

# An Empirical Study of State of the Art 3D Reconstruction Methods

## COS 529 Fall 2025

Harsha Rajendra  
Princeton University  
hr1751@princeton.edu

Maahin Rathinagiriswaran  
Princeton University  
mr1885@princeton.edu

### Abstract

*This project presents a comparative analysis of Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) using the nerfstudio framework across Mip-NeRF 360 and custom real-world datasets. We evaluate the trade-offs in rendering quality, computational efficiency, and robustness to data sparsity. Our empirical results demonstrate that while 3DGS achieves superior visual fidelity (6–10 dB improvement) and real-time rendering speeds (>100 FPS), it suffers from severe low-frequency structural degradation under sparse view conditions (24 views). In contrast, NeRF exhibits lower-quality but more stable degradation due to implicit regularization. We conclude by providing a practical decision framework to help researchers and practitioners select the optimal reconstruction model based on their specific needs and constraints including image/view count, hardware availability, and scene-type.*

## 1. Introduction

The field of 3D reconstruction is rapidly evolving and becoming increasingly significant. For decades, the field was dominated by traditional photogrammetry frameworks, Structure-from-Motion (SfM)[1] and Multi-View Stereo (MVS)[2]. Although these methods performed well in extracting geometry, they struggled with complex visual cues such as transparent surfaces, thin surfaces, and reflections, resulting in incomplete final reconstructions.

The landscape was revolutionized by the introduction of Neural Radiance Fields (NeRF) [3]. This shifted the paradigm from explicit mesh generation to implicit volumetric rendering. NeRF achieved an unprecedented level of photorealism and is capable of synthesizing novel views that were virtually similar to the original views. It performs this by continuously optimizing a function to represent scene density and color. However, this visual clarity came at the cost of significant computational resources often requiring hours of training to render a few frames.

3D Gaussian Splatting [4] uses discrete gaussian primitives that can be rendered in real-time. This approach proves to combine the visual fidelity of NeRF with fast rendering speeds which could prove useful in interactive applications and thus fundamentally changing the dominance of Neural field representations.

Despite their individual successes, the choice between these two state-of-the-art models remains a mystery for practitioners. This project aims to resolve this ambiguity in the decision-making process through a comprehensive comparative analysis. We aim to contribute to the following six research questions:

1. Qualitative Comparison: Which method achieves superior visual fidelity across indoor and outdoor scenes using rendered views vs. actual views?
2. Quantitative Comparison: How does each method perform quantitatively over the same dataset (PSNR, SSIM, and LPIPS)?
3. Sparse View Robustness: How does the reconstruction quality degrade for each method when training data is limited (we experiment with sparse 75 and 24 views)?
4. What are the quantitative differences in training time, and memory utilization.
5. Under what specific constraints (data, time, scene characteristics, hardware) should one choose NeRF over 3DGS or vice-versa?
6. How do the models perform when faced with a real-world custom dataset?

## 2. Related Work

The problem of 3D scene reconstruction from 2D images has traditionally been solved through Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipelines. The introduction of Neural Radiance Fields (NeRF) marked a paradigm shift toward implicit volumetric representations. However, the original NeRF suffered from long training times and slow rendering speeds making it incompatible for real-time rendering. Subsequent works like Mip-NeRF 360 [5] introduced anti-aliasing strategies for



Figure 1. Samples from the Mip-NeRF datasets

unbounded scenes, while Instant-NGP [6] utilized multi-resolution hash encodings to reduce training time. In contrast, 3DGS achieves real-time rendering speeds and faster training convergence compared to NeRF.

However, a critical challenge for both approaches is Sparse View synthesis. Neural fields often overfit when training with scarce views, leading to cloudy artifacts and geometric disasters. While specialized regularizers like RegNeRF [7] have been developed to yield better results in few-shot settings, standard 3DGS has been observed to degrade rapidly given insufficient diverse views. This is because the explicit primitives lack the inductive bias of continuous MLPs that help fill the unobserved regions [8].

Recent literature has begun to explore the performance trade-offs between implicit and explicit representations. An analysis [9] conducted by Blanchard et al. performed a comparative study of Instant-NGP and 3DGS, highlighting 3DGS’s superior efficiency in rendering complex scenes with non-Lambertian surfaces but this study primarily focuses on standard dense-view settings. Similarly, other qualitative studies [10] have evaluated these methods for UAV-based point cloud generation confirming 3DGS’s dominance in real-time applications but they also note its weaknesses to geometric artifacts in large-scale environments. This project analyzed these trade-offs, specifically focusing on the degradation behavior of NeRF and 3DGS under sparse view conditions. Unlike standard evaluations that base their results solely on metrics like PSNR and SSIM, we also perform a spectral analysis to pinpoint degradation in low-frequency versus high-frequency components.

