MoDGS: Dynamic Gaussian Splatting from Causually-captured Monocular Videos

ANONYMOUS AUTHOR(S)*

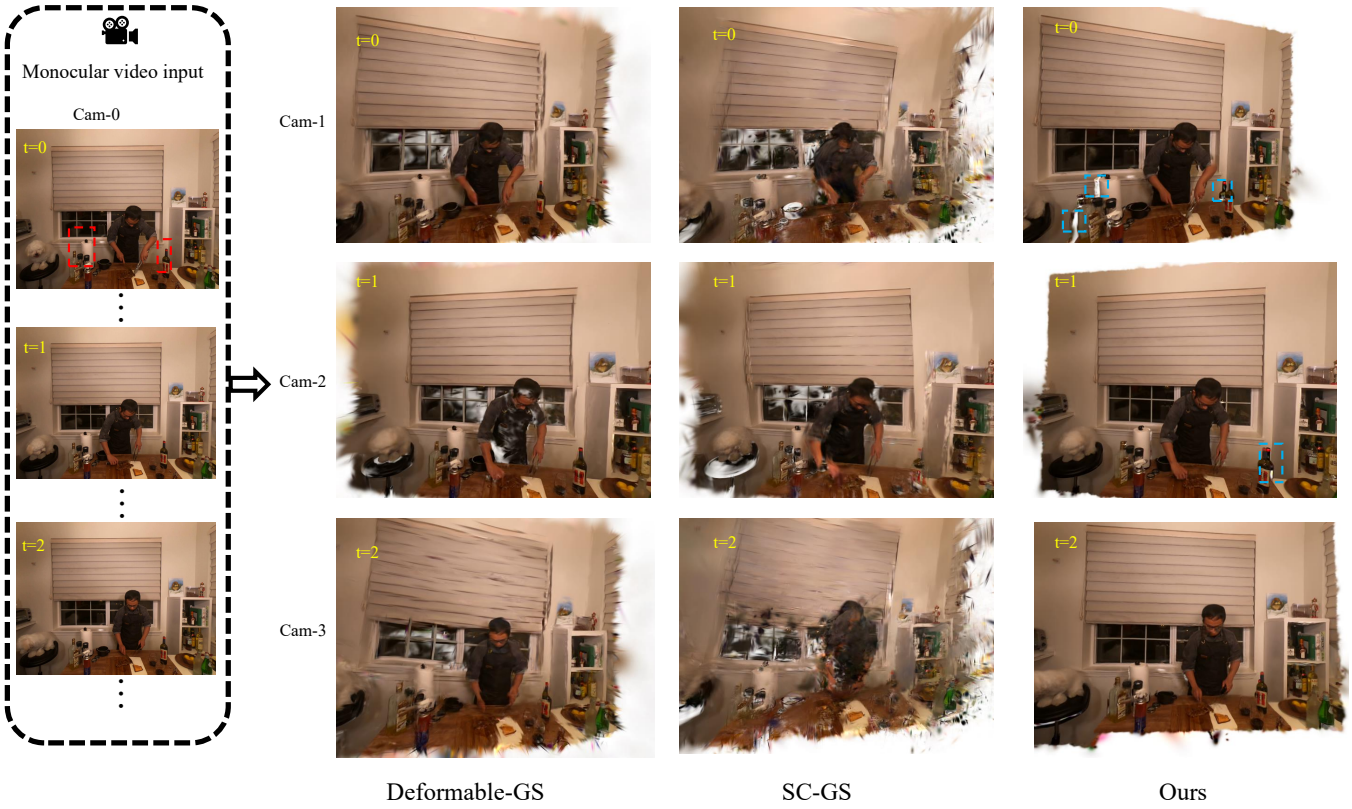


Fig. 1. Given a casually captured monocular video of a dynamic scene, **MoDGS** is able to synthesize high-quality novel-view images in this scene. Note that in this example, the camera of the input video is static. The baseline methods, i.e. Deformable-GS [Yang et al. 2023a] and SC-GS [Huang et al. 2024], fail to correctly reconstruct the 3D dynamic scenes on this static monocular video. The white regions in cyan bounding boxes are not visible in the input video (red bounding boxes) so there are some artifacts for these invisible regions.

In this paper, we propose MoDGS, a new pipeline to render novel-view images in dynamic scenes using only casually captured monocular videos. Previous monocular dynamic NeRF or Gaussian Splatting methods strongly rely on the rapid movement of input cameras to construct multiview consistency but fail to reconstruct dynamic scenes on casually captured input videos whose cameras are static or move slowly. To address this challenging task, MoDGS adopts recent single-view depth estimation methods to guide the learning of the dynamic scene. Then, a novel 3D-aware initialization method is proposed to learn a reasonable deformation field and a new robust depth loss is proposed to guide the learning of dynamic scene geometry. Comprehensive experiments demonstrate that MoDGS is able to render high-quality novel view images of dynamic scenes from just a casually captured monocular video, which outperforms baseline methods by a significant margin. Project page: https://MoDGS.github.io

Additional Key Words and Phrases: Novel View Synthesis, Monocular Video

1 INTRODUCTION

Novel view synthesis (NVS) is an important task in computer graphics and computer vision, which greatly facilitates downstream tasks such as augmentation or virtual reality. In recent years, the novel-view-synthesis quality on static scenes has witnessed great improvements thanks to the recent development of techniques such as NeRF [Mildenhall et al. 2020], Instant-NGP [Müller et al. 2022], and Gaussian Splatting [Kerbl et al. 2023], especially when there are sufficient input images. However, novel view synthesis in a dynamic scene with only one monocular video still remains a challenging task.

Dynamic View Synthesis (DVS) has achieved impressive improvements along with the emerging neural representations and Gaussian splitting techniques. Most of the existing DVS methods [Cao and Johnson 2023; Yang et al. 2023a] require multiview videos captured by dense synchronized cameras to achieve good rendering quality.

Though some works can process a monocular video for DVS, as pointed out by DyCheck [Gao et al. 2022], these methods require the camera of the monocular video to have extremely large movements, which is called "Teleporting Camera Motion" on different viewpoints, so these methods can utilize the multiview consistency provided by this pseudo multiview video to reconstruct the 3D geometry of the dynamic scene. However, such large camera movements are rarely seen in casually captured videos because casually captured videos are usually produced by smoothly moving or even static cameras. When the camera moves slowly or is static, the multiview consistency constraint will be much weaker and all these existing DVS methods fail to produce high-quality novel-view images, as shown in Fig. 1 (second and third column).

In this paper, we present Monocular Dynamic Gaussian Splatting (MoDGS) to render novel-view images from casually captured monocular videos in a dynamic scene. MoDGS addresses the weak multiview constraint problem by adopting a monocular depth estimation method [Fu et al. 2024]. The weak multiview constraint problem disables existing methods to correctly reconstruct the 3D dynamic scenes while the single view depth estimator provides prior depth information on the input video to help the 3D reconstruction. However, we find that simply applying a single-view depth estimator in DVS to supervise rendered depth maps is not enough for high-quality novel view synthesis. First, the depth supervision only provides information for each frame but does not help to associate 3D points between two frames. Thus, we still have difficulty in learning an accurate time-dependent deformation field. Second, the estimated depth values are not consistent among different frames.

