

Towards Alias-Free 4D Gaussian Representations with Motion-Aware Filtering

Ankit Dhiman^{1,2}, Kunal A Kathare¹, Pranav Vignesh¹, Lokesh R Boregowda²,
and Venkatesh Babu Radhakrishnan¹

¹ Indian Institute of Science, Bangalore

² Samsung R&D Institute India - Bangalore

<https://maaf-4dgs.github.io/>

Abstract. Novel-view synthesis of dynamic scenes, crucial for AR/VR applications, remains a challenging problem. Recent methods adapt representations like 3D Gaussian Splatting (3DGS) and Neural Radiance Fields (NeRF) for dynamic scenes by incorporating time as the fourth dimension (4D representations). These 4D representations still suffer from aliasing artifacts, especially when generating novel views from divergent viewpoints (zoom-in/zoom-out operations). While using 3D smoothing filters like those proposed in Mip-Splatting might seem like a possible solution, they fail to account for local motion and also exhibit aliasing. To address this, we propose a motion-aware 3D smoothing filter specifically designed for 4D representations. Our approach adapts the filter strength based on local motion information, effectively mitigating aliasing without compromising rendering quality. This is achieved by estimating the joint density function of time and focal-to-depth ratio using a non-parametric estimation method. During inference, we sample from this joint distribution to determine the appropriate smoothing filter. This flexible strategy can be integrated with various 4D representations. Our evaluations on standard datasets demonstrate superior performance compared to state-of-the-art methods.

1 Introduction

Learning the radiance field [40] of a scene from multi-view RGB images enables numerous applications, including novel-view synthesis (NVS) [1, 2, 8, 28, 41], 3D reconstruction [16, 17], mixed reality [21], and robotics [50, 56]. Generating novel views of dynamic scenes from multi-view camera observations holds immense potential for creating immersive experiences from everyday video recordings [14]. However, it is very challenging. Traditional NVS methods relied on implicit [1, 2, 40] or grid-based [3, 13, 41] representations, often employing ray-tracing and volumetric rendering. More recently, explicit primitive-based representations have become popular. In particular, 3D Gaussian Splatting (3DGS) [28] utilizes a differentiable rendering pipeline based on Gaussian splat rasterization, achieving real-time rendering of 1080p images on commercial GPUs.

To model dynamic scenes, extensions of NeRF and 3DGS employ the concept of deformation fields and canonical representations [18, 19, 33, 36, 46, 47, 64]. Alternative methods decompose the 4D scene into 3D projections of spatial-temporal



Fig. 1: We compare novel view synthesis results from our method (b) with SARO-GS (d), focusing on both dynamic regions (e.g., hand movements, cooking actions) and static regions (e.g., bottles, background). The zoomed-in regions (c) highlight that our method produces sharper details and fewer artifacts, demonstrating superior synthesis quality in motion and static areas.

and spatial-only 3D representations [4, 12, 35, 49, 59, 63], subsequently combining these reduced features to model the dynamic regions. Other methods extend the 3DGS representation to 4D Gaussians [5, 10, 27], directly optimizing them. However, in Fig. 2 all these representations exhibit limitations, notably aliasing artifacts when rendered from divergent viewpoints, which hinders accurate representation of dynamic scenes.

Recent work, Mip-Splatting [65], addresses the challenge of divergence in the novel-view viewpoints from training viewpoints (zoom-in and zoom-out operations) for static scenes. Specifically, it determines the maximum sampling interval for each Gaussian primitive and restricts the maximum frequency of the 3D representation by applying a low-pass filter to satisfy the Nyquist criterion [43]. While demonstrating impressive results for static scenes, this methodology exhibits limitations for dynamic scene representations



Fig. 2: We trained all models on training views with a resolution of 1352×1014 and rendered outputs by adjusting the focal length. Existing 4D representations exhibit significant artifacts during zoom-in and zoom-out operations. Applying static anti-aliasing filters, such as Mip-Splatting [65], is also ineffective as motion in the scene leads to incorrect filter estimation and persistent artifacts.

(See Fig. 2). A primary reason for this limitation is that the band-limited filter lacks motion awareness, failing to account for temporal changes in the scene. Consequently, we propose a motion-aware strategy to determine the maximum sampling interval, thereby enabling accurate rendering for dynamic scenes.

To mitigate aliasing artifacts observed in dynamic scenes, we first analyze the temporal distribution of the sampling interval to derive a motion-aware smoothing filter. We found that directly estimating the sampling interval at each timestamp led to noisy estimates due to temporal abruptness. To address this, we propose learning the joint distribution of discrete time and the focal-length-to-depth ratio for each Gaussian primitive. We employ a non-parametric estimation technique: the Parzen Window kernel density estimation method, to learn this joint distribution from accumulated data points. This also makes the estimates smoother compared to the trivial solution. During inference, we sample the most probable value from this learned distribution. Overall, this strategy makes the sampling interval estimation motion-aware and enables the design of a motion-aware smoothing filter for these 4D representations. Our main contributions are as follows:

- We introduce a motion-aware smoothing filter that resolves the aliasing artifacts in the 4D representations.
- We demonstrate the efficacy of our method in challenging real-world datasets [31, 46] and a synthetic dataset [47].
- Our modifications are invariant of the 4D representation and can be plugged in with other representations as well.

2 Related Work

Novel View Synthesis. With the advancement in neural rendering steered by Neural Radiance Fields (NeRF) [40], the task of novel-view synthesis has gained a lot of popularity. NeRF utilizes volumetric rendering and optimizes a multi-layer perception (MLP) to learn the radiance field of a scene. This approach is computationally expensive. To accelerate training and rendering speed subsequent works makes use of hash grids [41], voxel grids [13, 52], point-based methods [62] and depth supervision [6]. Further, this representation is extended to solve different tasks such as large-scale scenes [48, 60], text-to-3D generation [53, 57], hierarchical scenes [8] and editing tasks [9, 25, 66].

3DGS [28] models scenes as a mixture of 3D Gaussians that are projected as 2D splats and rasterized using alpha blending to get the projected image. Compared to NeRF based approaches, 3DGS enables high-quality rendering in real-time. Subsequent works have improved over this representation by introducing a hierarchical structure [37], generalizable exponential functions [20]. Further, other works have focused on improving rendering quality [24, 65], accelerated rendering [11, 15], memory optimization [42, 45], better initialization [26, 29, 44] and overall training time [7, 15, 22, 39].

