GaussianPrediction: Dynamic 3D Gaussian Prediction for Motion Extrapolation and Free View Synthesis

Boming Zhao*
bmzhao@zju.edu.cn
Zhejiang University
Hangzhou, China

Yuan Li*
yuan_li@zju.edu.cn
Zhejiang University
Hangzhou, China

Ziyu Sun
sunzy2121@mails.jlu.edu.cn
Jilin University
Changchun, China

Lin Zeng
22251265@zju.edu.cn
Zhejiang University
Hangzhou, China

Yujun Shen
shenyujun0302@gmail.com
Ant Group
Hangzhou, China

Rui Ma
ruim@jlu.edu.cn
Jilin University
Changchun, China

Yinda Zhang
yindaz@google.com
Google Inc.
Mountain View, USA

Hujun Bao
bao@cad.zju.edu.cn
Zhejiang University
Hangzhou, China

Zhaopeng Cui†
zhpcui@gmail.com
Zhejiang University
Hangzhou, China

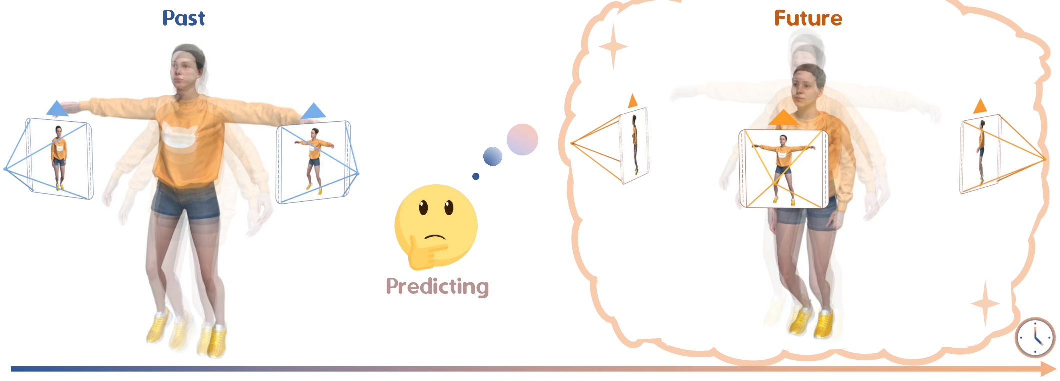


Figure 1: GaussianPrediction can reconstruct the entire dynamic scene with high quality from monocular dynamic images and reasonably predict what will happen in the future. In addition, unlike 2D video prediction, our approach can synthesize novel view images at future moments.

ABSTRACT

Forecasting future scenarios in dynamic environments is essential for intelligent decision-making and navigation, a challenge yet to be fully realized in computer vision and robotics. Traditional approaches like video prediction and novel-view synthesis either lack the ability to forecast from arbitrary viewpoints or to predict temporal dynamics. In this paper, we introduce GaussianPrediction, a novel framework that empowers 3D Gaussian representations with dynamic scene modeling and future scenario synthesis in dynamic environments. GaussianPrediction can forecast future states from any viewpoint, using video observations of dynamic scenes. To this end, we first propose a 3D Gaussian canonical space with deformation modeling to capture the appearance and geometry of dynamic scenes, and integrate the lifecycle property into Gaussians for irreversible deformations. To make the prediction feasible and efficient, a concentric motion distillation approach is developed by distilling the scene motion with key points. Finally, a Graph Convolutional Network is employed to predict the motions of key points, enabling the rendering of photorealistic images of future scenarios. Our framework shows outstanding performance on both synthetic and real-world datasets, demonstrating its efficacy in predicting and rendering future environments. Code is available on the project webpage: https://zju3dv.github.io/gaussian-prediction.

*Boming Zhao and Yuan Li contributed equally to this work.
†Corresponding author.

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657417

CCS CONCEPTS

• **Computing methodologies → Computer graphics**; **Rendering**.

KEYWORDS

novel view synthesis, dynamics modeling, future prediction

ACM Reference Format:

Boming Zhao, Yuan Li, Ziyu Sun, Lin Zeng, Yujun Shen, Rui Ma, Yinda Zhang, Hujun Bao, and Zhaopeng Cui. 2024. GaussianPrediction: Dynamic 3D Gaussian Prediction for Motion Extrapolation and Free View Synthesis. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers ’24 (SIGGRAPH Conference Papers ’24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3641519.3657417

1 INTRODUCTION

The ability to envision what is about to happen in the near future is critical for us, human beings, to survive in ubiquitous dynamic scenes, and equally crucial for computers to fulfill intelligent decision-making and navigation in complex 3D worlds. Specifically, such ability cannot be well achieved without: 1) predicting the dense motion in a short future time span, and 2) visualizing the scene in the future from arbitrary viewpoints. Despite extensive efforts in computer vision and robotics, imparting similar skills to intelligent agents remains a significant challenge.

One family of approaches aligned with this objective is video prediction, which endeavors to forecast the future dynamics of a scene from a specific viewpoint, based on past observations from that same viewpoint [Kwon and Park 2019; Neimark et al. 2021; Oprea et al. 2020]. Despite its potential, video prediction falls short in visualizing arbitrary viewpoints, thus constraining its effectiveness in understanding the near future. On the other hand, novel-view synthesis focuses on rendering images of a scene from a flexibly chosen viewpoint [Kerbl et al. 2023; Mildenhall et al. 2021; Zhang et al. 2022], but it does not incorporate the element of temporal prediction necessary for forecasting future states of the environment. Approaching in native 3D, 3D point cloud prediction aims to extrapolate future 3D point clouds from a sequence of past scans [Mersch et al. 2022; Wang et al. 2023]. This technique is particularly beneficial for decision-making in the context of intelligent vehicles; however, it lacks the capacity to generate high-quality images.

In this paper, we introduce GaussianPrediction, a novel framework that builds on 3D Gaussian representations for dynamic scene modeling and future scenario synthesis in dynamic environments. Given video observations of dynamic scenes, GaussianPrediction is capable of forecasting potential future scenarios from any viewpoint. Although the 3D Gaussian representation [Kerbl et al. 2023] appears to inherently bridge novel-view synthesis and motion prediction, designing such a system is far from straightforward. Firstly, temporal modeling is needed for the 3D Gaussian representation to address both general motions and irreversible deformations in dynamic scenes, which is absent in concurrent works involving dynamic 3D Gaussian modeling [Wu et al. 2023; Yang et al. 2023a]. Furthermore, accurately representing a scene requires a substantial number of 3D Gaussians, and their motions must be predicted in a cohesive manner to facilitate effective future projection and image rendering. However, prediction inaccuracies in even a small subset of these Gaussians can significantly degrade the quality of the rendered images (as shown in Fig. 7).