To learn a robust deformation field from monocular video, we propose a 3D-aware initialization scheme for the deformation field. Existing methods [Katsumata et al. 2023] solely rely on supervision from 2D flow estimation, which produces deteriorated results without sufficient multiview consistency. We find that directly initializing the deformation field in the 3D space greatly helps the subsequent learning of the 3D representations and improves the rendering quality in the end(Fig. 1,last column).

To better utilize the estimated depth maps for supervision, we propose a novel depth loss to address the scale inconsistency of estimated depth values across different frames. Previous methods [Li et al. 2023b; Liu et al. 2023a] supervise the rendered depth maps using a scale-invariant depth loss by minimizing the $L2$ distance of normalized rendered depth and depth priors, and the most recent method [Zhu et al. 2023]propose to supervise the rendered depth maps using a Pearson correlation loss to mitigate the scale ambiguity between the true scene scale and the estimate depth scale. However, the estimated depth maps of different frames are not even consistent after normalizing to the same scale. We observe that despite the inconsistency in values, the orders of depth values of different pixels in different frames are stable, which motivates us to propose an ordinal depth loss. This novel ordinal depth loss enables us to fully utilize the estimated depth maps for high-quality novel view synthesis.

To demonstrate the effectiveness of MoDGS, we conduct experiments on two widely used datasets, the Nvdia [Yoon et al. 2020] dataset and the DyNeRF [Li et al. 2022] dataset, and a self-collected dataset containing monocular videos from the Internet. We adopt an exact monocular DVS evaluation setting that only uses the video of one camera as input while evaluating the video of another camera. Results show that our method outperforms previous DVS methods by a large margin and achieves high-quality NVS on casually captured monocular videos.

2 RELATED WORKS

In recent years, numerous works have focused on the task of novel view synthesis in both static and dynamic scenes. The main representatives are Neural Radiance Field [Mildenhall et al. 2020] and Gaussian Splatting [Kerbl et al. 2023], along with their variants. In this paper, we primarily focus on view synthesis in dynamic scenes.

2.1 Dynamic NeRF

Recent dynamic NeRF methods can be roughly categorized into two groups. 1) Representing by time-varying neural radiance fields conditioned on time [Gao et al. 2021; Li et al. 2022; Park et al. 2023]. For example, Park et al. [2023] proposes a simple spatiotemporal radiance field by interpolating the feature vectors indexed by time. 2) Representing by a canonical space NeRF and deformation field [Guo et al. 2023; Li et al. 2021; Park et al. 2021a,b; Pumarola et al. 2021; Tretschk et al. 2021; Xian et al. 2021]. For example, NSFF [Li et al. 2021] models the dynamic components using forward and backward flow represented as 3D dense vector fields; Nerfies [Park et al. 2021a] and HyperNeRF [Park et al. 2021b] model the scene dynamics as a deformation field mapping to a canonical space. Recent advances in grid-based NeRFs [Chen et al. 2022; Müller et al. 2022; Sara Fridovich-Keil and Alex Yu et al. 2022] demonstrate that the training of static NeRFs can be significantly accelerated. Consequently, some dynamic NeRF works utilize these grid-based or hybrid representations for fast optimization [Cao and Johnson 2023; Fang et al. 2022; Fridovich-Keil et al. 2023; Guo et al. 2023; Shao et al. 2023; Song et al. 2023; Wang et al. 2023c,b; You and Hou 2023].

2.2 Dynamic Gaussian Splatting

The recent emergence of 3D Gaussian Splatting (3DGS) demonstrates its efficacy for super-fast real-time rendering attributed to its explicit point cloud representation. Recent follow-ups extend 3DGS to model dynamic 3D scenes. Luiten et al. [2023] and Katsumata et al. [2023] track dynamic 3D Gaussians by frame-by-frame training from synchronized multi-view videos. Yang et al. [2023a] propose a deformable version of 3DGS by introducing a deformation MLP network to model the 3D flows. Wu et al. [2023] and Duisterhof et al. [2023] also introduce a deformation field but using a more efficient Hexplane representation [Cao and Johnson 2023]. Yang et al. [2023b] proposes a dynamic representation with a collection of 4D Gaussian primitives, where the time evolution can be encoded by 4D spherical harmonics. Bae et al. [2024] encodes motions with a per-Gaussian feature vector. Some other works [Li et al. 2023a; Liang et al. 2023; Lin et al. 2023] also study how to effectively encode the motions for Gaussians with different bases. To effectively learn the motions of Gaussians, some works [Feng et al. 2023; Huang et al. 2024; Yu et al. 2023] resort to clustering the motions together for a compact representation.

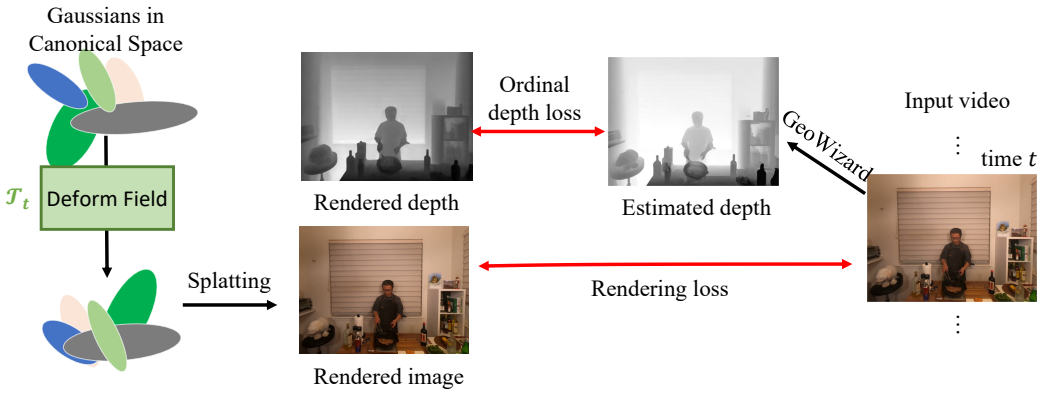


Fig. 2. **Overview**. Given a casually captured monocular video of a dynamic scene, MoDGS represents the dynamic scene with a set of Gaussians in a canonical space and a deformation field represented by an MLP network $\mathcal{T}$. To render an image at a specific timestep t, we deform all the Gaussians by $\mathcal{T}_t$ and then use the splatting technique to render images and depth maps. While in training MoDGS, we use a single-view depth estimator GeoWizard [Fu et al. 2024] to estimate a depth map for every frame and compute the rendering loss and an ordinal depth loss to learn MoDGS.