### 3. Methodology

#### 3.1. Datasets and Data Processing

Mip-NeRF 360 was primarily used to test the performance of both the models. We selected four scenes that put forward diverse challenges for reconstruction: Garden and Bicycle (outdoor scenes with complex backgrounds), and Bonsai and Kitchen (indoor scenes with minute details and controlled lighting). Figure 1 showcases samples from each dataset.

Data processing was standardized across all experiments to ensure fair comparison. We used the COLMAP Structure-from-Motion (SfM) framework to estimate camera poses. The frames presented in the dataset have varying but high resolution but to mitigate memory constraints and focus on high-frequency detail preservation, we standardized the processing stage by downscaling the input frames by a scale of 4 for all the scenes.

To evaluate the models’ ability to be robust across a varying number of views, we generated subsets of each scene from the dataset. We employed a uniform sub-sampling technique to create “sparse” views from the original dataset. This resulted in 2 different configurations for each scene: One with 24 views and another with 75 views. Only 2 scenes(Bonsai and Garden) were sub-sampled to 75 views while all the scenes were sub-sampled to 24. We also employed k-means to produce another version of the sparse 75 view dataset on the Bonsai and Garden scenes.

We also evaluated on custom real-world datasets recorded using an iPhone at 4k resolution and 30 fps. The subjects include:

- Scarlet Knight statue:** This scene captures a complex outdoor environment featuring uniform metallic textures to test the models on non-Lambertian surfaces.
- Medicine ball:** This constrained indoor scene focuses on a detailed and finely textured object that could test the model’s ability to reconstruct high-frequency surface details.
- Water bottle on a chair:** This scene is particularly challenging due to the presence of transparency and refraction of the water bottle.

#### 3.2. Experimental Setup

All experiments were performed on high-performance Nvidia RTX 4090 GPUs (24GB VRAM) to ensure consistent compute resources for timing benchmarks.

##### 3.2.1 Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS)

We utilized Nerfstudio’s nerfacto, which integrates hash encoding with a lightweight MLP to balance quality and speed. We trained the nerfacto model for 70000 iterations

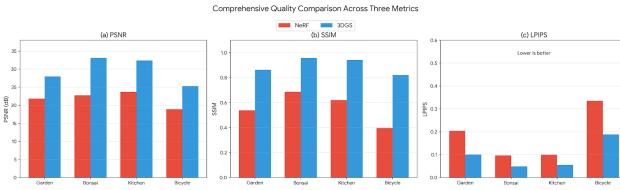


Figure 2. Quality comparison across PSNR, SSIM, and LPIPS for full dataset.

to ensure maximum convergence, using a batch size of 4096 rays.

For 3DGS, we used splatfacto. Unlike NeRF’s continuous volumetric function, 3DGS represents the scene as a set of discrete 3D Gaussian primitives. We trained this model for the same number of 70000 iterations.

### 3.2.2 Evaluation Metrics

In order to assess the quality of our novel synthesis and 3D renders, we utilized three quantitative metrics for both NeRF and 3DGS. Each metric captures different characteristics of reconstruction quality: pixel-wise accuracy, structural consistency, and perceptual realism.

As defined in equation 1, PSNR measures the pixel-level fidelity of the rendered images compared to the ground truth images. It is derived from the Mean Squared Error (MSE) between the two images.

$$\text{PSNR} = -10 \log_{10}(\text{MSE}) = 20 \log_{10} \left( \frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \quad (1)$$

Unlike PSNR, SSIM evaluates the degradation of structural information within local windows of the image. SSIM for two image patches  $x$  and  $y$  is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2)$$

Where  $\mu_x, \mu_y$  are local means,  $\sigma_x^2, \sigma_y^2$  are local variances,  $\sigma_{xy}$  is the covariance, and  $c_1, c_2$  are constants to stabilize division by zero.

We consider LPIPS to assess the perceptual quality as perceived by the humans. Both the rendered and ground truth images are passed through a pre-trained VGG network and LPIPS computes the distance between the extracted deep features. It calculates the Euclidean distance between feature maps across L layers:

$$\text{LPIPS}(I, I_{gt}) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (\phi^l(I)_{hw} - \phi^l(I_{gt})_{hw})||_2^2 \quad (3)$$

Where  $\phi^l(\cdot)$  represents the feature map at layer  $l$ , and  $w_l$  scales the activations channel-wise. A lower LPIPS score indicates better quality, capturing fine details and textures that PSNR often misses.

