K$ without further fine-tuning. The patch size P is experimentally determined to be 10. Through the experiment, we set $k = 25000$ in our experiment.

While the original SSIM is differentiable and allows for gradient-based optimisation, direct optimisation of SSIM can lead to suboptimal results. Unlike previous 3DGS [6] approaches that used SSIM directly with local patches, our method involves stochastic patch pairs that capture non-local structural information across the image which can produce superior results. This is demonstrated through our ablation studies which confirm that optimising P-SSIM via stochastic patches significantly outperforms conventional local patch-based SSIM optimisation. This approach captures relationships between distant pixels, thus providing a method for enhancing the quality of rendered images.

3.3. Dynamic Adjustment of Spherical Harmonics Initialisation

The conventional initialisation of spherical harmonics (SH) in 3D Gaussian Splatting typically sets the zeroth-degree coefficient (dc term) of the SH to the initial RGB values of the point cloud which leaves higher dimensions uninitialised in the early iterations. This approach often results in unnecessary computational overhead to optimise higher-order spherical harmonics dimensions. To address these inefficiencies and biases, we propose a method for dynamically adjusting the initialisation of spherical harmonics based on the opacity and distance metrics of the points within the point cloud data. Initially, the spherical harmonics coefficients are calculated based on the RGB colour data of the point cloud. The first three SH coefficients are initialised using the RGB values weighted by the opacity α_n of each point which allows these primary coefficients to encapsulate the fundamental colour information modified by the transparency of each point:

$$\text{SH}_{\text{RGB}}(0) = \text{RGB} \times \alpha_n. \tag{8}$$

Here, α_n is the opacity derived from the inverse sigmoid function. This design modifies the transparency based on the point's attributes, thereby tailoring the primary colour information embedded in the spherical harmonics.

The distance metric from each point to its neighbours significantly influences the level of detail needed in the spherical harmonics representation. Points that are closer to their neighbours are initialised with lower-degree SH terms, as the fine details are less discernible at small distances. Conversely, points that are farther away are initialised with higher-degree SH terms to capture more detail. The coefficients ν of the SH can be expressed as:

$$\text{D} = \max(1, \min(d, M)), \quad \nu = (\text{D} + 1)^2, \tag{9}$$

where d is the scaled Euclidean distance between points and M is a parameter that represents the upper bond degree of the spherical harmonics. Through our experiment, we set M as 5. The degree D of spherical harmonics is dynamically set to ensure detailed capture for distant points and less complexity for closer points. The spherical harmonics are dynamically adjusted based on both the opacity and the calculated distance. The method combines these attributes to initialise the

non-zero coefficients of SH for higher-order terms. For each point, its corresponding spherical harmonics coefficients beyond the first are initialised as:

$$\mathrm{SH}_{\mathrm{RGB}}(v) = \mathrm{RGB} \times \alpha_n \times (1 - e^{-sd}), \tag{10}$$

where n is the index for spherical harmonics coefficients, s is a scaling factor adjusting the impact of distance on higher-order terms, and e^{-sd} modulates the influence of distance on these terms. This dynamic initialisation approach seeks to optimise computational resources and accelerate the training process. Results demonstrate enhanced performance with fewer iterations required in training.

3.4. Multi-scale Residual Network

Current methods based on 3DGS-based methods [6, 8, 9] that render higher resolution images require training with higher resolution images. This not only increases memory usage but also significantly prolongs the training time. Our model addressed the limitation by employing a pre-trained Multi-scale Residual Network (MSRN) [11] F_θ to fit into the rendering process in 3DGS. It processes low-resolution (LR) pixel represented as $p^{(\mathrm{LR})}$ to generate super-resolution (SR) pixel denoted as $p^{(\mathrm{SR})}$. The input image undergoes a feature extraction process where the MSRN utilises K multi-scale residual blocks (MSRBs) to capture intrinsic patterns at various scales and depths. The LR input is fed directly into the network, avoiding initial upsampling, thus preserving the original low-resolution data for the learning process.

Multi-scale Residual Block (MSRB) To effectively identify features across multiple scales, the MSRB integrates two critical components: multi-scale feature fusion and local residual learning. In the process of multi-scale feature fusion, it leverages a dual-pathway architecture. Each pathway utilises convolutional kernels of different sizes, allowing for the interplay and fusion of feature maps at varying granularities. During the local residual learning procedure, each MSRB facilitates a shortcut connection that performs element-wise addition to blend the input and output. It enhances the flow of gradients and reduces the vanishing gradient problem which can be expressed as:

$$M_k = S + M_{k-1}, \tag{11}$$

where M_k and M_{k-1} are the input and output of the Kth MSRB and S is the output of the blocks inside the MSRB.