2.3 DVS from Casual Monocular Videos

As presented in DyCheck [Gao et al. 2022], most existing monocular dynamic view synthesis datasets commonly used for benchmarking, such as D-NeRF [Pumarola et al. 2021], HyperNeRF [Park et al. 2021b], and Nerfies [Park et al. 2021a], either "teleport" frames between multiple cameras (i.e., involve large camera movements between neighboring frames) or feature quasi-static object motions (i.e., small object motions over a long duration). This frame-capturing style facilitates the utilization of multi-view constraints, enabling easy dynamic 3D modeling; however, it is not common in daily casual video captures. When taking casual videos as input, quality degradations appear in the reconstruction results produced by the aforementioned works. A few works explore robust dynamic 3D scene modeling given monocular, casual videos. DynIBaR [Li et al. 2023b] enables long-sequence image-based rendering of dynamic scenes by aggregating features from nearby views following a scene motion-aware manner, but the training cost is large for long-time per-scene optimization. Lee et al. [2023] proposes a hybrid representation that combines static and dynamic components, enabling both fast training and rendering. However, it requires per-frame masks indicating the dynamic components as an additional input. RoDynRF [Liu et al. 2023a] enables robust dynamic NeRF reconstruction by jointly estimating NeRF parameters and camera pose parameters. DpDy [Wang et al. 2024] introduces additional supervision by fine-tuning a diffusion model and imposing this supervision using the SDS loss [Poole et al. 2022], where the quality heavily relies on diffusion models, which require a large amount of computational resources. In contrast, our method is more lightweight, as it does not introduce large neural networks and offers better computing efficiency.

3 METHODOLOGY

Given a casually captured monocular video, we aim to synthesize novel view images from this video. We propose MoDGS, which achieves this by learning a set of Gaussians $\{G_i | i = 1, \cdots, N\}$ in a canonical space and a deformation field $\mathcal{T}_t : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to deform these Gaussians to a specific timestamp t. Then, for a specific timestamp t and a camera pose, we will adopt the splitting technique to render an image.

Overview. As shown in Fig. 2, to train MoDGS, we split the monocular video into a sequence of images $\{I_t | t = 1, ..., T\}$ and we assume that all camera poses of all images are known. We denote our deformation field as a function $x_t = \mathcal{T}_t(x)$, which maps a 3D location $x \in \mathbb{R}^3$ in the canonical 3D space to a location $x_t \in \mathbb{R}^3$ in the 3D space on time t. For every image I_t, we utilize a single-view depth estimator GeoWizard [Fu et al. 2024] to estimate a depth map D_t for every image and utilize a flow estimation method RAFT [Teed and Deng 2020] to estimate a 2D optical flow $F_{t_i \rightarrow t_j}$ between I_{t_i} and I_{t_j} where t_i and t_j are two arbitrary timesteps. Then, we initialize our deformation field by a 3D-aware initialization scheme as introduced in Sec. 3.2. After initialization, we train our Gaussians and deformation field with a rendering loss and a new depth loss introduced in Sec. 3.3. In the following, we first begin with the definition of the Gaussians and the rendering process in MoDGS.

3.1 Gaussians and Deformation Fields

Gaussians in the canonical space. We define a set of Gaussians in the canonical space, we follow the original 3D GS [Kerbl et al. 2023] to define a 3D location, a scale vector, a rotation, and a color with spherical harmonics. Note this canonical space does not explictly correspond to any timestep but is just a virtual space that contains the canonical locations of all Gaussians.

Deformation fields. The deformation field $\mathcal{T}_t$ used in MoDGS follows the design of Omnimotion [Wang et al. 2023a] and Cadex [Lei and Daniilidis 2022] which is an invertible MLP network [Dinh et al. 2016]. This is an invertible MLP means that both $\mathcal{T}_t$ and $\mathcal{T}_t^{-1}$ can be directly computed from the MLP network. All $\mathcal{T}_t$ at different timesteps t share the same MLP network and the time t is normalized to $[0, 1]$ as input to the MLP network. We use this deformation

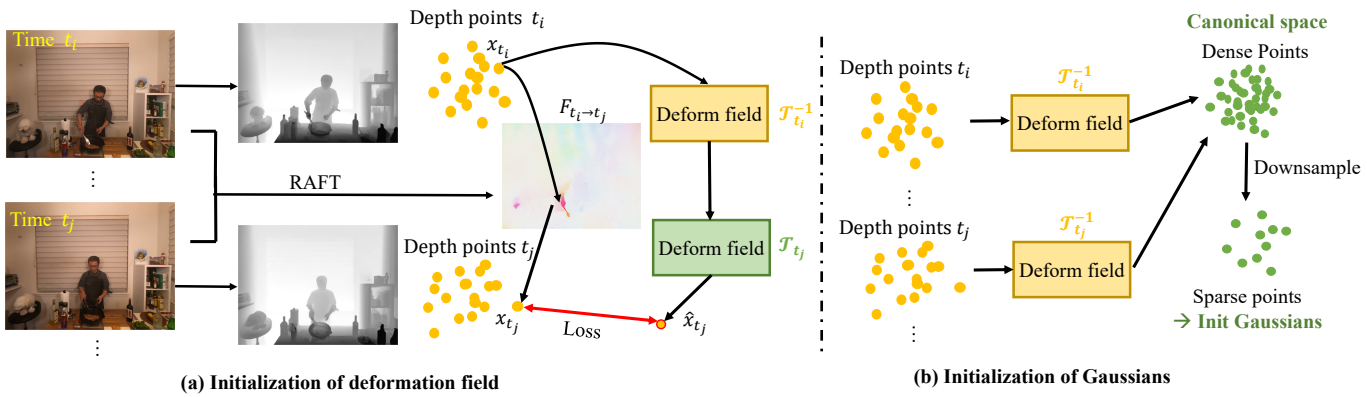


Fig. 3. (a) **Initialization of the deformation field**. We first lift the depth maps and a 2D flow to a 3D flow and train the deformation field for initialization. (b) **Initialization of Gaussians in the canonical space**. We use the initialized deformation field to deform all the depth points to the canonical space and downsample these depth points to initialize Gaussians.

field to deform the 3D locations of Gaussians in the canonical space but do not change the colors, orientations, scales, and rotations.

Render with MoDGS. After training both the Gaussians in canonical space and the deformation field, we will use the deformation field to deform the Gaussians in the canonical space to a specific time step t. Then, we follow exactly the splatting techniques in 3D GS [Kerbl et al. 2023] to render images from arbitrary viewpoints.

3.2 3D-aware Initialization

Original 3D Gaussian splatting [Kerbl et al. 2023] relies on the sparse points from Structure-from-Motion (SfM) to initialize all the locations of Gaussians. When we only have a casually captured monocular video, it is difficult to get an initial set of sparse points for initialization from SfM. Though it is possible to initialize all the Gaussians from the points of the estimated single-view depth of the first frame, we show that this leads to suboptimal results. At the same time, we need to initialize not only the Gaussians but also the deformation field. Thus, in the following, we propose a 3D-aware initialization scheme for MoDGS.

Initialize depth scales. Since the estimated depth maps on different timesteps would have different scales, we first estimate a coarse scale for every frame to unify the scales. We consider the scale of the first frame as 1.0 and normalize all other depth maps to the first frame. Since the input video is captured by a casual camera almost static without extremely large movements, we segment the pseudo-static regions on every timestep by thresholding on 2D flow $F_{t \to t+1}$. Then, on these pseudo-static regions, we reproject the depth values at a specific timestep to the first frame and minimize the difference between the projected depth and the depth of the first frame, which enables us to solve for a scale for every frame. We rectify all depth maps with the computed scales. In the following, we reuse D_t to denote the rectified depth maps by default.

