DeVRF: Fast Deformable Voxel Radiance Fields for Dynamic Scenes

Jia-Wei Liu[1,*], **Yan-Pei Cao**[2], **Weijia Mao**[1], **Wenqiao Zhang**[4], **David Junhao Zhang**[1], **Jussi Keppo**[5,6], **Ying Shan**[2], **Xiaohu Qie**[3], **Mike Zheng Shou**[1†]

[1] Show Lab, National University of Singapore [2] ARC Lab, [3] Tencent PCG
[4] National University of Singapore [5] Business School, National University of Singapore
[6] Institute of Operations Research and Analytics, National University of Singapore

Abstract

Modeling dynamic scenes is important for many applications such as virtual reality and telepresence. Despite achieving unprecedented fidelity for novel view synthesis in dynamic scenes, existing methods based on Neural Radiance Fields (NeRF) suffer from slow convergence (*i.e.*, model training time measured in *days*). In this paper, we present DeVRF, a novel representation to accelerate learning dynamic radiance fields. The core of DeVRF is to model both the 3D canonical space and 4D deformation field of a dynamic, non-rigid scene with explicit and discrete voxel-based representations. However, it is quite challenging to train such a representation which has a large number of model parameters, often resulting in overfitting issues. To overcome this challenge, we devise a novel *static* $\rightarrow$ *dynamic* learning paradigm together with a new data capture setup that is convenient to deploy in practice. This paradigm unlocks efficient learning of deformable radiance fields via utilizing the 3D volumetric canonical space learnt from multi-view static images to ease the learning of 4D voxel deformation field with only few-view dynamic sequences. To further improve the efficiency of our DeVRF and its synthesized novel view's quality, we conduct thorough explorations and identify a set of strategies. We evaluate DeVRF on both synthetic and real-world dynamic scenes with different types of deformation. Experiments demonstrate that DeVRF achieves two orders of magnitude speedup (**100× faster**) with on-par high-fidelity results compared to the previous state-of-the-art approaches. The code and dataset are released in `https://github.com/showlab/DeVRF`.

1 Introduction

Free-viewpoint photorealistic view synthesis techniques from a set of captured images unleash new opportunities for immersive applications such as virtual reality, telepresence, and 3D animation production. Recent advances in this domain mainly focus on static scenes, e.g., Neural Radiance Fields (NeRF) [21], which implicitly represent rigid static scenes using 5D (spatial locations (x, y, z) and view directions (θ, φ)) neural radiance fields. Although achieving unprecedented fidelity for novel view synthesis, NeRF were mainly exploited under static scenes. To unlock dynamic view synthesis, existing NeRF-based approaches either learn an additional MLP-based deformation field that maps coordinates in dynamic fields to NeRF-based canonical spaces [29, 25, 26, 40] or model dynamic scenes as 4D spatio-temporal radiance fields with relatively large MLPs [17, 7].

*Work is partially done during internship at ARC Lab, Tencent PCG.
†Corresponding Author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

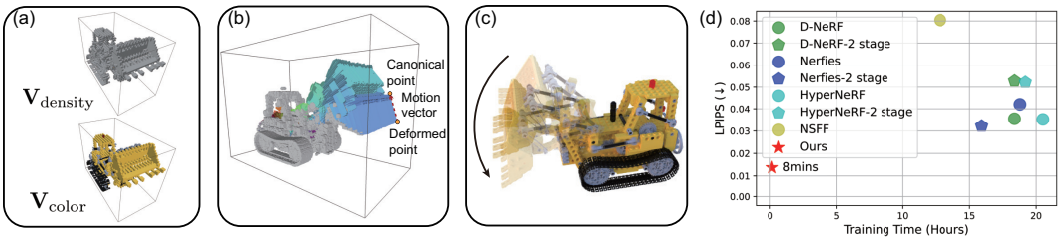

Figure 1: The 3D canonical space **(a)** and the 4D deformation field **(b)** of DeVRF for neural modeling of a non-rigid scene **(c)**. **(d)**: The comparison between DeVRF and SOTA approaches.