Hierarchical Feature Fusion Structure (HFFS) This structure aims to preserve and enhance the features extracted across the MSRBs. As the network depth increases, it is imperative to maintain the fidelity of the input characteristics throughout the network. This process's output F_{HFFS} can be formulated as:

$$F_{HFFS} = \mathbf{W} * [M_0, M_1, \ldots, M_K] + b. \tag{12}$$

Here, M_0 represents the initial feature maps produced by the first convolutional layer, with subsequent $M_i (i \neq 0)$ denoting the outputs from each respective MSRB. $\mathbf{W}$ and b are the weights and bias respectively.

MSRN Image Reconstruction Stage In the final stage, the image reconstruction module transforms the feature-enriched data back into high-resolution output using advanced upscaling techniques without initial upsampling. This approach

integrates the PixelShuffle [32] method to efficiently upscale the image by directly rearranging the output tensor to a higher resolution. This technique significantly minimises the introduction of redundant information typically associated with traditional upscaling methods like bi-cubic interpolation.

In the rendering stage, we employ the MSRN to render super-resolution images. The network F_θ encapsulates the entire transformation process from $p^{(\mathrm{LR})}$ to $p^{(\mathrm{SR})}$ can be formulated as:

$$p^{(SR)} = F_\theta(p^{(LR)}). \tag{13}$$

3.5. Training Losses

Following 3DGS [6], the loss function incorporates both L1 and D-SSIM terms. Also, our loss function contains our P-SSIM loss which is mentioned above and a total variant regulariser. The L1 term quantifies the absolute discrepancies between predictions and actual targets. The L_1 loss for the rendered colours is mathematically formulated as:

$$L_1 = \frac{1}{B}\sum_{i=1}^{B}\sum_{j}|p_{i,j} - \hat{p}_{i,j}|, \tag{14}$$

where $p_{i,j}$ and $\hat{p}_{i,j}$ are the pixels of the ground truth image G_i and the rendered image R_i, and B represents the total number of pixels in the image. We also employ the Structural Similarity Index Measure loss and it is a loss term specifically designed to significantly enhance the perceptual quality of digital images and videos. By focusing on the structural discrepancies between target and rendered images, this metric provides a superior and more perceptually relevant evaluation compared to traditional pixel-based metrics. The D-SSIM loss is defined as:

$$L_{\text{D-SSIM}} = 1 - \text{SSIM}(p_{i,j}, \hat{p}_{i,j}), \tag{15}$$

where SSIM [33] denotes the structural similarity between the target image and the rendered image.

To enhance image sharpness and mitigate excessive smoothness in the rendered image, our framework incorporates total variation (TV) regularisation. This regularisation technique is instrumental in preserving high-frequency details while maintaining overall image smoothness. The total variation loss is calculated by measuring the sum of absolute differences between adjacent pixels $\hat{p}$ in both horizontal and vertical directions in the rendered image. This approach emphasises maintaining edge sharpness by penalising large gradients in the image:

$$L_{tv} = \frac{\sum_{i,j}\left(|\hat{p}_{i,j} - \hat{p}_{i+1,j}| + |\hat{p}_{i,j} - \hat{p}_{i,j+1}|\right)}{\hat{c}\cdot h\cdot w}. \tag{16}$$

Here, $\hat{p}_{i,j}$ represents the pixel intensity at position (i, j) in the rendered image R_i. The normalisation by $\hat{c}\cdot h\cdot w$ (where $\hat{c}$, h, and w are the number of channels, height, and width of the image tensor, respectively) ensures that the loss calculation is scale-invariant with respect to the size of the image. The StructGS total loss function amalgamates the individual loss

	Mip-NeRF 360			Tanks&Temples			Deep Blending		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Instant-NGP	26.43	0.725	0.339	21.72	0.723	0.330	23.62	0.797	0.423
Plenoxels	23.62	0.670	0.443	21.08	0.719	0.379	23.06	0.795	0.510
Mip-NeRF 360	29.23	0.844	0.207	22.22	0.759	0.257	29.40	0.901	0.245
3DGS	28.69	0.870	0.182	23.14	0.841	0.183	29.41	0.903	0.243
Mip-Splatting	29.09	0.871	0.184	23.87	0.851	0.176	29.70	0.905	0.243
Scaffold-GS	28.84	0.848	0.220	23.96	0.853	0.177	30.21	0.906	0.254
Ours (without MSRN)	29.21	0.881	0.155	23.83	0.863	0.141	30.19	0.905	0.244
Ours (Full)	**30.69**	**0.928**	**0.036**	**24.55**	**0.893**	**0.051**	**30.28**	**0.908**	**0.079**

