Multi-Step Deformable Gaussian Splatting for Dynamic Scene Rendering

Jiaheng Hu[1], **Zhizhong Zhang**[1, 3], **Jingyu Gong**[1, 2, 3*], **Lizhuang Ma**[1, 4*], **Xin Tan**[1, 2], **Yuan Xie**[1, 2]

[1]School of Computer Science and Technology, East China Normal University, Shanghai, China
[2]Chongqing Key Laboratory of Precision Optics, Chongqing Institute of East China Normal University, Chongqing, China
[3]Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai, China
[4]School of Computer Science, Shanghai Jiao Tong University, Shanghai, China
51265901077@stu.ecnu.edu.cn, {zzzhang, jygong, lzma, xtan, yxie}@cs.ecnu.edu.cn

Abstract

Reconstructing dynamic scenes has long been a challenging task in 3D vision. Previous mainstream methods based on 3D Gaussian Splatting typically employ a single deformation field to directly model spatiotemporal changes. However, such one-step deformation struggles to capture diverse and complex motion patterns. To address this limitation, we propose decomposing the one-step deformation into a multi-step process, where each step is represented by a deformation layer. Additionally, we introduce a weight prediction mechanism for each layer to control the extent of deformation at every step. We provide two types of deformation layers based on implicit and explicit approaches. Moreover, while the deformation layer is time-conditioned, the Gaussians' behavior may still be influenced by their time-invariant properties. Therefore, we propose a fully time-agnostic scale modulation block to modulate the scaling changes of Gaussians. Extensive experiments on D-NeRF, Neu3D, and HyperNeRF demonstrate that our method achieves state-of-the-art performance.

1 Introduction

Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) have achieved significant advancements in novel view synthesis with the ability to accurately represent complex geometries and appearance properties. However, the real world is predominantly dynamic rather than static, thus rendering dynamic scenes presents a significant challenge, where both 3DGS (Kerbl et al. 2023) and NeRF (Mildenhall et al. 2021) often assume the geometry and appearance of the scene remain constant across all input views.

In this context, some research (Park et al. 2021a,b; Liu et al. 2023; Wang et al. 2023) has extended NeRF to dynamic scenes by employing a deformation Multi-Layer Perceptron (MLP), which constructs a warp field based on spatiotemporal coordinates, projecting 3D points from the observation space into the canonical NeRF field. Despite these efforts, NeRF-based methods suffer from slow training convergence. Numerous 3DGS-based dynamic methods (Yang et al. 2024a; Huang et al. 2024; Liang et al. 2025a) therefore emerged, which similarly introduce additional deformation fields to model the temporal changes in Gaussian coordinates and other attributes.

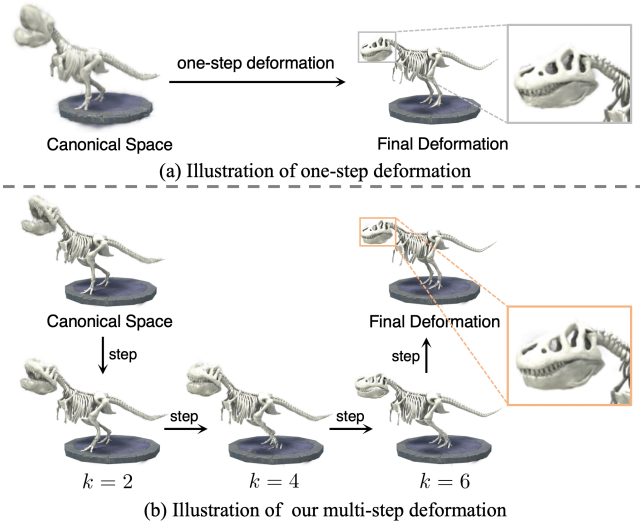


Figure 1: (a) Illustration of the previous one-step deformation method. (b) Illustration of the intermediate deformation stages from the canonical field to the target time. The images for $k = 2$, $k = 4$, and $k = 6$ show the intermediate results after 2, 4, and 6 deformation layers, respectively. Our method excels in capturing fine details. For comparison, both approaches employ MLPs to fit the deformation.

In such methods, learning an effective deformation field is crucial, especially when scenes are complex and exhibit various types of motion, such as rigid or non-rigid transformations, large-scale movements, and subtle changes. However, current deformation-based approaches often rely on a single deformation field to model the entire 4D spatiotemporal scene. For example, DeformGS (Yang et al. 2024a) and 4DGS (Wu et al. 2024) typically employ a direct mapping from (x, y, z, t) to changes in Gaussian attributes. This one-step deformation attempts to fit all scene variations at once, therefore struggles to capture diverse motions and high-frequency details, and lacks sufficient expressive power.

To address this limitation, we propose decomposing the

*Corresponding Author.

deformation field into a series of deformation steps. Each step is modeled by a small deformation field, referred to as a deformation layer. By stacking these deformation fields, we progressively achieve the final deformation by transforming the direct mapping into a more refined process.

However, naive stacking of multiple deformation fields often results in a degenerate composition, wherein the cumulative transformation collapses into an effective single-layer deformation, due to the lack of expressive capacity and the uniform impact in the stacked modules, leading to redundant displacement patterns that fail to capture hierarchical or progressive spatial transformations. To address this, we introduce a contribution weight prediction mechanism for each layer, allowing the model to adaptively determine the extent of deformation at each step. This enhanced deformation field demonstrates stronger capabilities in capturing complex motion and preserving fine details. Figure 1 illustrates the deformation processes of our approach compared to previous methods. For comparison, both approaches employ MLPs to fit the deformation functions. The canonical space learned through multi-step deformation appears significantly clearer, while one-step deformation results in a more chaotic 3D Gaussian distribution.