Drawing inspiration from D-NeRF [Pumarola et al. 2021a] and its variants [Liu et al. 2023], we first build the 3D Gaussian canonical space with deformation modeling to capture the appearance and geometry of the dynamic scene. We also design additional lifecycle properties for the Gaussians to model the irreversible deformations in a dynamic scene, such as those occurring on broken surfaces. To efficiently control the motion of the entire scene’s Gaussians, we employ a novel concentric motion distillation approach that utilizes key points to distill scene motions from the learned deformation fields. This markedly reduces the complexity of the following future prediction model, cutting down the number of nodes requiring prediction from hundreds of thousands to just a few hundred. Such a strategy makes the forecasting of future scenarios more efficient and feasible. In enhancing the model’s capability to accurately represent both appearances and motions with prediction capability, we select key points through feature clustering in a hyper-canonical space. This space is uniquely designed to encode both spatial and motion distances, thereby mitigating artifacts that often arise from discontinuous motion fields along object boundaries. Finally, our framework leverages a Graph Convolutional Network (GCN) to predict the future motion of these 3D key points. This prediction, in turn, drives the anticipated deformation of the entire scene, enabling our system to forecast dynamic environments and synthesize future photorealistic images freely.

Our contributions can be summarized as follows:

- We present GaussianPrediction, a novel framework that innovatively integrates 3D Gaussian representations with dynamic scene modeling and future scenario synthesis, leveraging video observations to forecast scenes shortly from any viewpoint.
- We develop a novel canonical space of 3D Gaussians with lifecycle properties, which can model both general motions and irreversible deformations in dynamic scenes, offering a more comprehensive representation of temporal dynamics.
- We introduce a novel future prediction strategy based on the concentric motion distillation, which enhances the efficiency and robustness.
- Experiments on both synthetic and real-world datasets demonstrate the efficacy of our proposed framework in predicting and rendering future scenarios.

2 RELATED WORK

2.1 4D Novel View Synthesis

Several studies address the reconstruction of dynamic scenes and the generation of free viewpoint renderings, employing explicit mesh representations [Broxton et al. 2020; Dou et al. 2016; Newcombe et al. 2015; Orts-Escolano et al. 2016], depth estimations [Bansal et al. 2020; Yoon et al. 2020] or implicit neural volumes [Lombardi et al. 2019]. Harnessing its capability to deliver photorealistic novel renderings, Neural Radiance Field (NeRF) [Mildenhall et al. 2021] has been incorporated into 4D dynamic scene reconstructions [Attal et al. 2023; Du et al. 2021; Li et al. 2022; Park et al. 2021a; Pumarola et al. 2021b], spanning various tasks such as monocular video reconstructions [Gao et al. 2021; Li et al. 2021, 2023; Tretschk et al. 2021], scene editings [Kania et al. 2022; Park et al. 2021b; Zheng et al. 2023a], human reconstructions [Peng et al. 2021a,b; Weng et al. 2022; Zielonka et al. 2023], fast reconstructions and renderings [Cao and Johnson 2023; Fang et al. 2022a; Fridovich-Keil et al. 2023; Geng et al. 2023; Lombardi et al. 2021; Peng et al. 2023; Shao et al. 2023; Song et al. 2023], as well as generalizable renderings [Lin et al. 2023, 2022]. Dynamic point clouds [Xu et al. 2023; Zhang et al. 2022; Zheng et al. 2023b] have garnered significant attention, particularly due to their rapid rendering speed. Notably, 3D Gaussian Splatting [Kerbl et al. 2023] has emerged as a technique that combines swift reconstruction and rendering speeds, all while preserving exceptional rendering quality. Hence, a series of studies have been undertaken to broaden the applicability of 3D Gaussian Splatting to dynamic reconstruction scenarios. This has been achieved through explicit extensions of time-variant Gaussian features [Luiten et al. 2023; Yang et al. 2023c] or the utilization of implicit deformation fields [Wu et al. 2023; Yang et al. 2023a]. By combining explicit 3D Gaussian representations with implicit neural representations like MLPs [Yang et al. 2023a] or HexPlanes [Wu et al. 2023], these endeavors demonstrate exceptional qualities in novel view synthesis. They achieve interactive rendering frame rates and offer flexible editing capabilities, including object insertions.

2.2 Dynamic Prediction

Generating successive frames following given video sequences [Oprea et al. 2020] proves invaluable in intelligent decision-making. Approaches that combine 3D-CNNs, LSTMs, or Transformers [Geng et al. 2023; Girdhar and Grauman 2021; Neimark et al. 2021; Villegas et al. 2018; Vondrick et al. 2016; Wang et al. 2018]seamlessly integrate spatial and temporal information during prediction. VAE-based methods [Babaeizadeh et al. 2017; Denton and Fergus 2018], utilizing stochastic latent samples, yield diverse future predictions. Another line of work [Kwon and Park 2019; Lu et al. 2017] employs GANs pre-trained on prior data to generate future predictions by analyzing historical frames. Other works [Höppe et al. 2022; Yang et al. 2023b] also produce realistic video predictions and infillings leveraging diffusion models.

Motion or dynamic predictions of 4D data inputs like human skeletons can also be conducted by RNNs [Corona et al. 2020; Martinez et al. 2017], VAEs [Petrovich et al. 2021], GANs [Barsoum et al. 2018; Martinez et al. 2017] or diffusion models [Alexanderson et al. 2023; Barquero et al. 2023]. In the literature, there are also studies demonstrating predictions for scene-level dynamic point clouds, such as LiDAR scans [Mersch et al. 2022] or common point clouds [Wang et al. 2023]. Approaches based on graph convolutional networks (GCN) [Mao et al. 2019; Sofianos et al. 2021] exhibit excellent generalizing abilities while providing rapid prediction speeds. Nonetheless, these methods fall short in rendering photorealistic novel views compared to NeRFs' or 3D Gaussian Splattings' derivatives. In contrast, our GCN-based method excels in both forecasting reasonable scene dynamics based on historical frames and rendering high-quality novel views.

3 PRELIMINARIES

3D Gaussian Splatting [Kerbl et al. 2023] employs a substantial number of explicit 3D Gaussians to represent a static 3D scene. Each 3D Gaussian G is defined by a full covariance matrix Σ and a center location μ:

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

For differentiable rendering optimization, 3D Gaussian splatting decomposes Σ into scaling matrix S and rotation matrix R: $\Sigma = RSS^TR^T$, where S and R are stored by a 3D vector s and a quaternion q respectively. To project these 3D Gaussians to 2D image, given a viewing transformation W, we obtain the 2D covariance matrix Σ' and 2D center location μ':

$$\Sigma' = JW\Sigma W^TJ^T, \mu' = JW\mu, \tag{2}$$

where J is the Jacobian of the affine approximation of the projective transformation. Then we can use the neural point-based α-blending to render the color C of each pixel with N ordered 3D Gaussians:

$$C = \sum_{i\in N} T_i c_i \alpha_i, \tag{3}$$

where T_i, α_i are calculated as:

$$\begin{aligned} T_i &= \textstyle\prod_{j=1}^{i-1}(1-\alpha_j), \\ \alpha_i &= \sigma_i e^{-\frac{1}{2}(x-\mu')^T\Sigma'^{-1}(x-\mu')}. \end{aligned} \tag{4}$$

Here σ_i is the opacity of the 3D Gaussian. Therefore, the 3D scene can be represented by the parameter set P of 3D Gaussians, where $P = \{G_i : \mu_i, q_i, s_i, c_i, \sigma_i\}$.

