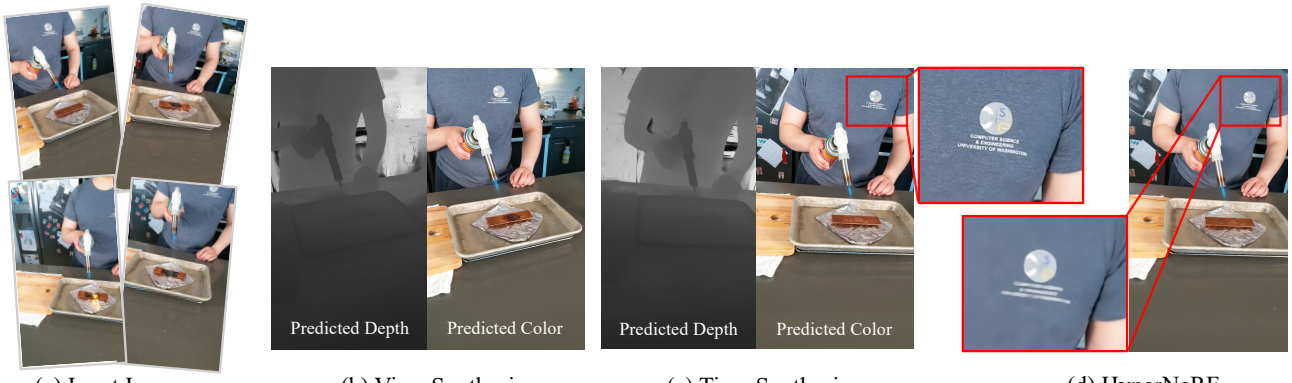
Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction

Ziyi Yang[1,2] Xinyu Gao[1] Wen Zhou[2] Shaohui Jiao[2] Yuqing Zhang[1] Xiaogang Jin[1†]

[1]Zhejiang University [2]ByteDance Inc.



(a) Input Images (b) View Synthesis (c) Time Synthesis (d) HyperNeRF

Figure 1. Given a set of monocular multi-view images and camera poses (a), our proposed method can reconstruct accurate dynamic scene geometry and render high-quality images in both the novel-view synthesis (b) and time interpolation (c) tasks. In real-world datasets with intricate details, our method outperforms *HyperNeRF* [37] (d) in terms of rendering quality and time performance.

Abstract

Implicit neural representation has paved the way for new approaches to dynamic scene reconstruction. Nonetheless, cutting-edge dynamic neural rendering methods rely heavily on these implicit representations, which frequently struggle to capture the intricate details of objects in the scene. Furthermore, implicit methods have difficulty achieving real-time rendering in general dynamic scenes, limiting their use in a variety of tasks. To address the issues, we propose a deformable 3D Gaussians splatting method that reconstructs scenes using 3D Gaussians and learns them in canonical space with a deformation field to model monocular dynamic scenes. We also introduce an annealing smoothing training mechanism with no extra overhead, which can mitigate the impact of inaccurate poses on the smoothness of time interpolation tasks in real-world scenes. Through a differential Gaussian rasterizer, the deformable 3D Gaussians not only achieve higher rendering quality but also real-time rendering speed. Experiments show that our method outperforms existing methods significantly in terms of both rendering quality and speed, making it well-suited for tasks such as novel-view synthesis, time interpolation, and real-time rendering. Our code is available at https://github.com/ingra14m/Deformable-3D-Gaussians.

1. Introduction

High-quality reconstruction and photorealistic rendering of *dynamic scenes* from a set of input images is critical for a variety of applications, including augmented reality/virtual reality (AR/VR), 3D content production, and entertainment. Previously used methods for modeling these dynamic scenes relied heavily on mesh-based representations, as demonstrated by methods described in [10, 18, 23, 46]. However, these strategies frequently face inherent limitations, such as a lack of detail and realism, a lack of semantic information, and difficulties in accommodating topological changes. With the introduction of neural rendering techniques, this paradigm has undergone a significant shift. Implicit scene representations, particularly as implemented by NeRF [34], have demonstrated commendable efficacy in tasks such as novel-view synthesis, scene reconstruction, and light decomposition.

To improve inference efficiency in NeRF-based static

scenes, researchers have developed a variety of acceleration methods, including grid-based structures [7, 9, 15, 52] and pre-computation strategies [50, 59]. Notably, by incorporating hash encoding, Instant-NGP [35] has achieved rapid training. In terms of quality, Mip-NeRF [2] pioneered an effective anti-aliasing method, which was later incorporated into the grid-based approach by Zip-NeRF [4]. 3D-GS [19] recently extended the point-based rendering to efficient CUDA implementation with 3D Gaussians, which has enabled a real-time rendering while matches or even exceeds the quality of Mip-NeRF [2]. However, this method is designed for modeling static scenes, and its highly customized rasterization pipeline diminishes its scalability.