Table 2: **Quantitative Comparison Results on the Mip-NeRF 360 [3], Tanks&Temples [34] and DeepBlending [35] Datasets.** Our method outperforms baselines and SOTA methods [9, 8]. The competing metrics are sourced from the respective papers (except for Mip-Splatting).

elements into a weighted sum which enhances the quality of reconstruction across diverse dimensions:

$$L_{\text{Total}} = \begin{cases} (1-\lambda)L_1 + \lambda L_{\text{D-SSIM}} + \beta L_{tv}, & \text{iterations} \leq k, \\ (1-\lambda)L_1 + \lambda L_{\text{P-SSIM}} + \gamma L_{tv}, & \text{otherwise}, \end{cases} \tag{17}$$

where k denotes the iteration where we start to employ the P-SSIM loss. The λ and β are the respective weights assigned to the loss components. Through experiments, we set $k = 25000$, λ as 0.5, β as 0.04 and γ as 0.02.

4. Experiments

4.1. Dataset and Metrics

In our experimental setup, we assessed the performance of our model using 18 scenes sourced from various publicly available datasets. Specifically, we evaluated data tested in Scaffold-GS [9], Mip-Splatting [8] and 3DGS [6] because they are our primary baselines. This included seven scenes from Mip-NeRF 360 [3], two scenes each from Tanks & Temples [34] and DeepBlending [35], six scenes from BungeeNeRF [36] and one scene labelled "Villa". The "Villa" scene is a self-collected dataset. These scenes were carefully selected to encapsulate a range of contents captured at multiple levels of detail (LODs), showcasing our model's capabilities in view-adaptive rendering. They also span a diverse array of environments, covering both indoor and outdoor settings and incorporating large-scale scenes to demonstrate the scalability and versatility of our approach.

To quantitatively evaluate our model, we employed three widely-used metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [33], and Learned Perceptual Image Patch Similarity (LPIPS) [37]. Additionally, we explored the model's performance through progressive training across these complex 3D scenarios, allowing us to further validate and refine our approach based on detailed feedback and metric scores.

4.2. Baselines and Implementation

For our experimental benchmarking, we selected Scaffold-GS [9], Mip-Splatting [8] and 3DGS [6] as our primary baselines due to their state-of-the-art (SOTA) performance in 3D reconstruction and novel view synthesis. To ensure a fair

	BungeeNeRF			Villa		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	24.89	0.827	0.211	25.40	0.862	0.160
Mip-Splatting	28.25	0.919	0.097	25.74	0.869	0.174
Scaffold-GS	27.03	0.899	0.101	25.81	0.869	0.164
Ours (without MSRN)	27.99	0.917	0.091	25.41	0.870	0.129
Ours (Full)	**31.01**	**0.960**	**0.025**	**25.99**	**0.912**	**0.043**

Table 3: **Quantitative Comparison Results on the BungeeNeRF [36] and Villa Datasets.** Our method outperforms baselines and SOTA methods [9, 8].