4 METHOD

Given a collection of images of a scene captured at different time instances from a monocular camera, GaussianPrediction aims to reconstruct the dynamic scenes and forecast future scenarios. As shown in Fig. 2, GaussianPrediction consists of three stages. At first, we recover the canonical space of 3D Gaussians with deformation fields to model the dynamic scenes from input images (Sec. 4.1). To model irreversible deformations (*e.g.*, cutting fruits or splitting cookies), we extend the 3D Gaussian with a novel lifecycle property. In the second stage, we employ a novel concentric motion distillation approach with key points to distill scene motion, significantly reducing the complexity of forecasting by decreasing the parameters from hundreds of thousands to a few hundred (Sec 4.2). Finally, in the third stage, we adopt Graph Convolutional Networks (GCN) to predict the future motion of these 3D key points, effectively forecasting the entire scene's deformation (Sec. 4.3).

4.1 Dynamic Modeling with Canonical Space

Hyper-canonical space of 3D Gaussians. One widely used strategy for reconstructing dynamic scenes is to build a canonical space and subsequently deform all the 3D information within this canonical space to match different time instances. Previous methods [Liu et al. 2023; Pumarola et al. 2021a; Yang et al. 2023a] employed Multilayer Perceptrons (MLP) to encode temporal deformation, utilizing the 3D location x and time t as inputs. However, due to the inherent similarity in spatial characteristics among adjacent 3D positions, these approaches may result in blurring when handling different

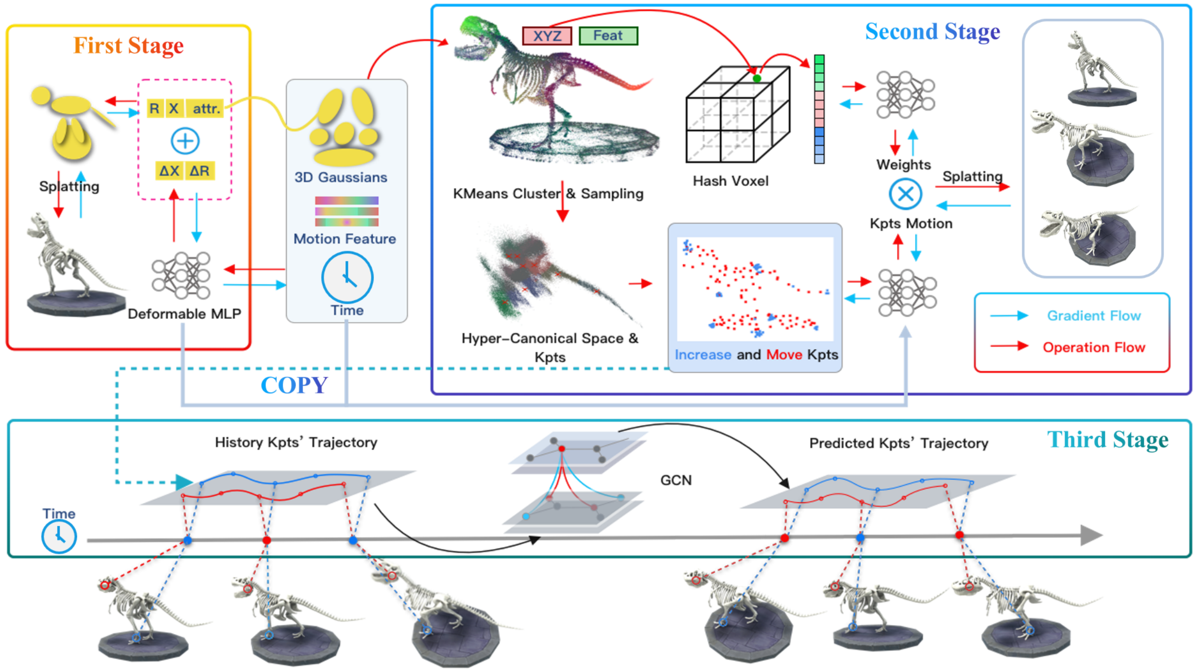


Figure 2: Optimization start with the initial 3D Gaussians. We then optimize the parameters of the 3D Gaussians, motion feature, and deformable MLP to build a Hyper-Canonical space. Next in the second stage, we first initialize the key points in the Hyper-Canonical space by a K-Means algorithm. Then we learn the time-independent weights for each Gaussian and deform the 3D Gaussian by key points motion. We employ a GCN (Graph Convolutional Network) to learn the relationships between key points, thereby predicting the future motion of key points, and rendering future scenes from a novel view.

motion patterns in neighboring locations. To address this problem, we utilize a motion feature m with dimension d for each Gaussian to encode the motion information. Given any timestamp t, we can obtain the deformation of each 3D Gaussian's center location μ and rotation q from the canonical space to the moment t:

$$\Delta\mu^t, \Delta q^t = D(\gamma(\mu), m, \gamma(t)), \tag{5}$$

where D is a Multilayer Perceptron (MLP) and γ denotes the positional encoding with frequency L:

$$\gamma(x) = (sin(2^l \pi x), cos(2^l \pi x))_{l=0}^{L-1}. \tag{6}$$

Therefore our deformed 3D Gaussians P_t at timestamp t can be defined as: $P^t = \{G_i^t : (\mu_i + \Delta\mu_i^t), (q_i \otimes \Delta q_i^t), s_i, c_i, \sigma_i, m_i\}$, where $\otimes$ represents quaternion multiplication. Then we define the hyper-canonical space C_h as the following:

$$C_h = \{(\mu, m) | \mu \in \mathbb{R}^3, m \in \mathbb{R}^d\}. \tag{7}$$

However, we found that optimizing both μ and the deformable MLP can easily get stuck in local optima. Therefore, we propose to introduce an annealing noise ε on μ:

$$\varepsilon(i) = \mathcal{N}(0, 1) \cdot N_s \cdot (1 - min(1, \frac{i}{10000})), \tag{8}$$

where i denotes the current training iteration and N_s is the scaling factor. In our experiments, we observed that decaying noise effectively guides the optimization of μ away from local optima. This leads to a more uniform distribution of μ in space and enhances the rendering quality of our model (See Sec. 5.4).

Lifecycle of Gaussians. Temporal motion often triggers situations where a portion of the original surface disappears (*e.g.*, gluing up toy pieces together) or new surfaces emerge (*e.g.*, cutting a lemon into two halves). This deformation is irreversible, which is different from the general deformation in that correspondences exist over the whole sequence. An intuitive idea is that we should allow 3D Gaussians to exhibit the same property—being renderable until a certain point in time, after which they lose their rendering abilities. Therefore, we propose to add a lifecycle ψ to the opacity of 3D Gaussians:

$$\begin{aligned} \psi(G_i, t) &= \frac{1}{1 + e^{(-10\Delta_o(G_i,t))}}, \\ \Delta_o &= D_o(\gamma(\mu_i), m_i, \gamma(t)). \end{aligned} \tag{9}$$

Here Δ_o is calculated by the opacity deformable MLP D_o. We then multiply this lifecycle ψ with the opacity of the 3D Gaussian which makes the opacity mostly either 0 or 1 in the majority of cases, indicating that Gaussians are involved in rendering at certain moments while being invisible at other times. Our experiment shows that this strategy efficiently improves the rendering quality both quantitatively and qualitatively (See Sec. 5.4).

