MCGS: Markov Chain Gaussian Splatting for Dynamic Scenes Reconstruction

Yuzhong Wang[1], **Wenmin Wang**[1*], **Shixiong Zhang**[2], **Xinxing Yu**[1], **Zhongheng Chen**[1]

[1]Macau University of Science and Technology
[2]National UHD Video Innovation Center (Shenzhen)
{yuzhongw01,xinxingy,zhonghengc01}@student.must.edu.mo, wmwang@must.edu.mo, zhangsx@bohuauhd.com

Abstract

We present MCGS (Markov Chain Gaussian Splatting), a novel approach for high-fidelity dynamic scene reconstruction via combining Markov chain and 3D Gaussian splatting. Our method addresses the critical challenge of artifact-free temporal consistency in dynamic neural rendering. By integrating a Markov chain-based deformation network with multi-head temporal attention, MCGS effectively captures motion patterns and temporal dependencies, producing more accurate and stable 3D representations over time. The key innovations include: (1) a Markov Deform Network that models state transitions while preserving temporal coherence, (2) a temporal attention mechanism that adaptively weights historical states within a sliding window, and (3) strategic noise injection during training to enhance model robustness and generalization. Experiments on representative dynamic scene datasets demonstrate that MCGS outperforms previous methods in both visual quality and temporal coherence, while maintaining competitive rendering speed and efficiency. These results suggest the practical applicability of our approach to real-world dynamic scene understanding and synthesis.

Code — https://github.com/joseclipse/MCGS

Introduction

High-quality novel view synthesis (NVS) for dynamic scenes has important applications in many fields today, such as video games and film production. Dynamic scene novel view synthesis can render images from camera perspectives or timestamps beyond the sparse input data, while modeling complex motion scenes remains a challenging task.

Neural Radiance Field (NeRF) (Mildenhall et al. 2021) pioneered novel view synthesis for captured static scenes using Multi-Layer Perceptron (MLP) networks. Similarly, NeRF-based frameworks such as (Pumarola et al. 2021; Yan, Li, and Lee 2023) have achieved impressive results for novel view synthesis in dynamic scenes and dynamic specular scenes, but face numerous issues with training speed and robustness. Guo et al. (2023) introduced a forward flow field to inject temporal information into NeRF and a differentiable warping process to improve coherence in motion areas of dynamic scene NVS. However, this approach is highly memory-intensive with slow training speeds, and these disadvantages are amplified as the resolution of the scene increases. Since NeRF-based methods rely on pixel-level volumetric rendering, achieving real-time rendering remains extremely difficult.

Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has demonstrated real-time rendering speeds with near-photorealistic quality. Unlike the implicit representation of NeRF frameworks, 3DGS adopts an explicit point-based representation, revolutionarily representing scenes as collections of 3D Gaussians. Subsequently, Kheradmand et al. (2024) first reconstructed 3DGS rendering as an MCMC sampling process, enabling more elegant and efficient scene rendering. Their insight made 3DGS more graceful and concise while achieving better rendering quality, although these improvements targeted static scene NVS rather than dynamic scenes. 3DGS provides a new framework for dynamic scene NVS, and consequently, several 3DGS-based methods (Yang et al. 2024; Wu et al. 2024; Wan, Lu, and Zeng 2024; Huang et al. 2024; Duan et al. 2024) have pushed real-time high-fidelity dynamic rendering to new heights.

However, NVS for monocular dynamic scenes faces a significant challenge: motion artifacts in rendered dynamic scenes caused by temporal inconsistency. For portions of dynamic scenes, there exist moments when they are occluded, meaning our camera perspective temporarily cannot see them. Current approaches suffer from degraded rendering quality in new viewpoints when dealing with temporarily occluded objects.

To address this issue, we take a step back to reconsider the underlying logic of object motion, viewing the movement process of 3D Gaussians from a Markov chain perspective. In dynamic scenes, the motion states of most objects follow physical laws—their position changes and rotational changes exhibit reasonable, continuous linear acceleration and angular acceleration. Within a continuous time period, motion states remain continuous, with rare instances where a moving object's position in the next frame is extremely distant from its previous position. Furthermore, scene motion processes follow complex distributions that exhibit both

*Corresponding author

temporal continuity (such as smooth motion trajectories) and high complexity (such as object interactions and non-rigid deformations). This perfectly aligns with the Markov chain concept, where the state at the next moment can be inferred from past states. Therefore, we only need to view the dynamic process of 3D Gaussians as a Markov process and treat the movement of 3D Gaussians as state transitions in a Markov chain.

More specifically, we use a Markov chain with memory, which is a variation of the Markov chain, to describe the motion process of 3D Gaussians. We employ Multi-Head Attention to adaptively process the historical states of 3D Gaussians from previous moments. By learning short-term memory, incorporating physical constraints and noise, we derive the final predicted state transition through weighted combination of the Markov chain and MLP predictor.