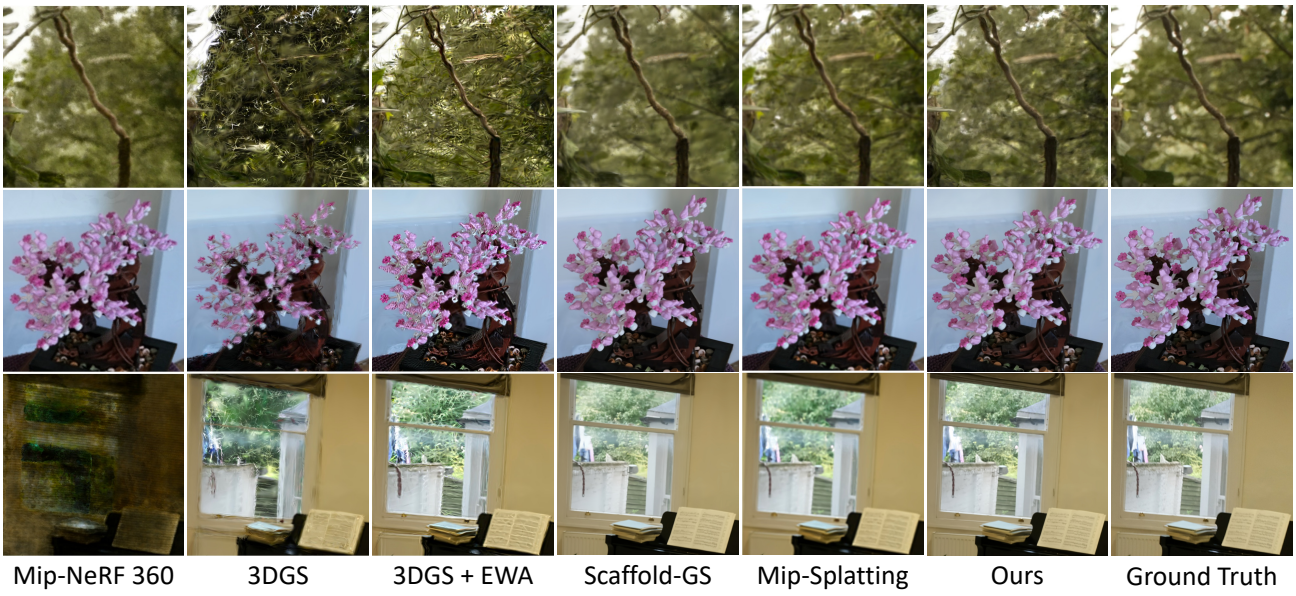

Figure 3: **Qualitative Comparison Results on the Mip-NeRF 360 Dataset [3].** These models were trained on images with a resolution of 1.6k and we simulated the zoom-in situation. Unlike previous approaches, our model attains a higher degree of accuracy and detail than other models, rendering images that closely match the ground truth.

comparison, we trained our model along with all selected baselines for 30k iterations. Additionally, we included results from Mip-NeRF 360 [3], Instant-NGP [4], Plenoxels [24] and 3DGS enhanced with EWA [38] to provide a comprehensive overview of current capabilities in the field.

In our implementation, we set $P = 10$, where P represents the number of stochastic patch pairs assessed during training, and $k = 25,000$ to control the transition to Patch-based Structural Similarity Index Measure (P-SSIM). We structured the training process around three primary loss weights: $\lambda = 0.5, \beta = 0.04$, and $\gamma = 0.02$. The reason why we select λ as 0.5 is because we aim to increase the importance of P-SSIM. The selection of weights for TV loss is motivated by a desire to enhance the effect of TV Loss for the initial k iterations, to suppress noise and irregularities in the model more effectively. Following this phase, we aim to emphasise the importance of P-SSIM by reducing the weight of TV loss. This reduction is also intended to preserve some fine structures and details. Furthermore, our refinement process includes a pruning step where an anchor is eliminated if the accumulated opacity of its Gaussians falls below 0.005, ensuring that only significant features contribute to the final model structure.