Implicit representations have been increasingly harnessed for modeling dynamic scenes. To handle the motion part in a dynamic scene, entangled methods [49, 56] conditioned NeRF on a time variable. Conversely, disentangled methods [28, 36, 37, 40, 45] employ a deformation field to model a scene in canonical space by mapping point coordinates at a given time to this space. This decoupled modeling approach can effectively represent scenes with non-dramatic action variations. However, irrespective of the categorization, adopting an implicit representation for dynamic scenes often proves both inefficient and ineffective, manifesting slow convergence rates coupled with a marked susceptibility to overfitting. Drawing inspiration from seminal NeRF acceleration research, numerous studies on dynamic scene modeling have integrated discrete structures, such as voxel-grids [12, 44], or planes [6, 42]. This integration amplifies both training speed and modeling accuracy. However, challenges remain. Techniques leveraging discrete structures still grapple with the dual constraints of achieving real-time rendering speeds and producing high-quality outputs with adequate detail. Multiple facets underpin these challenges: Firstly, ray-casting, as a rendering modality, frequently becomes inefficient, especially when scaled to higher resolutions. Secondly, grid-based methods rely on a low-rank assumption. Dynamic scenes, in comparison to static ones, exhibit a higher rank, which hampers the upper limit of quality achievable by such approaches.

In this paper, to address the aforementioned challenges, we extend the static 3D-GS and propose a deformable 3D Gaussian framework for modeling dynamic scenes. To enhance the applicability of the model, we specifically focus on the modeling of monocular dynamic scenes. Rather than reconstructing the scene frame by frame [31], we condition the 3D Gaussians on time and jointly train a purely implicit deformation field with the learnable 3D Gaussians in canonical space. The gradients for these two components are derived from a customized differential Gaussian rasterization pipeline. Furthermore, to solve the jitter in temporal sequences during the reconstruction process caused by inaccurate poses, we incorporate an annealing smoothing training (AST) mechanism. This strategy not only improves the smoothness between frames in the time interpolation task but also allows for greater rendering details.

In summary, the major contributions of our work are:

- A deformable 3D-GS framework for modeling monocular dynamic scenes that can achieve real-time rendering and high-fidelity scene reconstruction.
- A novel annealing smoothing training mechanism that ensures temporal smoothness while preserving dynamic details without increasing computational complexity.
- The first framework to extend 3D-GS for dynamic scenes through a deformation field, enabling the learning of 3D Gaussians in canonical space.

2. Related Work

2.1. Neural Rendering for Dynamic Scenes

Neural rendering, due to its unparalleled capability to generate photorealistic images, has seen an uptick in scholarly interest. Recently, NeRF [34] facilitates photorealistic novel view synthesis through the use of MLPs. Subsequent research has expanded the utility of NeRF to various applications, encompassing tasks such as mesh reconstruction from a collection of images [25, 32, 51, 55], inverse rendering [5, 30, 62], optimization of camera parameters [26, 53, 54], few-shot learning [11, 58], and editing [16, 17, 60].

Constructing radiance fields for dynamic scenes is a critical branch in the advancement of NeRF, with significant implications for real-world applications. A cardinal challenge in rendering these dynamic scenes lies in the encoding and effective utilization of temporal information, especially when addressing the reconstruction of monocular dynamic scenes, a task inherently involves sparse reconstruction from a single viewpoint. One class of dynamic NeRF approaches models scene deformation by adding time t as an additional input to the radiance field. However, this strategy couples the positional variations induced by temporal changes with the radiance field, lacking the geometric prior information regarding the influence of time on the scene. Consequently, substantial regularization is required to ensure temporal consistency in the rendering results. Another category of methods [36, 37, 40] introduces a deformation field to decouple time and the radiance field, mapping point coordinates to the canonical space corresponding to time t through the deformation field. This decoupled approach is conducive to the learning of pronounced rigid motions and is versatile enough to cater to scenes undergoing topological shifts. Other methods seek to enhance the quality of dynamic neural rendering from various aspects, including segmenting static and dynamic objects in the scene [45, 48], incorporating depth information [1] to introduce geometric prior, introducing 2D CNN to encode scene priors [27, 39], and leveraging the redundant information in