Specifically, our contributions can be summarized as follows:

- We propose MCGS, a framework that combines Markov chain with dynamic scenes rendering based on Gaussian splatting, enabling robust novel view synthesis in challenging complex dynamic scenes.
- We incorporate Multi-Head Attention to adaptively select and weight historical information, learning dependencies between different timestamps, enabling robust learning of complex motion patterns from data.
- We integrate physical constraints and noise into deformation prediction to solve the reconstruction discontinuity problem of occluded objects.

Related Work

Novel View Synthesis via Markov Chain Kheradmand et al. (2024) first conceptualized 3D Gaussian Splatting as an Markov Chain Monte Carlo (MCMC) sampling process of physical scene distributions, rather than relying on original heuristic cloning and splitting strategies. They equated conventional Gaussian updates to noisy Stochastic Gradient Langevin Dynamics (SGLD) updates, naturally introducing MCMC sampling mechanisms without requiring manually tuned density control and pruning. Furthermore, they reformulated the 'cloning' operation as a deterministic state transition that preserves sample probability along the Markov chain, and relocated the low-opacity 3D Gaussians. They enhanced the reusability of 3D Gaussians, improving rendering quality and initialization robustness. However, this work focuses on improving static scenes rather than dynamic ones.

Zhu, Xie, and Li (2023) combines Energy-Based Models (EBMs) with NeRF, performing maximum likelihood estimation via MCMC to learn 3D structures from monocular images and synthesize novel views. Specifically, this framework approximates the posterior distribution of latent variables through MCMC sampling, enabling robust learning without adversarial training or variational bounds. Consequently, this work achieves high-quality novel view generation from single or incomplete 2D images with strong generalization capabilities, both with known and unknown camera poses.

Dynamic Scenes via Gaussian Splatting Since the introduction of 3D Gaussian Splatting (3DGS), numerous 3DGS-based approaches have achieved novel view synthesis for dynamic scenes. The most direct approach is to train 3D Gaussians step-by-step for each timestamp in the time sequence (Luiten et al. 2024), treating dynamic scene reconstruction as a dense 6-DOF tracking task. However, this obviously leads to multiplicative increases in training time and memory requirements. D-3D-GS (Yang et al. 2024) is the first 3DGS-based work for dynamic scenes reconstruction through a deformation field. They decouple 3D Gaussians and the deformation field, using the positions of 3D Gaussians and time as inputs to an MLP.

Subsequently, 4D-GS (Wu et al. 2024) drew inspiration from Cao and Johnson (2023); Fridovich-Keil et al. (2023); Fang et al. (2022); Shao et al. (2023), combining position information (x,y,z) and time (t) in pairs to create 6 multi-resolution planes, which are then fed into a lightweight MLP for encoding. A multi-head decoder is then used to decode the deformation of 3D Gaussians. 4DRotorGS (Duan et al. 2024) motivated by Ten Bosch (2020), using 4D rotors to describe 3D spatial rotation and spatio-temporal rotation of 4D Gaussians. However, these approaches apply deformation to each 3D Gaussian individually, requiring more computational resources while neglecting the correlations between Gaussian points.

SC-GS (Huang et al. 2024) and SP-GS (Wan, Lu, and Zeng 2024) can be seen as an improvement based on the framework of D-3D-GS. It defines a set of sparse control points, connecting nearby 3D Gaussians via KNN to ensure local rigidity. They used an MLP to obtain the deformation of each control point, and nearby 3D Gaussians were subjected to deformation accordingly. Their method considers the connections between local 3D Gaussians at the same timestamp, achieving excellent reconstruction quality while enabling custom dragging and rotation to edit trained 3D scenes. However, they did not consider the connections between 3D Gaussians across adjacent timestamps.

Our method not only preserves the connections between local 3D Gaussians but also considers the connections between 3D Gaussians across adjacent timestamps, which are modeled through a Markov chain. We use a lightweight MLP to learn deformation features from past frames to predict deformation in the next timestamp, improving rendering quality while achieving faster training speeds.

Method

We introduce Markov Chain Gaussian Splatting (MCGS), as shown in Figure 1, the first method that combines 3D Gaussian splatting with Markov chain with memory for novel view synthesis of monocular dynamic scenes. In this section, we first briefly review Markov chain with memory and 3D Gaussian Splatting to establish the foundation for our approach. Then we introduce our Markov Deform Network, followed by the Temporal Attention mechanism that adaptively processes local historical information, and our optimization strategy.