Table 1. Comparative Results: NeRF vs. 3DGS

Experiment	Method	Views	PSNR↑	SSIM↑	LPIPS↓	$\Delta$ vsBaseline
Garden-Full	NeRF	100	21.80	0.538	0.204	-
Garden-Full	3DGS	100	27.97	0.862	0.100	+6.17 dB
Garden-Sparse-24	NeRF	24	16.00	0.284	0.538	-5.80 dB
Garden-Sparse-24	3DGS	24	17.20	0.404	0.365	-7.01 dB
Bonsai-Full	NeRF	100	22.75	0.687	0.096	-
Bonsai-Full	3DGS	100	33.11	0.957	0.048	+10.36 dB
Bonsai-Sparse-24	NeRF	24	14.39	0.322	0.619	-8.36 dB
Bonsai-Sparse-24	3DGS	24	15.95	0.433	0.494	-17.16 dB
Kitchen-Full	NeRF	100	23.69	0.620	0.099	-
Kitchen-Full	3DGS	100	32.37	0.941	0.055	+8.68 dB
Kitchen-Sparse-24	NeRF	24	12.52	0.256	0.764	-11.17 dB
Kitchen-Sparse-24	3DGS	24	15.73	0.423	0.493	-16.64 dB
Bicycle-Full	NeRF	100	18.87	0.397	0.335	-
Bicycle-Full	3DGS	100	24.21	0.754	0.155	+5.34 dB
Bicycle-Sparse-24	NeRF	24	12.68	0.193	0.633	-6.19 dB
Bicycle-Sparse-24	3DGS	24	14.59	0.187	0.557	-9.62 dB
AVERAGE (Full)	NeRF	-	21.78	0.560	0.183	-
AVERAGE (Full)	3DGS	-	28.48	0.851	0.103	+6.70 dB
AVERAGE (Sparse)	NeRF	-	13.90	0.264	0.639	-7.88 dB
AVERAGE (Sparse)	3DGS	-	15.87	0.362	0.477	-12.61 dB

## 4. Analysis and Results

We compare two leading 3D reconstruction methods: Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS). Both were tested under two conditions: full dataset (roughly 100 views) and sparse dataset (just 24 views).

The experiments used Nerfstudio’s implementations on four Mip-NeRF 360 benchmark scenes, each with different challenges. We measured quality using PSNR (pixel accuracy), SSIM (structural similarity), and LPIPS (perceptual quality). Computational metrics like training time, memory usage, and rendering speed were also tracked.

### 4.1. Full Dataset Performance

With roughly 100 training views, 3DGS beats NeRF across all quality metrics. Figure 2 shows the differences.

In our experiments, the average PSNR improvement sits around 6 to 7 dB. Structural similarity and perceptual quality follow similar patterns and the performance gap varies across scenes. Bonsai shows the strongest improvement, over 10 dB, likely because those intricate needles challenge continuous representations. This was an unexpected result given the scene’s relatively compact structure. Kitchen follows with approximately 8 dB advantage, which appears connected to how 3DGS handles sharp edges and flat surfaces. Garden and Bicycle show smaller gains, around 5 to 6 dB each. The quantitative results are in Table 1.

To understand why 3DGS performs better, we look at the representational differences. NeRF uses a continuous neural network that learns smooth functions mapping 3D coordinates to colors and densities. This creates spectral bias: the network naturally favors low-frequency features

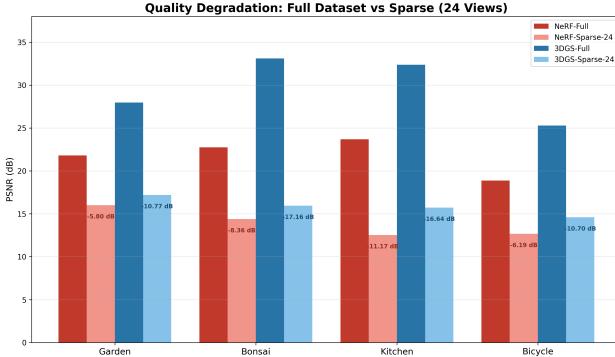


Figure 3. Quality degradation from full to sparse (24 views).

and struggles with sharp details. 3DGS takes a different approach with explicit Gaussian primitives. Each primitive independently represents a small piece of geometry. This makes capturing sharp edges and intricate details easier. The architectural differences explain the quality advantages we see across diverse scene types.

## 4.2. Sparse View Performance

The sparse-view scenario on the other hand shows major limitations. When we reduce the dataset to just 24 views, quality drops severely for both methods. Figure 3 shows this decline.

NeRF drops by roughly 8 dB on average while 3DGS experiences a steeper decline of around 12 to 13 dB. Kitchen shows the worst performance loss, with both methods falling below 16 dB PSNR (essentially unusable quality). Despite this decline, 3DGS holds a slight edge, though the performance gap narrows from the comfortable 6 to 7 dB lead seen with full data.