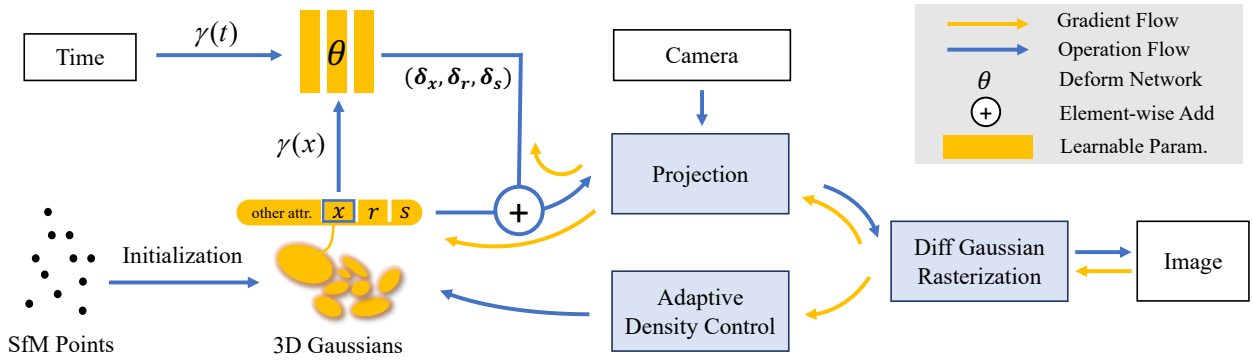


Figure 2. **Overview of our pipeline.** The optimization process begins with Structure from Motion (SfM) points derived from COLMAP or generated randomly, which serve as the initial state for the 3D Gaussians. We use the position (detached) of 3D Gaussians $\gamma(\mathrm{sg}(\boldsymbol{x}))$ and time $\gamma(t)$ with positional encoding as input to a deformation MLP network to obtain the offset $(\delta\boldsymbol{x}, \delta\boldsymbol{r}, \delta\boldsymbol{s})$ of dynamic 3D Gaussians in canonical space. We use a warm-up phase for the 3D Gaussians during the first 3k iterations without optimizing the deformation field. Following that, we use the fast differential Gaussian rasterization pipeline to perform joint optimization of the deformation field and the 3D Gaussians, as well as to adaptively control the density of the set of Gaussians.

multi-view videos [24] to set up keyframe compression storage, thereby accelerating the rendering speed.

However, the rendering quality of existing dynamic scene modeling based on MLP (Multilayer Perceptron) remains unsatisfactory. In this work, we will focus on the reconstruction of monocular dynamic scenes. We continue to decouple the deformation field and the radiance field. To enhance the editability and rendering quality of intermediate states in dynamic scenes, we have adapted this modeling approach to fit within the framework of differentiable point-based rendering.

2.2. Acceleration of Neural Rendering

Real-time rendering has long been a pivotal objective in the field of computer graphics, a goal that is also pursued in the domain of neural rendering. Numerous studies dedicated to NeRF acceleration have meticulously navigated the trade-off between spatial and temporal efficiency.

Pre-computed methods [13, 41] utilize spatial acceleration structures such as spherical harmonics coefficients [59] and feature vectors [14], cached or distilled from implicit neural representation, as opposed to directly employing the neural representations themselves. A prominent technique [8] in this category transforms NeRF scenes into an amalgamation of coarse meshes and feature textures, thereby enhancing rendering velocity in contemporary mobile graphics pipelines. However, this pre-computed approach may necessitate significant storage capacities for individual scenes. While it offers advantages in terms of inference speed, it demands protracted training durations and exhibits considerable overhead.

Hybrid methods [4, 7, 29, 33, 47, 52] incorporate a neural component within the explicit grid. The hybrid approaches confer the dual benefits of expediting both training and inference phases while producing outcomes on par with advanced frameworks [2, 3]. This is primarily attributed to the robust representational capabilities of the grid. This grid or plane-based strategy has been extended to the acceleration [12] or representation of time-conditioned 4D feature [6, 42, 44] in dynamic scene modeling and time-conditioned compact 4D dynamic scene modeling.

Recently, several studies [20, 21, 61] have evolved the continuous radiance field from implicit representations to differentiable point-based radiance fields, markedly enhancing the rendering speed. 3D-GS [19] further innovates point-based rendering by introducing a customized CUDA-based differentiable Gaussian rasterization pipeline. This approach not only achieves superior outcomes in tasks like novel-view synthesis, but also facilitates rapid training times on the order of minutes, and supports real-time rendering surpassing 100 FPS. However, the method employs a customized differential Gaussian rasterization pipeline, which complicates its direct extension to dynamic scenes. Inspired by this, our work will leverage the point-based rendering framework, 3D-GS, to expedite both the training and rendering speeds for dynamic scene modeling.