Initialization of the deformation field. As shown in Fig. 3 (left), given two depth maps D_{t_i} and D_{t_j} along with the 2D flow $F_{t_i \to t_j}$, we lift them to a 3D flow $F^{3D}_{t_i \to t_j}$. This is achieved by first converting the depth maps into 3D points in the 3D space. Then, the estimated 2D flow $F_{t_i \to t_j}$ actually associate two sets of 3D points, which results in a 3D flow $F^{3D}_{t_i \to t_j}$. After getting this 3D flow, we then train our deformation field $\mathcal{T}$ with this 3D flow. Specifically, for a pixel in I_{t_i} whose corresponding 3D point is x_{t_i}, we query $F^{3D}_{t_i \to t_j}$ to find its target point x_{t_j} in the t_j timestep. Then, we minimize the difference by

$$\ell_{\text{init}} = \sum \| \mathcal{T}_{t_j} \circ \mathcal{T}_{t_i}^{-1}(x_{t_i}) - x_{t_j} \|^2. \tag{1}$$

We train the MLP networks in $\mathcal{T}$ for a fixed number of steps to initialize the deformation field.

Initialization of Gaussians. After getting the initialized deformation field, we will initialize a set of 3D Gaussians in the canonical space as shown in Fig. 3 (right). We achieve this by first converting all the depth maps to get 3D points. Then, these 3D points are deformed backward to the canonical 3D space. This means that we transform all the depth points of all timesteps to the canonical space, which results in a large amount of points. We then evenly downsample these points with a predefined interval to reduce the point number and we initialize all our Gaussians with the locations of these downsampled 3D points in the canonical space.

3.3 Ordinal Depth Loss

Pearson correlation loss. Existing dynamic Gaussian Splatting or NeRF methods also adopt a depth loss to supervise the learning of their 3D representations. One possible solution [Li et al. 2021; Liu et al. 2023a; Zhu et al. 2023] is to maximize a Pearson correlation between the rendered depth and the estimated single-view depth

$$\text{Corr}\left(\hat{D}_t, D_t\right) = \frac{\text{Cov}\left(\hat{D}_t, D_t\right)}{\sqrt{\text{Var}\left(\hat{D}_t\right) \text{Var}\left(D_t\right)}}, \tag{2}$$

where $\text{Cov}(\cdot,\cdot)$ and $\text{Var}(\cdot,\cdot)$ means the covariance and variance respectively, D_t and $\hat{D}_t$ are the estimated depth and the rendered depth respectively. Since the estimated single-view depth has ambiguity in scale, the Pearson correlation loss avoids the negative

effects of the scale ambiguity. Note that in [Li et al. 2021; Liu et al. 2023a], the loss is called normalized depth loss, which is equivalent to Pearson correlation here as shown in the supplementary material.

Problem in Pearson correlation loss. However, we find that this Pearson correlation depth loss is still suboptimal. As shown in Fig. 4, the estimated depth maps at two different timesteps are still not consistent with each other after normalization. Making two depth maps consistent after normalization actually requires these two depth maps to be related by a linear transformation, i.e. $D_{t+1} = aD_t + b$ with a and b two constants. However, the single-view depth estimation method is not accurate enough to guarantee the linear relationship between two estimated depth maps at different timesteps. In this case, the Pearson correlation loss still brings inconsistent supervision for training the 3D representation.

Ordinal depth loss. To address this problem, our observation is that though we cannot guarantee depth consistency after normalization, as shown in Fig. 4, the order of depth value is consistent among two different frames. Thus, this motivates us to ensure the order of depth is correct by a new ordinal depth loss. We first define an order indicator function

$$\mathcal{R}(D_t(u_1), D_t(u_2)) = \begin{cases} +1, & D_t(u_1) > D_t(u_2) \\ -1, & D_t(u_1) < D_t(u_2) \\ 0, & D_t(u_1) = D_t(u_2) \end{cases}, \tag{3}$$

where $\mathcal{R}$ is the order indicator function on depth map D_t which indicates the order between the depth values of pixels $u_1 \in \mathbb{R}^2$ and $u_2 \in \mathbb{R}^2$, and $D_t(u)$ means the depth value on the pixel u. Then, we define our ordinal depth loss based on the depth order by

$$\ell_{\text{ordinal}} = \|\tanh\left(\alpha(\hat{D}_t(u_1) - \hat{D}_t(u_2))\right) - \mathcal{R}\left(D_t(u_1), D_t(u_2)\right)\|, \tag{4}$$

where $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\hat{D}_t$ means the rendered depth map at time step t, $\hat{D}_t(u)$ is the depth value of this rendered depth map on the pixel u, α is a predefined constant. Eq. 4 means we transform the depth difference between $\hat{D}_t(u_1)$ and $\hat{D}_t(u_2)$ to 1 or -1 by tanh function. Then, we force the depth order of the rendered depth map $\hat{D}_t$ to be consistent with the order in the predicted depth map D_t. In the implementation, we randomly sample 100k pairs (u_1, u_2) to compute the ordinal depth loss.

3.4 Training of MoDGS

After initializing the Gaussians and the deformation fields, we use MoDGS to render at a specific timestep and compute the rendering loss ℓ_{render} and the ordinal depth loss ℓ_{ordinal}. So the total training loss for MoDGS is

$$\ell = \lambda_{\text{ordinal}}\ell_{\text{ordinal}} + \lambda_{\text{render}}\ell_{\text{render}}. \tag{5}$$

4 EXPERIMENTS

4.1 Implementation Details

We implement our MoDGS with PyTorch. To initialize the deformation field, we train it with 30k steps as stated in Sec. 3.2. Subsequently, we jointly train the 3D Gaussians and the deformation field with the rendering loss and the ordinal depth loss for another 30k steps. The downsampling voxel size for Gaussian initialization is 0.004.

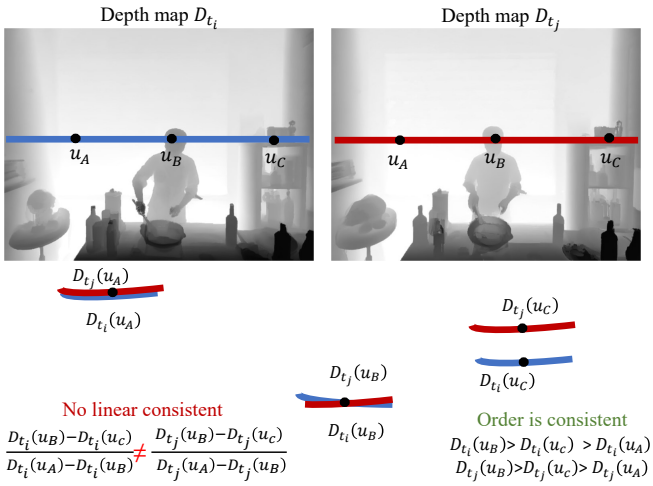


Fig. 4. We show the estimated single-view depth maps at two different timesteps D_{t_i} and D_{t_j} after normalization to the same scale. Since the single-view depth estimator is not accurate enough, the depth maps are not linear related so the scale normalization does not perfectly align them. However, the order of depth values on three corresponding pixels is stable for these two depth maps, which motivates us to propose an ordinal depth loss for supervision.