Despite being promising, NeRF is notoriously known for suffering from lengthy optimization time. This issue becomes particularly prominent for non-rigid, dynamic scenes because the aforementioned NeRF-based approaches require extra computation for the deformation MLPs [29, 25, 26, 40] or time-varying texture/density querying [17, 7], resulting in quite long training time (in "days").

This motivates us to improve the learning efficiency of dynamic radiance fields. Recent advances in static NeRF [37, 47] show that employing voxel grids, such a volumetric representation, can achieve fast convergence. To adapt for dynamic scenes, one straightforward approach is to incorporate such a volumetric representation into the dynamic radiance field for fast neural modeling. In this paper, we present a novel deformable voxel radiance field (DeVRF) that models both the 3D canonical space and 4D deformation field of a non-rigid, dynamic scene with explicit and discrete voxel-based representations, as illustrated in Fig. 1 (a-c). However, we empirically observe that recklessly learning such a representation in dynamic radiance fields tends to plunge into the local optimum, *i.e.*, the overfitting issue, due to the large number of parameters in DeVRF.

To overcome this overfitting issue, we power our DeVRF with two novel designs: **(1)** We devise an efficient and practical learning paradigm, *i.e.*, **static** $\rightarrow$ **dynamic**, for learning deformable radiance fields. The key idea behind this is that the 3D volumetric canonical space learned from multi-view static images can introduce *inductive bias* [3] to unlock efficient learning of deformable radiance fields. Further, with such 3D priors, a dynamic scene can be effectively modeled with only a few fixed cameras. We argue that such a few-fixed-cameras setup for dynamic scene data capture is more convenient than the moving camera (such as the setup used in D-NeRF [29]) in practice. **(2)** Based on the *static* $\rightarrow$ *dynamic* paradigm, we conduct extensive explorations and identify a set of strategies customised for DeVRF to improve its efficiency and effectiveness. These include a coarse-to-fine training strategy for the 4D deformation field to further improve efficiency, and three objectives to encourage our DeVRF to reconstruct dynamic radiance fields with high fidelity: deformation cycle consistency, optical flow supervisions, and total variation regularization.

Fig. 1 (d) shows that on five inward-facing synthetic scenes, two forward-facing real-world scenes and one inward-facing real-world scene, our approach enables fast dynamic radiance field modeling in about 10 minutes on a single NVIDIA GeForce RTX3090 GPU. This is **100**$\times$ faster than SOTA approaches with comparable novel view synthesis quality.

To summarize, the major contributions of our paper are:

- A novel perspective of DeVRF is presented that enables fast non-rigid neural scene reconstruction, which achieves an impressive 100$\times$ speedup compared to SOTA approaches with on-par high-fidelity.
- To the best of our knowledge, we are the first to incorporate the 4D voxel deformation field into dynamic radiance fields.
- We devise a *static* $\rightarrow$ *dynamic* learning paradigm that can boost performances with a low-cost yet effective capture setup.

2 Related Work

Novel View Synthesis for Static Scenes. Earlier approaches [15, 33, 4, 6, 9, 22] tackle novel view synthesis by first building an explicit 3D reconstruction of a static scene, such as voxels and meshes, and then rendering novel views based on the reconstructed model. On the other hand, multi-plane

images [50, 20] represent a scene with multiple images at different depths and can reconstruct scenes with complex structures. Recently, NeRF [21] achieves unprecedented fidelity for novel view synthesis by modeling static scenes with neural radiance fields. Subsequent works have extended NeRF to different scenarios, such as few-view novel view synthesis [12], multi-scale representation [2], and larger scenes [38, 45, 31]. However, these methods mainly focus on static scenes while the dynamic radiance fields reconstruction is more practical.