3. Method

The overview of our method is illustrated in Fig. 2. The input to our method is a set of images of a monocular dynamic scene, together with the time label and the corresponding camera poses calibrated by SfM [43] which also produces a sparse point cloud. From these points, we create a set of 3D Gaussians $G(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{s}, \sigma)$ defined by a center position $\boldsymbol{x}$, opacity σ, and 3D covariance matrix Σ obtained from quaternion $\boldsymbol{r}$ and scaling $\boldsymbol{s}$. The view-dependent ap-

pearance of each 3D Gaussian is represented via spherical harmonics (SH). To model the dynamic 3D Gaussians that vary over time, we decouple the 3D Gaussians and the deformation field. The deformation field takes the positions of the 3D Gaussians and the current time t as inputs, outputting $\delta \boldsymbol{x}$, $\delta \boldsymbol{r}$, and $\delta \boldsymbol{s}$. Subsequently, we put the deformed 3D Gaussians $G(\boldsymbol{x}+\delta \boldsymbol{x}, \boldsymbol{r}+\delta \boldsymbol{r}, \boldsymbol{s}+\delta \boldsymbol{s}, \sigma)$ into the efficient differential Gaussian rasterization pipeline, which is a tile-based rasterizer that allows α-blending of anisotropic splats. The 3D Gaussians and deformation network are optimized jointly through the fast backward pass by tracking accumulated α values, together with the adaptive control of the Gaussian density. Experimental results show that after 30k training iterations, the shape of the 3D Gaussians stabilizes, as does the canonical space, which indirectly proves the efficacy of our design.

3.1. Differentiable Rendering Through 3D Gaussians Splatting in Canonical Space

To optimize the parameters of 3D Gaussians in canonical space, it is imperative to differentially render 2D images from these 3D Gaussians. In this work, we employ the differential Gaussian rasterization pipeline proposed by [19]. Following [63], the 3D Gaussians can be projected to 2D and rendered for each pixel using the following 2D covariance matrix Σ':

$$\Sigma' = JV\Sigma V^T J^T, \tag{1}$$

where J is the Jacobian of the affine approximation of the projective transformation, V symbolizes the view matrix, transitioning from world to camera coordinates, and Σ denotes the 3D covariance matrix.

To make learning the 3D Gaussians easier, Σ is divided into two learnable components: the quaternion $\boldsymbol{r}$ represents rotation, and the 3D-vector $\boldsymbol{s}$ represents scaling. These components are then transformed into the corresponding rotation and scaling matrices R and S. The resulting Σ can be expressed as:

$$\Sigma = RSS^T R^T. \tag{2}$$

The color of the pixel on the image plane, denoted by $\mathbf{p}$, is rendered sequentially with point-based volume rendering:

$$\begin{aligned} C(\mathbf{p}) &= \sum_{i \in N} T_i \alpha_i c_i, \\ \alpha_i &= \sigma_i e^{-\frac{1}{2}(\mathbf{p}-\mu_i)^T \Sigma'(\mathbf{p}-\mu_i)}, \end{aligned} \tag{3}$$

where T_i is the transmittance defined by $\Pi_{j=1}^{i-1}(1-\alpha_j)$, c_i signifies the color of the Gaussians along the ray, and μ_i represents the uv coordinates of the 3D Gaussians projected onto the 2D image plane.

During the optimization, adaptive density control emerges as a pivotal component, enabling the rendering of 3D Gaussians to achieve desirable outcomes. This control serves a dual purpose: firstly, it mandates the pruning of transparent Gaussians based on σ. Secondly, it necessitates the densification of Gaussian distribution. This densification fills regions void of geometric intricacies, while simultaneously subdividing areas where Gaussians are large and exhibit significant overlap. Notably, such areas tend to display pronounced positional gradients. Following [19], we discern the 3D Gaussians that demand adjustments using a threshold given by $t_{pos} = 0.0002$. For diminutive Gaussians inadequate for capturing geometric details, we clone the Gaussians and move them a certain distance in the direction of the positional gradients. Conversely, for those that are conspicuously large and overlapping, we split them and divide their scale.

It is clear that 3D Gaussians are only appropriate for static scenes. Applying a time-conditioned learnable parameter for each 3D Gaussian not only contradicts the original intent of the differentiable Gaussian rasterization pipeline, but also results in the loss of spatiotemporal continuity of motion. To enable 3D Gaussians to model dynamic scenes while retaining the physical meaning, we decided to learn 3D Gaussians in canonical space and use an additional deformation field to learn the position and shape variations.

3.2. Deformable 3D Gaussians

An intuitive solution to model dynamic scenes using 3D Gaussians is to separately train 3D-GS set in each time-dependent view collection and then perform interpolation between these sets as a post-processing step. While such an approach is feasible for Multi-View Stereo (MVS) captures at discrete time, it falls short for continuous monocular captures within a temporal sequence. To deal with the latter, a more general case, we jointly learn a deformation field along with 3D Gaussians.