Dynamic Scenes. To model dynamic scenes, NeRF based methods [18, 36, 46, 47] model a scene as a canonical scene and a deformed scene and then use

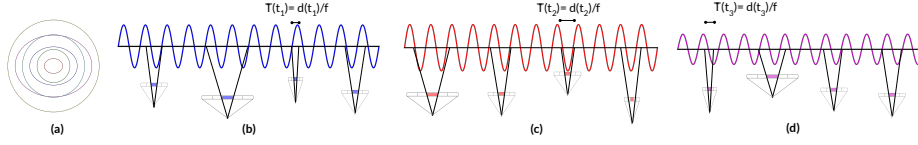


Fig. 3: Consider a 4D signal (represented as a 2D signal in (a)). Its different projections at timestamps t_1 , t_2 & t_3 , form 3D signals (represented as 1D signals in (b), (c), and (d), respectively). These continuous 3D signals are modeled by 3D Gaussian primitives. When observed by multiple cameras at each timestamp, these primitives are band-limited by different sampling intervals. Hence, it is essential to design a motion-aware low-pass filter for dynamic scenes.

deformation field network to model the relationship between static canonical space and the deformed points. In contrast, [4, 12, 35, 49, 55] decomposes the 4D space into set of planes or hash-grids to effectively model the spatio-temporal relations. However, these approaches require dense sampling and suffer from same issue as the static scene counterparts such as low rendering speed.

Building on 3DGS, recent methods extend to dynamic scenes. Dynamic 3D Gaussians [38] model a dynamic scene as a collection of moving and rotating Gaussians frame by frame. 4DGS [59] utilizes Hex-planes to model the changes in Gaussian primitives over time. Other methods [19, 33, 64] utilize a time-conditioned deformation network to wrap a canonical set of Gaussians into each timeframe. These methods fail to handle scenarios with significant motion and often suffer from artifacts such as object appearances/disappearances. Further works [5, 10, 27] extend 3D Gaussians across space-time, i.e. means and covariances of Gaussian primitives are in 4D. Other works [5, 27, 51] attempt the challenging task of novel-view synthesis of casual monocular videos for dynamic scenes. In comparison, our work addresses the aliasing artifacts in these representations.

Aliasing in Radiance Field Methods. Earlier NeRF-based [1–3, 23, 68] approaches used pre-filtering to mitigate aliasing, but these methods do not extend to 3DGS representations. EWA splatting [69] filters 2D Gaussians in screen space, but this rendering-time filter is prone to the choice of filter size. Conversely, Mip-Splatting [65] designs a 3D filter during optimization, allowing alias-free rendering at different scales, but this fails for dynamic scenes. In contrast, we propose applying a 3D band-limited filter based on the Gaussian primitive’s motion.

3 Preliminaries

In this section, we provide a brief overview of 3D Gaussian Splatting (3DGS) [28] and a recent dynamic representation SARO-GS [63]. We then discuss Mip-Splatting [65], an anti-aliasing 3D filter for static scenes.

3.1 3D Gaussian Splatting

3DGS [28] represents scene geometry using a collection of 3D Gaussian primitives and each primitive is defined as:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (1)$$

where μ is the center position and Σ is an anisotropic covariance matrix. A combination of rotational R and scaling S transformations, represented as $RSS^T R^T$, constructs the covariance matrix, guaranteeing that it is positive semi-definite. Each Gaussian primitive has an opacity value σ , and spherical harmonic coefficients to represent a view-dependent color. During rendering, Gaussians are projected as 2D splats, sorted by depth, and combined with α blending using a tile-based rasterizer. The resulting pixel color C , is determined by:

$$C(x') = \sum_i c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \quad \sigma_i = \alpha_i G'_i(x') \quad (2)$$

where x' is the queried pixel, c_i is the color of the i -th Gaussian, α_i is the learned opacity for the i -th Gaussian, and $G'_i(x')$ is the 2D projection of the Gaussian at pixel x' .

3.2 Dynamic Representations using 3DGS

We choose SARO-GS [63] as the base 4D representation. SARO-GS models a dynamic scene using a set of 4D Gaussian primitives, $\mathcal{G}^{4D}$, and a Scale-aware Residual Field, $\mathcal{M}$. Each 4D Gaussian $\mathcal{G}^{4D}$ is parameterized by a 4D location $\mu^{4D} = (\mu, \tau)$ and other attributes covariance Σ^{4D} , color c^{4D} and opacity σ^{4D} . Here, $\mathcal{M}$ is a lightweight MLP that encodes the spatio-temporal variation of the scene: for every primitive it predicts a time-dependent residual to its attributes. Concretely, the 4D Gaussian primitives are first projected into 3D space at the initial time t_0 . Then, $\mathcal{M}$ is queried at a given time-stamp t to obtain residual features, which are added to the projected 3D Gaussian primitives before rasterization using Eq. 2. The final loss is given as follows:

$$\mathcal{L} = (1 - \lambda_1)\mathcal{L}_1 + \lambda_1\mathcal{L}_{D-SSIM} + \lambda_2\mathcal{L}_{SR} \quad (3)$$

where $\mathcal{L}_1$ and $\mathcal{L}_{D-SSIM}$ are the standard photometric losses used in 3DGS [28], computed between the ground-truth and the rasterized image, and $\mathcal{L}_{SR}$ is the regularization term on the scaling residuals introduced by SARO-GS [63].