Novel View Synthesis for Dynamic Scenes. In order to capture dynamic scenes with non-rigidly deforming objects, traditional non-rigid reconstruction approaches require depth information as additional input or only reconstruct sparse geometry [24, 11, 46, 5, 36]. VolumeDeform [11] introduces a unified volumetric representation to encode both the scene's geometry and its motion but requires RGB-D sequences as input. Neural Volumes [19] represents dynamic objects with a 3D voxel grid plus an implicit warp field, but requires an expensive multi-view capture rig and days to train. Recent studies have built upon NeRF [21] and extended it to dynamic neural radiance field reconstruction by learning a mapping from dynamic to canonical field [29, 25, 26, 40] or building a 4D spatio-temporal radiance field [44, 17, 7, 16]. D-NeRF [29] learns a deformation field that maps coordinates in a dynamic field to a NeRF-based canonical space. Nerfies [25] further associates latent codes in the deformation MLP and the canonical NeRF to tackle more challenging scenes such as moving humans. HyperNeRF [26] proposes to model the motion in a higher dimension space, representing the time-dependent radiance field by slicing through the hyperspace. In contrast, Video-NeRF [44] models the dynamic scene as 4D spatio-temporal radiance fields and addresses motion ambiguity using scene depth. Sharing a similar idea on the 4D spatio-temporal field, NSFF [17] represents a dynamic scene as a time-variant function of geometry and appearance, and warps dynamic scene with 3D scene motion. Lastly, several NeRF-based approaches have been proposed for modeling dynamic humans [8, 43, 18, 28, 34] but can not directly generalize to other scenes. Although achieving promising results, existing methods require days of GPU training time, which is undesirable in real-world applications.

NeRF Acceleration. In the light of NeRF's substantial computational requirements for training and rendering, recent papers have proposed methods to improve its efficiency. A line of work [48, 30, 10] focuses on NeRF rendering acceleration and has achieved encouraging results. As for training acceleration, DVGO [37] models the radiance field with explicit and discretized volume representations, reducing training time to *minutes*. Plenoxels [47] employs sparse voxel grids as the scene representation and uses spherical harmonics to model view-dependent appearance, reaching a similar training speedup. Finally, Instant-ngp [23] proposes multiresolution hash encoding; together with a highly optimized GPU implementation, it can produce competitive results after *seconds* of training. However, existing acceleration methods only focus on static scenes, while hardly any research, to our best knowledge, has studied NeRF acceleration for dynamic scenes. Very recently, Fourier PlenOctrees [41] extends PlenOctrees [48] to dynamic scenes by processing time-varying density and color in the frequency domain; however, the data capturing setup is expensive, and it still requires hours of training. Instead, our proposed algorithm, DeVRF, offers a superior training speed while only requires a few cameras for data capture.

3 Method

3.1 Capture Setup

Deformable scenes undergo various types of deformations and motions, which can result in different scene properties such as object poses, shapes, and occlusions. Therefore, capturing and modeling deformable scenes is nontrivial even for professional photographic studios. Existing approaches [41, 13, 51, 1] attempt to capture 360° inward-facing dynamic scenes with multi-view sequences and thus require dozens of high-quality cameras. On the other hand, D-NeRF [29] reconstructs deformable radiance fields from a sparse set of synthetic images rendered from a moving monocular camera. However, in practice, it is particularly challenging to capture real-world 360° inward-facing dynamic scenes with a single moving camera due to various types of deformations and resulting occlusions in dynamic scenes, especially for scenes undergoing fast deformations. As a result, subsequent studies [25, 26, 17, 7] only capture forward-facing videos of real-world dynamic scenes with a monocular camera.

Table 1: Comparisons of capture setups for dynamic scenes.

Approach	No. of cameras	Cost	Supported real-world use cases
D-NeRF[29], Nerfies[25]	Monocular	Low	Forward-facing scenes, slow reconstruction in *days*.
Neural Volumes[19]	Multiple (34)	High	360° inward-facing scenes, slow reconstruction in *days*.
Fourier PlenOctrees[41]	Multiple (60)	High	360° inward-facing scenes, fast reconstruction in 2hrs.
Ours	Few (4)	Low	360° inward-facing and forward-facing scenes, super-fast reconstruction in 10mins.