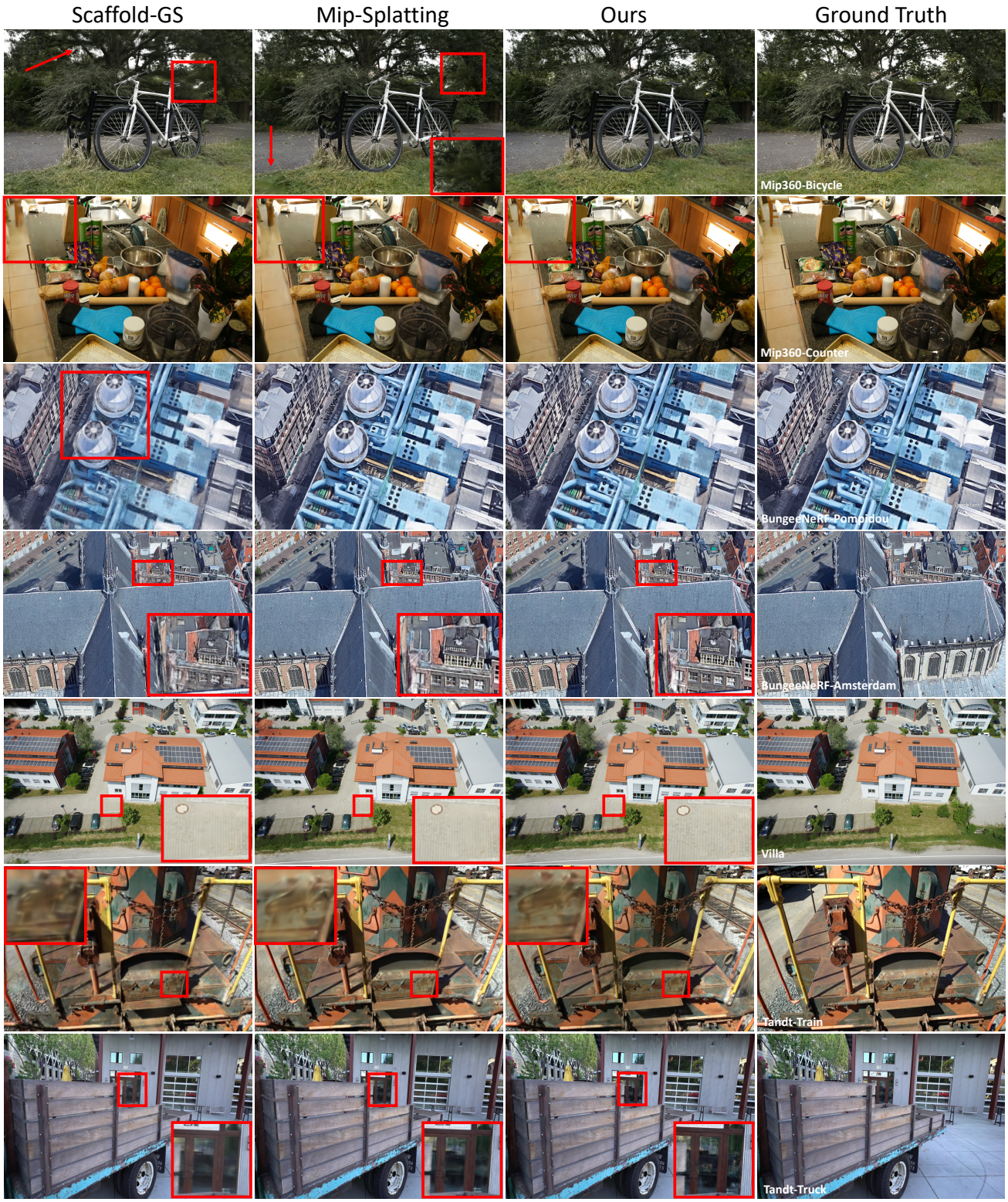

Figure 4: **Qualitative Comparison Results across diverse datasets [3, 36, 34].** The red boxes and arrows highlight artifacts rendered by state-of-the-art models [8, 9], which cause certain areas of the rendered images to appear blurry. The results above demonstrate that our model outperforms state-of-the-art models, achieving superior performance in rendering details with significantly fewer artifacts.

	Bicycle			Counter			Kitchen			Tanks&Temples		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
None	25.13	0.747	0.245	29.13	0.915	0.183	31.47	0.932	0.116	23.33	0.845	0.179
w/ L_{P-SSIM}	25.64	0.785	0.178	29.40	0.920	0.161	31.90	0.937	0.106	23.79	0.855	0.151
w/ L_{tv}	25.18	0.750	0.229	29.21	0.916	0.177	31.55	0.933	0.110	23.48	0.850	0.163
w/ MSRN	27.51	0.832	0.091	29.66	0.927	0.087	32.11	0.954	0.044	24.11	0.878	0.099
Full	**28.07**	**0.884**	**0.047**	**30.13**	**0.939**	**0.045**	**33.37**	**0.965**	**0.022**	**24.55**	**0.893**	**0.051**

Table 4: **Ablation Study of Components in Our Model.** We present quantitative results for the Bicycle, Counter and Kitchen scenes from the Mip-NeRF 360 dataset [3] and Tanks&Temples dataset [34]. All scenes are trained for 30k iterations and the input image is at a resolution of 1.6k.

4.3. Results Comparison

In evaluating the efficacy of our method, we conducted comparisons against several baselines including Scaffold-GS, Mip-Splatting, 3D-GS, Mip-NeRF360, Instant-NGP, and Plenoxels using real-world datasets. Qualitative results are presented in Tab. 2 and 3. The performance metrics for Mip-NeRF 360 [3], Instant-NGP [4] and Plenoxels [24] are consistent with those presented in the previous study [6]. We conducted the training for Scaffold-GS [9] and Mip-Splatting [8] using default settings provided by their respective implementations. Notably, our model outperformed all other models, including state-of-the-art (SOTA) methods [9, 8], across all dataset scenes. Despite the removal of the Multi-scale Residual Network (MSRN) from our model, it still outperformed SOTA 3D-GS-based methods [6, 9, 8] in the Mip-NeRF 360 dataset [3] and achieved comparable results in other datasets as detailed in Tab. 2 and 3.