4.2 Concentric Motion Distillation with Key Points

A direct way to predict the future deformation of the scenes is to extrapolate the input time t to D in Eq. 5. However, each 3D Gaussian in the canonical space is independent and unconstrained, and as a result, direct extrapolation would cause 3D Gaussians to lose their original geometric properties. To address this problem, we design a key points driven framework inspired by [Zheng et al. 2023a] to deform the whole 3D Gaussians by key points $K = \{k_i = (\mu_i^k \in \mathbb{R}^3, m_i^k \in \mathbb{R}^d)\}, i \in \{1, 2, ..., N_k\}$ defined in the hyper-canonical space. Here μ_i^k denotes the 3D position and m_i^k is the motion feature of key points k_i. For each key point k_i, we calculate the 3DoF translation vector T_i^t and 3DoF rotation quaternion Q_i^t at time t by the deformable MLP D trained in Sec. 4.1:

$$T_i^t, Q_i^t = D(\gamma(\mu_i^k), m_i^k, \gamma(t)). \tag{10}$$

Note that the representation of motion for key points here is identical to the representation of motion for 3D Gaussians in the first stage. Therefore, we can utilize the deformable MLP trained in the first stage as the deformable MLP for the second stage. Then we can render the image at time t using the deformed 3D Gaussians P_t^{key} which can be represented as:

$$\begin{aligned} P_t^{\text{key}} &= \{G_i^t : (\mu_i + \Delta\mu_i^t), (q_i \otimes \Delta q_i^t), s_i, c_i, \sigma_i, m_i\}, \\ \Delta\mu_i^t &= \sum_{k \in K} (w_{i \leftarrow k}^T \cdot T_k^t), \Delta q_i^t = \sum_{k \in K} (w_{i \leftarrow k}^Q \cdot Q_k^t), \end{aligned} \tag{11}$$

where $w_{i \leftarrow k}^T$, $w_{i \leftarrow k}^Q$ represent the translation and rotation weights of 3D Gaussian G_i with respect to key point k. In summary, our concentric motion distillation comprises three key steps:

1. **Initializing key points.** This foundational step will initialize k_{init} key points in the hyper-canonical space.
2. **Adaptive increasing key points.** For those complex motion areas, the initial key points may not be enough. Therefore we need to increase the key points in complex motion areas adaptively.
3. **Time-independent weights learning.** This step involves understanding how each key point affects each 3D Gaussian, which transforms the motion of key points into the motion of 3D Gaussians.

In this way, we can deform more than 200k 3D Gaussians using hundreds of key points, which simplifies the prediction process. Next, we will introduce the details of each step.

Initializing key points. Once we have trained the hyper-canonical space, we can sample k_{init} 3D points as the initial key points. Gaussians driven by the same key point should demonstrate both motion similarity and spatial proximity. To achieve this, we employ clustering techniques in the hyper-canonical space to organize 3D Gaussians into k_{init} classes as $S = \{S_1, S_2, ..., S_{k_{init}}\}$. Formally, the objective is to find:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{K_{init}} \frac{1}{|S_i|} \sum_{x,y \in S_i} \|x - y\|^2 , \tag{12}$$

where x,y is the vector $\in \mathbb{R}^{3+d}$ in C_h. We then utilize the 3D center of each class as the initial position for the k_{init} key points.

Adaptive increasing key points. Following [Kerbl et al. 2023], we also present an adaptive increasing strategy to add key points near the complex motion 3D Gaussians. To identify the areas that require additional key points, we calculate the Gaussian gradient norm and select Gaussians with a norm greater than the gradient threshold. A large gradient means these Gaussians show poor reconstruction results. Therefore, we use FPS [Eldar et al. 1997] uniformly downsampling the large gradient Gaussians by a factor of 100 to generate the locations for the newly added key points. However, to facilitate predicting scene motion, we limit the number of added points to avoid increasing complexity. Thus the maximum number of adaptive increasing key points is $N - k_{init}$. In this way, our model can flexibly handle complex motions.

Time-independent weights learning. In Eq. 11, we introduced the time-independent weights w^T, w^Q and discussed how to drive the 3D Gaussian by key points motion. In this step, we will discuss how to learn these weights for each 3D Gaussian.

Given a 3D Gaussian G_i in the canonical space, its motion is primarily influenced by the movements of the nearest N_{near} key points, rather than those that are far away. As a result, we can represent the weights w_i^T, w_i^Q for the Gaussian G_i w.r.t. the key points as:

$$w_i^T = \operatorname{softmax}(\sum_{k \in K_{\text{near}}^i} w_{i \leftarrow k}^T), w_i^Q = \operatorname{softmax}(\sum_{k \in K_{\text{near}}^i} w_{i \leftarrow k}^Q), \tag{13}$$

where k_{near}^i represents the N_{near} key points closest to the G_i. However, we found that directly finding the nearest neighbors in space without considering motion information may not completely separate Gaussians with different motions but are spatially adjacent. To address this issue, we propose searching for the nearest key points to each Gaussian point in the Hyper-Canonical space C_h taking into account both spatial proximity and motion similarity. Our experiments demonstrate the effectiveness of this nearest key points strategy which significantly reduces artifacts in the real-world dataset (See Sec. 5.4).

EditableNeRF [Zheng et al. 2023a] takes canonical coordinate x as input and outputs a weight vector w by a large MLP. However, this method is inefficient when the number of queries x increases. Inspired by [Müller et al. 2022], we present a novel method that uses hash encoding to map the coordinate x to trainable feature vectors and then decode to w by a tiny MLP. In this way, our model remains efficient even with a substantial increase in input coordinates.

4.3 GCN-based Motion Prediction

After the optimization in the second stage, we obtain several key points that encapsulate the distilled motion information of the scene. All the scene motion details are implicitly encoded within these key points. Considering the relationships between these key points, we utilize the capabilities of Graph Convolution Network (GCN) to model and predict the dynamic movement patterns of key points within a given scene. At different time steps, we use the 3D positions of key points as supervision. We employ a GCN to extract relational features between key points across multiple frames. A single-layer MLP is then utilized to decode these features and predict the positions of key points at the next time step. Then, we

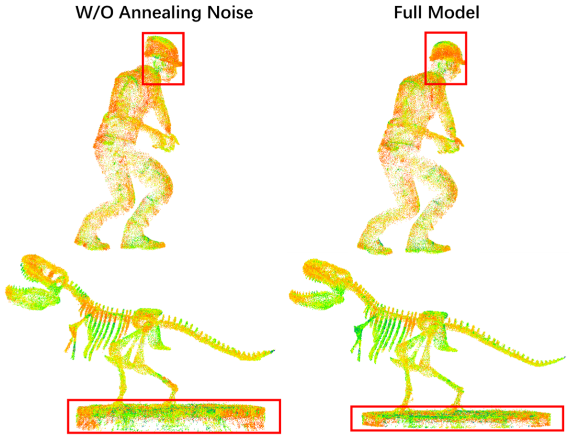
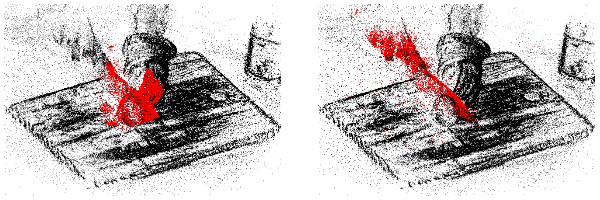


Figure 3: Canonical space point cloud with different training strategies.