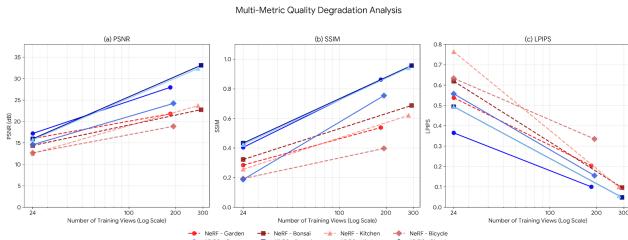


Figure 4. Multi-metric degradation curves across varying view counts.

Figure 4 shows how quality declines as view count decreases. The degradation curves follow power-law behavior. 3DGS degrades roughly linearly while NeRF shows a square-root pattern. NeRF’s degradation curve is smoother despite its lower baseline accuracy. This happens because NeRF’s continuous representation provides implicit regularization under sparse conditions.

Table 2. Efficiency metrics comparing training time, memory, rendering speed, and model size

Scene	Method	Train Time ↓	Peak Memory ↓	Render FPS ↑	Model Size ↓
Garden	NeRF	21.97 min	5.00 GB	1.6	168 MB
Garden	3DGS	32.78 min	13.88 GB	90.0	649 MB
Bonsai	NeRF	20.28 min	4.78 GB	4.2	168 MB
Bonsai	3DGS	11.93 min	1.93 GB	163.7	230 MB
Kitchen	NeRF	20.12 min	4.77 GB	4.2	168 MB
Kitchen	3DGS	10.59 min	1.94 GB	157.2	226 MB
Bicycle	NeRF	21.80 min	5.27 GB	1.7	168 MB
Bicycle	3DGS	36.57 min	5.56 GB	98.4	1229 MB
AVERAGE	NeRF	21.04 min	4.96 GB	2.9	168 MB
AVERAGE	3DGS	22.97 min	5.83 GB	127.3	584 MB
SPEEDUP	3DGS	0.92×	0.85×	43.53×	-

Both methods show steep drops between 24 and 100 views, indicating that data quantity matters critically in this range. The main issue is geometric: without sufficient rays crossing from different angles, the reconstruction problem becomes ambiguous. Many different 3D configurations can satisfy identical 2D observations when viewpoints are sparse. Sufficient view count is therefore essential for stable reconstruction.

## 4.3. Computational Efficiency

Training time is surprisingly similar between methods. NeRF takes around 21 minutes on average and 3DGS takes about 23 minutes (Table 2). However, 3DGS varies by scene. Garden requires over 32 minutes due to massive Gaussian densification for vegetation. Simpler scenes like Bonsai and Kitchen complete in under 12 minutes.

The difference in rendering speed is more pronounced. 3DGS achieves 127 FPS compared to NeRF’s 3 FPS (a 43x speedup). For applications requiring real-time rendering, 3DGS is the only practical choice. 3DGS uses GPU rasterization pipelines efficiently, while NeRF evaluates a neural network for every sample along every ray.

Memory usage shows different scaling behaviors. NeRF stays constant at 5 GB regardless of scene complexity. 3DGS scales with geometric complexity, ranging from as low as 2 GB for simple scenes to nearly 14 GB for complex outdoor environments. This variability can help (lower memory for simple scenes) or hurt (unpredictable requirements).

Training convergence patterns (Figure 5) show that 3DGS reaches usable quality faster through its Adaptive Density Control mechanism. On Bonsai, 3DGS hits 30 dB by 10,000 iterations while NeRF plateaus around 23 dB. The tradeoff appears in stability: 3DGS shows volatility as primitives are continuously managed, whereas NeRF shows steady improvement across all scenes. As noted earlier in Figure 4, this stability difference matters when considering deployment constraints. Method choice depends on

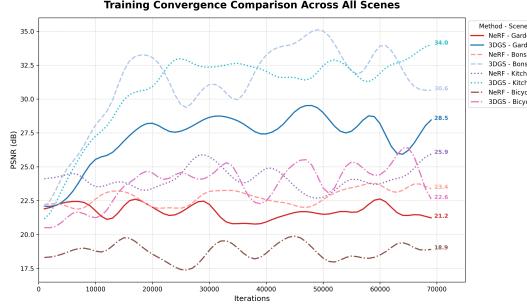


Figure 5. Training convergence patterns across all scenes.

application needs: real-time rendering favors 3DGS, while resource-constrained environments may prefer NeRF’s predictable behavior and fixed memory overhead.