Besides, current methods often overlook time-invariant factors when modeling spatiotemporal changes, relying instead on time-dependent deformation fields to capture transformations. However, different objects exhibit distinct properties that influence their deformation, particularly in terms of shape changes. For instance, non-rigid objects undergo shape changes during movement, while rigid bodies maintain their shape. To address this, we explicitly model time-invariant properties by attaching a learnable embedding to each Gaussian and introduce a fully spatiotemporal-agnostic module to modulate the scaling changes of Gaussians during multi-step deformation, as the scaling matrix of Gaussians determines their shape. Since its forward is consistent across all time steps and depends only on the Gaussians' additional properties, it can be removed after training, with a low-dimensional modulation vector cached for efficient inference.

Our contributions are summarized as follows:

- We introduce a multi-step deformation field with lightweight layers, each enhanced by adaptive control over the deformation process, enabling precise and flexible modeling of complex spatiotemporal dynamics.
- Our time-agnostic scale modulation module explicitly decouples time-invariant properties from dynamic motion, enabling more fine-grained motion modeling.
- Through experiments on D-NeRF (Pumarola et al. 2021), Neu3D (Li et al. 2022), and HyperNeRF (Park et al. 2021b) datasets, we demonstrate the effectiveness of our approach, achieving state-of-the-art (SOTA) performance compared to previous deformation methods.

2 Related Work

2.1 Novel View Synthesis

Novel view synthesis aims to generate unseen perspectives from multi-view images. NeRF (Mildenhall et al. 2021) pioneered a new implicit scene reconstruction for photorealistic rendering. However, dense point sampling and a large MLP result in high training costs and low computational efficiency. Thus, (Garbin et al. 2021; Reiser et al. 2021; Gong et al. 2021; Fridovich-Keil et al. 2022) focus on accelerating training or inference for faster rendering. The emergence of 3DGS (Kerbl et al. 2023) marks significant progress, using explicit representation and highly parallel rendering pipelines to achieve better quality than NeRF (Mildenhall et al. 2021) and enable real-time rendering. Subsequent works, such as (Yu et al. 2024; Yan et al. 2024; Liang et al. 2025b), further improve rendering quality while some studies (Radl et al. 2024; Hamdi et al. 2024) focus on more efficient rendering techniques. Today, 3DGS (Kerbl et al. 2023) is widely applied in various 3D vision tasks, such as human reconstruction (Moreau et al. 2024; Kocabas et al. 2024), autonomous driving (Zhou et al. 2024; Tian et al. 2024), and AIGC (Tang et al. 2024; Ling et al. 2024).

2.2 Dynamic Scene Rendering

Extending novel view synthesis to dynamic scenes is a significant challenge due to complex motions. NeRF-based methods like (Pumarola et al. 2021; Park et al. 2021a,b; Liu et al. 2023; Wang et al. 2023) employ a deformable field to warp 3D coordinates back to a canonical space. To accelerate NeRF, approaches leverage hybrid grid-MLP representations like TiNeuVox (Fang et al. 2022) or decompose spacetime into explicit planes (Cao and Johnson 2023; Fridovich-Keil et al. 2023; Shao et al. 2023).

Recent work leveraging 3DGS (Kerbl et al. 2023) for dynamic scenes falls into two main categories. The first type directly learns 4D representations, for instance, by using spacetime Gaussian primitives (Yang et al. 2024b; Duan et al. 2024; Li et al. 2024). The other, often more effective category, comprises deformation-based methods (Yang et al. 2024a; Wu et al. 2024; Huang et al. 2024; Xu et al. 2024; Lin et al. 2024; Bae et al. 2024; Lu et al. 2024; Liang et al. 2025a). These methods model temporal residuals of Gaussian attributes using either implicit MLPs (Yang et al. 2024a; Huang et al. 2024; Liang et al. 2025a; Wan, Lu, and Zeng 2024; Zhao et al. 2024) or explicit representations (Lin et al. 2024; Wu et al. 2024; Xu et al. 2024), such as decomposing spacetime into low-rank planes 4DGS (Wu et al. 2024) or hash grids Grid4D (Xu et al. 2024).

Furthermore, to better model 4D scenes, some methods combine deformation fields with other techniques. These include separating the scene into static and dynamic regions (Liang et al. 2025a), using sparse control points to reduce complexity (Huang et al. 2024), and employing 4D information fusion (Lu et al. 2024). However, these approaches rarely consider incorporating time-invariant properties when modeling motion.

3 Method

In this section, we introduce our solution for novel view synthesis in dynamic scenes. We first outline key concepts behind 3D Gaussian Splatting (Kerbl et al. 2023) and dynamic fields, then detail our multi-step deformation in Sec. 3.2.