3.3 Anti-aliasing filter for 3DGS

Mip-Splatting [65] builds upon the principle that image resolution, camera focal length, and scene depth define the effective sampling rate for 3D scene reconstruction from multi-view images. Their major insight is that primitive smaller

Algorithm 1 Motion-Aware Sampling Rate Estimation (per Gaussian primitive)

Require: Observations $\{(t_j, x_j)\}_{j=1}^J$, $x_j = f_j/d_j$ (empty slots flagged by $t_j < 0$); bandwidths h_x, h_t ; Weibull shape k , scale λ ; grid sizes D, T

Ensure: Sampling rate $\hat{v}(t)$ for a query time t

- 1: Kernels: $K_w(u) = \frac{k}{\lambda} (\frac{u}{\lambda})^{k-1} e^{-(u/\lambda)^k} \mathbb{1}[u \geq 0]$, $K_g(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$
- 2: $v_j \leftarrow \mathbb{1}[t_j \geq 0]$ {mask empty observation slots}
- 3: $x_{\max} \leftarrow \max_j v_j x_j$; $\bar{x}_j \leftarrow x_j/x_{\max}$ {normalize f/d to $[0, 1]$ per primitive}
- 4: $g^x \leftarrow \text{Linspace}(0, 1, D)$; $g^t \leftarrow \text{Linspace}(0, 1, T)$ {query grids}
- 5: **for** $d = 1$ to D **do**
- 6: **for** $s = 1$ to T **do**
- 7: $\rho[d, s] \leftarrow \frac{1}{2h_x h_t} \sum_{j=1}^J v_j K_w\left(\frac{g_d^x - \bar{x}_j}{h_x}\right) K_g\left(\frac{g_s^t - t_j}{h_t}\right)$ {joint KDE}
- 8: **end for**
- 9: **end for**
- 10: *Inference at a query time t :*
- 11: $s^* \leftarrow \arg \min_s |g_s^t - t|$ {nearest time grid point}
- 12: $d^* \leftarrow \arg \max_d \rho[d, s^*]$ {mode of the conditional density $\rho[\cdot, s^*]$ }
- 13: $\hat{v}(t) \leftarrow x_{\max} \cdot g_{d^*}^x$ {rescale to f/d units}

than twice the world-space sampling interval $\hat{T}$ can introduce aliasing, where $\hat{T} = \frac{d}{f}$, where d is distance from the camera and f is the focal length of the camera. To mitigate this, they approximate the depth-dependent sampling rate for each primitive based on its center and camera visibility and is given by following:

$$\hat{v}_k = \max \left(\left\{ \mathbb{1}_n(\mathbf{p}_k) \cdot \frac{f_n}{d_n} \right\}_{n=1}^N \right) \quad (4)$$

where N is the total number of images, $\mathbb{1}_n(\mathbf{p})$ is an indicator function that assesses the visibility of a primitive. Then, they apply a Gaussian low-pass filter $\mathcal{G}_{\text{low}}$ to a 3D Gaussian primitive $\mathcal{G}_k$ before projecting it onto screen space:

$$\mathcal{G}_k(\mathbf{x})_{\text{reg}} = (\mathcal{G}_k \otimes \mathcal{G}_{\text{low}})(\mathbf{x}) \quad (5)$$

Result of convolving two Gaussians is a Gaussian and is given by:

$$\mathcal{G}_k(\mathbf{x})_{\text{reg}} = \sqrt{\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_k + \frac{s}{\hat{v}_k} \cdot \mathbf{I}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T (\boldsymbol{\Sigma}_k + \frac{s}{\hat{v}_k} \cdot \mathbf{I})^{-1} (\mathbf{x}-\boldsymbol{\mu})} \quad (6)$$

where s controls the size of the filter. Note that $\frac{s}{\hat{v}_k}$ is different for each Gaussian primitive and remains constant post-training.

4 Proposed Method

Aliasing in Dynamic Representations Fig. 2 demonstrates the aliasing artifacts present in these 4D representations. After training 4D representations

on input views, we perform zoom-in and zoom-out operations. During zoom-in, 4DGS [59], SARO-GS [63], and Grid4D [61] suffer from disappearing primitives in both dynamic and static regions. During zoom-out, these methods produce noticeable aliasing. For example, in Grid4D, the human merges with the wall, and SARO-GS misses fine floor details. These results indicate that aliasing occurs in 4D representations when novel-view camera viewpoints deviate significantly from the training data.

Limitations of Static Anti-Aliasing Filters Aliasing artifacts pose a significant challenge in reconstructing dynamic scenes with dynamic representations, particularly in areas of motion (Fig. 2).

While Mip-Splatting [65] effectively addresses aliasing in static scenes by introducing a low-pass filter, it proves ineffective for dynamic scenarios, as evidenced by the results for “SARO-GS + Mip-Splatting” in Fig. 2. Therefore, we propose a motion-aware smoothing filter tailored for dynamic scene reconstruction.

Analysis of Static and Dynamic Gaussian primitives. Analysis of the sampling rate (f/d) distribution over time reveals key insights for dynamic scenes. As presented in Fig. 4 for the “cook-spinach” scene from the Plenoptic Video dataset [31], static regions exhibit a consistently stable distribution, as expected. In contrast, dynamic regions show significant temporal variation, demonstrating that a single, static smoothing filter is inadequate and results in artifacts (Fig. 2). This observation directly motivates the design of our motion-aware smoothing filter.

Necessity of Time-Varying Sampling Rates. Fig. 3 illustrates the challenge of estimating sampling intervals for dynamic scenes. Unlike static scenes, dynamic scenes are continuous 4D signals. While Mip-Splatting can provide an overall estimate of the maximal sampling rate for this 4D signal, a single value is insufficient. Since each timestamp t yields a unique 3D projection of the 4D signal, it is necessary to determine a separate maximal sampling rate for each projection. Therefore, we propose a novel method to estimate the maximal sampling interval for a timestamp. We emphasize that here “sampling rate” denotes the *spatial* sampling requirement f/d of a primitive at a given timestamp. Although the video timestamps are fixed, a primitive’s depth changes with motion, which makes its f/d time-varying. Our method thus computes a separate maximal sampling rate for each spatio-temporal projection of the 4D signal, allowing

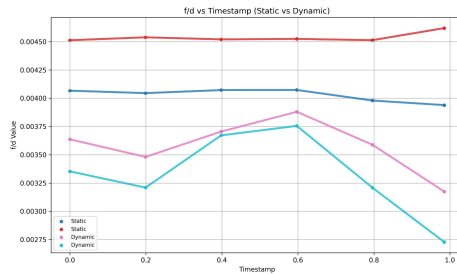


Fig. 4: Analysis of the sampling rate f/d over time for the “cook-spinach” scene from the Plenoptic Video dataset [31]. Static primitives maintain a nearly constant f/d , whereas dynamic primitives vary substantially over time, as motion changes their depth. We detail how static and dynamic primitives are identified in the supplementary material.