#### 4.4. Qualitative Analysis

The qualitative impact of data sparsity is evident in the zoomed regions highlighted in Figure 6, where models trained on only 75 views fail to resolve fine-scale characteristics of the scene. In the Garden scene (Figure 6 (a)), the degradation appears in the form of “cloudy” artifacts on top of the rendered image, where the high-frequency texture of the brick surface is washed out into a low-frequency approximation. This indicates that without sufficient views to supplement the reconstruction process, the model struggles to resolve depth and texture on distant surfaces, reverting to an average density representation.

Similarly, the Bonsai scene (Figure 6 (b)) contains “cloudy” and translucent artifacts around the bicycle spokes and door frame. This suggests that the model could not effectively replicate the empty space between fine geometric details (bicycle spokes) leaving behind a visual haze.

#### 4.5. Scene-Specific Analysis

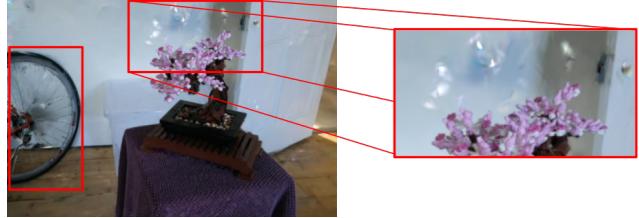
Kitchen has the steepest sparse performance loss across all scenes (both methods falling below 13 dB). We look at how each scene’s geometry interacts with each method’s representation.

**Garden:** 3DGS gets approximately 6 dB advantage. The poor structural similarity scores for NeRF come from spectral bias (discussed earlier) blurring fine leaf details. Under 24-view training, both methods drop to around 16 to 17 dB due to insufficient ray intersections from diverse viewpoints.

**Bonsai:** This scene has our largest gap, over 10 dB, with 3DGS getting near-photorealistic quality. The fine needle structure tests high-frequency reconstruction, and training efficiency also favors 3DGS, which finishes 40% faster. Under sparsity, both fail because portions of the tree occlude each other. 3DGS creates “floaters” positioned to satisfy single viewpoints but looking wrong from other angles.



(a) Sample render from Garden-75



(b) Sample render from Bonsai-75

Figure 6. Degradation on the renders from sparse garden and bonsai dataset.

**Kitchen:** 3DGS holds roughly 8 dB advantage with full data. Explicit Gaussians flatten along walls and align with sharp edges. Under sparse conditions, NeRF fills empty space with semi-transparent fog while 3DGS has “background collapse” (walls and objects positioned incorrectly).

**Bicycle:** 3DGS holds approximately 5 dB advantage by using anisotropic stretched ellipsoids for thin wires. Under sparsity, NeRF’s structural similarity slightly beats 3DGS’s despite lower PSNR. The difference is in failure modes: NeRF creates cohesive blur while 3DGS produces disjointed floaters. Hence, scene geometry plays an important role in determining method performance.

### 5. Frequency Domain Analysis

PSNR provides a single measure of reconstruction error, but it does not indicate whether the error arises from overall scene layout or from fine texture. To examine these aspects separately, we evaluate reconstructions in the frequency domain. Broad variations in intensity fall into the low frequency range, while sharper transitions such as edges appear in the high frequency range.

#### 5.1. Frequency Breakdown

Each image is transformed using the Fast Fourier Transform:

$$F(u, v) = \sum_{x,y} I(x, y) \cdot e^{-2\pi i(ux/M+vy/N)} \quad (4)$$

A circular mask centered at the origin divides the spectrum into low and high frequency regions. Values within 30% of the radius form the low frequency band, and values outside this region form the high frequency band. We apply

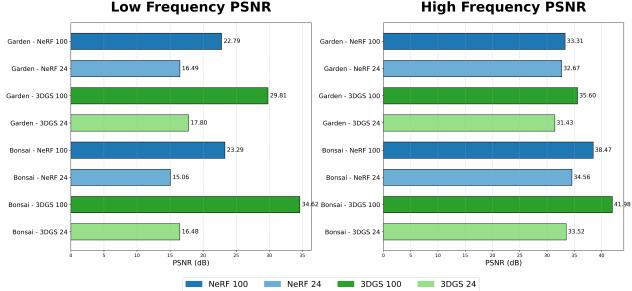


Figure 7. Low and high frequency PSNR under full and sparse training. 3DGS maintains strong low frequency accuracy with full data, while performance declines sharply with fewer views.

the inverse transform to each band separately and measure reconstruction accuracy using

$$\text{PSNR}_{\text{band}} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}_{\text{band}}} \right) \quad (5)$$

This study is applied to the Garden and Bonsai scenes using two training configurations: a full setting with approximately 100 views and a sparse setting with 24 views.

## 5.2. Frequency Band Results

Looking at each frequency band separately (Figure 7), 3DGS clearly outperforms NeRF in the low frequencies when trained on the full dataset. The improvement is about 7 decibels on Garden and exceeds 11 decibels on Bonsai. Since these components describe the broad structure of the scene, errors here have a stronger effect on overall image quality than errors in fine detail.