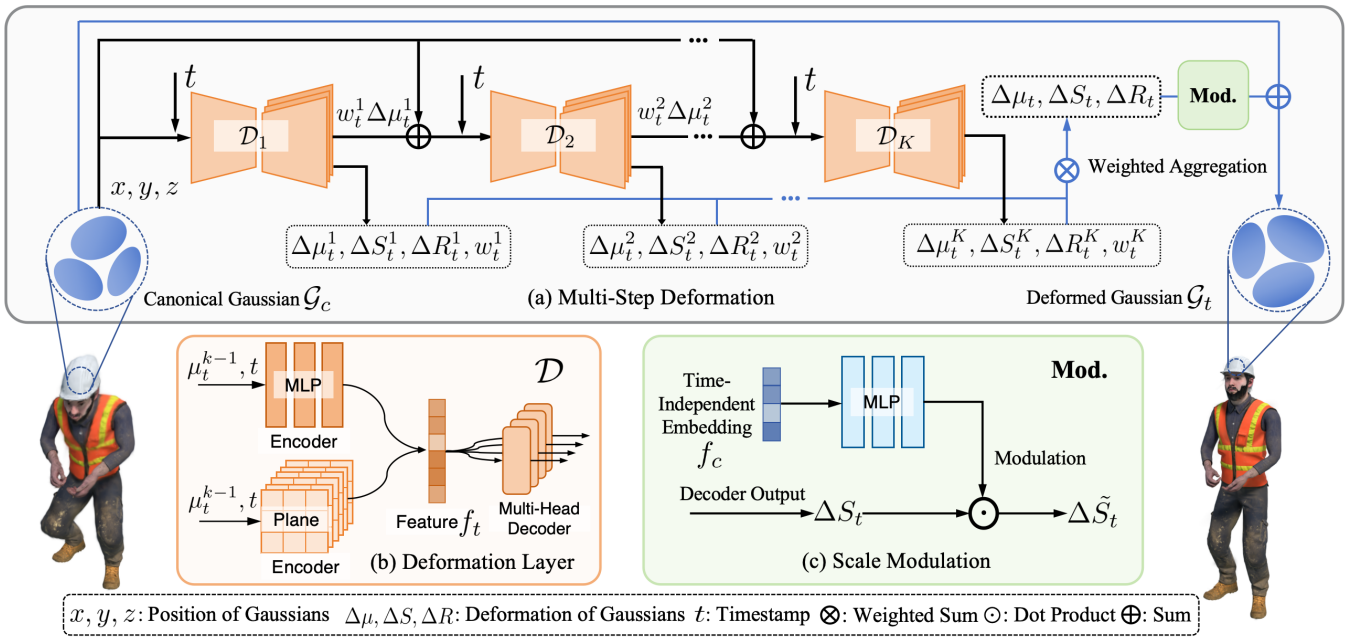


Figure 2: Overview of our pipeline. (a) The Gaussian coordinates x, y, z and time t are passed into the deformation layer, and the updated coordinates along with t are fed into the next layer. After K deformation layers, the attribute variations and contribution weights are aggregated to compute the final deformation. The scaling variations are further processed by the Mod. (modulation) module and then the deformed Gaussians are rendered to generate the image at time t. (b) The illustration of two types of deformation layers: an implicit MLP feature extractor and an explicit 6-planes feature extractor, followed by a multi-head decoder predicting deformations and weights. (c) Scaling variations ΔS are additionally modulated by a time-independent module to produce $\Delta\tilde{S}$.

Sec. 3.3 describes two deformation layer implementations, followed by a time-invariant scale modulation in Sec. 3.4. Finally, Sec. 3.5 discusses the model optimization process.

3.1 Preliminaries

3D Gaussian Splatting (Kerbl et al. 2023) uses a set of anisotropic Gaussian primitives for scene representation. Each primitive has five learnable attributes, including the 3D position μ (or x, y, z), the opacity o, the spherical harmonics coefficients (SH), the scaling matrix S and the rotation matrix R. The matrices S and R are decomposed from the covariance matrix $\Sigma = RSS^TR^T$. Gaussians are defined by their positions and 3D covariance matrices as follows:

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

Simply, we use $\mathcal{G}$ to represent a set of Gaussians. For rendering, 3D Gaussians are projected onto the image plane as 2D Gaussians G', and the color C is computed using alpha blending of the $\mathcal{N}$ overlapping Gaussians at each pixel:

$$C = \sum_{i\in\mathcal{N}} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where α is a blending weight, defined as the product of opacity and the projected Gaussian $G'(x)$. c represents the per-Gaussian color, computed using spherical harmonics (SH). After rendering, the RGB ground truth is used to guide the optimization of Gaussian attributes.

Although 3DGS (Kerbl et al. 2023) demonstrates fast and high-quality rendering in static scenes, it is incapable of handling dynamic scenes. The key to improving the quality of dynamic scene reconstruction lies in accurately modeling the motion of each point in the 3D space. Previous methods are often based on deformation fields $\mathcal{D}$, built upon 3DGS (Kerbl et al. 2023), expressed in the following form:

$$\mathcal{G}_t = \mathcal{G}_c + \Delta\mathcal{G}_t, \quad \Delta\mathcal{G}_t = \mathcal{D}(\mu, t), \tag{3}$$

where $\mathcal{G}_c$, $\mathcal{G}_t$ represent Gaussians in the canonical space and at time t respectively, while $\Delta\mathcal{G}_t$ denotes the changes in the Gaussians attributes at time t. These methods decompose dynamic scenes into a canonical space and a deformation field, which are jointly optimized. Finally, the time-dependent gassuians are rendered for dynamic scenes.

3.2 Multi-Step Deformation

However, deformation fields $\mathcal{D}$ (Yang et al. 2024a; Wu et al. 2024; Huang et al. 2024; Xu et al. 2024; Wu et al. 2024) typically rely on a single network (e.g., an MLP or HexPlane (Cao and Johnson 2023)) to model temporal variations of hundreds of thousands of Gaussians in dynamic scenes. In complex scenes where diverse motion types coexist—such as linear and nonlinear, fast and slow, large-scale motions and fine-grained variations—modeling a di-

rect mapping from (x, y, z, t) to variations in Gaussian attributes becomes highly challenging.