We decouple the motion and static structure by leveraging a deformation network, converting the time-independent 3D Gaussians optimization into a canonical space. This decoupling approach introduces geometric priors of the scene, associating the changes in the positions of the 3D Gaussians with both time and coordinates. The core of the deformation network is an MLP. In our study, we did not employ the grid/plane structures applied in static NeRF that can accelerate rendering and enhance its quality. This is because such methods operate on a **low-rank assumption**, whereas dynamic scenes possess a higher rank. Moreover, we believe that explicit structures are appropriate for directly modeling 4D scenes, while smooth MLP is more suited for a temporally smooth deformation field related to a canonical space, leading to the performance gap.

Given time t and center position $\boldsymbol{x}$ of 3D Gaussians as inputs, the deformation MLP produces offsets, which subsequently transform the canonical 3D Gaussians to the de-

Method	Hell Warrior PSNR↑	SSIM↑	LPIPS↓	Mutant PSNR↑	SSIM↑	LPIPS↓	Hook PSNR↑	SSIM↑	LPIPS↓	Bouncing Balls PSNR↑	SSIM ↑	LPIPS↓
3D-GS	29.89	0.9155	0.1056	24.53	0.9336	0.0580	21.71	0.8876	0.1034	23.20	0.9591	0.0600
D-NeRF	24.06	0.9440	0.0707	30.31	0.9672	0.0392	29.02	0.9595	0.0546	38.17	0.9891	0.0323
TiNeuVox	27.10	0.9638	0.0768	31.87	0.9607	0.0474	30.61	0.9599	0.0592	40.23	0.9926	0.0416
Tensor4D	31.26	0.9254	0.0735	29.11	0.9451	0.0601	28.63	0.9433	0.0636	24.47	0.9622	0.0437
K-Planes	24.58	0.9520	0.0824	32.50	0.9713	0.0362	28.12	0.9489	0.0662	40.05	0.9934	0.0322
Ours	41.54	0.9873	0.0234	42.63	0.9951	0.0052	37.42	0.9867	0.0144	41.01	0.9953	0.0093
Method	Lego PSNR↑	SSIM↑	LPIPS↓	T-Rex PSNR↑	SSIM↑	LPIPS↓	Stand Up PSNR↑	SSIM↑	LPIPS↓	Jumping Jacks PSNR↑	SSIM ↑	LPIPS↓
3D-GS	22.10	0.9384	0.0607	21.93	0.9539	0.0487	21.91	0.9301	0.0785	20.64	0.9297	0.0828
D-NeRF	25.56	0.9363	0.0821	30.61	0.9671	0.0535	33.13	0.9781	0.0355	32.70	0.9779	0.0388
TiNeuVox	26.64	0.9258	0.0877	31.25	0.9666	0.0478	34.61	0.9797	0.0326	33.49	0.9771	0.0408
Tensor4D	23.24	0.9183	0.0721	23.86	0.9351	0.0544	30.56	0.9581	0.0363	24.20	0.9253	0.0667
K-Planes	28.91	0.9695	0.0331	30.43	0.9737	0.0343	33.10	0.9793	0.0310	31.11	0.9708	0.0468
Ours	33.07	0.9794	0.0183	38.10	0.9933	0.0098	44.62	0.9951	0.0063	37.72	0.9897	0.0126

Table 1. **Quantitative comparison on synthetic dataset**. We compare our method to several previous approaches: 3D-GS [19], D-NeRF [40], TiNeuVox [12], Tensor4D [44] and K-Planes [42] on full resolution (800x800) test images. This may cause some methods to perform worse than the original paper because they downsample images by default. We report PSNR, SSIM, LPIPS(VGG) and color each cell as best, second best and third best. It is worth noting that we observed a discrepancy in the scenarios presented in the training and test sets of the Lego in D-NeRF dataset. This can be substantiated by examining the flip angles of the Lego shovels. To ensure a meaningful comparison, we opted to utilize the validation set of Lego as the test set in our experiments. See more in supplementary materials.

formed space:

$$(\delta \boldsymbol{x}, \delta \boldsymbol{r}, \delta \boldsymbol{s}) = \mathcal{F}_\theta(\gamma(\text{sg}(\boldsymbol{x})), \gamma(t)), \tag{4}$$

where $\text{sg}(\cdot)$ indicates a stop-gradient operation, γ denotes the positional encoding:

$$\gamma(p) = (sin(2^k \pi p), cos(2^k \pi p))_{k=0}^{L-1}, \tag{5}$$

where $L = 10$ for x and $L = 6$ for t in synthetic scenes, while $L = 10$ for both x and t in real scenes. We set the depth of the deformation network $D = 8$ and the dimension of the hidden layer $W = 256$. Experiments demonstrate that applying positional encoding to the inputs of the deformation network can enhance the details in rendering results.