The high frequency band shows a smaller gap between the two methods. On Garden, both reach nearly the same value, approximately 33 decibels. On Bonsai, 3DGS is slightly higher, but the gap is still small compared to the low frequency range. NeRF tends to preserve local texture reasonably well but does not maintain consistent large scale structure. This result differs from the common expectation that neural fields are primarily limited by sharp detail.

## 5.3. Effect of Limited Training Views

Reducing the number of views to 24 greatly affects the low frequency band. Figure 8 shows the decrease in accuracy. NeRF drops by 6 to 8 decibels, and 3DGS drops by 12 to 18 decibels. Both methods fall to roughly 15 to 17 decibels in the structural components, a range where clear distortions appear.

High frequency accuracy drops more gradually. For NeRF, the change on Garden is very small, and the change on Bonsai is moderate. For 3DGS, the decrease ranges from 4 to 8 decibels. Reconstructing the overall shape of the

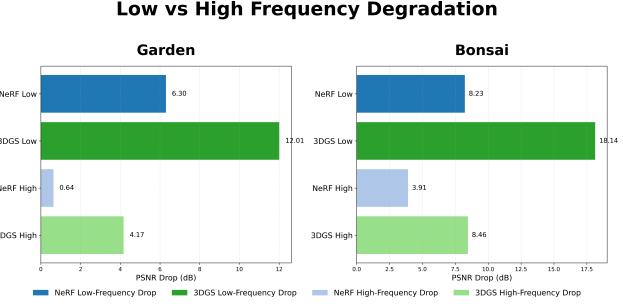


Figure 8. Decrease in PSNR when training with only 24 views. The Structural accuracy falls the most under sparse data, with 3DGS showing the largest reduction.

scene requires viewpoints that cover the space from different directions. Fine textures, however, can be reproduced as long as they are visible from at least one direction.

The trend can be described using :

$$\text{Degradation ratio} = \frac{\Delta_{\text{low}}}{\Delta_{\text{high}}} \quad (6)$$

In Garden, NeRF shows a structural decrease that is about 10 times larger than its change in local detail, while Bonsai shows a ratio of roughly 2. 3DGS follows a similar trend but with larger overall reductions.

## 5.4. NeRF and 3DGS Degradation Behavior

NeRF represents the scene as a continuous function. When trained with limited data, the network tends to oversmooth broad structures while still reproducing some local detail. This explains the stronger decline in low frequency components, in contrast to 3DGS which uses explicit Gaussian primitives. Many views help us put these primitives in the exact position. With limited coverage, however, their placement becomes uncertain, which accounts for the larger drop in structural accuracy.

Both methods become less reliable when the training set falls below roughly 75 to 100 views. This threshold likely relates to the number of viewing directions required to determine the arrangement of objects in the scene.

## 6. Adaptive View Sampling

### 6.1. Sampling Techniques

Uniform sampling selects frames at regular intervals and preserves the order of the original capture. Circular camera paths often provide broad angular coverage when sampled uniformly.

However, K-means sampling uses a different principle. Camera positions are grouped into clusters using :

$$\text{maximize } C(V) = \int_{\Omega} \min_{v \in V} d(x, v) dx \quad (7)$$

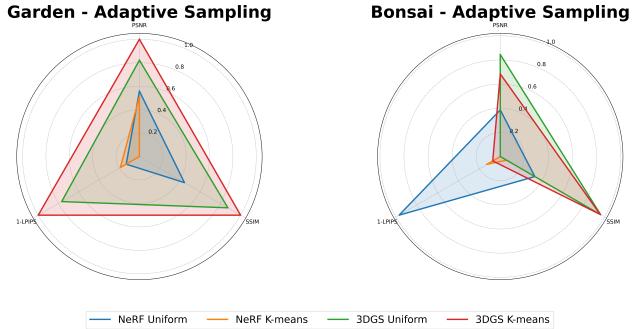


Figure 9. Comparison of uniform and K-means sampling for Garden and Bonsai. Larger regions correspond to better reconstruction quality.

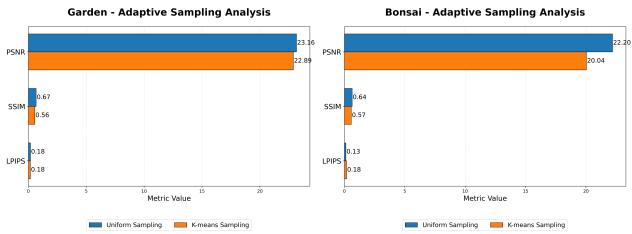


Figure 10. Direct comparison of uniform and K-means sampling. Garden shows mixed behavior, while Bonsai strongly favors uniform sampling for both methods.

where  $\Omega$  is the scene volume and  $V$  is the chosen set of views. Because computing the exact solution is not feasible, we cluster the available viewpoints into 75 groups and select one representative from each cluster. Prior studies suggest that improved spatial spread can provide measurable gains of up to one and a half decibels.