Therefore, we propose to decompose the dynamic field of Eq. 3 into multiple stages of deformation, as shown in Figure 1. Our core idea is to break down the direct mapping into a series of deformation steps, where each step meticulously adjusts both spatial positions and other attributes of the Gaussians. Through multiple motion steps, we can obtain the final temporal deformation. The process is depicted in Figure 2(a). More precisely, we define a single deformation step as a deformation layer in our framework, which is given by:

$$\Delta\mathcal{G}_t^k = \mathcal{D}_k(\mu_t^{k-1}, t), \quad \text{for } k = 1, 2, \ldots, K, \tag{4}$$

where $\mu_t^0 = \mu_c$. The input to the k-th deformation layer consists of the Gaussian coordinates after the $(k-1)$-th step and the time t. The deformation layer $\mathcal{D}_k$ models the variation $\Delta\mathcal{G}_t^k = (\Delta\mu_t^k, \Delta S_t^k, \Delta R_t^k, \Delta o_t^k, \Delta SH_t^k)$, representing the k-th deformation at time t. Then, the output of the k-th layer is used to compute the coordinate input for the $(k+1)$-th layer as follows:

$$\mu_t^k = \mu_t^{k-1} + \Delta\mu_t^k. \tag{5}$$

Note that the time t remains consistent across all layers as input and K is a fixed parameter independent of t. After applying K deformation layers, we obtain the set $\{\Delta\mathcal{G}_t^i\}_{i=1}^K$. Therefore, the Gaussian at time t is given by:

$$\mathcal{G}_t = \mathcal{G}_c + \sum_{i=1}^{K} \Delta\mathcal{G}_t^i. \tag{6}$$

However, directly fusing the motion outputs from each layer through simple summation can lead to poor performance, as demonstrated in the ablation study (see Table 5). This process causes Gaussians to undergo large, uncontrolled cumulative transformations as well as the uniform impact in the stacked modules that treats each layer's contribution equally, failing to account for their varying importance at different steps of deformation. To enhance the flexibility of the progressive deformation, each deformation layer is designed to additionally output a scalar weight w. This allows the model to autonomously learn and apply the appropriate influence from each layer at each step, preserving motion coherence and fine-grained details.

Therefore, we update Eq. 4 and Eq. 5 as follows:

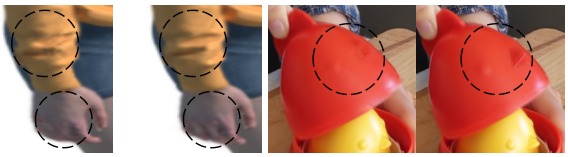
$$\Delta\mathcal{G}_t^k, w_t^k = \mathcal{D}_k(\mu_t^{k-1}, t), \tag{7}$$

$$\mu_t^k = \mu_t^{k-1} + w_t^k \Delta\mu_t^k, \tag{8}$$

where w_t^k denotes the contribution of the current step. The final deformation is computed as the weighted sum of the attribute changes. To illustrate this, we modify the second part of Eq. 6 and take the scaling S as an example:

$$\Delta\mathcal{G}_t = \sum_{i=1}^{K} w_t^i \Delta\mathcal{G}_t^i, \quad \Delta S_t = \sum_{i=1}^{K} w_t^i \Delta S_t^i. \tag{9}$$

In all our implementations, we only deform three attributes: μ, S and R, following the practice of most existing methods. Subsequently, the Gaussians are transformed from the canonical field to their state at time t.



(a) w/o mod. (b) w mod. (a) w/o mod. (b) w mod.

Figure 3: The rendering results of (a) our method without scale modulation and (b) our method with scale modulation.

3.3 Deformation Layer

Each deformation layer is a compact deformation field composed of a spacetime feature encoder and a multi-head prediction decoder. We propose two simple yet effective designs based on SOTA deformation field methods: an implicit field and an explicit field, as illustrated in Figure 2(b).

The first is inspired by the MLP-based DeformGS (Yang et al. 2024a). In contrast to DeformGS, which uses an 8-layer MLP with a width of 512 as its encoder, we employ a shallower and narrower MLP while retaining its multi-head prediction network. This process is defined as:

$$f_t = MLP(\gamma(\mu), \gamma(t)), \quad \Delta A_t = Head_A(f_t) \tag{10}$$

where $\gamma(\cdot)$ denotes positional encoding, and $Head_A$ represents the prediction head for a given attribute A.

The second simplifies 4DGS (Wu et al. 2024), which models 4D deformation using multi-resolution 6-planes $(xy, xz, xt, ...)$ and a subsequent MLP decoder. Our simplification uses a single, lower-resolution 6-plane encoder to derive features via interpolation.

$$f_t = Plane(\mu, t) \tag{11}$$

As described in Sec. 3.2, we integrate a weight prediction head into each deformation layer's decoder. This module learns the contribution weight w for its respective deformation step, using a $sigmoid$ activation function. Our multi-step framework is flexible, allowing different deformation layers to be tailored to specific needs—such as using implicit fields for model compactness or explicit fields for efficiency. Furthermore, each layer is designed as a lightweight deformation field. As shown in Table 4, our method proves its effectiveness by outperforming baselines with fewer parameters.

3.4 Time-Invariant Scale Modulation

During object motion, each object exhibits time-invariant properties that influence its deformation patterns, especially its shape variations. For example, rigid objects typically exhibit minimal shape changes, whereas deformable objects often undergo significant shape transformations. The scaling matrix S of a Gaussian, derived from the covariance matrix, controls the three principal axes of the Gaussian ellipsoid, thereby defining its shape. Each Gaussian undergoes varying magnitudes of scaling changes.