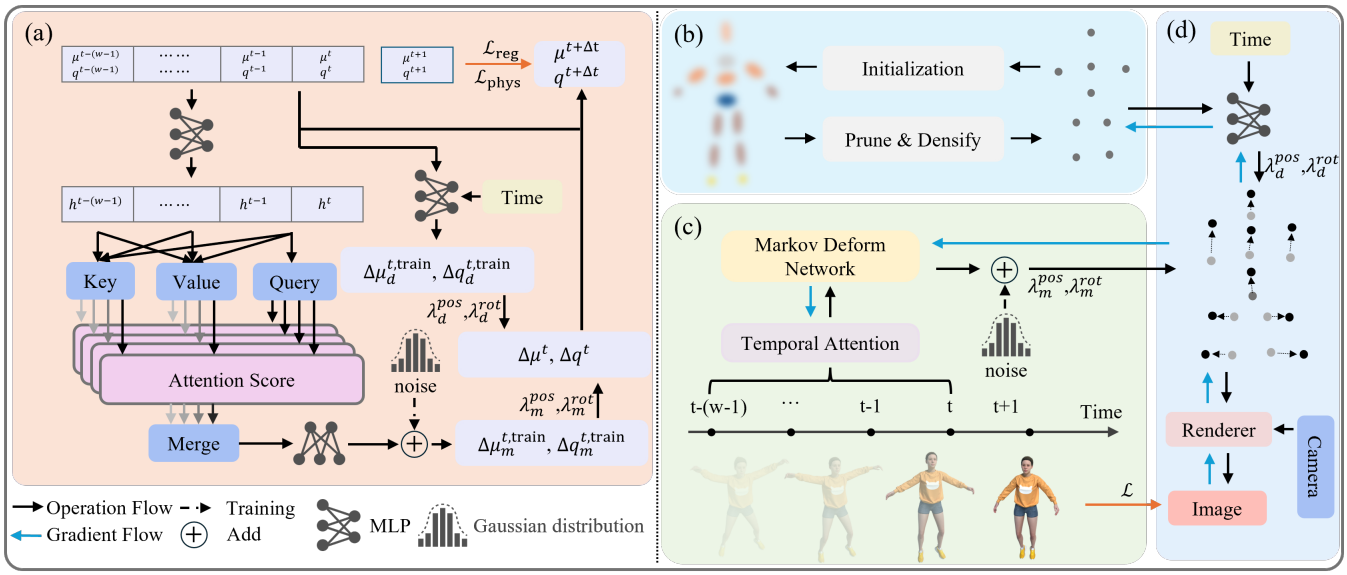

Figure 1: **Overview of our MCGS framework**. (a) The left side shows the detail of our approach. (b) We initialize Markov points by sampling from the canonical space. (c) Our approach encodes the position and rotation of Markov points into a unified representation, processes historical states through multi-head temporal attention, and predicts state transitions while accounting for a gradually decreasing Gaussian-distributed noise. The predicted state transitions are then decoded into deformations of the Markov points. (d) In parallel, a direct MLP prediction branch captures static mapping relationships between position, rotation, and time. The two prediction branches are combined with adaptive weighting, generating the final deformation that is applied to Markov points and propagated to neighboring 3D Gaussians.

Foundation

Markov Chain with Memory A Markov chain is a stochastic process $\{S_t \mid S_t \in S \text{ and } t \in T\}$ over a finite state space S and index set T, satisfying the Markov Property, which the conditional probability distribution of future states depends only on the current state and is irrelevant to the past states (Wang 2025). It is expressed as:

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_1 = s_1) = P(S_{t+1} = s_{t+1} \mid S_t = s_t), \tag{1}$$

where $s_t \in S$. This process is typically described as memoryless.

However, in certain scenarios, data sequences benefit from dependence on historical states, where additional historical memory can provide more accurate prediction advantages. For example, in dynamic scenes, we can infer the next position of moving objects using a finite history of states without explicitly adding velocity and acceleration attributes. Markov chain with memory is a variation of Markov chain that, for a finite memory length m, satisfies:

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_1 = s_1) = P(S_{t+1} = s_{t+1} \mid S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_{t-(m-1)} = s_{t-(m-1)}) \tag{2}$$

for $t \geq m$. This can also be referred to as a Markov chain of order m. In other words, future states depend on the past m states, and when $m = 1$, it reduces to a conventional Markov chain.

3D Gaussian Splatting 3D Gaussian splatting (Kerbl et al. 2023) represents scenes using a collection of 3D Gaussians. For our dynamic scene modeling, we focus on the key properties of these Gaussians that will be subject to our Markov deformation framework. Each 3D Gaussian is parameterized by:

- A 3D center position $\mu \in \mathbb{R}^3$,
- A rotation quaternion $q \in S^3$ that defines orientation,
- A scaling vector $s \in \mathbb{R}^3$ that defines size along principal axes,
- An opacity value σ,
- Spherical Harmonic coefficients for view-dependent color rendering.

The complete scene is represented as a set of Gaussians $\mathcal{G} = \{(\mu, q, s, \sigma, sh)\}$. During rendering, these 3D Gaussians are projected to 2D based on camera parameters, and their contributions are accumulated using alpha blending to generate the final image.