3.3. Annealing Smooth Training

A prevalent challenge with numerous real-world datasets is the inaccuracies in pose estimation, a phenomenon markedly evident in dynamic scenes. Training under imprecise poses can lead to overfitting on the training data. Moreover, as also mentioned in HyperNeRF [37], the imprecise poses from colmap for real datasets can cause spatial jitter between each frame w.r.t. the test or train set, resulting in a noticeable deviation when rendering the test image compared to the ground truth. Previous methods that used implicit representations benefited from the MLP's inherent smoothness, making the impact of such minor offsets on the final rendering results relatively inconspicuous. However, explicit point-based rendering tends to amplify this effect. For monocular dynamic scenes, novel-view rendering at a fixed time remains unaffected. However, for the task involving interpolated time, this kind of inconsistent scene at different times can lead to irregular rendering jitters.

To address this issue, we propose a novel annealing smooth training (AST) mechanism specifically designed for real-world monocular dynamic scenes:

$$\begin{aligned} \boldsymbol{\Delta} &= \mathcal{F}_{\boldsymbol{\theta}}\left(\boldsymbol{\gamma}(\text{sg}(\boldsymbol{x})), \boldsymbol{\gamma}(t) + \mathcal{X}(i)\right), \\ \mathcal{X}(i) &= \mathbb{N}(0,1) \cdot \beta \cdot \Delta t \cdot \max((1 - i/\tau), 0), \end{aligned} \tag{6}$$

where $\mathcal{X}(i)$ represents the linearly decaying Gaussian noise at the i-th training iteration, $\mathbb{N}(0,1)$ denotes the standard Gaussian distribution, β is an empirically determined scaling factor with a value of 0.1, Δt represents the mean time interval, and τ is the threshold iteration for annealing smooth training (empirically set to $20k$).

Compared to the smooth loss introduced by methods of [40, 44], our approach does not incur additional computational overhead. It can enhance the model's temporal generalization in the early stages of training, as well as prevent excessive smoothing in the later stages, thus preserving the details of objects in dynamic scenes. Concurrently, it reduces the jitter observed in real-world datasets during time interpolation tasks.

4. Experiment

In this section, we present the experimental evaluation of our method. To give proof of effectiveness, we evaluate our approach on the benchmark which consists of the synthetic dataset from D-NeRF [40] and real-world datasets sourced from HyperNeRF [37] and NeRF-DS [57]. The division on training and testing part, as well as the image resolution, aligns perfectly with the original paper.

Figure 3. **Qualitative comparisons of baselines and our method on monocular synthetic dataset.** We visualize each scene using baselines and our method. Experimental results indicate that our approach recovers more details when rendering novel viewpoints and can reconstruct more delicate structures over time, such as hands or skeletons. The efficacy of Deformable-GS can be attributed to its capability to equally back-propagate the gradient to both the Deformation Field and the 3D Gaussians. Larger gradients in the dynamic portion of the Deformation Field can further assist the 3D Gaussians in achieving better densification in dynamic regions.

(a) 6pe w/o AST (b) 10pe w/o AST (c) ours

Figure 4. **Ablation study.** We conduct ablation studies focusing on the annealing smooth training scheme within real-world datasets, wherein *pe* signifies the positional encoding over time. Compared with the reduced order (a) and the original order (b) of positional encoding over time, it becomes evident that the annealing smooth training strategy (c) effectively preserves high-frequency information. Simultaneously, it mitigates the temporal overfitting challenges instigated by imprecise pose estimations.