To solve this, we further modulate the scaling variations ΔS of Gaussians, ensuring that the direct output of the network adapts to the time-invariant properties of each Gaussian, as shown in Figure 2(c). We achieve this by attaching a

Method	Bouncingballs			Hellwarrior			Hook			Jumpingjacks		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	40.36	0.992	0.031	24.30	0.944	0.073	28.26	0.955	0.052	31.74	0.974	0.036
TiNeuVox	40.28	0.992	0.042	27.29	0.964	0.076	30.51	0.959	0.060	33.46	0.977	0.041
4DGS	40.75	0.994	0.015	28.61	0.973	0.036	32.89	0.976	0.026	35.33	0.985	0.020
DeformGS	41.01	0.995	0.009	41.54	0.987	0.023	37.42	0.986	0.014	37.72	0.989	0.012
SC-GS	41.34	0.995	0.008	42.43	**0.990**	0.017	39.53	0.991	0.009	40.09	0.993	0.008
Grid4D	42.48	0.995	**0.007**	42.99	**0.990**	**0.015**	39.01	0.990	0.009	39.47	0.992	**0.007**
Ours(Plane)	39.99	0.994	0.009	41.14	0.986	0.027	37.20	0.986	0.014	36.85	0.988	0.014
Ours(MLP)	**43.54**	**0.996**	**0.007**	**43.14**	**0.990**	0.016	**40.39**	**0.992**	**0.008**	**41.22**	**0.994**	**0.007**

Method	Mutant			Standup			Trex			Mean		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	33.66	0.982	0.028	34.12	0.983	0.019	31.01	0.976	0.028	31.92	0.972	0.038
TiNeuVox	32.07	0.961	0.048	34.46	0.980	0.033	31.43	0.967	0.047	32.78	0.972	0.050
4DGS	37.57	0.987	0.016	38.07	0.989	0.014	34.14	0.984	0.021	35.33	0.984	0.021
DeformGS	42.63	0.995	0.005	44.62	0.995	0.006	38.10	0.993	0.009	40.43	0.991	0.011
SC-GS	43.51	0.995	0.004	46.86	0.996	0.004	40.17	0.994	0.008	41.99	0.993	0.008
Grid4D	43.95	**0.996**	0.004	46.39	0.996	**0.003**	39.93	0.994	0.008	42.03	0.993	**0.007**
Ours(Plane)	42.14	0.994	0.007	43.93	0.994	0.008	36.96	0.991	0.011	39.74	0.990	0.013
Ours(MLP)	**44.00**	**0.996**	**0.003**	**47.54**	**0.997**	**0.003**	**41.31**	**0.995**	**0.007**	**43.02**	**0.994**	**0.007**

Table 1: Quantitative comparison to previous methods on D-NeRF (Pumarola et al. 2021) datasets with best results highlighted in bold. Both of our proposed methods demonstrate significant improvements over their respective baselines.

Method	Coffee Martini		Cook Spinach		Cut Beef		Flame Salmon		Flame Steak		Sear Steak	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DeformGS	26.71	0.890	31.35	0.943	31.89	0.943	26.74	0.899	30.96	0.953	31.59	0.951
4DGS	28.86	0.909	32.89	0.947	28.92	0.927	28.70	0.912	33.12	0.953	32.84	0.954
Grid4D	28.36	0.899	31.49	0.939	33.27	0.947	28.74	0.906	33.41	0.956	**33.99**	0.958
Ours(MLP)	26.92	0.891	31.61	0.943	31.36	0.942	27.91	0.907	31.68	0.952	32.95	0.954
Ours(Plane)	**29.42**	**0.923**	**33.14**	**0.953**	**33.65**	**0.955**	**29.88**	**0.926**	**33.43**	**0.960**	33.89	**0.960**

Table 2: Quantitative comparison to previous methods on Neu3D (Li et al. 2022) datasets with best results highlighted in bold.

learnable embedding f_c to each Gaussian, which is decoded by a tiny MLP F into a scale modulation vector $\in R^3$. This vector, activated using the exponential function, is then multiplied with the original scaling variations ΔS, yielding the modulated variations $\Delta\tilde{S}$, as shown in the formula below:

$$\Delta\tilde{S} = F(f_c) \cdot \Delta S. \tag{12}$$

This modulation is independent of the deformation field, and both the embeddings and their decoding remain invariant over time. The effectiveness of this module is demonstrated through visual results in Figure 3. By incorporating time-invariant properties, scale modulation enhances the preservation of high-frequency details.

After training, the embeddings and the decoding neural network can be discarded, and only the low-dimensional modulation vectors for each Gaussian need to be retained. Therefore, no significant additional parameters are introduced during inference.

3.5 Optimization

We employ the vanilla Gaussian's photometric loss for model optimization, which consists of $\mathcal{L}_1$ and D-SSIM losses between the rendered image and the ground truth:

$$L = \lambda\mathcal{L}_1 + (1-\lambda)\mathcal{L}_{D-SSIM} \tag{13}$$

In particular, each plane-based deformation layer additionally incorporates a grid-based TV loss (Cao and Johnson 2023; Fang et al. 2022; Fridovich-Keil et al. 2023; Sun, Sun, and Chen 2022) $\mathcal{L}_{reg}$ to regularize the grid weights for spatiotemporal smoothness.