For our deformation modeling, we focus primarily on how the positions μ and rotations q of Gaussians change over time, as these parameters most directly capture the motion dynamics of the scene.

Markov Deform Network

Our Markov Deform Network models the historical states of points as a state space using a Markov chain, and employs a lightweight MLP to predict state transitions based on these historical states. We begin by randomly sampling points from a canonical space, which we define as a set of Markov points $\mathcal{M} = \{M_i \in \mathcal{G} \mid i = 1, 2, \ldots, N\}$, inspired by the control points in Huang et al. (2024) and superpoints in Wan, Lu, and Zeng (2024), where N is the number of Markov points. These Markov points use KNN to connect to nearby 3D Gaussians and drive the deformation of these neighboring Gaussians within the deformation network.

For dynamic scenes represented by 3D Gaussians, we propose a Markov-based deformation network that models position and rotation jointly in a unified Markov chain framework. This unified approach captures the intrinsic correlation between positional and rotational changes, which is crucial for modeling complex dynamic scenes. Given a Markov point $M_i^t : (\mu_i^t, q_i^t, s_i, \sigma_i, sh_i)$ at time t, which is also a 3D Gaussian with center $\mu_i^t \in \mathbb{R}^3$ and rotation quaternion $q^t \in S^3$, we encode these parameters together using a unified encoder:

$$h^t = \text{MLP}([\mu^t, q^t]) \tag{3}$$

The encoder employs a two-layer MLP with LayerNorm and ReLU activations, mapping the position and rotation to a d-dimensional latent vector. This joint encoding allows our model to learn the interdependencies between position and rotation changes that might be missed when treating them independently.

Our Markov Deform Network implements a Markov chain with memory. To predict the future state of each Markov point, we maintain a sliding window of length w to store the past w encodings as the state of Markov chain:

$$H^t = [h^{t-(w-1)}, \ldots, h^{t-1}, h^t] \tag{4}$$

The historical state sequence H^t captures the temporal dynamics of both position and rotation in a unified representation.

We designed a dual-channel weighted deformation prediction network structure. Specifically, our deformation prediction is jointly accomplished by Markov chain prediction and MLP direct prediction. The Markov chain prediction can be represented as:

$$\begin{aligned} f_{\text{Markov}}(H^t) &= \text{MLP}_\phi(\text{Attention}_{h^t}(H^t)) \\ &= \Delta\mu_m^{t,\text{train}}, \Delta q_m^{t,\text{train}} \end{aligned} \tag{5}$$

where MLP_ϕ is a Markov predictor which consists of three linear layers with ReLU activations, enabling conversion of historical information processed by temporal attention into deformation prediction. Markov chain prediction excels at capturing temporal dependencies; when occlusions exist or observations are incomplete, Markov chain with memory can provide more reliable deformation predictions. Attention is the multi-head attention mechanism used to adaptively process historical information, see more details in Temporal Attention.

During training, the output of $f_{\text{Markov}}(H^t)$ is a matrix composed of $\Delta\mu_m^{t,\text{train}}$ and $\Delta q_m^{t,\text{train}}$, which contains the displacement and rotation offsets predicted by the Markov Deform Network. We deliberately introduce controlled noise into the deformation prediction process to enhance robustness against occlusions and incomplete observations:

$$\Delta\mu_m^{t,\text{train}} = \Delta\mu_m^{t,\text{train}} + \eta_\mu \mathcal{N}(0, \sigma^2) \tag{6}$$

$$\Delta q_m^{t,\text{train}} = \text{SLERP}(\Delta q_m^{t,\text{train}}, \Delta q_m^{t,\text{train}} + \mathcal{N}(0, \sigma^2), \eta_q) \tag{7}$$

The weights η_μ and η_q increase noise in regions likely to be occluded (determined through visibility analysis), and $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with adaptive variance σ^2 that gradually decreases during training according to an annealing schedule. This noise injection serves two critical purposes: (1) it simulates the uncertainty present in real-world observations with occlusions, and (2) it acts as a regularizer that prevents overfitting to specific motion patterns.

Our ablation studies demonstrate that this noise-augmented training significantly improves the model's ability to generalize to complex motion sequences and handle regions with limited observations.

We utilize spherical linear interpolation (SLERP) on the quaternion hypersphere, which ensures smooth transitions along the shortest path while avoiding non-uniform velocity issues inherent in linear interpolation. This improves the model's ability to robustly represent rotational dynamics.