(a) Searching nearest key points in 3D space. **(b) Searching nearest key points in Hyper-Canonical space.**

Figure 4: Influenced 3D Gaussians by a key point on the knife. We compare two different search methods and show influenced Gaussian points in red.

calculate the motion of each 3D Gaussian using Eq. 11, obtaining the predicted 3D Gaussians. Our approach allows for continuous predictions using a sliding window, leveraging past time steps to generate new predictions. Please refer to our supp. material for more details.

5 EXPERIMENTS

In this section, we first introduce our implementation details and the test datasets in Sec. 5.1. Then we evaluate the dynamic scene rendering quality of our method in Sec. 5.2 and compare the prediction results with different baselines in Sec. 5.3. Lastly, we perform ablation studies to analyze the effectiveness of our method in Sec. 5.4.

5.1 Datasets and Implementation Details

We conduct evaluations on both synthetic and real datasets.

Synthetic Dataset. The D-NeRF dataset [Pumarola et al. 2021a] contains 8 dynamic objects, and comprises 100-200 training images and 20 test images, with timestamps ranging from 0 to 1 for all images. We render images of this dataset at an 800 × 800 resolution. Following Deform-GS [Yang et al. 2023a], we evaluate the rendering results with a black background and exclude the "*Lego*" data because its test data and model did not align with the training data.

Real-World Dataset. From the Hyper-NeRF real-world dataset [Park et al. 2021b], we chose three scenes (*cut-lemon, split-cookie, and chickchicken*) captured by a camera and one scene (*3D-printer*) captured by two Google Pixel 3 phones. Timestamps range from 0 to 1 for all images and rendering resolution is set to 960 × 540.

Implementation details. We develop a composited training strategy, which learns scene hyper-canonical space, key points, and scene prediction in a three-step fashion. We train for 30k iterations in the first step. During the initial 1k iterations, we exclusively optimized the parameters of the 3D Gaussians for warm-up purposes. Subsequently, we jointly train for 27k iterations on the deformation MLP and the motion feature in Eq. 5. In the second step, we conduct training on the hash-encoding, the deformation MLP, and the position and motion feature of the key points for 10k iterations. Then in the third step, we train all the parameters together for 20k iterations using the synthetic dataset and 30k iterations for the real-world dataset. All our experiments are evaluated on an NVIDIA GeForce RTX 4090 GPU.

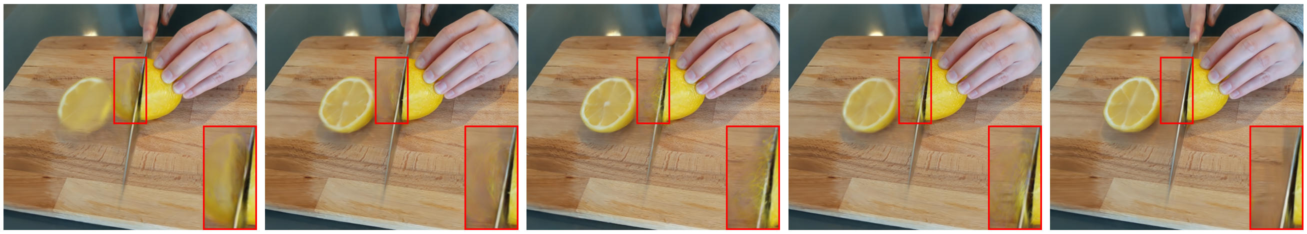
5.2 Comparison of Dynamic Scene Rendering Quality

We first compare our method with the current state-of-the-art dynamic scene reconstruction methods: TiNeuVox [Fang et al. 2022b], 4D-GS [Wu et al. 2023] and Deform-GS [Yang et al. 2023a] on the synthetic dataset using the same data setting. The quantitative results are shown in Table 2. We present the PSNR/SSIM/LPIPS(VGG) values on this dataset. The results demonstrate that our method outperforms either existing NeRF-based or Gaussian-based methods. We also show the rendering results in Fig. 8. It can be seen that our method achieves higher quality than other methods and reconstructs more details of dynamic scenes.

We also evaluate our method on the real-world dataset following Hyper-NeRF's [Park et al. 2021b] setting. We compare with four state-of-the-art methods and report the PSNR/MS-SSIM values in Table 3. As shown in Fig. 6, our method achieves better rendering quality on the real-world dataset. However, our method does not surpass the previous NeRF-based rendering [Fang et al. 2022b; Park et al. 2021b] in quantitative results due to inaccurate ground truth camera poses and the misalignment of timestamps across all images in the real dataset. Despite this, ours still outperforms existing Gaussian-based methods [Wu et al. 2023; Yang et al. 2023a], which demonstrates the effectiveness of our approach.

5.3 Comparison of Future Synthesis

We now conduct the comparison on the future synthesis task in Table 1. In this experiment, to synchronize the time intervals, we divide the original training data into new test and training sets. We utilize the data from the original training set with image timestamps less than 0.8s as the training data, and those greater than 0.8s as the test data. We directly input the time $t \sim [0.8 - 1.0]$ to each method and calculate the PSNR/SSIM/LPIPS(VGG) as presented in Table 1. Additionally, we also compare with a couple of variants of GaussianPrediction: "Ours" is our full model which uses the GCN to predict the motion of key points (Sec. 4.3), and "Ours-MLP" removes GCN and directly input the time t to the deformable MLP in the second stage. Quantitative results demonstrate that our



(a) w/o Hyper Space Init. (b) w/o Adap Increasing. (c) w/o Hyper Space K-NN. (d) w/o Lifecycle. (e) Full Model.

Figure 5: We analyze the effectiveness of each component.

Table 1: Quantitative motion prediction results comparison on D-NeRF dataset. Following HyperNeRF [Park et al. 2021b], the average metrics are calculated using a weighted average. Best results are highlighted as first, second.