4 Experiments

4.1 Datasets

We evaluate our method on both synthetic and real-world datasets (D-NeRF (Pumarola et al. 2021), Neu3D (Li et al. 2022), and HyperNeRF (Park et al. 2021b)) using three metrics: PSNR, SSIM, and LPIPS (Zhang et al. 2018). D-NeRF

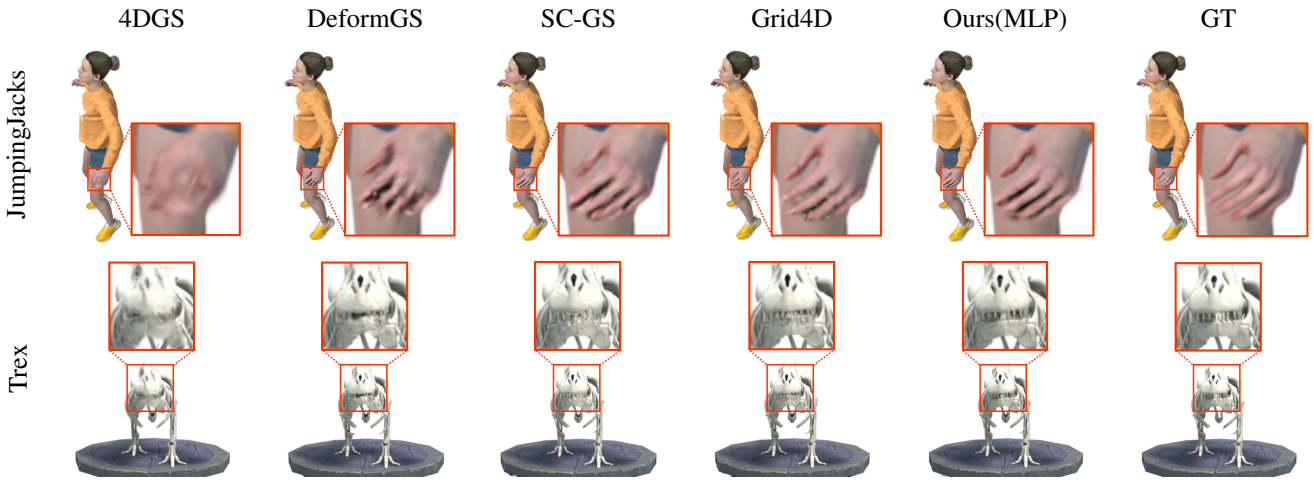


Figure 4: Qualitative comparison on the synthetic dataset D-NeRF.

Figure 5: Qualitative comparisons on the Neu3D real dataset.

(Pumarola et al. 2021) is a synthetic dataset comprising 8 scenes (800×800 resolution) with large motions and realistic non-Lambertian materials. Following previous methods, we exclude the problematic *Lego* scene and initialize with random point clouds. Neu3D (Li et al. 2022) is a real-world dataset consisting of 6 scenes, captured by 21 fixed cameras (original 2704×2028, downsampled 2× for experiments). Gaussians are initialized using COLMAP (Schonberger and Frahm 2016) point clouds. HyperNeRF (Park et al. 2021b) dataset consists of real dynamic scenes captured by monocular cameras. We conduct experiments on 4 $vrig$ sequences at 540×960 resolution for fair comparison.

4.2 Implementation Details

Across all datasets, we uniformly set the number of deformation layers K to 10. The learnable time-invariant embedding is 32-dimensional and is decoded by a two-layer MLP with a hidden layer width of 32. All experiments were conducted on a single RTX 3090. For more implementation details, please refer to the supplementary material.

4.3 Comparisons

To evaluate our model's performance, we conducted quantitative and qualitative comparisons with state-of-the-art NeRF-based methods (HexPlane (Cao and Johnson 2023), TiNeuVox (Fang et al. 2022)) and Gaussian-based methods (DeformGS (Yang et al. 2024a), 4DGS (Wu et al. 2024), SC-GS (Huang et al. 2024), Grid4D (Xu et al. 2024)). Further results are provided in the supplementary material.

	Broom		3DPrinter		Chicken		Peel Banana	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
4DGS	21.93	0.365	22.05	0.706	28.70	0.815	27.92	0.854
Grid4D	**21.99**	0.409	22.34	0.723	29.26	0.846	**28.52**	**0.875**
Base	21.01	0.335	20.86	0.682	24.02	0.712	27.20	0.855
Ours	21.96	**0.425**	**22.41**	**0.727**	**29.63**	**0.867**	28.29	0.873

Table 3: Quantitative results on HyperNeRF datasets.

Table 1 compares our method to SOTA approaches on the D-NeRF (Pumarola et al. 2021) dataset. It shows that both our deformation layers achieve significant improvements: +2.59 dB over the MLP baseline (DeformGS (Yang et al. 2024a)) and +4.41 dB over the Plane baseline (4DGS (Wu et al. 2024)). Our method also attains state-of-the-art performance across all scenes. As shown in Figure 4, our method yields superior visual results compared to previous methods.

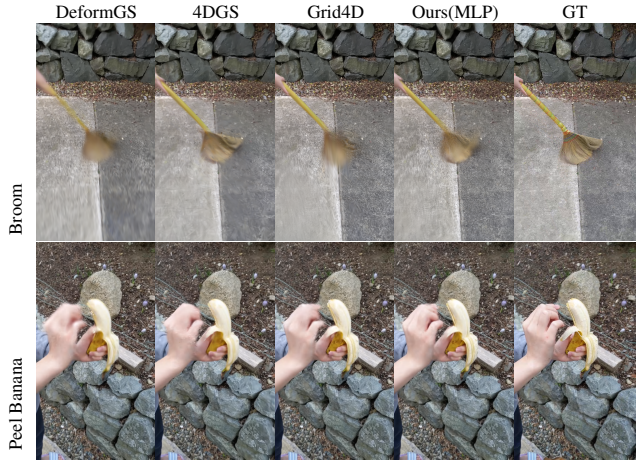


Figure 6: Qualitative comparisons on the HyperNeRF real-world dataset.