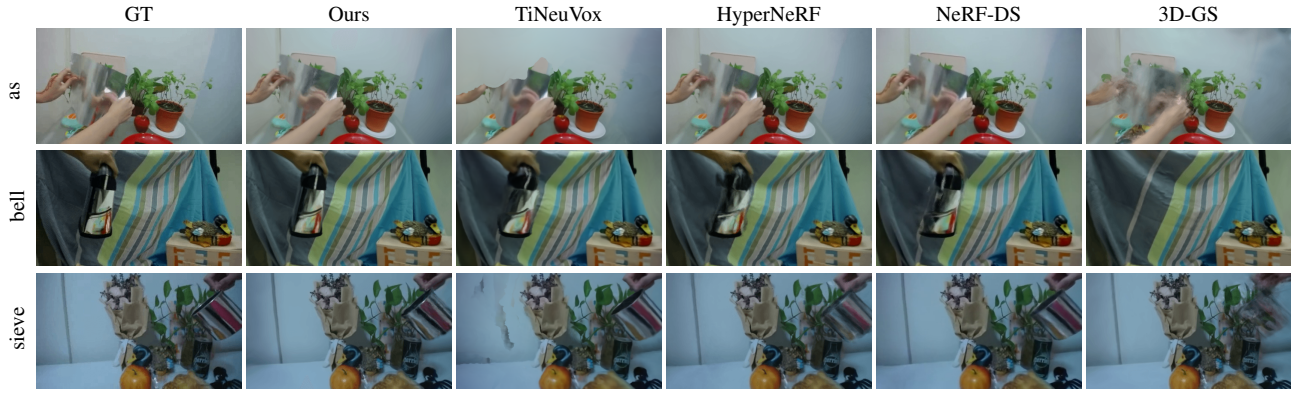


Figure 5. **Qualitative comparisons of baselines and our method on NeRF-DS real-world dataset.** Experimental results indicate that our method can achieve superior rendering quality on real-world datasets where the pose is not absolutely precise.

4.1. Implementation Details

We implement our framework using PyTorch [38] and modify the differentiable Gaussian rasterization by incorporating depth visualization. For training, we conducted training for a total of 40k iterations. During the initial 3k iterations, we solely trained the 3D Gaussians to attain relatively stable positions and shapes. Subsequently, we jointly train the 3D Gaussians and the deformation field. For optimization, a single Adam optimizer [22] is used but with a different learning rate for each component: the learning rate of 3D Gaussians is exactly the same as the official implementation, while the learning rate of the deformation network undergoes exponential decay, ranging from 8e-4 to 1.6e-6. Adam's β value range is set to (0.9, 0.999). Experiments with synthetic datasets were all conducted against a black background and at a full resolution of 800x800. All the experiments were done on an NVIDIA RTX 3090.

4.2. Results and Comparisons

Comparisons on Synthetic Dataset. In our experiments, we benchmarked our method against several baselines using the monocular synthetic dataset introduced by D-NeRF [40]. The quantitative evaluation, presented in Tab. 1, offers compelling evidence of the superior performance of our approach over the current state-of-the-art. Notably, metrics pertinent to structural consistency, such as LPIPS and SSIM, demonstrate our method's pronounced superiority.

For a more visual assessment, we provide qualitative results in Fig. 3. These visual comparisons underscore the capability of our method in delivering high-fidelity dynamic scene modeling. It's evident from the results that our approach ensures enhanced consistency and captures intricate rendering details in novel-view renderings.

Comparisons on Real-world Dataset. We compare our method with the baselines using the monocular real-world dataset from NeRF-DS [57] and HyperNeRF [37]. It should be noted that the camera poses for some of the scenes in HyperNeRF are very inaccurate. Given that metrics like PSNR are inclined to penalize slight deviations more than blurring, we have refrained from incorporating HyperNeRF in our quantitative analysis. The quantitative and qualitative evaluations for the NeRF-DS dataset are detailed in Tab. 2

and Fig. 5, respectively. These results attest to the robustness of our method when applied to real-world scenes, even when the associated poses are not perfectly accurate.

Rendering Efficiency. The rendering speed is correlated with the quantity of 3D Gaussians. Overall, when the number of 3D Gaussians is below 250k, our method can achieve real-time rendering over 30 FPS on an NVIDIA RTX 3090. Detailed results can be found in the supplementary material.

Depth Visualization. We visualized the depth of synthetic scenes in Fig. 6 to demonstrate that our deformation network is well fitted to produce temporal transformation rather than relying on color-based hard-coding. The precise depth underscores the accuracy of our geometric reconstruction, proving highly advantageous for the novel-view synthesis task.

4.3. Ablation Study

Annealing Smooth Training. As illustrated in Fig. 4 and Tab. 2, AST fosters improved convergence towards intricate regions, effectively mitigating the overfitting tendencies in real-world datasets. Furthermore, it is unequivocally clear from our observations that this strategy significantly bolsters the temporal smoothness of the deformation field. See more ablations in supplementary materials.

5. Limitations

Through our experimental evaluations, we observed that the convergence of 3D Gaussians is profoundly influenced by the diversity of perspectives. As a result, datasets characterized by sparse viewpoints and limited viewpoint coverage may lead our method to encounter overfitting challenges. Additionally, the efficacy of our approach is contingent upon the accuracy of pose estimations. This dependency was evident when our method did not achieve optimal PSNR values on the Nerfies/HyperNeRF dataset, attributable to deviations in pose estimation via COLMAP. Furthermore, the temporal complexity of our approach is directly proportional to the quantity of 3D Gaussians. In scenarios with an extensive array of 3D Gaussians, there is a potential escalation in both training duration and memory consumption. Lastly, our evaluations have predominantly revolved around scenes with moderate motion dynamics. The method's adeptness at handling intricate human motions, such as nuanced facial expressions, remains an open question. We perceive these constraints as promising directions for subsequent research endeavors.