Fig. 5: Single-Scale Training and Multi-Scale Evaluation on Plenoptic Video [31] dataset. All baseline methods and our approach were trained at a resolution downsampled by a factor of 4, while novel views are evaluated at the original. We provide zoomed-in regions highlighted by green boxes. Notice that the fine details of the martini (Row 3) being poured into the glass are not preserved in any baseline method, whereas our method successfully retains this detail.

the anti-aliasing filter to adapt per primitive and per timestep while satisfying the Nyquist criterion.

Motion-Aware Smoothing Filter We assume that f/d is sampled from a distribution $\mathcal{X}$ and t is sampled from a distribution $\mathcal{T}$. We do not know the underlying distribution of these variables and want to estimate the joint density function $\mathcal{P}(\mathcal{X}, \mathcal{T})$. Hence, we utilize the Parzen window density estimation method to estimate the density function $\mathcal{P}(\mathcal{X}, \mathcal{T})$. Parzen window methods yield smooth and continuous density estimates, in contrast to histograms which are discontinuous and sensitive to bin size. We choose following kernel functions:

$$\mathcal{X} \sim \text{Weibull}(k, \lambda) \quad \mathcal{T} \sim \mathcal{N}(\mu_T, \sigma_T^2) \quad (7)$$

We assume a Weibull distribution because, for $k > 1$, it becomes biased toward maximum values. This aligns with Eq. 4, where the sampling rate for each primitive is defined as the maximum of all entries.

With these assumptions, we now describe our method. First, for each Gaussian primitive k we create data $\left\{ (t_i, \frac{f_i}{d_i}) \right\}_{i=1}^n$ by projecting primitives at every time stamp for all the training views. To make the estimate invariant to the abso-

	PSNR $\uparrow$				SSIM $\uparrow$				LPIPS $\downarrow$			
	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.
SARO-GS [63]	32.75	29.40	27.40	29.85	0.956	0.913	0.884	0.918	0.086	0.167	0.234	0.160
Spacetime-GS [32]	32.50	29.20	27.09	29.60	0.957	0.918	0.892	0.922	0.087	0.169	0.232	0.163
4DGS [59]	31.46	28.30	26.36	28.71	0.946	0.901	0.881	0.909	0.095	0.177	0.230	0.167
4DRotorGS [10]	32.29	29.60	29.10	30.33	0.951	0.920	0.877	0.916	0.092	0.187	0.212	0.162
Grid4D [61]	31.03	28.13	26.64	28.60	0.799	0.773	0.765	0.779	0.095	0.177	0.234	0.169
Ours	32.52	30.75	29.79	31.01	0.952	0.925	0.903	0.926	0.089	0.162	0.216	0.155

Table 1: Single-Scale Training and Multi-Scale Evaluation on the Plenoptic Video Dataset [31]. All methods are trained at the smallest scale (1 $\times$) that is 676x507 and evaluated across multiple scales (1 $\times$, 2 $\times$, and 4 $\times$), where increasing resolution simulates zoom-in effects. While our approach performs comparably to existing methods at the training resolution (1 $\times$), it demonstrates superior fidelity across all other scales, significantly outperforming prior work.

	PSNR $\uparrow$				SSIM $\uparrow$				LPIPS $\downarrow$			
	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.	1 $\times$	Res. 2 $\times$	Res. 4 $\times$	Res. Avg.
Deformable3DGS [64]	39.57	30.09	27.30	32.32	0.989	0.957	0.934	0.960	0.008	0.034	0.061	0.034
SARO-GS [63]	36.61	29.59	27.15	31.12	0.982	0.952	0.931	0.955	0.015	0.040	0.064	0.039
4DGS [59]	35.65	28.03	25.58	29.75	0.987	0.958	0.937	0.960	0.012	0.035	0.059	0.035
4DRotorGS [10]	32.17	26.77	24.84	27.92	0.965	0.931	0.919	0.938	0.024	0.050	0.075	0.050
Grid4D [61]	40.30	30.07	28.54	32.97	0.990	0.956	0.933	0.960	0.007	0.035	0.062	0.035
GauFRe [34]	34.05	27.59	25.12	28.92	0.981	0.951	0.928	0.953	0.025	0.058	0.079	0.054
Ours	37.10	34.05	32.36	34.50	0.982	0.972	0.960	0.971	0.017	0.033	0.054	0.035

Table 2: Single-Scale Training and Multi-Scale Evaluation on the Synthetic Dataset [47]. All methods are trained on the smallest scale (1 $\times$) that is 200x200 and evaluated across multiple scales (1 $\times$, 2 $\times$, and 4 $\times$), where increasing resolution simulates zoom-in effects. Our approach outperforms baselines across all other scales.

lute scale of each primitive’s sampling rate, we normalize its f/d values by their per-primitive maximum $x_{\max}$ so that they lie in $[0, 1]$, and define the density grids over $[0, 1]$; the value recovered at inference is rescaled by $x_{\max}$ to return to f/d units. Then, we estimate the joint density function $\mathcal{P}_k(\mathcal{X}, \mathcal{T})$ for each Gaussian primitive. We describe the algorithm in more detail in Algorithm 1. To find the value of f/d given a specific t , we determine it maximizing the conditional density $P(\mathcal{X}|\mathcal{T} = t)$. Specifically, we extract the estimated density values for a specific t ; $\mathcal{P}_k(\mathcal{X}, \mathcal{T} = t)$ and then find the value corresponding to the highest density. Finally, this estimated sampling rate is applied to a deformed Gaussian primitive $\mathcal{G}'_k$ as shown in Eq. 5, which ensures that the highest frequency component of any Gaussian does not exceed half of its maximal sampling rate.