Method	Trex			Jumpingjacks			Bouncingballs			Hellwarrior		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	20.72	.9284	.0751	19.87	.9115	.0954	25.92	.9677	.0853	29.36	.9097	.1138
4D-GS	20.72	.9401	.0579	20.28	.9176	.0825	29.42	.9753	.0433	31.48	.9266	.0929
Deformable-GS	20.81	.9426	.0461	20.21	.9150	.0800	28.90	.9784	.0271	29.82	.9141	.0834
Ours-MLP	21.51	.9444	.0452	20.68	.9194	.0742	29.58	.9816	.0225	29.99	.9176	.0789
Ours	21.09	.9406	.0461	20.51	.9184	.0760	26.63	.9714	.0361	30.75	.9281	.0729
Methods	Mutant			Standup			Hook			Average		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	24.40	.9282	.0700	21.77	.9169	.0927	21.05	.8817	.1033	22.83	.9229	.0886
4D-GS	24.61	.9269	.0582	22.25	.9140	.0870	23.93	.9042	.0755	23.98	.9305	.0697
Deformable-GS	24.32	.9300	.0469	21.38	.9133	.0837	21.41	.8872	.0824	23.35	.9285	.0623
Ours-MLP	25.05	.9359	.0409	23.04	.9250	.0700	22.6	.8971	.0702	24.14	.9339	.0560
Ours	28.16	.9560	.0256	25.96	.9403	.0481	23.42	.9089	.0573	24.62	.9387	.0514

Table 2: Quantitative rendering results comparison on D-NeRF dataset. Best results are highlighted as first, second.

Method	Trex			Jumpingjacks			Bouncingballs			Hellwarrior		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	31.24	.9771	.0326	34.29	.9799	.0360	35.00	.9835	.0391	39.20	.9763	.0508
4D-GS	33.60	.9863	.0188	35.59	.9844	.0210	37.69	.9919	.0150	38.52	.9754	.0524
Deformable-GS	38.10	.9933	.0098	37.72	.9897	.0126	41.01	.9953	.0093	41.54	.9873	.0234
Ours	37.39	.9926	.0110	37.93	.9906	.0099	41.57	.9954	.0086	41.73	.9874	.0214
Methods	Mutant			Standup			Hook			Average		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	35.07	.9768	.0307	38.11	.9854	.0208	33.34	.9711	.0458	35.18	.9786	.0365
4D-GS	38.80	.9857	.0212	40.43	.9890	.0164	33.83	.9728	.0338	36.92	.9836	.0255
Deformable-GS	42.63	.9951	.0052	44.62	.9951	.0063	37.42	.9867	.0144	40.43	.9918	.0116
Ours	42.90	.9954	.0049	45.09	.9954	.0057	37.44	.9868	.0137	40.58	.9919	.0107

method achieves more realistic effects in future synthesis tasks, thanks to our key points distilled motion strategy. The experiments show that our GCN can learn the relationships between key points and predict more reasonable results which improves the rendering results of novel views in the future. We also show prediction quantitative evaluations of the real-world HyperNeRF dataset in Sec. B of the supp. material. Note that in real-world scene-level datasets, the predicted results cannot be perfectly aligned with the ground truth images due to ill camera poses and inaccurate timestamps, which makes the quantitative comparison less meaningful than the qualitative comparison. Therefore, relying solely on quantitative evaluations to assess prediction performance is limited. Please refer to our video for more information. In short sequence prediction, our approach maintains better coherence, consistency, and reasonable.

5.4 Ablation Studies

Table 3: Quantitative results comparison with TiNeuVox [Fang et al. 2022b], HyperNeRF [Park et al. 2021b], 4D-Gs [Wu et al. 2023], and Deform-GS [Yang et al. 2023a] on Hyper-NeRF real-dataset. Best results are highlighted as first , second , third.

Method	CHICKEN (113 images)		CUT LEMON (415 images)		SPLIT COOKIE (134 images)		3D PRINTER (207 images)		AVERAGE	
	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)
TiNeuVox-B	27.7	.951	28.6	.955	28.9	.965	22.8	.839	27.2	.928
HyperNeRF	28.7	.948	31.8	.956	30.9	.967	20.0	.821	28.4	.924
4D-GS	26.9	.911	30.0	.929	32.5	.975	22.0	.808	28.1	.905
Deform-GS	26.1	.902	29.1	.937	32.8	.981	20.3	.756	27.2	.896
Ours	27.1	.920	31.1	.952	34.0	.983	22.2	.814	28.9	.920

Table 4: Ablation studies of the annealing noise on the D-NeRF dataset.

Config.	D-NeRF		
	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Annealing Noise	39.71	0.991	0.012
Full Model	**40.58**	**0.992**	**0.011**

Table 5: Ablation study of each step discussed in Sec. 4.2 on the real-world Hyper-NeRF dataset. Best results are highlighted as first , second.

Config.	Hyper-NeRF		
	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Hyper Initialization	27.9	.911	.232
w/o Adaptively Increasing	28.6	.917	.225
w/o Lifecycle	28.6	.918	.218
Full Model	28.9	.920	.184

Annealing noise. We first inspect the effectiveness of the annealing noise during training hyper-canonical space (Sec. 4.1). We show the location of 3D Gaussians in the canonical space in Fig. 3. When training with annealing noise, our model can preserve better geometry. The quantitative results in Table 4 indicate that training with annealing noise can improve the reconstruction results. This proves the necessity of annealing noise for training.

Initial key points in the hyper-canonical space. We also compared the impact of initializing key points in different spaces. As shown in Fig. 5 and Table 5, directly initializing key points in 3D space results in significant blurring in regions with complex motion, leading to a decrease in quantitative values. However, our approach initializes key points in hyper-space, considering not only spatial information but also motion information, effectively improving rendering quality.

Adaptive increasing key points. To demonstrate the effectiveness of the adaptively increasing key points strategy, we compared the results of directly initializing N_k key points with adaptively increasing $N_k - k_{init}$ key points in Table 5. The results demonstrate that our method excels in identifying regions that require additional key points, whereas direct initialization cannot adjust based on rendering outcomes.

Searching nearest key points in the hyper-canonical space. We then analyze the impact of searching the nearest key points in the hyper-canonical space. We choose a key point on the knife in *Cut-Lemon* scene and then select all 3D Gaussians that are influenced by this key point using two different selecting strategies. As shown in Fig. 4a, the key point on the knife is the nearest key point of 3D Gaussians on the lemon. However, these two motions are entirely different yet influenced by the same key point, resulting in blurriness during rendering. Finding the nearest key point in the hyper-canonical space can prevent this issue. Fig. 5 and Table 5 further demonstrate the effectiveness of our approach.

Lifecycle opacity. We also study the effectiveness of lifecycle strategy in Fig. 5 and Table 5. We can observe that when the lemon is sliced, some 3D Gaussians still have residues at the cut (within the red box), while the addition of the lifecycle strategy significantly eliminates these Gaussians, improving the rendering quality.

6 CONCLUSION

In this paper, we present a novel framework, *i.e.*, GaussianPrediction, for forecasting future scenarios in dynamic scenes. GaussianPrediction employs a 3D Gaussian canonical space with deformation modeling, coupled with a lifecycle property, to effectively represent changes in dynamic scenes. Additionally, a novel concentric motion distillation technique with key points is developed to simplify complex scene motion prediction with a Graph Convolutional Network.

Since GaussianPrediction learns to predict the dynamics of key points solely based on the input observations without any pretraining, it can only predict meaningful and short-term future scenarios. For long-term prediction, incorporating motion priors in our framework presents a promising direction for future research.