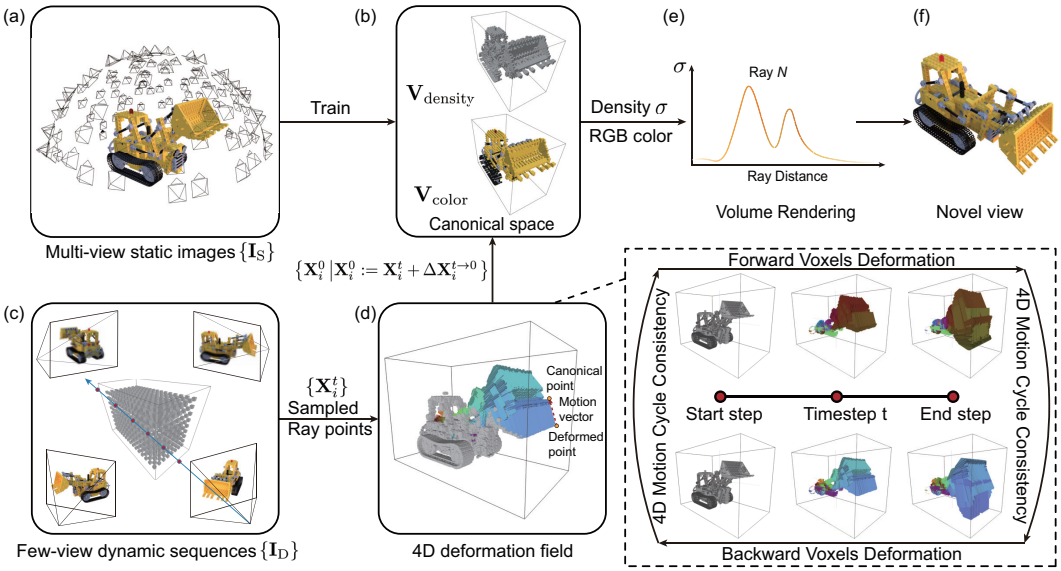

Figure 2: **Overview of our method.** In the first stage, DeVRF learns a 3D volumetric canonical prior **(b)** from multi-view static images **(a)**. In the second stage, a 4D deformation field **(d)** is jointly optimized from taking few-view dynamic sequences **(c)** and the 3D canonical prior **(b)**. For ray points sampled from a deformed frame, their deformation to canonical space can be efficiently queried from the 4D backward deformation field **(d)**. Therefore, the scene properties (*i.e.*, density, color) of these deformed points can be obtained through linear interpolation in the 3D volumetric canonical space, and novel views **(f)** can be accordingly synthesized by volume rendering **(e)** using these deformed sample points.

Compared to dynamic scenes, it is much easier in practice to do multi-view capture for real-world static scenes with a monocular moving camera. Therefore, we propose to separate the capture process of a dynamic scene into two stages: the first stage captures a static state using a moving monocular camera, and the second stage captures the scene in motion using a few fixed cameras. In this capture setup, the multi-view static images provide complete 3D geometry and appearance information of the scene, while few-view dynamic sequences show how the scene deforms in 3D space over time; the entire capture process only requires a few cameras. Tab. 1 compares our capture process with existing approaches in terms of the number of required cameras, cost, and supported real-world use cases.

3.2 Deformable Voxel Radiance Fields

As illustrated in Fig. 2, we present DeVRF to model both the 3D canonical space and 4D deformation field of a non-rigid scene with explicit and discrete voxel-based representations. This volumetric representation allows us to efficiently query the deformation, density, and color of any 3D point at any time step in a deformable scene, thus largely improving the training and rendering efficiency. In addition, we devise a *static* $\to$ *dynamic* learning paradigm that first learns a 3D volumetric canonical prior from multi-view static images (Fig. 2(a-b)) and transfers such prior to dynamic radiance fields reconstruction (Fig. 2(c-f)).

3D Volumetric Canonical Space. We take inspiration from the volumetric representation of DVGO [37] and model the scene properties such as density and color of our 3D canonical space into voxel

grids. Such representation enables us to efficiently query the scene property of any 3D point via trilinear interpolation of its neighboring voxels,

$$\text{Tri-Interp}\left([x, y, z]\,,\, \mathbf{V}_p\right) : \left(\mathbb{R}^3,\, \mathbb{R}^{C \times N_x \times N_y \times N_z}\right) \rightarrow \mathbb{R}^C\,, \forall p \in \{\text{density},\, \text{color}\} \tag{1}$$

where C is the dimension of scene property $\mathbf{V}_p$. N_x, N_y, and N_z are the voxel resolutions of $\mathbf{V}_p$ in x-, y-, z- dimension.