5 Experiments

Implementation Details We use SARO-GS [63] as the base 4DGS representation. We apply the smoothing filter on the projected 3D Gaussians and train our model for 15K iterations for all plenoptic scenes and 20K iterations for synthetic scenes. We use the loss function described in Eq. 3. The adaptive density control happens after 600 iterations at every 100th iteration until 5K iterations for plenoptic scenes and until 15K iterations for synthetic scenes. We compute the motion-aware smoothing filter after 600 iterations at every 100th iteration

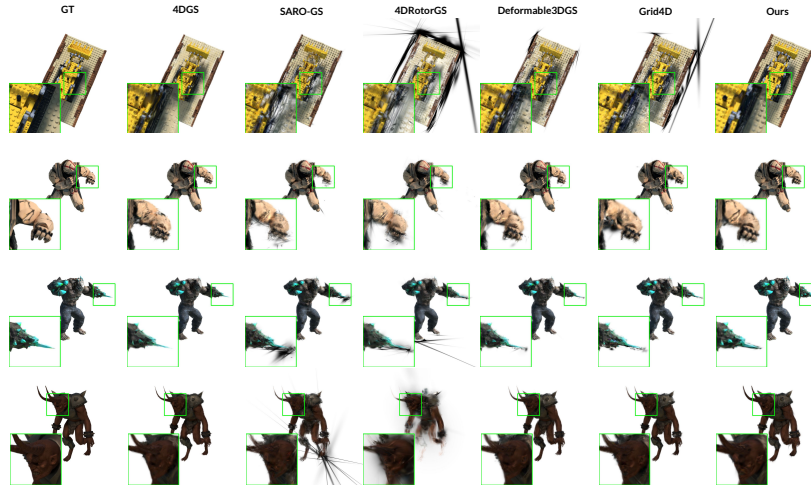


Fig. 6: Single-Scale Training and Multi-Scale Evaluation on D-NeRF [47] dataset. All baseline methods and our method were trained at 200×200 image resolution, while novel views are evaluated at 800×800 . Notice the artifacts in the baselines: the arm in “mutant” (Row 3), the arm in “hook” (Row 2), and the top part in “lego” (Row 1). In contrast, our method renders with high fidelity.

except the final training iteration. We use the hyperparameter settings similar to SARO-GS. Further, we also use the 2D filter proposed in Mip-Splatting with variance of 0.1.

Dataset We evaluate our method on six real-world scenes from the Plenoptic Video dataset [31], each captured by 15 to 20 cameras. Additionally, we assess performance on eight synthetic scenes from D-NeRF [47] dataset. These datasets were utilized for single-scale training and multi-scale testing. We also show evaluation results on the HyperNeRF [46] dataset.

Metrics. We employ the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [58], and Learned Perceptual Image Patch Similarity (LPIPS) [67] to assess the quality of the rendered novel-views. Further, we evaluate the temporal stability of a video based on the flow warping error between two frames [30]. We provide more details on this temporal stability metrics in the supplementary material.

Baselines. We compare our method with the Gaussian splatting-based methods SARO-GS [63], 4DGS [59], Spacetime-GS [32], 4DRotorGS [10] and Grid4D [61] and use the official implementations of all baselines.

5.1 Results and Analysis

Single-Scale and Multi-Scale Testing To evaluate zoom-in performance, we adopt the methodology from Mip-Splatting. Specifically, we train the model on training-images downsampled by a factor of four. Then, we evaluate the

generated novel views at ($1\times$, $2\times$ and $4\times$) resolution. We train baseline methods using this setting.

Evaluation on Plenoptic dataset. Tab. 1 shows a multi-scale performance comparison using the training strategy discussed above. At $1\times$ resolution, our method achieves comparable results to the baselines. However, at $2\times$ and $4\times$ resolutions, we significantly outperform the baseline methods. Notably, we achieve a 2.11 dB improvement over SARO-GS, which is significant as we build upon their method. In contrast, 4DGS, Spacetime-GS, and Grid4D exhibit subpar performance at $4\times$ resolution. This trend is also evident in the qualitative results presented in Fig. 5. Observe the blurriness in the baseline methods’ output for the “sear steak” scene. Our method, on the contrary, preserves high-frequency details when rendered at higher resolutions. This ability to maintain high-frequency content, even when trained on lower-resolution data, is a crucial advantage missing in other 4D representations. Further, our method preserves more high-frequency details than baseline methods on the HyperNeRF dataset in Fig. 8.

Temporal Error Analysis. We compare the quality of rendered views by evaluating their consistency across consecutive frames in Tab. 3. This is done by warping a frame to align with the next one and then calculating the error between the warped frame and the actual subsequent frame.

	W-Temporal Error	W-LPIPS	W-SSIM
Spacetime-GS [32]	1.127	0.003803	0.989175
4DGS [59]	0.780	0.002454	0.991685
4DRotorGS [10]	0.769	0.002653	0.992745
Grid4D [61]	0.840	0.002689	0.991504
Ours	0.608	0.001806	0.994581

Metrics like W-Temporal Error, W-LPIPS (lower is better), and W-SSIM (higher is better) quantify this consistency. We observe that the renderings from our approach are the most coherent over time compared to the other techniques.

Table 3: Temporal error analysis. We present a quantitative evaluation of our method’s performance against existing techniques, utilizing warped error, warped LPIPS, and warped SSIM as key metrics. Results demonstrate the superior performance of our approach.

Evaluation on D-NeRF synthetic dataset. We also evaluated our method against baseline methods on D-NeRF dataset with Single-Scale training and Multi-scale testing setting. Consistent with results on real-world dataset, our method achieved comparable performance at $1\times$ resolution, but demonstrated significant improvements at $2\times$ and $4\times$ resolution. In particular, our method outperformed several baselines, including a 5.06 dB increase over Deformable3DGS, a 5.21 dB in-

	PSNR	SSIM	LPIPS
Deformable3DGS [64]	38.23	0.985	0.017
SARO-GS [63]	35.30	0.978	0.024
4DGS [59]	34.19	0.979	0.026
4DRotorGS [10]	34.22	0.974	0.024
Grid4D [61]	39.88	0.987	0.012
Ours	34.82	0.974	0.028

Table 4: Single-scale Training and Same-scale Testing on Synthetic Dataset [47]. We observe that the baselines perform better but our method excels in Multi-Scale Evaluation.(Tab. 2)

crease over SARO-GS, a 3.83 *dB* increase over Grid4D, a 6.78 *dB* increase over 4DGS, and a 7.53 *dB* increase over 4DRotorGS. Furthermore, the baseline methods exhibited a substantial decline in performance at higher resolutions. For example, Grid4D’s PSNR dropped by 10.24 *dB* from $1\times$ to $4\times$ resolution.