ACKNOWLEDGMENTS

We would like to acknowledge the support from NSFC (No. 62102356), Ant Group, Information Technology Center, and State Key Lab of CAD&CG, Zhejiang University. We also express our gratitude to all the anonymous reviewers for their professional and insightful comments. The authors from Zhejiang University are also affiliated with the State Key Lab of CAD&CG.

REFERENCES

Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. 2023. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM*

Transactions on Graphics (TOG) 42, 4 (2023), 1–20.

Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. 2023. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16610–16620.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. 2017. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252* (2017).

Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 2020. 4d visualization of dynamic events from unconstrained multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5366–5375.

German Barquero, Sergio Escalera, and Cristina Palmero. 2023. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2317–2327.

Emad Barsoum, John Kender, and Zicheng Liu. 2018. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 1418–1427.

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 86–1.

Ang Cao and Justin Johnson. 2023. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 130–141.

Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. 2020. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6992–7001.

Emily Denton and Rob Fergus. 2018. Stochastic video generation with a learned prior. In *International conference on machine learning*. PMLR, 1174–1183.

Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. 2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–13.

Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. 2021. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 14304–14314.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. 1997. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing* 6, 9 (1997), 1305–1315.

Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022a. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022b. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. 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.

Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. 2021. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5712–5721.

Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. 2023. Learning neural volumetric representations of dynamic humans in minutes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8759–8770.

Rohit Girdhar and Kristen Grauman. 2021. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 13505–13515.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. 2022. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696* (2022).

Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. 2022. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18623–18632.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023).

Yong-Hoon Kwon and Min-Gyu Park. 2019. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1811–1820.

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

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6498–6508.

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. 2023. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4273–4284.

Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. 2023. Im4D: High-Fidelity and Real-Time Novel View Synthesis for Dynamic Scenes. *arXiv preprint arXiv:2310.08585* (2023).

Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2022. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. 2023. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13–23.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751* (2019).

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–13.

Chaochao Lu, Michael Hirsch, and Bernhard Scholkopf. 2017. Flexible spatio-temporal networks for video prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6523–6531.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. 2023. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713* (2023).

Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9489–9497.

Julieta Martinez, Michael J Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2891–2900.

Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. 2022. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*. PMLR, 1444–1454.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

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

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. 2021. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3163–3172.

Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 343–352.

Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. 2020. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2020), 2806–2826.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*. 741–754.

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

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. 2021b. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228* (2021).

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14314–14323.

Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2023. Representing Volumetric Videos as Dynamic MLP Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4252–4262.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9054–9063.

Mathis Petrovich, Michael J Black, and Gül Varol. 2021. Action-conditioned 3D human motion synthesis with transformer VAE. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10985–10995.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021a. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10318–10327.

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

Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. 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.

Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. 2021. Space-time-separable graph convolutional network for pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11209–11218.

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

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. 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*. 12959–12970.

Ruben Villegas, Dumitru Erhan, Honglak Lee, et al. 2018. Hierarchical long-term video prediction without supervision. In *International Conference on Machine Learning*. PMLR, 6038–6046.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. *Advances in neural information processing systems* 29 (2016).

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. 2018. Eidetic 3D LSTM: A model for video prediction and beyond. In *International conference on learning representations*.

Zifan Wang, Zhuorui Ye, Haoran Wu, Junyu Chen, and Li Yi. 2023. Semantic Complete Scene Forecasting from a 4D Dynamic Point Cloud Sequence. *arXiv preprint arXiv:2312.08054* (2023).

Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. 2022. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*. 16210–16220.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2023. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528* (2023).

Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 2023. 4k4d: Real-time 4d view synthesis at 4k resolution. *arXiv preprint arXiv:2310.11448* (2023).

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. 2023b. Diffusion probabilistic modeling for video generation. *Entropy* 25, 10 (2023), 1469.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. 2023a. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101* (2023).

Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. 2023c. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642* (2023).

Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5336–5345.

Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. 2022. Differentiable point-based radiance fields for efficient view synthesis. In *SIGGRAPH Asia 2022 Conference Papers*. 1–12.

Chengwei Zheng, Wenbin Lin, and Feng Xu. 2023a. Editablenerf: Editing topologically varying neural radiance fields by key points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8317–8327.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. 2023b. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21057–21067.

Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4574–4584.

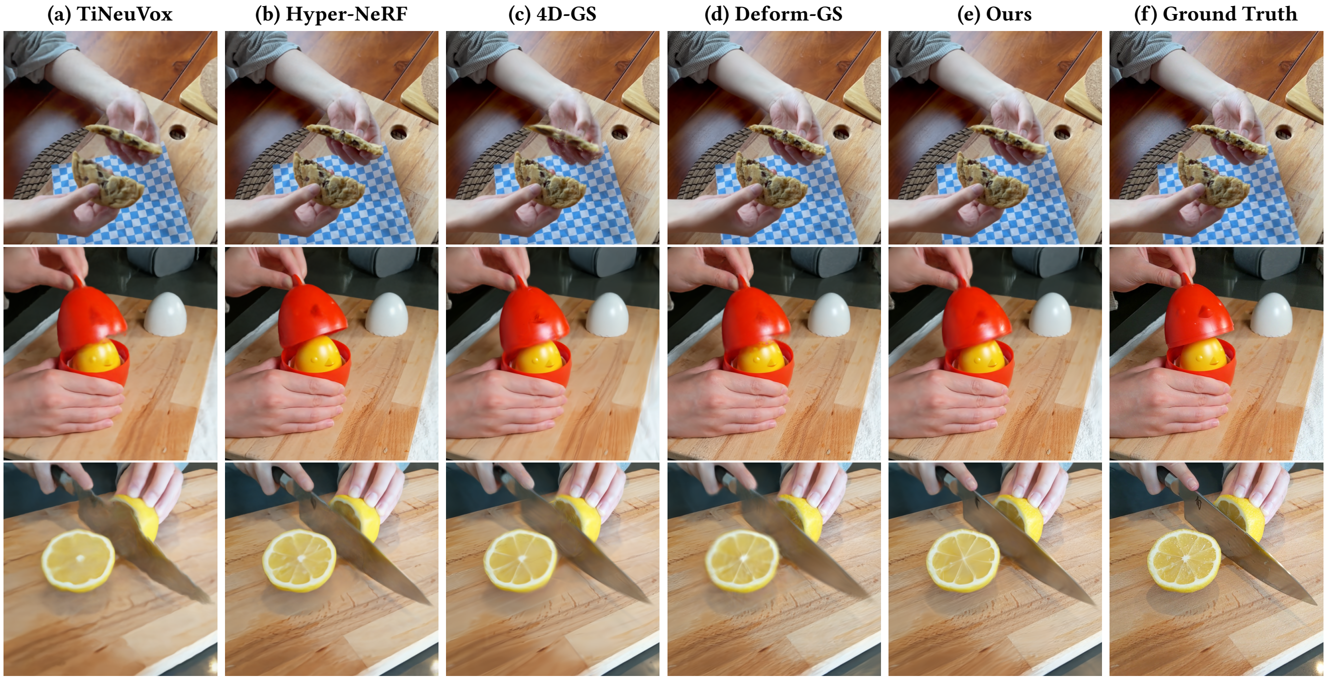


Figure 6: Qualitative results on real-world scenes. We compare our methods with TiNeuVox [Fang et al. 2022b], Hyper-NeRF [Park et al. 2021b], 4D-GS [Wu et al. 2023], and Deform-GS [Yang et al. 2023a].