We adopt an Adam optimizer for optimization. The learning rate for 3D Gaussians exactly follows the official implementation of 3D GS [Kerbl et al. 2023], while the learning rate of the deformation network undergoes exponential decay from 1e-3 to 1e-4 in initialization and from 1e-4 to 1e-6 in the subsequent optimization. We set $\alpha = 100$ for ℓ_{ordinal}. The weight of our depth order loss is 0.1. The whole training takes around 6 hours to converge (3 hours for the initialization and 3 hours for the subsequent optimization) on an NVIDIA RTX A6000 GPU, which uses about 14G memory. The rendering speed of MoDGS is about 70 FPS.

4.2 Evaluation Protocols

Datasets. We conducted experiments on three datasets to demonstrate the effectiveness of our method. The first dataset is the DyNeRF [Li et al. 2022] dataset which consists of 6 scenes. On each scene, we have 18-20 synchronized cameras capturing 10-30 second videos. In these videos, there is mainly a man working on a desktop, like cutting beef or dumping water. We use 5 scenes for the evaluation of the DyNeRF dataset. We use camera0 for training and evaluate the results on camera5 and camera6. The second dataset is the Nvidia [Yoon et al. 2020] dataset which contains more diverse dynamic subjects like jumping, playing with balloons, and opening an umbrella. The Nvidia dataset contains 8 scenes, which also has 12 synchronized cameras. We use 3 scenes for quantitative evaluation. We train all methods on camera4 and evaluate with camera3 and camera5. Other than these widely-used benchmarks, we also collect 6 online videos to construct an in-the-wild dataset, called the Monocular Casual Video (MCV) Dataset, to demonstrate our method can generalize to in-the-wild casual videos. The MCV dataset contains diverse subjects like skating, a dog eating food,

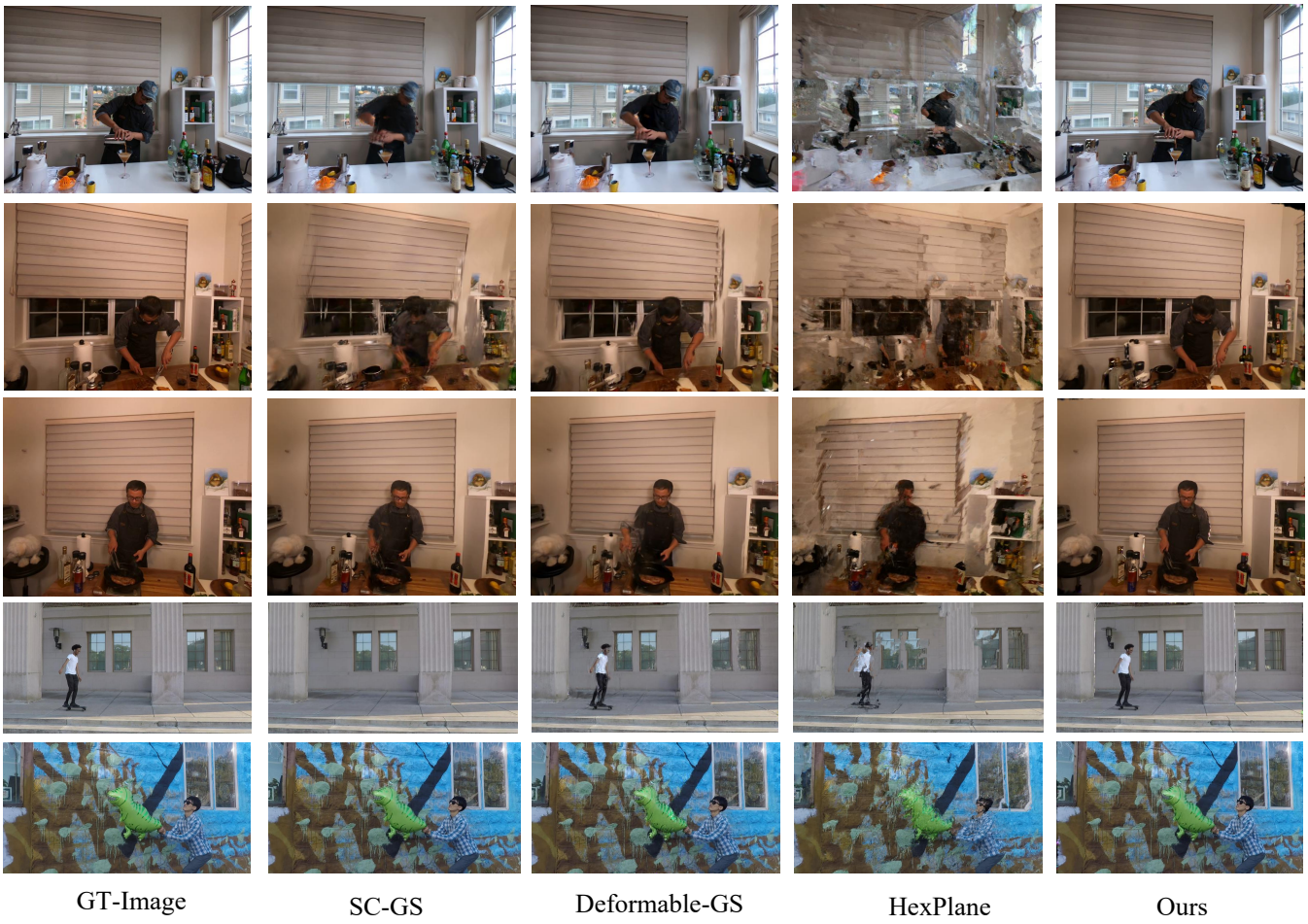


Fig. 5. Qualitative comparison on the novel-view renderings of the DyNeRF [Li et al. 2022] and Nvidia [Yoon et al. 2020] datasets. We compare MoDGS with SC-GS [Huang et al. 2024], Deformable-GS [Yang et al. 2023a], and Hexplane [Cao and Johnson 2023].

YOGA, etc. The MCV dataset only contains a single video for each scene, so we cannot evaluate the quantitative results but only report the qualitative results on this dataset.

Evaluation setting. Previous DVS methods [Cao and Johnson 2023; Gao et al. 2021; Yang et al. 2023a] all use different cameras to train the dynamic NeRF or Gaussian Splatting. Even though they only use one camera at a specific timestep, they use different cameras at different timesteps so that a pseudo multiview video can be constructed to learn the 3D structures of the scene. Since our target is to conduct novel-view synthesis on the casually captured images, we do not adopt such "teleporting camera motions" to construct training videos but just adopt one static camera to record training videos. Then, we render the images from the viewpoints of another camera for evaluation.