As shown in Fig. 2(a-b), we learn the 3D volumetric canonical prior, *i.e.*, density grid $\mathbf{V}_{\text{density}}$ and color grid $\mathbf{V}_{\text{color}}$, with multi-view static images $\{\mathbf{I}_{\text{S}}\}$ via volumetric rendering. Following DVGO [37], we employ softplus and post-activation after the trilinear interpolation of a 3D point in $\mathbf{V}_{\text{density}}$, as they are critical for sharp boundary and high-frequency geometry reconstruction. We also apply a shallow MLP after the trilinear interpolation of a 3D point in $\mathbf{V}_{\text{color}}$ to enable view-dependent color effects [37]. In our static $\rightarrow$ dynamic learning paradigm, the learned 3D volumetric canonical prior provides critical knowledge of the 3D geometry and appearance of the target dynamic scene, as few-view dynamic sequences alone struggle to reconstruct a complete deformable radiance field with high fidelity (as shown in Section 4).

4D Voxel Deformation Field. We employ a 4D voxel deformation field $\mathbf{V}_{\text{motion}}$ to efficiently represent the motion of a deformable scene. As shown in Fig. 2(d), the arrow directions represent the motions of voxels, the color denotes the motion direction, and the arrow magnitude denotes the motion scale. To synthesize a novel view at time step t, we shoot rays through image pixels and sample ray points $\mathcal{X}_t = \{\mathbf{X}_i^t\}$ in 3D space. The 3D motion $\Delta\mathcal{X}_{t\rightarrow 0} = \{\Delta\mathbf{X}_i^{t\rightarrow 0}\}$ from $\mathcal{X}_t$ to their corresponding 3D points in the canonical space $\mathcal{X}_0 = \{\mathbf{X}_i^0 \mid \mathbf{X}_i^0 = \mathbf{X}_i^t + \Delta\mathbf{X}_i^{t\rightarrow 0}\}$ can be efficiently queried through quadruple interpolation of their neighboring voxels at neighboring time steps in the 4D backward deformation field,

$$\text{Quad-Interp}\left([x, y, z, t]\,,\, \mathbf{V}_{\text{motion}}\right) : \left(\mathbb{R}^4,\, \mathbb{R}^{N_t \times C \times N_x \times N_y \times N_z}\right) \rightarrow \mathbb{R}^C\,, \tag{2}$$

where C is the degrees of freedom (DoFs) of the sample point motion. We use $C = 3$ in this paper, *i.e.*, assign a displacement vector to each sample point. N_t is the number of key time steps that can be user-defined based on the scene motion properties.

Therefore, scene properties of $\mathcal{X}_t$ can then be obtained by querying the scene properties of their corresponding canonical points $\mathcal{X}_0$ through trilinear interpolation in the volumetric canonical space. Finally, pixel colors can be calculated through volume rendering with the sampled scene properties along each ray, as illustrated in Fig. 2(e-f).

3.3 Optimization

Training the DeVRF is quite challenging, mainly because a large number of model parameters may lead to overfitting or suboptimal solutions. This section describes the training strategy and optimization losses that we design to facilitate fast optimization of the DeVRF.

Coarse-to-Fine Optimization. For a dense 4D voxel deformation field with $N_t \times C \times N_x \times N_y \times N_z$ resolution, there could be millions of free parameters, which are prone to overfitting and suboptimal solutions. To solve this problem, we employ a coarse-to-fine training strategy. Specifically, in our experiments, we progressively up-scale the x-y-z resolution of the 4D voxel deformation field from $10 \times 10 \times 10$ to $160 \times 160 \times 160$. With this strategy, the 4D voxel deformation field first learns a rough motion at the coarse stage, which is thereafter progressively refined in finer stages. Our experiments demonstrate that the coarse-to-fine strategy can effectively smoothen the optimization landscape of the 4D voxel deformation field and remove most suboptimal solutions, thus largely improving the training efficiency and accuracy.