Fig. 3 illustrates the zoom-in performance comparison of our model with the baselines, where our model visibly surpasses the visual quality of SOTA [9, 8]. Both 3DGS [6] and 3DGS + EWA [38] exhibited notable erosion artifacts due to dilation operations and some high-frequency artifacts. In comparison to Scaffold-GS [9] and Mip-Splatting [8], our model rendered images with clearer details and higher resolution, while Scaffold-GS and Mip-Splatting exhibited various degrees of blurry artifacts. Our model avoids such artifacts and closely matches the ground truth. Fig. 4 provides a more detailed view of our model's rendering quality compared to Scaffold-GS [9] and Mip-Splatting [8]. During training, it was observed that Scaffold-GS and Mip-Splatting often produced blurry outcomes and Scaffold-GS was particularly susceptible to high-frequency artifacts. As depicted in Fig. 4, which represents different scenes from various datasets, our model demonstrates its ability to preserve superior detail. Even when zooming into specific areas, it maintains complete and crisp details. In contrast, Mip-Splatting and Scaffold-GS frequently lose detail and exhibit blurriness.

4.4. Ablation Study

4.4.1. Loss terms and Regulariser

We conducted an ablation study to evaluate the impact of the designed loss terms which are L_{P-SSIM} and L_{tv}. We trained our model on three scenes from the Mip-NeRF 360 dataset [3] and two scenes from the Tanks&Temples dataset [34], including the "Bicycle", "Counter", "Kitchen", "Train" and "Truck" scenes for 30k iterations. They are the most commonly used scenes in 3D reconstruction. As indicated in Tab. 4, the inclusion of L_{P-SSIM} and L_{tv} in our original model led to improvements across all performance metrics. Notably, L_{P-SSIM} had the most significant impact due to its ability to help the model capture more non-local structural information with stochastic characteristics during training. L_{tv}, serving as an

auxiliary regulariser, further enhanced the model's performance by promoting smoothness and reducing artifacts.

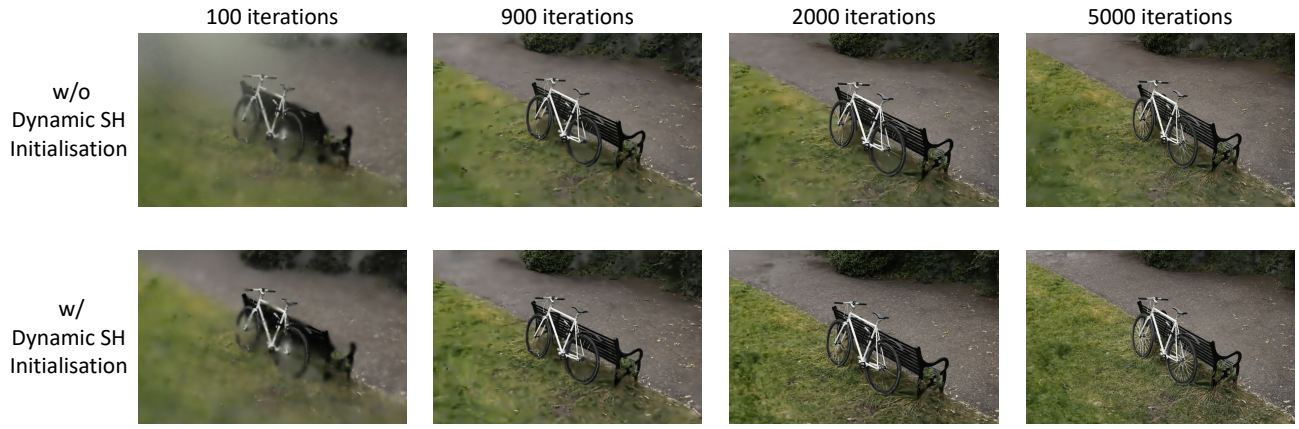

Figure 5: **Ablation of Dynamic Spherical Harmonics Initialisation.** We present an ablation study of the training progression of the bicycle scene [3]. The result shows that this strategy improves the training quality in the early iterations.