Metrics. To evaluate the rendering quality, we have to render images from a new viewpoint and compare them with the ground-truth images. However, if the input video is almost static, the input video will contain insufficient 3D information and there would exist an ambiguity in scale. Thus, different DVS methods would choose different scales in reconstructing dynamic scenes so that the rendered images on the novel viewpoints are not aligned with the given ground-truth images. To address this problem, we manually label correspondences between the training images and ground-truth novel-view images. Then, we render a depth map on the training image using the reconstructed dynamic scene and optimize for a scale factor to scale the depth value to satisfy these labeled correspondences. After aligning the scale factors of different methods with the ground-truth images, we compute the SSIM, LPIPS, and PSNR between the rendered images and the ground-truth images.

Baseline methods. We compare MoDGS with 4 baseline methods to demonstrate the superior ability of MoDGS to synthesize novel-view images with casually captured monocular videos. These methods can be categorized into two classes. The first is the NeRF-based methods including HexPlane [Cao and Johnson 2023] and RoDynRF [Liu et al. 2023a]. HexPlane represents the scene with six feature planes in both 3D space and time-space. We find that HexPlane does not produce reasonable results if only a monocular video

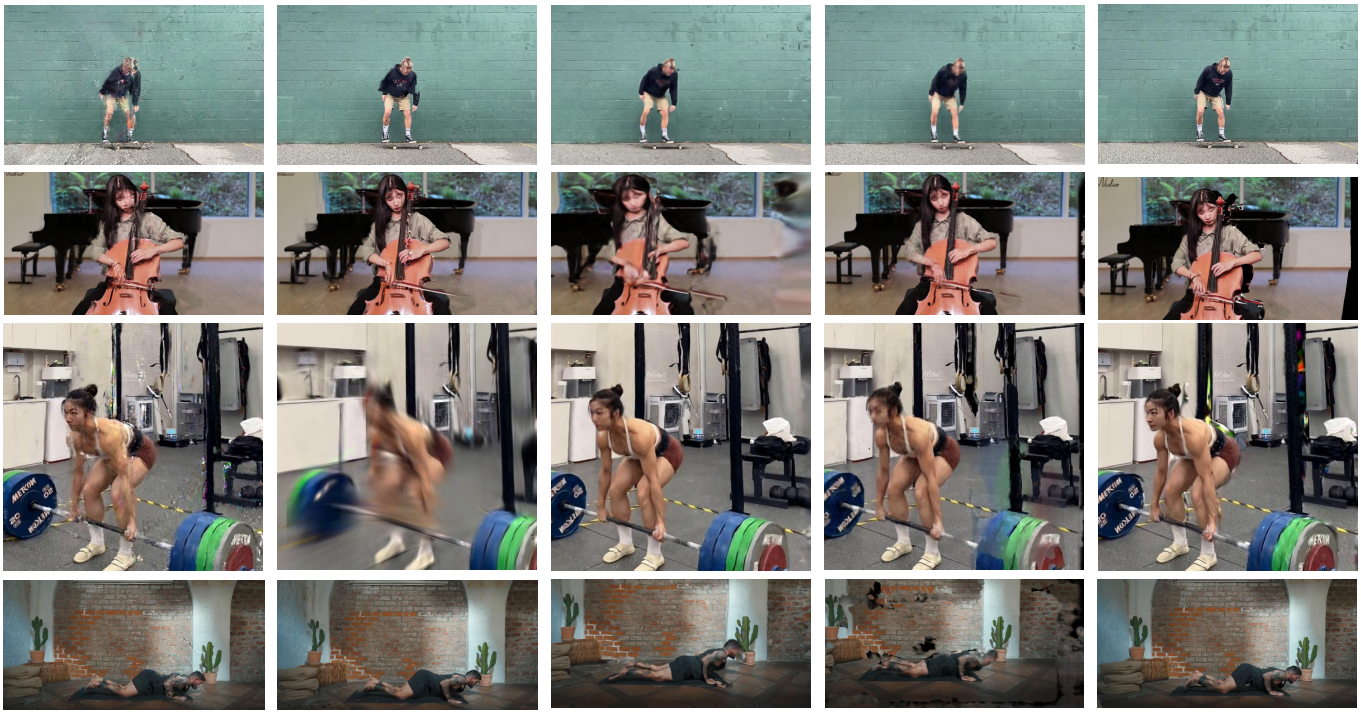


Fig. 6. Qualitative comparison of DVS quality on the MCV dataset. We compare MoDGS with SC-GS [Huang et al. 2024], Deformable-GS [Yang et al. 2023a], Hexplane [Cao and Johnson 2023], and RoDynRF [Liu et al. 2023a].

Table 1. Quantitative results on the Nvidia [Yoon et al. 2020] dataset. We compare our method with SC-GS [Huang et al. 2024], Deformable GS [Yang et al. 2023a] (D-GS) and HexPlane [Cao and Johnson 2023] in PSNR↑, SSIM↑, and LPIPS↓.

Nvidia Dataset												
	Ours			SC-GS			D-GS			Hexplane		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
balloon2-2	**20.47**	**0.5275**	**0.2408**	18.28	0.3893	0.3671	18.57	0.3941	0.3768	18.42	0.3159	0.4881
skatting2	**25.64**	**0.7996**	**0.1518**	23.07	0.7186	0.2002	24.47	0.7582	0.2073	21.39	0.6403	0.3983
truck-2	**23.69**	**0.7455**	**0.1551**	21.48	0.6374	0.2078	21.38	0.6321	0.2142	20.85	0.5687	0.3653
Avg.	**23.27**	**0.6908**	**0.1826**	20.94	0.5817	0.2584	21.47	0.5948	0.2661	20.22	0.5083	0.4172

Table 2. Quantitative results on the DyNeRF [Li et al. 2022] dataset. We compare our method with SC-GS [Huang et al. 2024], Deformable GS [Yang et al. 2023a] (D-GS) and HexPlane [Cao and Johnson 2023] in PSNR↑, SSIM↑, and LPIPS↓.

DyNeRF Dataset												
	Ours			SC-GS			D-GS			HexPlane		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
cut_beef	**23.98**	**0.8221**	**0.1438**	20.69	0.7414	0.2625	22.20	0.7808	0.1931	16.76	0.5382	0.5054
sear_steak	23.53	**0.8126**	**0.1642**	21.23	0.7870	0.2188	**23.56**	0.8101	0.1773	16.89	0.5897	0.5049
coffee_martini	**21.37**	**0.7962**	**0.1473**	19.02	0.7124	0.2151	19.23	0.7013	0.2270	13.26	0.4049	0.5835
cook_spinach	**22.40**	**0.7823**	**0.1728**	16.70	0.7377	0.2117	17.20	0.7195	0.2329	16.95	0.7286	0.2223
flame_steak	**23.23**	**0.8083**	**0.1592**	17.31	0.7532	0.2527	16.62	0.7523	0.2559	16.97	0.7528	0.2543
Avg.	**22.90**	**0.8043**	**0.1575**	18.99	0.7463	0.2322	19.76	0.7528	0.2172	16.17	0.6028	0.4141

from a single camera is given as input. Thus, other than the input monocular video, we use another video from a different viewpoint to train HexPlane for the DVS task. RoDynRF is a SoTA NeRF-based DVS method that also adopts single-view depth estimation as supervision for the 3D dynamic representations. We train it with the same single-view depth estimator GeoWizard [Fu et al. 2024] as ours.