For MLP direct prediction, we use conventional position encoding (PE) to capture geometric structures and static mapping relationships in the scene, then use the encoded position and time as input through a multilayer perceptron MLP_θ consisting of 8 fully connected layers with 256 neurons each to directly predict the deformation of Markov points:

$$f_{\text{direct}}(\mu^t, t) = \text{MLP}_\theta(\text{PE}(\mu^t), \text{PE}(t)) \tag{8}$$

$$f_{\text{direct}}(q^t, t) = \text{MLP}_\theta(\text{PE}(q^t), \text{PE}(t)) \tag{9}$$

The position and rotation deformation of each Markov point can be predicted as:

$$\Delta\mu^t = (1 - \lambda_{pos}) f_{\text{direct}}(\mu^t, t) + \lambda_{pos} \Delta\mu_m^{t,\text{train}} \tag{10}$$

$$\Delta q^t = \text{SLERP}[f_{\text{direct}}(q^t, t), \Delta q_m^{t,\text{train}}, \lambda_{rot}] \tag{11}$$

where λ_{pos} and λ_{rot} represent the weighting coefficients employed by the Markov Deform Network during deformation prediction, governing positional and rotational transformations, respectively. These predicted deformations are then applied to update the current state:

$$\mu^{t+\Delta t} = \mu^t + \Delta\mu^t \tag{12}$$

$$q^{t+\Delta t} = q^t \otimes \Delta q^t \tag{13}$$

where $\otimes$ denotes quaternion multiplication for composing rotations.

Our method updates the position and rotation offsets for each 3D Gaussian through the nearest Markov points, enabling real-time dynamic scene rendering while maintaining temporal coherence.

Temporal Attention

To enhance the network's ability to focus on relevant historical information, we incorporate a temporal attention mechanism that adaptively weights past states. For each current state encoding h^t, the attended state is:

$$\text{Attention}_{h^t}(H^t) = \sum_{j=0}^{K-1}\Big(\sum_{i=t-(w-1)}^{t} \alpha^i (W_j^v h^i)\Big) W^o \quad (14)$$

where the attention weights α^i are computed as:

$$\alpha^i = \text{softmax}\left(\frac{(W_j^q h^t)(W_j^k h^i)^T}{\sqrt{d_k}}\right) \quad (15)$$

We implement Multi-Head Attention (Vaswani et al. 2017) with K heads, where each head has independent transformation matrices W_q^j, W_k^j, W_v^j and W^o is the output projection matrix, which integrates the outputs of the distributed attention heads into a unified feature representation. This allows different attention heads to focus on different aspects of the temporal relationship. $\text{Attention}_{h^t}(H^t)$ is the feature vector that concatenates all the outputs of the attention heads through the output projection. This allows each head to focus on different feature patterns and learn how to optimally combine these different features.

Optimization Strategy

For optimization, we train our model using a combination of photometric loss, regularization terms, and physical constraints. The overall loss function is as follows:

$$\mathcal{L} = (1-\lambda_0)\mathcal{L}_1 + \lambda_0 \mathcal{L}_{\text{D-SSIM}} + \lambda_1 \mathcal{L}_{\text{reg}} + \lambda_2 \mathcal{L}_{\text{phys}} \quad (16)$$

where $\mathcal{L}_1$ is the pixel-wise L1 loss and $\mathcal{L}_{\text{D-SSIM}}$ is the structural similarity loss, jointly measuring rendering quality, following 3DGS (Kerbl et al. 2023). $\mathcal{L}_{\text{reg}}$ is the loss to maintain local rigidity, following Huang et al. (2024), and $\mathcal{L}_{\text{phys}}$ introduces physical constraints to ensure realistic motion.

The physical constraints loss $\mathcal{L}_{\text{phys}}$ specifically encourages motion patterns that adhere to physical principles:

$$\mathcal{L}_{\text{phys}} = \lambda_a \mathcal{L}_{\text{accel}} + \lambda_s \mathcal{L}_{\text{smooth}} \quad (17)$$

where $\mathcal{L}_{\text{accel}}$ penalizes unrealistic accelerations:

$$\mathcal{L}_{\text{accel}} = \|\mu^{t+1} - 2\mu^t + \mu^{t-1}\|_2^2 \quad (18)$$

and $\mathcal{L}_{\text{smooth}}$ encourages smooth transitions in both position and rotation:

$$\mathcal{L}_{\text{smooth}} = \|\nabla_t \mu\|_1 + \alpha \|\nabla_t q\|_1 \quad (19)$$

where ∇_t denotes the temporal gradient operator used to compute the rate of change of physical quantities with respect to time. These physical constraints are particularly important for generating plausible motion in regions with limited observations or occlusions, where photometric supervision alone is insufficient.

A key challenge in monocular dynamic scene reconstruction is handling temporal occlusions, where objects temporarily disappear from view. Our Markov Deform Network is particularly well-suited to addressing this problem, as it can learn to predict motion patterns even when observations are temporarily missing.

We implement a progressive training strategy where the weighting parameters λ_{pos} and λ_{rot} in Eq. (10) and Eq. (11) start from small values and gradually increase during training. This allows the model to first learn basic scene geometry and then progressively incorporate temporal information to refine the dynamics.