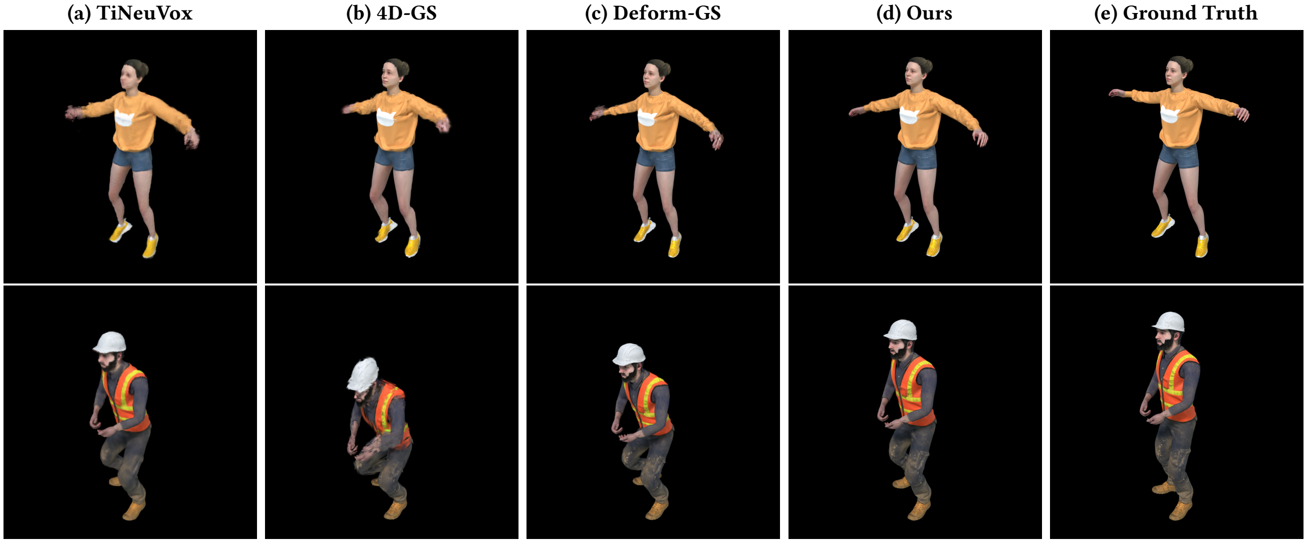


Figure 7: To distinguish from reconstructions results, we show prediction results in black background. Qualitative results on real-world scenes. We compare our methods with TiNeuVox [Fang et al. 2022b], 4D-GS [Wu et al. 2023], and Deform-GS [Yang et al. 2023a]. Our method not only renders highly detailed novel views but also predicts motion close to ground truth. Please refer to our supplementary video for more comparisons.

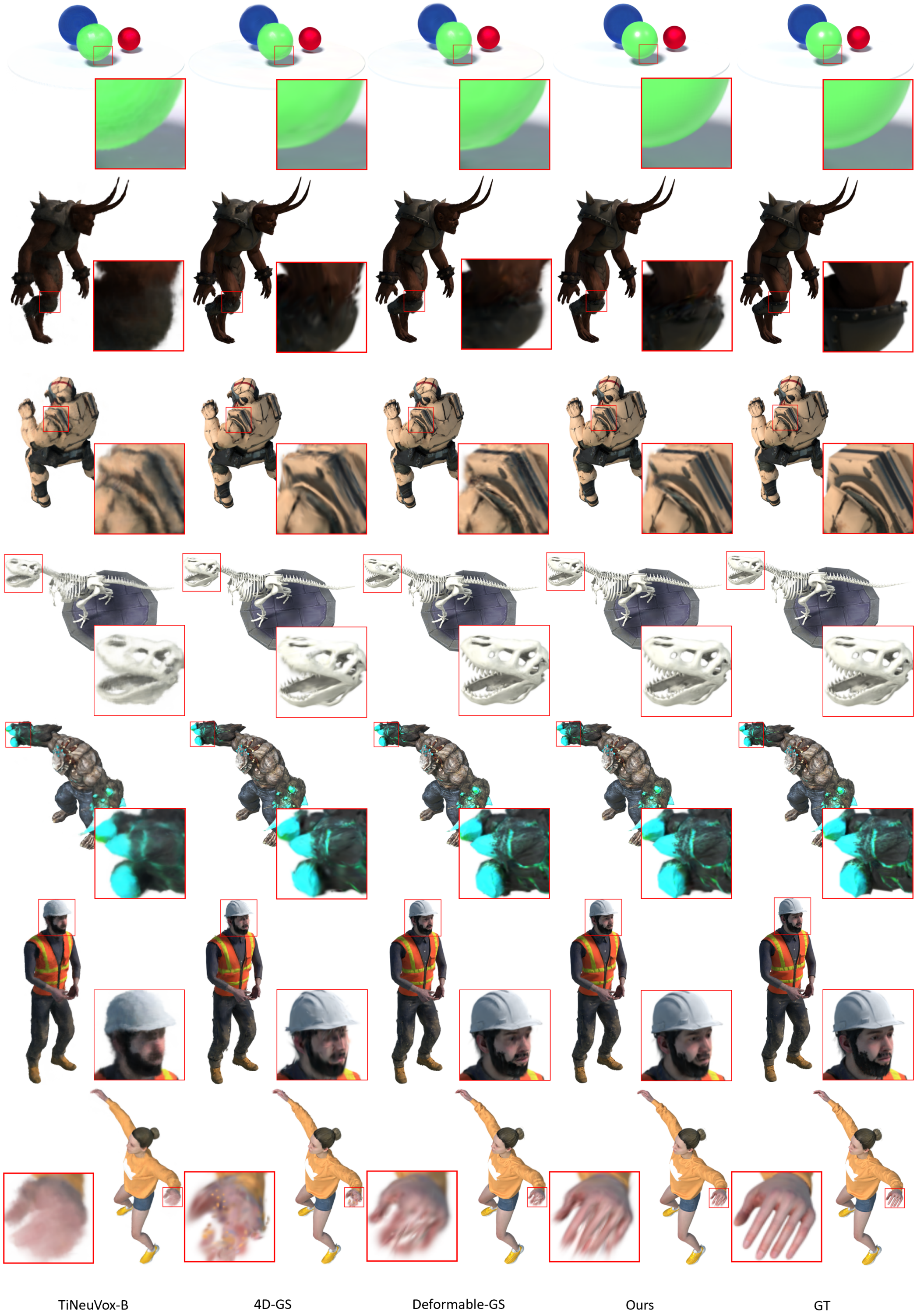


Figure 8: Qualitative results on synthetic dynamic scenes. We compare our methods with TiNeuVox [Fang et al. 2022b], 4D-GS [Wu et al. 2023], and Deform-GS [Yang et al. 2023a].