4.4.2. Dynamic Spherical Harmonics Initialisation

We evaluated our method for the dynamic adjustment of spherical harmonics initialisation, as described in Section 4. In Fig. 5, we present the training progression of the Bicycle scene with and without applying this strategy, showing results at 100, 900, 2000 and 5000 training iterations. It is evident that the model utilising our dynamic initialisation strategy outperforms the standard approach in early training phases. This advantage is particularly noticeable in aspects such as richer colours and more realistic details. Our method addresses the issue of high-degree spherical harmonics not being initialised and engaged during early training iterations. By dynamically adjusting the initialisation of spherical harmonics, our model achieves better performance in the initial stages of training.

4.4.3. Multi-scale Residual Network

We assessed our model both with and without the implementation of the Multi-scale Residual Network. The results of this comparison are presented in Tab. 2, Tab. 3 and Tab. 4, where we display the metric scores for all datasets, including scenarios where the Multi-scale Residual Network (MSRN) was removed. As evident from these tables, the inclusion of the Multi-scale Residual Network (MSRN) significantly enhances performance across various datasets. This enhancement is attributed to the network's capability to render higher resolution images during the rendering phase of our model, which in turn improves the expressiveness of the pixels.

4.5. Limitation and Future Work

Our model leverages a patch-based SSIM loss, which has significantly improved performance; however, this approach requires the image to be segmented into numerous patches to compute SSIM for each segment, thereby increasing computational resource demands and extending overall training duration. Currently, we apply the P-SSIM loss only to RGB values. In the future, we may extend the application of structural similarity metrics like S3IM to non-RGB losses, such as those involving depth information. Another limitation involves our method of Dynamic Adjustment of Spherical Harmonics

Initialisation, which heavily relies on initial estimates of opacity and RGB colours. Poor initial estimates can adversely affect the efficacy of this strategy. We are considering future enhancements that would allow dynamic adjustments of spherical harmonic values throughout the training process, making it more adaptive rather than confined to the initialisation phase. Lastly, the use of the Multi-scale Residual Network (MSRN) in the rendering phase increases rendering times by approximately 30%. Despite this, it still supports relatively fast rendering. Future iterations of our model may integrate MSRN with 3DGS during training, applying super-resolution information directly to enhance both training efficiency and rendering quality.

5. Conclusion

In this paper, we introduced StructGS, an enhanced model based on the original 3D Gaussian Splatting framework. Our approach includes three main innovations: Non-Local Structural Information P-SSIM, Dynamic Adjustment of Spherical Harmonics Initialisation and the incorporation of a Multi-scale Residual Network (MSRN) during the rendering phase. These innovations aim to address the limitations of 3DGS-based models in capturing stochastic, non-local structural information during training and to improve the early training performance as well as the rendering quality of the model. Experimental results demonstrate that StructGS significantly outperforms state-of-the-art (SOTA) models across various datasets, including indoor, outdoor and large-scale scenes. It delivers higher quality, higher resolution images with more detailed content and fewer high-frequency artifacts.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.

[4] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[5] C. Sun, M. Sun, and H.-T. 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*, 2022, pp. 5459–5469.

[6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[7] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[8] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-splatting: Alias-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 447–19 456.

[9] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.

[10] Z. Xie, X. Yang, Y. Yang, Q. Sun, Y. Jiang, H. Wang, Y. Cai, and M. Sun, "S3im: Stochastic structural similarity and its unreasonable effectiveness for neural fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 024–18 034.

[11] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 517–532.

[12] P. Debevec, Y. Yu, and G. Boshokov, "Efficient view-dependent ibr with projective texture-mapping," in *EG Rendering Workshop*, vol. 4, no. 11, 1998.

[13] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[14] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 501–518.

[15] H. Shum and S. B. Kang, "Review of image-based rendering techniques," *Visual Communications and Image Processing 2000*, vol. 4067, pp. 2–13, 2000.

[16] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.

[17] H. Qin, J. Li, Y. Jiang, Y. Dai, S. Hong, F. Zhou, Z. Wang, and T. Yang, "Bullet-time video synthesis based on virtual dynamic target axis," *IEEE Transactions on Multimedia*, vol. 25, pp. 5178–5191, 2022.

[18] D. S. Alexiadis, D. Zarpalas, and P. Daras, "Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 339–358, 2012.

[19] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[20] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021.

[21] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3295–3306.

[22] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European conference on computer vision*. Springer, 2022, pp. 333–350.

[23] Z. Huang, S. M. Erfani, S. Lu, and M. Gong, "Efficient neural implicit representation for 3d human reconstruction," *Pattern Recognition*, vol. 156, p. 110758, 2024.

[24] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.