On the Neu3D (Li et al. 2022) dataset, Table 2 presents quantitative results comparing to previous SOTA methods. Our method significantly outperforms the corresponding baselines and achieves the highest performance overall. Figure 5 demonstrates that our method produces sharper and more detailed images.

For the HyperNeRF (Park et al. 2021b) dataset, we provide a comparison against DeformGS (Yang et al. 2024a) as a baseline and employing the MLP-based deformation layer for comparative evaluation. As shown in Table 3, our method shows consistent improvements over previous methods. In Figure 6, our approach better captures fine details, like the ground in the $broom$ and the fingers in the $peel\_banana$.

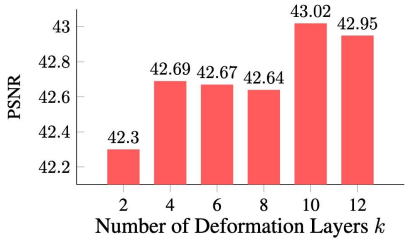


Figure 7: Ablation study of the effect of varying layer numbers on the average PSNR in the synthetic dataset.

Method	Params(M)	PSNR↑	SSIM↑	LPIPS↓
DeformGS	0.52	40.43	0.991	0.011
Ours(k=2, w/o Mod.)	0.47	41.96	0.992	0.009
Ours(k=2)	0.47 + 3.41	**42.30**	**0.993**	**0.008**
4DGS	3.38	35.33	0.984	0.021
Ours(k=2, w/o Mod.)	0.26	38.89	0.989	0.015
Ours(k=2)	0.26 + 1.69	**39.56**	**0.990**	**0.013**

Table 4: Results with fewer trainable parameters.

Method	PSNR↑	SSIM↑	LPIPS↓
w/o weight	35.69	0.978	0.029
w/o Mod. on ΔS	42.60	0.993	**0.007**
Mod. on $\Delta\mu$	42.08	0.992	0.009
Mod. on ΔR	42.56	**0.994**	**0.007**
Mod. on $\Delta\mu$ ΔS ΔR	42.38	0.993	0.008
ours full	**43.02**	**0.994**	**0.007**

Table 5: Ablation study of our proposed components on the synthetic dataset.

4.4 Ablation Study

Comparisons with Fewer Parameters. To demonstrate our method's effectiveness, we compare our multi-step approach (specifically, the two-step model without modulation) against one-step baselines on the D-NeRF dataset. As detailed in Table 4, our model achieves superior results despite using fewer trainable parameters.

Effect of the Number of Deformation Layers. As shown in the Figure 7, we investigate the impact of the number of deformation layers and report the average PSNR on the D-NeRF dataset. We observe that increasing the number of layers leads to an improvement.

Effectiveness of Scale Modulation. The scale modulation module learns time-invariant Gaussians properties to modulate scaling deformations. As shown in Figure 3, our approach effectively preserves fine-grained details and ablation study in Table 5 further explores the modulation of other attributes ($\Delta\mu$ and ΔR), revealing that these time-invariant properties primarily govern Gaussian shape variations. More visual results are provided in the supplementary material.

Effect of the Weight Prediction Head. Each deformation layer incorporates a head that predicts a weight for its contribution, making the overall deformation field more flexible. As demonstrated in Table 5, removing this head results in degraded reconstruction accuracy, confirming the module's critical role in improving the model's fitting capability.

5 Conclusion

In this paper, we propose a novel multi-step deformation framework for dynamic scene reconstruction. Unlike previous methods relying on a single deformation field, our approach decomposes the process into multiple steps. Each step is a deformation layer with a weight prediction mechanism to adaptively control its contribution. We also introduce a time-agnostic scale modulation module to capture time-invariant features. Our method excels at modeling complex motion patterns, resulting in robust and accurate reconstructions.

Limitations. Our use of stacked deformation layers can slow rendering, particularly as the number of layers increases. While reducing layers improves efficiency, it creates a trade-off between rendering speed and quality.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62222602, 62302167, U23A20343, 62476090, 62472282,62502159), Natural Science Foundation of Shanghai (Grant No. 25ZR1402135), Shanghai Sailing Program (Grant No. 23YF1410500), Young Elite Scientists Sponsorship Program by CAST (Grant No. YESS20240780), the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission (Grant No. 23CGA34), Natural Science Foundation of Chongqing (Grant No. CSTB2023NSCQ-JQX0007,CSTB2023NSCQ-MSX0137, CSTB2025NSCQ-GPX0445), Fundamental Research Funds for the Central Universities (Grant No. YG2023QNA35),Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2501), Zhejiang University, Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science-MOE, ECNU.

References

Bae, J.; Kim, S.; Yun, Y.; Lee, H.; Bang, G.; and Uh, Y. 2024. Per-Gaussian Embedding-Based Deformation for Deformable 3D Gaussian Splatting. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XV*, 321–335. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-72632-3.

Cao, A.; and Johnson, J. 2023. HexPlane: A Fast Representation for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 130–141.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024. 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24. New York, NY, USA: Association for Computing Machinery. ISBN 9798400705250.

Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; and Tian, Q. 2022. Fast Dynamic Radiance Fields with Time-Aware Neural Voxels. In *SIGGRAPH Asia 2022 Conference Papers*, SA '22. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394703.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12479–12488.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance Fields Without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5501–5510.

Garbin, S. J.; Kowalski, M.; Johnson, M.; Shotton, J.; and Valentin, J. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14346–14355.