Qualitative results in Fig. 6 further support our quantitative findings. For example, in the “hell-warrior” scene, observe the artifacts present in the SARO-GS output, which are absent in our results. This is significant, as we build on SARO-GS as our base representation. Likewise, the “lego” scene exhibits blurry and aliasing artifacts across all baseline methods. Overall, our method effectively preserves high-frequency details, even when trained at a lower resolution.

	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
SARO-GS [63]	32.09	0.946	0.138
Spacetime-GS [32]	31.64	0.948	0.139
4DGS [59]	31.02	0.935	0.150
4DRotorGS [10]	31.10	0.937	0.145
Grid4D [61]	30.09	0.932	0.148
Ours	31.43	0.943	0.140

Table 5: Single-scale Training and Same-scale Testing on Plenoptic Video Dataset [31]. Our approach is comparable to baseline methods, with only a 0.003 SSIM difference from the base SARO-GS representation.

Single-Scale Training Tab. 5 presents the results on the Plenoptic Video dataset, where all baselines were trained and evaluated at the same resolution of 1352×1014 . Our method performed comparably to the other baselines, although it showed a slight performance decrease compared to SARO-GS. This behaviour is expected and consistent with anti-aliasing filters such as Mip-Splatting [65]: by band-limiting each primitive to satisfy the Nyquist criterion, the filter suppresses the high-frequency components that would otherwise overfit the training-resolution views. This is a small trade-off at the training scale for substantially better fidelity at unseen scales. SSIM decreases by only 0.004 relative to the base SARO-GS representation in Tab. 4.

Comparison with SARO-GS + static 3D filter. To further assess the robustness of our method, we construct an additional baseline by integrating the static 3D filtering module from Mip-Splatting [65] with SARO-GS. This baseline aims to simulate the effect of applying spatially consistent 3D filtering to dynamic content. Quantitative results presented in Tab. 6 demonstrate that our approach substantially outperforms this baseline across all evaluation metrics (6.4 *dB* improvement). Further, Fig. 7 shows that this baseline introduces noticeable blur and temporal inconsistencies, particularly around moving

	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
SARO-GS + 3D Filter	24.92	0.840	0.270
Ours	31.40	0.941	0.117

Table 6: Quantitative comparison of our method against SARO-GS + 3D Filter. Our approach significantly outperforms the baseline. This highlights the effectiveness of our method in handling dynamic scenes.

Table 7: Integration with SpeeDe3DGS [54] on the D-NeRF dataset [47] (single-scale training, $4\times$ testing). Our motion-aware filter improves SpeeDe3DGS by 4.45 dB while using fewer primitives and adding only ~ 2 minutes of training. Lower rows ablate the bandwidths h_x, h_t and grid size T ($\dagger$ marks the default configuration). #G is the number of Gaussian primitives.

	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Train (min)	#G
SpeeDe3DGS [54]	27.83	0.940	0.067	4.45	2508
+ Ours ($\dagger$)	32.28	0.956	0.064	6.48	2330
0.1	32.13	0.955	0.065	6.41	2314
h_x 0.2	32.20	0.956	0.064	6.31	2312
0.3 $\dagger$	32.28	0.956	0.064	6.48	2330
0.05 $\dagger$	32.28	0.956	0.064	6.48	2330
h_t 0.1	32.13	0.956	0.065	6.47	2339
0.2	32.07	0.956	0.065	6.47	2352
50	32.25	0.956	0.064	6.41	2290
T 100 $\dagger$	32.28	0.956	0.064	6.48	2330

objects such as the human subject. This degradation arises because the static 3D filter fails to account for temporal motion. In contrast, our method effectively preserves fine-grained details and temporal coherence, producing sharper reconstructions and consistent visual quality over time.

Integration with Other 4D Representations. A key advantage of our motion-aware smoothing filter is that it is decoupled from the underlying 4D representation: it operates on the projected 3D Gaussians and merely replaces the constant sampling rate $\hat{\nu}_k$ with a time-varying $\hat{\nu}_k(t)$. To demonstrate this, we integrate it with SpeeDe3DGS [54], a recent and architecturally different 4D representation. Tab. 7 reports single-scale training and $4\times$ testing on the D-NeRF dataset [47]. Our filter improves SpeeDe3DGS by 4.45 dB in PSNR while using fewer Gaussian primitives, confirming that the gains are not specific to SARO-GS. The overhead is modest: training increases by about two minutes, and the per-primitive density estimate ($\mathcal{O}(NDT)$) adds only 17.7 MB of memory; compressing this is left for future work. The lower part of Tab. 7 ablates the kernel bandwidths h_x, h_t and the temporal grid size T , showing that our method is robust to these choices. Unless stated otherwise, we use $h_x = 0.3$, $h_t = 0.05$, $T = 100$, and Weibull shape $k = 10$ in all experiments.

Zoom-Out (Sub- $1\times$) Evaluation. The evaluations before are for the resolutions above the training resolution (zoom-in). We now assess the opposite regime, rendering *below* it. Using SpeeDe3DGS on the D-NeRF dataset, we train at $1\times$ (800×800) and render at $1\times$, $\frac{1}{2}\times$ (400×400), and $\frac{1}{4}\times$ (200×200). As shown in Tab. 8, the two methods are on par at the training resolution, but as the render resolution decreases the base representation degrades sharply—its PSNR falls from 35.0 to 28.1 dB at $\frac{1}{4}\times$ —whereas our motion-aware filter stays nearly con-

Table 8: Zoom-out (sub-1 $\times$) evaluation with SpeeDe3DGS [54] on the D-NeRF dataset [47] (averaged over 8 scenes). All models are trained at 1 $\times$ (800 $\times$ 800) and rendered at lower resolutions. Our filter keeps quality nearly flat across scales, whereas the base degrades as the render resolution drops.