## 6.2. Quality Changes Based on Sampling

Figure 9 summarizes the reconstruction results obtained under uniform sampling and K-means sampling for both scenes.

On Garden, 3DGS improves by roughly one decibel with K-means sampling. The wider spatial distribution of viewpoints appears to support more accurate placement of primitives. NeRF shows a slight decrease with K-means sampling. This may reflect the fact that NeRF benefits from smooth variation across neighboring views along the capture trajectory.

On Bonsai, both methods perform better with uniform sampling. NeRF drops by more than 2 decibels under K-means sampling, and 3DGS drops by approximately 0.9 decibels. Figure 10 provides a direct comparison. This scene requires viewpoints that surround the central object, and even spacing in three dimensional space does not guarantee this type of angular coverage.

The improvement on Garden for 3DGS represents a significant portion of the theoretical upper limit. The absence

Table 3. Quantitative comparison on custom object-centric datasets.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	FPS↑	VRAM↓	ΔvsBaseline
Scarlet Knight	NeRF	28.27	0.918	0.039	20.6 min	3.2	4.94 GB	-
Scarlet Knight	3DGS	36.28	0.975	0.033	9.0 min	160.9	1.91 GB	+8.01 dB
Medicine Ball	NeRF	27.65	0.877	0.068	22.1 min	3.2	4.94 GB	-
Medicine Ball	3DGS	38.10	0.977	0.038	8.8 min	163.4	1.94 GB	+10.45 dB
Bottle Close	NeRF	23.70	0.839	0.134	22.0 min	3.2	4.55 GB	-
Bottle Close	3DGS	32.20	0.960	0.056	8.7 min	159.2	1.79 GB	+8.50 dB
AVERAGE	NeRF	26.54	0.878	0.080	21.6 min	3.2	4.81 GB	-
AVERAGE	3DGS	35.53	0.971	0.042	8.8 min	161.2	1.88 GB	+8.99 dB

of improvement on Bonsai reflects differences in scene geometry. Explicit methods such as 3DGS benefit from spatially distributed viewpoints in outdoor settings, while NeRF relies more on continuity along the capture path.

## 6.3. View Selection Best Practices

For scenes with large spatial extent, distributing viewpoints with a clustering-based method can benefit 3DGS. For indoor or object-centric scenes, uniform temporal sampling is more reliable for both methods. These settings require views from many angles around the object rather than an even spread in global space.

## 6.4. Frequency Relationship

The sampling results are consistent with the observations from the frequency domain study. Wider viewpoint coverage supports accurate recovery of structural components, while gaps in angular coverage lead to errors similar to those seen under sparse view conditions. In both evaluations, the spatial distribution of training views strongly influences reconstruction quality at the structural level.

## 7. Custom Dataset Validation

We selected three object-centric scenes for this analysis: *Scarlet Knight* (a statue with complex geometry), *Medicine Ball* (a textured spherical object), and *Bottle* (close up scan of a water bottle on top of a chair)

### 7.1. Quality and Fidelity Analysis

Table 3 reveals a larger performance gap on custom datasets than on Mip-NeRF 360. While 3DGS outperformed NeRF by 6-7 dB on standardized scenes, the gap widened to 9-10 dB on custom objects.

On the *Medicine Ball* scene, 3DGS achieved a remarkable PSNR of 38.10 dB compared to NeRF's 27.65 dB. This difference in performance further highlights the capability of Gaussian Splatting to model textured surfaces in object-centric scenes effectively. Similarly, for the *Bottle* scene, NeRF struggled significantly (PSNR: 23.70 dB, SSIM: 0.839), which is likely due to the challenges in resolving the object's boundaries against the background. However, 3DGS maintained high fidelity (PSNR: 32.20

dB, SSIM 0.960), suggesting that its explicit representation is more robust to peculiarities found in casual, handheld recordings.

## 7.2. Efficiency Analysis

A critical finding from these experiments contradicts the trend observed in our unbounded scene analysis earlier. In the large-scale *Garden* dataset, 3DGS consumed nearly 14 GB of VRAM at one point, which was significantly more than NeRF. However, in these bounded, object centric scenes, the relationship flipped.