Gong, J.; Xu, J.; Tan, X.; Song, H.; Qu, Y.; Xie, Y.; and Ma, L. 2021. Omni-Supervised Point Cloud Segmentation via Gradual Receptive Field Component Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11673–11682.

Hamdi, A.; Melas-Kyriazi, L.; Mai, J.; Qian, G.; Liu, R.; Vondrick, C.; Ghanem, B.; and Vedaldi, A. 2024. GES : Generalized Exponential Splatting for Efficient Radiance Field Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19812–19822.

Huang, Y.-H.; Sun, Y.-T.; Yang, Z.; Lyu, X.; Cao, Y.-P.; and Qi, X. 2024. SC-GS: Sparse-Controlled Gaussian Splatting for Editable Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4220–4230.

Kerbl, B.; Kopanas, G.; Leimkuehler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4).

Kocabas, M.; Chang, J.-H. R.; Gabriel, J.; Tuzel, O.; and Ranjan, A. 2024. HUGS: Human Gaussian Splats. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 505–515.

Li, T.; Slavcheva, M.; Zollhöfer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; and Lv, Z. 2022. Neural 3D Video Synthesis From Multi-View Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5521–5531.

Li, Z.; Chen, Z.; Li, Z.; and Xu, Y. 2024. Spacetime Gaussian Feature Splatting for Real-Time Dynamic View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8508–8520.

Liang, Y.; Khan, N.; Li, Z.; Nguyen-Phuoc, T. H.; Lanman, D.; Tompkin, J.; and Xiao, L. 2025a. GauFRe: Gaussian Deformation Fields for Real-Time Dynamic Novel View Synthesis. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2642–2652.

Liang, Z.; Zhang, Q.; Hu, W.; Zhu, L.; Feng, Y.; and Jia, K. 2025b. Analytic-Splatting: Anti-Aliased 3D Gaussian Splatting via Analytic Integration. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 281–297. Cham: Springer Nature Switzerland. ISBN 978-3-031-72643-9.

Lin, Y.; Dai, Z.; Zhu, S.; and Yao, Y. 2024. Gaussian-Flow: 4D Reconstruction with Dynamic 3D Gaussian Particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21136–21145.

Ling, H.; Kim, S. W.; Torralba, A.; Fidler, S.; and Kreis, K. 2024. Align Your Gaussians: Text-to-4D with Dynamic 3D Gaussians and Composed Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8576–8588.

Liu, Y.-L.; Gao, C.; Meuleman, A.; Tseng, H.-Y.; Saraf, A.; Kim, C.; Chuang, Y.-Y.; Kopf, J.; and Huang, J.-B. 2023. Robust Dynamic Radiance Fields. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13–23.

Lu, J.; Deng, J.; Zhu, R.; Liang, Y.; Yang, W.; Zhang, T.; and Zhou, X. 2024. DN-4DGS: Denoised Deformable Network with Temporal-Spatial Aggregation for Dynamic Scene Rendering. arXiv:2410.13607.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1): 99–106.

Moreau, A.; Song, J.; Dhamo, H.; Shaw, R.; Zhou, Y.; and Pérez-Pellitero, E. 2024. Human Gaussian Splatting: Real-time Rendering of Animatable Avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 788–798.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021a. Nerfies: Deformable Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5865–5874.

Park, K.; Sinha, U.; Hedman, P.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Martin-Brualla, R.; and Seitz, S. M. 2021b. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6).

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10318–10327.

Radl, L.; Steiner, M.; Parger, M.; Weinrauch, A.; Kerbl, B.; and Steinberger, M. 2024. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Trans. Graph.*, 43(4).

Reiser, C.; Peng, S.; Liao, Y.; and Geiger, A. 2021. KiloNeRF: Speeding Up Neural Radiance Fields With Thousands of Tiny MLPs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14335–14345.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-From-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; and Liu, Y. 2023. 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 (CVPR)*, 16632–16642.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct Voxel Grid Optimization: Super-Fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5459–5469.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2024. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In *The Twelfth International Conference on Learning Representations*.

Tian, Q.; Tan, X.; Xie, Y.; and Ma, L. 2024. DrivingForward: Feed-forward 3D Gaussian Splatting for Driving Scene Reconstruction from Flexible Surround-view Input. arXiv:2409.12753.

Wan, D.; Lu, R.; and Zeng, G. 2024. Superpoint Gaussian splatting for real-time high-fidelity dynamic scene reconstruction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Wang, C.; MacDonald, L. E.; Jeni, L. A.; and Lucey, S. 2023. Flow Supervision for Deformable NeRF. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21128–21137.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20310–20320.

Xu, J.; Fan, Z.; Yang, J.; and Xie, J. 2024. Grid4D: 4D Decomposed Hash Encoding for High-fidelity Dynamic Gaussian Splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yan, Z.; Low, W. F.; Chen, Y.; and Lee, G. H. 2024. Multi-Scale 3D Gaussian Splatting for Anti-Aliased Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20923–20931.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024a. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20331–20341.

Yang, Z.; Yang, H.; Pan, Z.; and Zhang, L. 2024b. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In *The Twelfth International Conference on Learning Representations*.

Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19447–19456.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, B.; Li, Y.; Sun, Z.; Zeng, L.; Shen, Y.; Ma, R.; Zhang, Y.; Bao, H.; and Cui, Z. 2024. GaussianPrediction: Dynamic 3D Gaussian Prediction for Motion Extrapolation and Free View Synthesis. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24. New York, NY, USA: Association for Computing Machinery. ISBN 9798400705250.

Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; and Yang, M.-H. 2024. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21634–21643.