Experiments

Datasets

To validate the effectiveness of our method, we conduct extensive experiments on D-NeRF dataset (Pumarola et al. 2021) which consists of eight dynamic scenes with comprehensive camera viewpoints, and vrig scenes of HyperNeRF (Park et al. 2021) dataset. Notably, however, the Lego scene of D-NeRF in the test set exhibits a certain bias; therefore, we exclude it from quantitative comparisons.

To evaluate the performance of our method and compare it with existing approaches on D-NeRF dataset, we adopt the following metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). For the real-world HyperNeRF dataset, we employ PSNR, Multiscale SSIM (MS-SSIM), and LPIPS as evaluation metrics.

Comparisons

All experiments are conducted on a GeForce RTX 4070 Ti Super with 16GB of VRAM, based on the PyTorch (Paszke et al. 2019) framework. We conduct comprehensive comparisons with state-of-the-art methods on D-NeRF (Pumarola et al. 2021) datasets and HyperNeRF (Park et al. 2021) datasets to evaluate our approach.

As highlighted in Figure 3, our method achieves superior rendering quality while effectively eliminating 3D Gaussian artifacts in dynamic scenes. Furthermore, as demonstrated in Figure 2, our approach exhibits exceptional performance on the more challenging real-world dataset HyperNeRF, successfully capturing and reconstructing fine-grained details with high fidelity. The qualitative results demonstrate that our approach produces more realistic and temporally consistent reconstructions compared to previous methods across both synthetic and real-world scenarios.

Quantitative results on D-NeRF and HyperNeRF datasets are presented in Table 1 and Table 2. We retrained SC-GS at full resolution (800×800). Our method consistently outperforms previous approaches across all metrics on most scenes. Furthermore, our method maintains high rendering quality while being computationally efficient, as evidenced by the competitive training time and rendering speed shown in the ablation study.

Tables 3 and 4 present the number of 3D Gaussians required for rendering, training time, and rendering speed (FPS) for each scene in the D-NeRF and HyperNeRF

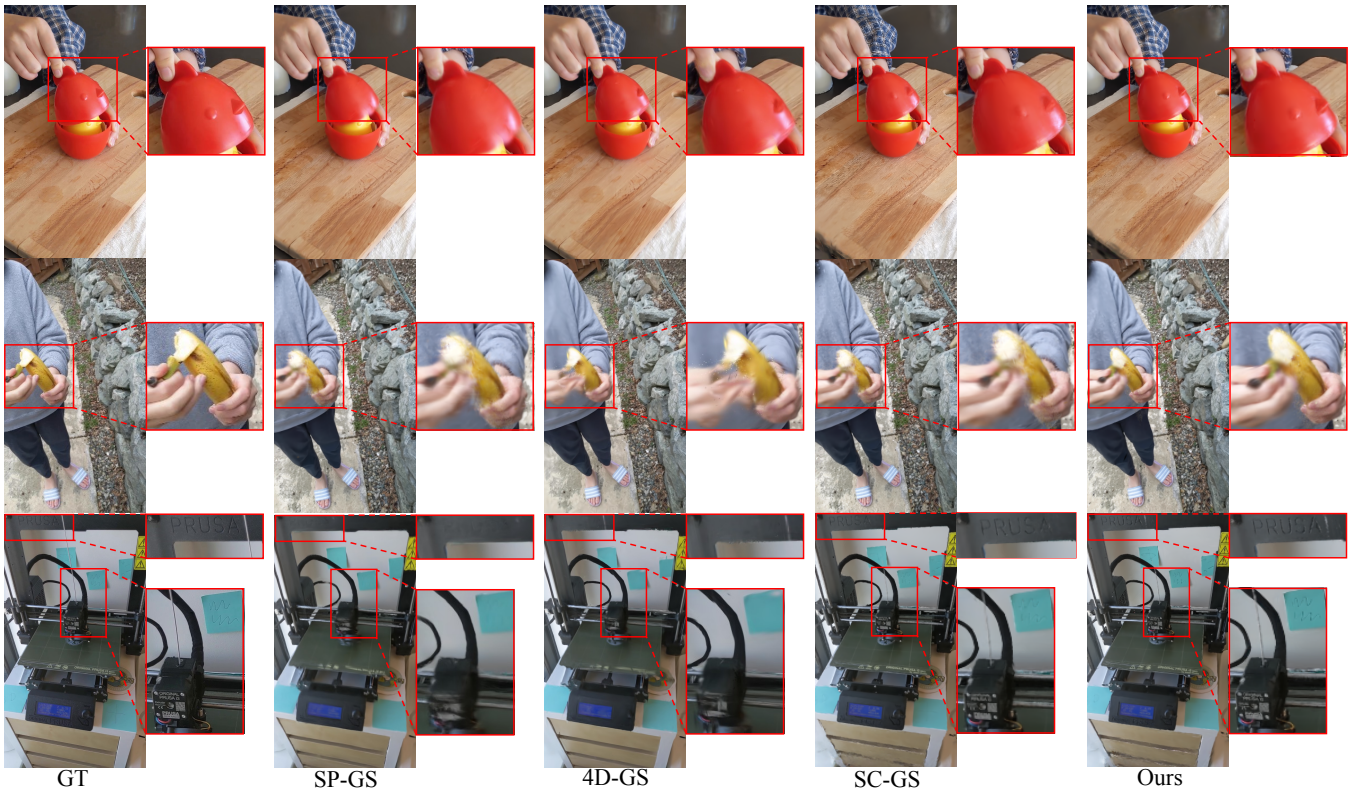

Figure 2: Comparison of different methods on HyperNeRF (Park et al. 2021) datasets.