The second class is the Gaussian Splatting-based DVS methods including Deformable-GS [Yang et al. 2023a] and SC-GS [Huang et al. 2024]. Deformable-GS also associates a deformation field with a set of canonical Gaussians for DVS. SC-GS learns a set of keypoints and uses the deformation of these keypoints to blend the deformation of arbitrary 3D points.

4.3 Comparison with Baselines

The qualitative results on the DyNeRF and Nvidia datasets are shown in Fig. 5. Other qualitative results on our MCV dataset are shown in Fig. 6. The quantitative results on the Nvidia and DyNeRF datasets are shown in Table 1 and Table 2 respectively.

Synthesizing novel views from a casually captured monocular video is a challenging task. As shown in Fig. 5, though baseline methods achieve impressive results on these benchmarks with "teleporting camera motions", these methods fail to correctly reconstruct the 3D geometry of the dynamic scenes and produce obvious artifacts on both dynamic foreground and static background. The main reason is that the monocular camera is almost static and does not provide enough multiview consistency to reconstruct high-quality 3D geometry for novel view synthesis. In the third row of Fig. 5, SC-GS [Huang et al. 2024] fails to reconstruct the dynamic foreground subject because SC-GS has an initialization process that treats the whole scene as a static scene and trains on the scene for a number of steps. When the foreground subject is moving with a large motion (like skating from left to right), it would be ignored by the static scene initialization and then we fail to reconstruct in the subsequent steps.

In comparison, our method relies on a 3D-aware initialization which provides a strong basis for the subsequent optimization. Meanwhile, our ordinal depth loss enables the 3D prior from the single-view depth estimator for an accurate reconstruction of the dynamic scenes. The quantitative results on both Table 2 and Table 1 also show that our method achieves the best performances in all metrics on both datasets. Note that there are still some artifacts on occlusion boundaries because the input monocular camera is almost static and these regions are not visible in our input videos.

4.4 Ablation studies

We conduct ablation studies with our initialization and depth loss on the DyNeRF [Li et al. 2022] dataset to demonstrate their effectiveness. The qualitative results are shown in Fig. 7 and Fig. 8 while the quantitative results are shown in Table 3. We also provide additional results in the supplementary material to demonstrate MoDGS achieves much better rendering quality than simple depth warping.

Table 3. Ablation studies with the 3D-aware initialization ("3D-aware Init") and Depth Loss on the DyNeRF [Li et al. 2022] dataset. "Ordinal" means the ordinal depth loss while "Pearson" means the Pearson correlation loss.

3D-aware Init	Loss	PSNR↑	SSIM↑	LPIPS↓
×	Ordinal	21.27	0.7655	0.1984
✓	Pearson	21.77	0.7938	0.1680
✓	Ordinal	**22.96**	**0.8103**	**0.1518**

4.4.1 3D-aware initialization. To show the effectiveness of our 3D-aware initialization, we adopt a random initialization for the deformation field. Based on the random initialization, we still deform all the depth points backward to the canonical space and downsample these points to initialize the Gaussians. Then, we follow the exact same training procedure to train the randomly initialized baseline method. The final results of this random initialization are shown in Fig. 7. In comparison with our 3D-aware initialization, this random initialization produces more artifacts on the dynamic foreground human, which demonstrates that our 3D-aware initialization provides a good initial point for subsequent 3D dynamic scene reconstruction.

4.4.2 Ordinal depth loss. To demonstrate the effectiveness of our proposed ordinal depth, we experiment with two baseline settings, removing the depth loss and utilizing the Pearson correlation as the depth loss. As shown in Fig. 8, when there is no depth loss, the reconstructed 3D geometry contains much noise and obvious artifacts exist in the scene. Adopting the Pearson depth loss improves the quality by linear correlating the rendered depth maps and input depth maps but still produces noisy depth inside a region. In comparison, our ordinal depth loss enables a smooth reconstruction of the depth map in the interior while maintaining sharp edges at boundaries. Thus, the proposed ordinal depth loss enables a more robust reconstruction of the dynamic scene.

4.5 Limitations

Though our method can conduct dynamic view synthesis from casually captured monocular videos, the task is still extremely challenging. One limitation is that our method can only reconstruct the visible 3D parts but cannot imagine the unseen parts, which leads to artifacts when rendering novel view videos on these unseen parts. Incorporating recent 3D-related diffusion generative models [Chung et al. 2023; Liu et al. 2023b; Long et al. 2023] could be a promising direction to solve this problem, which we leave for future works. Another limitation is that the current training time is comparable with existing DVS methods, which could take several hours for a single scene. How to efficiently reconstruct the dynamic field would be an interesting and promising future research topic. Meanwhile, when the camera is completely static, our method strongly relies on the single-view depth estimator to estimate the 3D depth maps. Though existing single-view depth estimators [Bhat et al. 2023; Fu et al. 2024; Ke et al. 2023; Yang et al. 2024] are trained on large-scale datasets and predict reasonable depth maps for most cases, these depth estimators may fail to capture some details which degenerate the quality.

5 CONCLUSION

In this paper, we have presented a novel dynamic view synthesis paradigm called MoDGS. In comparison with existing DVS methods which requires “teleporting camera motions", MoDGS is designed to render novel view images from a casually captured monocular video. MoDGS introduces two new designs to finish this challenging task. First, a new 3D-aware initialization scheme is proposed, which directly initializes the deformation field to provide a reasonable starting point for subsequent optimization. We further analyze the problem of depth loss and propose a new ordinal depth loss to supervise the learning of the scene geometry. Extensive experiments

Fig. 7. Visualization of rendering results using and without using our 3D-aware initialization.

Fig. 8. Visualization of the rendered depth and RGB images using our ordinal depth loss and the Pearson correlation loss.

on three datasets demonstrate superior performances of our method on in-the-wild monocualr videos over baseline methods.

REFERENCES

Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. 2024. Per-Gaussian Embedding-Based Deformation for Deformable 3D Gaussian Splatting. *arXiv preprint arXiv:2404.03613* (2024).

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. 2023. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288* (2023).

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.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*. Springer, 333–350.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. 2023. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384* (2023).

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016).

Bardienus P Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Mike Zheng Shou, Shuran Song, and Jeffrey Ichnowski. 2023. Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes. *arXiv preprint arXiv:2312.00583* (2023).

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

Qi-Yuan Feng, Hao-Xiang Chen, Qun-Ce Xu, and Tai-Jiang Mu. 2023. SLS4D: Sparse Latent Space for 4D Novel View Synthesis. *arXiv preprint arXiv:2312.09743* (2023).

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.

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. 2024. GeoWizard: Unleashing the Diffusion Priors for 3D Geometry Estimation from a Single Image. *arXiv preprint arXiv:2403.12013* (2024).

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.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. 2022. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems* 35 (2022), 33768–33780.

Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. 2023. Forward flow for novel view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16022–16033.