[25] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[26] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[27] M. Schütz, K. Krösl, and M. Wimmer, "Real-time continuous level of detail rendering of point clouds," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2019, pp. 103–110.

[28] D. Rückert, L. Franke, and M. Stamminger, "Adop: Approximate differentiable one-pixel point rendering," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–14, 2022.

[29] G. Kopanas, J. Philip, T. Leimkühler, and G. Drettakis, "Point-based neural rendering with per-view optimization," in *Computer Graphics Forum*, vol. 40, no. 4. Wiley Online Library, 2021, pp. 29–43.

[30] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "Synsin: End-to-end view synthesis from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7467–7477.

[31] G. Arvanitis, E. I. Zacharaki, L. Vâŝa, and K. Moustakas, "Broad-to-narrow registration and identification of 3d objects in partially scanned and cluttered point clouds," *IEEE Transactions on Multimedia*, vol. 24, pp. 2230–2245, 2021.

[32] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[34] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[35] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 6, pp. 1–15, 2018.

[36] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *European conference on computer vision*. Springer, 2022, pp. 106–122.

[37] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[38] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 29–538.

Method | Scenes	bicycle	garden	stump	room	counter	kitchen	bonsai
Instant-NGP	0.491	0.649	0.574	0.855	0.798	0.818	0.890
Plenoxels	0.496	0.606	0.523	0.842	0.759	0.648	0.814
Mip-NeRF 360	0.685	0.813	0.744	0.913	0.894	0.920	0.941
3DGS	0.771	0.868	0.775	0.914	0.905	0.922	0.938
Scaffold-GS	0.705	0.842	0.784	0.925	0.914	0.928	0.946
Mip-Splatting	0.747	0.858	0.770	0.927	0.915	0.932	0.947
Ours (without MSRN)	0.785	0.865	0.783	0.927	0.920	0.937	0.951
Ours (Full)	**0.884**	**0.935**	**0.854**	**0.951**	**0.939**	**0.965**	**0.968**

Table A.5: SSIM of baselines and our method for Mip-NeRF 360 dataset [3].

Method | Scenes	bicycle	garden	stump	room	counter	kitchen	bonsai
Instant-NGP	22.19	24.60	23.63	29.27	26.44	28.55	30.34
Plenoxels	21.91	23.49	20.66	27.59	23.62	23.42	24.67
Mip-NeRF 360	24.37	26.98	26.40	31.63	29.55	32.23	**33.46**
3DGS	25.25	27.41	26.55	30.63	28.70	30.32	31.98
Scaffold-GS	24.50	27.17	26.27	31.93	29.34	31.30	32.70
Mip-Splatting	25.13	27.38	26.64	31.50	29.13	31.47	32.39
Ours (without MSRN)	25.64	27.18	26.79	31.05	29.40	31.90	32.48
Ours (Full)	**28.07**	**29.60**	**28.18**	**32.10**	**30.13**	**33.37**	33.41

Table A.6: PSNR of baselines and our method for Mip-NeRF 360 dataset [3].

Appendix A. Experiments and Result

In the appendix, we present the evaluation metrics for our baselines and our model across all scenes in the Mip-NeRF 360 dataset [3]. It is evident that our model outperforms the state-of-the-art (SOTA) models based on 3DGS [8, 9] in terms of PSNR and SSIM [33] metrics in almost all scenes. This is true even without employing MSRN for super-resolution rendering. In a few scenes, the results are comparable. Table A.7 illustrates that our model's images, whether MSRN is used or not, significantly surpass all other models in LPIPS [37] scores.

Method | Scenes	bicycle	garden	stump	room	counter	kitchen	bonsai
Instant-NGP	0.487	0.312	0.450	0.301	0.342	0.254	0.227
Plenoxels	0.506	0.386	0.503	0.419	0.441	0.447	0.398
Mip-NeRF 360	0.301	0.170	0.261	0.211	0.204	0.127	0.176
3DGS	0.205	0.103	0.210	0.220	0.204	0.129	0.205
Scaffold-GS	0.306	0.146	0.284	0.202	0.191	0.126	0.185
Mip-Splatting	0.245	0.123	0.243	0.197	0.183	0.116	0.180
Ours (without MSRN)	0.178	0.104	0.201	0.178	0.161	0.106	0.156
Ours (Full)	**0.047**	**0.026**	**0.057**	**0.040**	**0.045**	**0.022**	**0.018**

Table A.7: LPIPS of baselines and our method for Mip-NeRF 360 dataset [3].