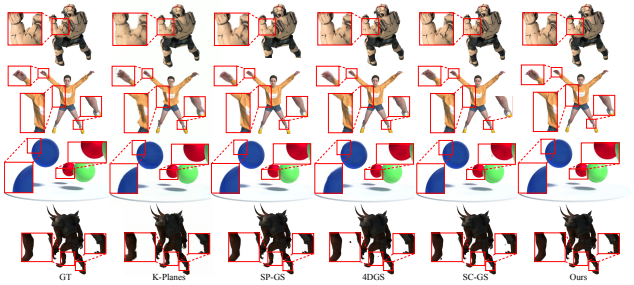

Figure 3: Comparison of different methods on D-NeRF (Pumarola et al. 2021) datasets.

Methods	PSNR(↑)	SSIM(↑)	LPIPS(↓)
D-NeRF	31.69	.975	.0575
K-Planes	31.41	.970	.0470
TiNeuVox	32.31	.969	.0501
FF-NVS	33.73	.979	.0357
D-3D-GS	40.30	.991	<u>.0116</u>
4D-GS	35.34	.985	.0185
SP-GS	<u>40.53</u>	.983	.0326
SC-GS	38.63	<u>.997</u>	.0148
Ours	**42.74**	**.998**	**.0074**

Table 1: **Qualitative results comparison on D-NeRF datasets.** We compare our method with several prior methods (Pumarola et al. 2021; Fridovich-Keil et al. 2023; Fang et al. 2022; Guo et al. 2023; Yang et al. 2024; Wu et al. 2024; Wan, Lu, and Zeng 2024; Huang et al. 2024) at full resolution (800×800). We highlight the **best** and <u>second-best</u> values for each column.

datasets, respectively. It is evident that larger scenes and more complex motions require a greater number of Gaussian points for modeling, while the temporal duration of dynamic scenes also significantly impacts the training efficiency of our method.

The 3D Printer scene in HyperNeRF features a continuously moving occluder, making it particularly suitable for validating our method's capability to address temporal occlusions, a capability where, while quantitative metrics provide overall quality assessment, visual comparison offers the most intuitive evidence of superior detail preservation. As illustrated in Figure 4, our method achieves the clearest reconstruction of the graffiti and provides reconstruction of the connecting wires of the extruder.

Ablation Study

In our method, Multi-Head Attention is employed to adaptively assign importance weights to historical states within a sliding window, while noise is introduced during training to enhance model robustness and prevent overfitting. We con-

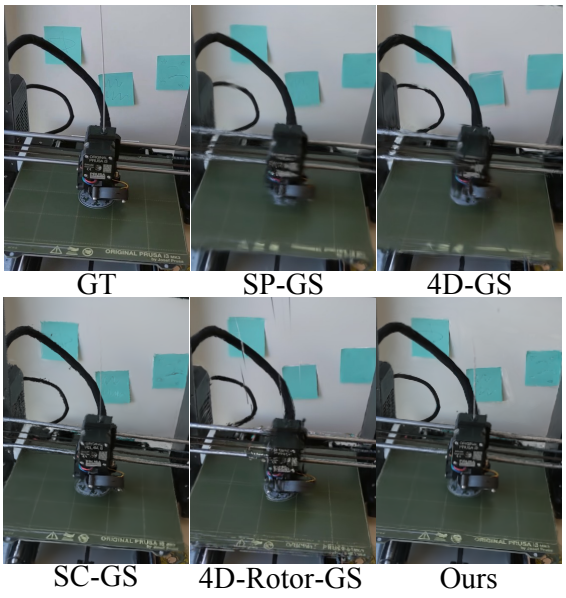

Figure 4: Comparison of details in the 3D Printer scene of HyperNeRF.

Methods	PSNR(↑)	MS-SSIM(↑)	LPIPS(↓)
HyperNeRF	23.2	.843	.2600
TiNeuVox-B	24.3	.837	-
4D-GS	25.02	.833	.2915
SP-GS	25.22	.838	**.2404**
SC-GS	24.96	.842	.2804
Ours	**25.76**	**.852**	.2427

Table 2: **Qualitative results comparison on HyperNeRF's vrig datasets.** We highlight the **best** and second-best values for each column. '-' denotes that LPIPS is not reported in Fang et al. (2022).

ducted ablation studies by removing these key components to evaluate their individual contributions. Specifically, we evaluated MCGS without Temporal Attention and without noise injection on the same dataset under identical experimental conditions. The results are shown in Tab 5.