6. Conclusions

We introduce a novel deformable 3D Gaussian splatting method, specifically designed for monocular dynamic scene modeling, which surpasses existing methods in both quality and speed. By learning the 3D Gaussians in canonical space, we enhance the versatility of the 3D-GS differentiable rendering pipeline for dynamically captured monocular scenes. It's crucial to recognize that point-based methods, in comparison to implicit representations, are more editable and better suited for post-production tasks. Additionally, our method incorporates an annealing smooth training strategy, aimed at reducing overfitting associated with time encoding while maintaining intricate scene details, without adding any extra training overhead. Experimental results demonstrate that our method not only achieves superior rendering effects but is also capable of real-time rendering.

	PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS	20.29	0.7816	0.2920
TiNeuVox	21.61	0.8234	0.2766
HyperNeRF	23.45	0.8488	0.1990
NeRF-DS	23.60	0.8494	0.1816
Ours (w/o AST)	23.97	0.8346	0.2037
Ours	24.11	0.8525	0.1769

Table 2. **Metrics on NeRF-DS dataset**. We computed the mean of the metrics across all seven scenes. Cells are highlighted as follows: best , second best , and third best . For individual metrics about each scene, please refer to the supplementary material.

Figure 6. **Depth Visualization.** We visualized the depth map of the D-NeRF dataset. The first row includes bouncing-balls, hell-warrior, hook, and jumping-jacks, while the second row includes lego, mutant, standup, and trex.

7. Acknowledgments

Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant No. 62036010), the Key R&D Program of Zhejiang (No. 2023C01047), and the FDCT under Grant 0002/2023/AKP. We sincerely thank the support of Bytedance MMLab, as well as Zihao Wang for the debugging in the early stage, preventing this work from sinking. We are also very grateful to Yi-Hua Huang, Yang-Tian Sun, Xiaoyang Lyu, and Xiaojuan Qi from the University of Hong Kong for their kind support and help.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in Neural Information Processing Systems*, 34:26289–26301, 2021. 2

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 2, 3

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 3

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2, 3

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. *CVPR*, 2023. 2, 3

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, pages 333–350, 2022. 2, 3

[8] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16569–16578, 2023. 3

[9] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4182–4194, 2023. 2

[10] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 1

[11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[12] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2, 3, 5

[13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 3

[14] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 3

[15] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 2

[16] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[17] Yi-Hua Huang, Yan-Pei Cao, Yu-Kun Lai, Ying Shan, and Lin Gao. Nerf-texture: Texture synthesis with neural radiance fields. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[18] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997. 1

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 2, 3, 4, 5

[20] Leonid Keselman and Martial Hebert. Approximate differentiable rendering with algebraic surfaces. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[21] Leonid Keselman and Martial Hebert. Flexible Techniques for Differentiable Rendering with 3D Gaussians, 2023. 3

[22] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 7

[23] Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, and Szymon Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics (TOG)*, 31(1):1–11, 2012. 1

[24] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 3

[25] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[26] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[27] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia Conference Proceedings*, 2022. 2

[28] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie,

and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022. 2

[29] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[30] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In *SIGGRAPH*, 2023. 2

[31] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *preprint*, 2023. 2

[32] Xiaoyang Lyu, Peng Dai, Zizhang Li, Dongyu Yan, Yi Lin, Yifan Peng, and Xiaojuan Qi. Learning a room with the occ-sdf hybrid: Signed distance function mingled with occupancy aids scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8940–8950, 2023. 2

[33] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (SIGGRAPH)*, 40(4), 2021. 3

[34] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41 (4):102:1–102:15, 2022. 2

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

[37] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics*, 40(6):1–12, 2021. 1, 2, 5, 7

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[39] Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Representing volumetric videos as dynamic mlp maps. In *CVPR*, 2023. 2

[40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 5, 7

[41] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 3

[42] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2, 3, 5

[43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4104–4113, 2016. 3

[44] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2, 3, 5

[45] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 2

[46] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. 1

[47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3

[48] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 2

[49] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 2

[50] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13524–13534, 2022. 2

[51] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2

[52] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. *CVPR*, 2023. 2, 3

[53] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4170–4179, 2023. 2

[54] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[55] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[56] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 2

[57] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 5, 7

[58] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[59] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2, 3

[60] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: Geometry editing of neural radiance fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[61] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. *arXiv preprint arXiv:2205.14330*, 2022. 3

[62] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2

[63] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 4