Eval resolution	PSNR $\uparrow$		SSIM $\uparrow$		LPIPS $\downarrow$	
	SpeeDe3DGS	+Ours	SpeeDe3DGS	+Ours	SpeeDe3DGS	+Ours
1 $\times$ (800 $\times$ 800)	35.01	35.02	0.974	0.974	0.040	0.041
$\frac{1}{2}\times$ (400 $\times$ 400)	32.89	35.88	0.975	0.981	0.025	0.024
$\frac{1}{4}\times$ (200 $\times$ 200)	28.10	35.87	0.951	0.985	0.029	0.014



Fig. 7: Comparison with SARO-GS + static 3D Filter. Incorporating the static 3D filter from Mip-Splatting [65] into SARO-GS leads to spatial blurring and temporal flickering, especially around the moving human subject. In contrast, our method maintains temporal consistency and preserves high-frequency details.

stant (~ 35.9 dB) across all scales, i.e., it is scale-consistent in both directions. We note that the large PSNR gap at $\frac{1}{4}\times$ is concentrated at high-frequency regions (check analysis in supplementary material).

6 Limitations

The main limitation of our method is its memory footprint. For each Gaussian primitive, we store a discrete density matrix over the joint $(f/d, t)$ grid; with N primitives and a grid of size $D \times T$, where D and T are the numbers of discretization bins along f/d and time respectively, the total memory scales as $\mathcal{O}(N \times D \times T)$. This overhead grows with the grid resolution and the number of primitives, which can become significant for large scenes. Several directions can mitigate this in future work. Since inference only requires the per-timestamp maximum of the conditional density, the full joint matrix can be discarded and replaced by the resulting time-varying sampling-rate curve, reducing the foot-

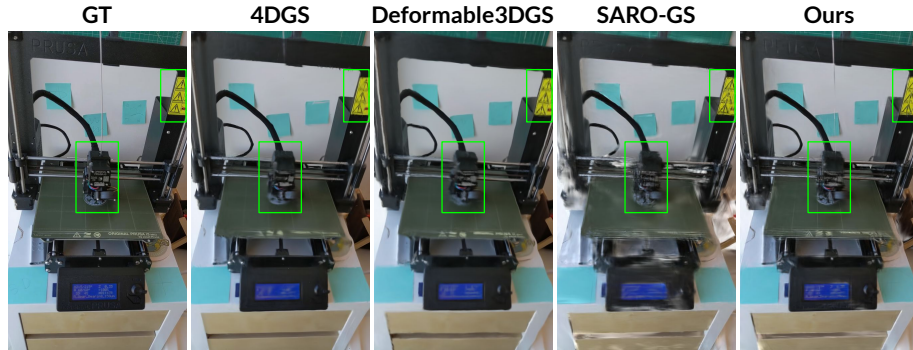


Fig. 8: Qualitative Results on HyperNeRF [46] dataset. We use the “Single-Scale Training and Multi-Scale Testing” setting, where training is performed at a resolution downsampled by a factor of 4, and novel views are rendered at the original resolution. Notice the blurriness in the “printer head” and “yellow sticker” (highlighted with a green box) in the baseline methods. In contrast, our method better preserves high-frequency details.

print from $\mathcal{O}(NDT)$ to $\mathcal{O}(NT)$. Another approach can be that a small shared network predicts the sampling rate directly from a primitive’s features and the query time, removing the need to store a matrix per primitive. Finally, primitives undergoing similar motion can be clustered to share a single density estimate. We leave these memory-efficient extensions to future work.

7 Conclusion and Future Work

We present a motion-aware smoothing filter designed to achieve alias-free rendering for 4D representations, enabling effective modelling of dynamic scenes. Our filter estimates the appropriate sampling interval by estimating a joint distribution and, during inference, selects the value corresponding to the maximum probability for a given query timestamp. Experimental results demonstrate that the proposed method outperforms baseline 4D representations when trained at a lower resolution and evaluated at higher resolutions. This quantitative and qualitative analysis further corroborates our claim that this smoothing filter excels at zoom-in/zoom-out rendering. Furthermore, this strategy is flexible and can be applied to any 4D representation. In future work, we will explore its application to more challenging dynamic scenes characterized by rapid, complex motion, pushing the boundaries of high-fidelity 4D reconstruction for per-scene methods and feed-forward methods.

Acknowledgement

Venkatesh Babu Radhakrishnan is supported by Google and Kotak IISc AI-ML Centre (KIAC).

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19697–19705 (2023)
4. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2023)
5. Das, D., Wewer, C., Yunus, R., Ilg, E., Lenssen, J.E.: Neural parametric gaussians for monocular non-rigid object reconstruction. arXiv preprint arXiv:2312.01196 (2023)
6. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12882–12891 (2022)
7. Dhiman, A., Lu, T., Srinath, R., Arslan, E., Xing, A., Xiangli, Y., Radhakrishnan, V.B., Sridhar, S.: Turbo-gs: Accelerating 3d gaussian fitting for high-resolution radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15454–15464 (2026)
8. Dhiman, A., Srinath, R., Rangwani, H., Parihar, R., Boregowda, L.R., Sridhar, S., Babu, R.V.: Strata-nerf: Neural radiance fields for stratified scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17603–17614 (2023)
9. Dhiman, A., Srinath, R., Sarkar, S., Boregowda, L.R., Babu, R.V.: Chromadistill: Colorizing monochrome radiance fields with knowledge distillation (2024), <https://arxiv.org/abs/2309.07668>
10. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)
11. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. arXiv preprint arXiv:2311.17245 (2023)
12. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023)
13. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5501–5510 (2022)
14. Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5712–5721 (2021)