Influence of Temporal Attention The removal of Temporal Attention leads to a noticeable performance degradation, with PSNR dropping by 2.29 dB. This shows that attention plays a crucial role in capturing temporal dependencies by adaptively weighting historical states, thereby enhancing the model's understanding of motion patterns and maintaining temporal consistency in the generated results.

Influence of Noise Injection When noise injection is removed, the model's performance decreases slightly. This indicates that noise injection helps improve the robustness and generalization ability. The noise acts as a regularizer during training, preventing the model from overfitting to the training data and enabling better performance.

Scenes	Gaussians	Train(mm:ss↓)	FPS(↑)
Hook	81.30k	15:19	209
Jumpingjacks	71.18k	17:56	198
Trex	65.27k	15:47	176
Bouncingballs	48.97k	17:44	132
Hellwarrior	31.73k	13:24	204
Mutant	87.39k	14:45	158
Standup	60.35k	14:25	195
Lego	180.89k	17:52	126
Average	78.39k	15:54	174.75

Table 3: Number of Gaussians, training time and rendering speed on D-NeRF dataset at full resolution(800x800).

Scenes	Gaussians	Train(↓)	FPS(↑)
3D Printer	137.78k	1.27 hour	53
Broom	520.24k	1.98 hour	30
Chicken	231.39k	1.52 hour	51
Peel Banana	301.04k	1.87 hour	34
Average	297.61k	1.66 hour	42

Table 4: Number of Gaussians, training time and rendering speed on vrig HyperNeRF dataset at 536x960 resolution.

Methods	PSNR(↑)	Train(↓)	FPS(↑)
SP-GS	40.53	32:53	95
SC-GS	38.63	28:36	186
Ours w/o Attention	40.45	**14:48**	**209**
Ours w/o Noise	41.59	15:28	170
Ours (Full)	**42.74**	15:54	174.75

Table 5: **Ablation Study on D-NeRF dataset.** To quantitatively evaluate the effectiveness of our approach, we analyze the PSNR, training time and rendering speed of 30,000 iterations.

Limitations

While our method achieves brilliant reconstruction quality for details, the model architecture inevitably leads to increased training time when modeling large-scale and long-duration dynamic scenes. Additionally, similar to other neural rendering approaches, our method exhibits challenges when handling scenes with complex specular reflections.

Conclusion

In this paper, we introduce MCGS, a novel approach that combines Markov chain modeling with 3D Gaussian splatting for real-time high-fidelity dynamic scenes reconstruction. Extensive evaluations show MCGS achieves SOTA results, effectively addresses the temporal artifacts and motion inconsistencies that plague existing dynamic scenes reconstruction approaches.

Acknowledgments

This work was supported by Fund for the Development of Science and Technology (FDCT) of Macau (Grant No. 0010/2024/AGJ). We thank Associate Professor Dagang Li for his kindly support.

References

Cao, A.; and Johnson, J. 2023. HexPlane: A Fast Representation for Dynamic Scenes. *CVPR*.

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 *Proc. SIGGRAPH*.

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*.

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*, 12479–12488.

Guo, X.; Sun, J.; Dai, Y.; Chen, G.; Ye, X.; Tan, X.; Ding, E.; Zhang, Y.; and Wang, J. 2023. Forward flow for novel view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16022–16033.

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*, 4220–4230.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Kheradmand, S.; Rebain, D.; Sharma, G.; Sun, W.; Tseng, Y.-C.; Isack, H.; Kar, A.; Tagliasacchi, A.; and Yi, K. M. 2024. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37: 80965–80986.

Luiten, J.; Kopanas, G.; Leibe, B.; and Ramanan, D. 2024. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, 800–809. IEEE.

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. *Communications of the ACM*, 65(1): 99–106.

Park, K.; Sinha, U.; Hedman, P.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Martin-Brualla, R.; and Seitz, S. M. 2021. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

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*, 10318–10327.

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*, 16632–16642.

Ten Bosch, M. 2020. N-dimensional rigid body dynamics. *ACM Transactions on Graphics (TOG)*, 39(4): 55–1.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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*, 49957–49972.

Wang, W. 2025. Probabilistic Framework. In *Principles of Machine Learning*, 69–123. Singapore: Springer Nature Singapore. ISBN 978-981-9753-32-1 978-981-9753-33-8.

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.

Yan, Z.; Li, C.; and Lee, G. H. 2023. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8285–8295.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20331–20341.

Zhu, Y.; Xie, J.; and Li, P. 2023. Likelihood-based generative radiance field with latent space energy-based model for 3D-aware disentangled image representation. *arXiv preprint arXiv:2304.07918*.