Shi-Sheng Huang, Zixin Zou, Yichi Zhang, Yan-Pei Cao, and Ying Shan. 2024. Sc-neus: Consistent neural surface reconstruction from sparse and noisy views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 2357–2365.

Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. 2023. An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. *arXiv preprint arXiv:2311.12897* (2023).

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. 2023. Repurposing diffusion-based image generators for monocular depth estimation. *arXiv preprint arXiv:2312.02145* (2023).

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).

Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. 2023. Fast View Synthesis of Casual Videos. *arXiv preprint arXiv:2312.02135* (2023).

Jiahui Lei and Kostas Daniilidis. 2022. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6624–6634.

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.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. 2023a. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812* (2023).

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. 2023b. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4273–4284.

Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. 2023. GauFRe: Gaussian Deformation Fields for Real-time Dynamic Novel View Synthesis. *arXiv preprint arXiv:2312.11458* (2023).

Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. 2023. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. *arXiv preprint arXiv:2312.03431* (2023).

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023b. SyncDreamer: Generating Multiview-consistent Images from a Single-view Image. *arXiv preprint arXiv:2309.03453* (2023).

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

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. 2023. Wonder3D: Single Image to 3D using Cross-Domain Diffusion. arXiv:2310.15008 [cs.CV]

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).

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*. Springer, 405–421.

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.

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).

Sungheon Park, Minjung Son, Seokhwan Jang, Young Chun Ahn, Ji-Yeon Kim, and Nahyup Kang. 2023. Temporal interpolation is all you need for dynamic neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4212–4221.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

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

Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *CVPR*.

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.

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.

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 402–419.

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.

Chaoyang Wang, Peiye Zhuang, Aliaksandr Siarohin, Junli Cao, Guocheng Qian, Hsin-Ying Lee, and Sergey Tulyakov. 2024. Diffusion Priors for Dynamic View Synthesis from Monocular Videos. *arXiv preprint arXiv:2401.05583* (2024).

Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. 2023c. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19706–19716.

Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. 2023b. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 76–87.

Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. 2023a. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19795–19806.

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).

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

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891* (2024).

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. 2023b. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642* (2023).

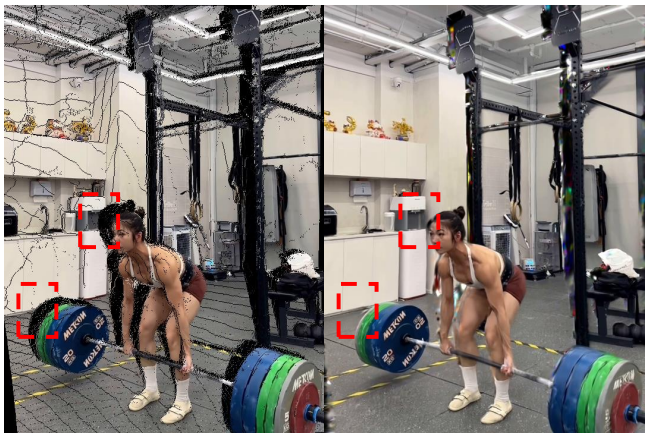
Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5336–5345.

Meng You and Junhui Hou. 2023. Decoupling Dynamic Monocular Videos for Dynamic View Synthesis. *arXiv preprint arXiv:2304.01716* (2023).

Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. 2023. Cogs: Controllable gaussian splatting. *arXiv preprint arXiv:2312.05664* (2023).

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. 2023. FSGS: Real-Time Few-shot View Synthesis using Gaussian Splatting. *arXiv preprint arXiv:2312.00451* (2023).

Fig. 9. We provide a pair of images on the iPhone [Gao et al. 2022] dataset. We draw two correspondences in the same color and their corresponding epipolar lines that are computed from the provided camera poses. The epipolar lines deviate far from the correspondences, which demonstrate that the camera poses are not accurate enough.



Depth Warping Ours

Fig. 10. Comparison of our renderings and depth warping. Our method accumulates information among different timesteps and thus is able to render more completed images.

A SUPPLEMENTARY MATERIALS

A.1 Real Depth Recovery

The depth prediction of GeoWizard [Fu et al. 2024] is normalized depth values in [0,1] and we have to transform them into real depth values. We follow their official implementation to estimate a scaling factor and an offset value on the normalized normal maps by minimizing the normal maps estimated from the transformed real depth values and the estimated normal maps from GeoWizard. The optimization process takes just several seconds. Note that even after this normalization, the depth maps of different timesteps still differ from each other in scale.

A.2 Pose Errors in the iPhone dataset

DyCheck [Gao et al. 2022] adopts the iPhone dataset as the evaluation dataset for the DVS on casually captured videos. We do not adopt this dataset because we find that the poses on this dataset are not very accurate. An example is shown in Fig. 9, where we draw two correspondences and their corresponding epipolar line in the same color. The epipolar line is computed from the provided camera poses. As we can see, the epipolar line does not pass through the correspondence, which means that the provided poses are not accurate enough. We have tried to rerun COLMAP on the iPhone dataset but cannot get reasonable results.

A.3 Difference from Depth Map Warping

MoDGS learns a set of 3D Gaussians in a canonical space and a deformation field to transform it to an arbitrary timestep. This means that MoDGS is able to accumulate information among different timesteps to reconstruct a more completed scene than just using a single-view depth estimation. We show the difference between our renderings and just warping the training view using the estimated single-view depth map in Fig. 10. As we can see, MoDGS produces more completed reconstruction on contents that are not visible on this timestep. Meanwhile, MoDGS rectifies the single-view depth maps to be more accurate so the rendering quality is much better.

A.4 Pearson Depth Loss and Scale-shift Invariant Depth Loss

Previous works [Li et al. 2021; Liu et al. 2023a] use scale and shift invariant depth loss by minimizing the $L2$ distance of normalized ground truth depth map $\text{Norm}(D_t)$ and normalized rendered depth map $\text{Norm}(\hat{D}_t)$

$$\ell_{\text{depth}} = \|\text{Norm}(D_t) - \text{Norm}(\hat{D}_t)\| \tag{6}$$

where $\text{Norm}(D_t)$ denotes $\dfrac{D_t - u_{D_t}}{\sigma_{D_t}}$.

$$\begin{aligned}&\left(\frac{D_t - u_{D_t}}{\sigma_{D_t}} - \frac{D_t - u_{\hat{D}_t}}{\sigma_{\hat{D}_t}}\right)^2 \\ &= -\frac{2(D_t - u_{D_t})(D_t - u_{\hat{D}_t})}{\sigma_{D_t}\sigma_{\hat{D}_t}} + \left(\frac{D_t - u_{D_t}}{\sigma_{D_t}}\right)^2 + \left(\frac{\hat{D}_t - u_{\hat{D}_t}}{\sigma_{\hat{D}_t}}\right)^2.\end{aligned} \tag{7}$$

where D_t is the predicted depth prior, σ_{D_t} and u_{D_t} denotes the standard deviation and means respectively. The second term and third term are constant. The first term is a simple transformation of the Pearson correlation coefficient. Thus, minimizing the $L2$ distance is equivalent to maximizing the coefficient.