15. Girish, S., Gupta, K., Shrivastava, A.: Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In: European Conference on Computer Vision. pp. 54–71. Springer (2024)
16. Guédon, A., Lepetit, V.: Gaussian frosting: Editable complex radiance fields with real-time rendering. In: European Conference on Computer Vision. pp. 413–430. Springer (2024)
17. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5354–5363 (2024)
18. Guo, X., Sun, J., Dai, Y., Chen, G., Ye, X., Tan, X., Ding, E., Zhang, Y., Wang, J.: Forward flow for novel view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16022–16033 (2023)
19. Guo, Z., Zhou, W., Li, L., Wang, M., Li, H.: Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology* (2024)
20. Hamdi, A., Melas-Kyriazi, L., Mai, J., Qian, G., Liu, R., Vondrick, C., Ghanem, B., Vedaldi, A.: Ges: Generalized exponential splatting for efficient radiance field rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19812–19822 (2024)
21. Henriques, H., de Oliveira, A., Oliveira, E., Trevisan, D., Clua, E.: Foveated path culling: A mixed path tracing and radiance field approach for optimizing rendering in xr displays. *Journal on Interactive Systems* **15**(1), 576–590 (2024)
22. Höllein, L., Božič, A., Zollhöfer, M., Nießner, M.: 3dgs-lm: Faster gaussian-splatting optimization with levenberg-marquardt. *arXiv preprint arXiv:2409.12892* (2024)
23. Hu, W., Wang, Y., Ma, L., Yang, B., Gao, L., Liu, X., Ma, Y.: Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19774–19783 (2023)
24. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 conference papers. pp. 1–11 (2024)
25. Huang, Y.H., He, Y., Yuan, Y.J., Lai, Y.K., Gao, L.: Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18342–18352 (2022)
26. Jung, J., Han, J., An, H., Kang, J., Park, S., Kim, S.: Relaxing accurate initialization constraint for 3d gaussian splatting. *arXiv preprint arXiv:2403.09413* (2024)
27. Katsumata, K., Vo, D.M., Nakayama, H.: A compact dynamic 3d gaussian representation for real-time dynamic view synthesis. In: European Conference on Computer Vision. pp. 394–412. Springer (2024)
28. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139–1 (2023)
29. Kheradmand, S., Rebain, D., Sharma, G., Sun, W., Tseng, J., Isack, H., Kar, A., Tagliasacchi, A., Yi, K.M.: 3d gaussian splatting as markov chain monte carlo. *arXiv preprint arXiv:2404.09591* (2024)
30. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: Proceedings of the European conference on computer vision (ECCV). pp. 170–185 (2018)

31. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5521–5531 (2022)
32. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8508–8520 (2024)
33. Liang, Y., Khan, N., Li, Z., Nguyen-Phuoc, T., Lanman, D., Tompkin, J., Xiao, L.: Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. arXiv preprint arXiv:2312.11458 (2023)
34. Liang, Y., Khan, N., Li, Z., Nguyen-Phuoc, T., Lanman, D., Tompkin, J., Xiao, L.: Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2642–2652. IEEE (2025)
35. Lin, H., Peng, S., Xu, Z., Xie, T., He, X., Bao, H., Zhou, X.: High-fidelity and real-time novel view synthesis for dynamic scenes. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–9 (2023)
36. Liu, J.W., Cao, Y.P., Mao, W., Zhang, W., Zhang, D.J., Keppo, J., Shan, Y., Qie, X., Shou, M.Z.: Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems* **35**, 36762–36775 (2022)
37. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)
38. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 2024 International Conference on 3D Vision (3DV). pp. 800–809. IEEE (2024)
39. Mallick, S.S., Goel, R., Kerbl, B., Steinberger, M., Carrasco, F.V., De La Torre, F.: Taming 3dgs: High-quality radiance fields with limited resources. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
41. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* **41**(4), 1–15 (2022)
42. Navaneet, K., Pourahmadi Meibodi, K., Abbasi Koohpayegani, S., Pirsiavash, H.: Compgs: Smaller and faster gaussian splatting with vector quantization. In: European Conference on Computer Vision. pp. 330–349. Springer (2024)
43. Nyquist, H.: Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers* **47**(2), 617–644 (2009)
44. Paliwal, A., Ye, W., Xiong, J., Kotovenko, D., Ranjan, R., Chandra, V., Kalantari, N.K.: Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In: European Conference on Computer Vision. pp. 19–37. Springer (2024)
45. Papantonakis, P., Kopanas, G., Kerbl, B., Lanvin, A., Drettakis, G.: Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **7**(1), 1–17 (2024)
46. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)* **40**, 1 – 12 (2021), <https://api.semanticscholar.org/CorpusID:235624060>

47. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10318–10327 (2021)
48. Rematas, K., Liu, A., Srinivasan, P.P., Barron, J.T., Tagliasacchi, A., Funkhouser, T., Ferrari, V.: Urban radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12932–12942 (2022)
49. Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., Liu, Y.: Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16632–16642 (2023)
50. Söti, G., Huang, X., Wurl, C., Hein, B.: 6-dof grasp pose evaluation and optimization via transfer learning from nerfs. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 9495–9501. IEEE (2024)
51. Stearns, C., Harley, A., Uy, M., Dubost, F., Tombari, F., Wetzstein, G., Guibas, L.: Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
52. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5459–5469 (2022)
53. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
54. Tu, A., Ying, H., Hanson, A., Lee, Y., Goldstein, T., Zwicker, M.: Speede3dgs: Speedy deformable 3d gaussian splatting with temporal pruning and motion grouping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26083–26093 (2026)
55. Wang, F., Chen, Z., Wang, G., Song, Y., Liu, H.: Masked space-time hash encoding for efficient dynamic scene reconstruction. *Advances in Neural Information Processing Systems* **36** (2024)
56. Wang, H., Zhang, Y., Wang, Y., Li, J.: 6-dof grasp detection in clutter with enhanced receptive field and graspable balance sampling. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3000–3007. IEEE (2024)
57. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems* **36** (2024)
58. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
59. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20310–20320 (2024)
60. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: European conference on computer vision. pp. 106–122. Springer (2022)
61. Xu, J., Fan, Z., Yang, J., Xie, J.: Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) *Advances in Neural Information Processing Systems*. vol. 37, pp. 123787–123811. Curran Associates, Inc. (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/df8b8c12afcaf65639aa686ececbbf00c-Paper-Conference.pdf

62. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5438–5448 (2022)
63. Yan, J., Peng, R., Tang, L., Wang, R.: 4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 7871–7880 (2024)
64. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20331–20341 (2024)
65. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19447–19456 (2024)
66. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields. In: European Conference on Computer Vision. pp. 717–733. Springer (2022)
67. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
68. Zhuang, Y., Zhang, Q., Feng, Y., Zhu, H., Yao, Y., Li, X., Cao, Y.P., Shan, Y., Cao, X.: Anti-aliased neural implicit surfaces with encoding level of detail. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)
69. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: Proceedings Visualization, 2001. VIS’01. pp. 29–538. IEEE (2001)