NeRF’s memory usage remained constant at approximately 4.9 GB regardless of the scene characteristics, owing to its fixed MLP architecture and hash-grid capability as mentioned before. On the other hand, 3DGS handled the scenes more efficiently, requiring less than 2 GB VRAM across all three custom scenes. This indicates that while 3DGS is VRAM hungry for large environments with a lot of intricate details as it would have to generate millions of gaussian primitives, it is exceptionally lightweight for close up object scans. This makes it an exceptional choice for deployment on lower-end consumer devices where the scope of reconstruction is limited to singular objects.

## 7.3. Training and Inference Speed

The training and inference speed were consistent with our prior finding from the standardized datasets. 3DGS demonstrated superior computational efficiency. Training times for 3DGS reached an average of 8.8 minutes across all three scenes, which is 2.45x faster than NeRF which reached an average of 21.6 minutes across the same three scenes. Inference speeds remained real-time for 3DGS (>160 FPS), while NeRF remained quite slow for interactive viewing at ( $\sim 3$  FPS).

## 7.4. Method Selection Criteria

We recommend 3DGS when sufficient training data is available (100+ views). It delivers better quality (typically 6 to 7 dB higher) with real-time rendering (40x+ speedup) and strong handling of high-frequency details.

3DGS works best for complex scenes with intricate structures (Bonsai’s 10+ dB advantage) and indoor environments with hard surfaces (Kitchen’s 8+ dB advantage). NeRF fits limited capture situations (under 50 views) where NeRF degrades more smoothly, and resource-tight deployment where NeRF’s constant memory footprint is predictable.

At 24 views, both methods produce unusable reconstructions. The steep drop between 24 and 100 views suggests practical deployment needs at least 75 to 100 views for complex scenes. Scene type affects relative performance: 3DGS’s advantage is largest on object-focused complex

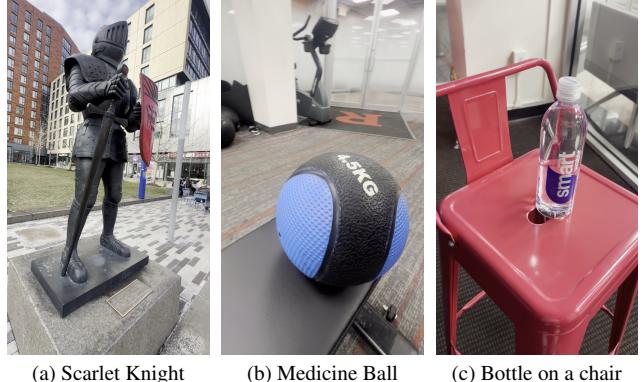


Figure 11. Rendered samples from our custom dataset.

scenes and indoor bounded environments, while the gap shrinks on outdoor unbounded scenes.

From our analysis on a custom dataset with object scans from a handheld device, we found that 3DGS was the clear winner and should be chosen by default for its superior quality and speed, unless the data is very limited and stability is prioritized. In such cases, NeRF is the safer option.

## 7.5. Ablation Study

Our work includes three ablation studies to isolate factors influencing reconstruction quality. *First*, we varied training view count across 100, 75, and 24 views to quantify the impact of data quantity on performance. *Second*, we decomposed reconstructions into frequency bands to determine whether degradation occurs in structural components or fine details. *Third*, we compared uniform temporal sampling against K-means spatial clustering to evaluate view selection strategies under constrained capture budgets. Each ablation varied only one factor while keeping others constant.

## 7.6. Conclusion

With sufficient training data, 3DGS produces higher quality reconstructions & renders much faster than NeRF, making it ideal for virtual reality, video games & architectural visualization. Both methods struggle below 75-100 views, where scene structure degrades more than fine details. 3DGS works best for product visualization and digital archives when capture is controlled, while NeRF suits mobile devices or systems with limited data. Our future work will include generating 3D meshes for the custom dataset and exploring hybrid methods that adapt based on available data.

## 7.7. Project Documentation

GitHub: <https://github.com/rmaahin/3D-reconstruction-using-nerf-and-3dgs>

## References

- [1] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [1](#)
- [2] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. [1](#)
- [3] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. [1](#)
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. [1](#)
- [5] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields, 2022. [1](#)
- [6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 2022. [2](#)
- [7] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, 2021. [2](#)
- [8] Shuting He, Peilin Ji, Yitong Yang, Changshuo Wang, Jia-ayi Ji, Yinglin Wang, and Henghui Ding. A survey on 3d gaussian splatting applications: Segmentation, editing, and generation, 2025. [2](#)
- [9] Shailesh Nanisetty Cordell Blanchard, Lakshya Gupta. Analyzing 3d gaussian splatting and neural radiance fields: A comparative study on complex scenes and sparse views., 2024. [2](#)
- [10] M.E. Atik. Comparative assessment of neural radiance fields and 3d gaussian splatting for point cloud generation from uav imagery, 2025. [2](#)