Re-rendering Loss. With sampled properties at $\mathcal{X}_t$, the color of a pixel can be calculated through volume rendering, *i.e.*, by integrating the density and color of $\mathcal{X}_t$ along a ray $\mathbf{r}$ [21]:

$$\hat{C}\left(\mathbf{r}\right) = \sum_{i=1}^{N_r} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) c_i + T_{N_r+1}\mathbf{c}_{\text{bg}},\; T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)\,, \tag{3}$$

where N_r is the number of sampled deformed points along the ray, T_i represents the probability of light transmitting through ray $\mathbf{r}$ to the i-th sampled point, and $1 - \exp\left(-\sigma_i \delta_i\right)$ is the probability

that light terminates at the i-th point. δ_i is the distance between adjacent sampled points, and σ_i, c_i denote the density and color of deformed point i, respectively. $\mathbf{c}_{\mathrm{bg}}$ is the pre-defined background color.

Given the few-view training dynamic sequences with calibrated poses $\{\mathbf{I}_{\mathrm{D}}\}$, DeVRF is optimized by minimizing the photometric MSE loss between the observed pixels color $C(\mathbf{r})$ and the rendered pixels color $\hat{C}(\mathbf{r})$:

$$\mathcal{L}_{\mathrm{Render}} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 , \tag{4}$$

where $\mathcal{R}$ is the set of rays in a mini-batch.

4D Deformation Cycle Consistency. As illustrated in Fig. 2(d), the forward deformation is modeled as 4D forward deformation voxels with the same resolution as the 4D backward deformation voxels, and we enforce 4D deformation cycle consistency between backward and forward motion, which regularizes the learned deformation field. In the 4D deformation cycle, backward motion vectors $\Delta\mathcal{X}_{t\to 0}$ models the motion from $\mathcal{X}_t$ to $\mathcal{X}_0$; in contrast, forward motion vectors $\Delta\mathcal{X}_{0\to t}$ models the motion from $\mathcal{X}_0$ to their corresponding 3D points in the dynamic space $\tilde{\mathcal{X}}_t = \{\tilde{\mathbf{X}}_i^t \mid \tilde{\mathbf{X}}_i^t = \mathbf{X}_i^0 + \Delta\mathbf{X}_i^{0\to t}\}$. The 4D motion cycle consistency can now be realized by minimizing the following cycle consistency loss $\mathcal{L}_{\mathrm{Cycle}}(t)$,

$$\mathcal{L}_{\mathrm{Cycle}}(t) = \frac{1}{2N_s} \sum_{i=1}^{N_s} \left\| \mathbf{X}_i^t - \tilde{\mathbf{X}}_i^t \right\|_2^2 , \tag{5}$$

where N_s is the number of sampled 3D points in a mini-batch.

Optical Flow Supervision. The DeVRF is indirectly supervised by 2D optical flows estimated from consecutive frames of each dynamic sequence using a pre-trained RAFT model [39]. For $\mathcal{X}_t$ and their corresponding $\mathcal{X}_0$, we first compute the corresponding 3D points of $\mathcal{X}_0$ at $t-1$ time step via forward motion $\tilde{\mathcal{X}}_{t-1} = \{\tilde{\mathbf{X}}_i^{t-1} \mid \tilde{\mathbf{X}}_i^{t-1} = \mathbf{X}_i^0 + \Delta\mathbf{X}_i^{0\to t-1}\}$. After that, we project $\tilde{\mathcal{X}}_{t-1}$ onto the reference camera and get their pixel locations $\tilde{\mathcal{P}}_{t-1} = \{\tilde{\mathbf{P}}_i^{t-1}\}$, and compute the induced optical flow with respect to the pixel location $\mathcal{P}_t = \{\mathbf{P}_i^t\}$ from which the rays of $\mathcal{X}_t$ are cast. We enforce the induced flow to be the same as the estimated flow by minimizing $\mathcal{L}_{\mathrm{Flow}}(t)$,

$$\mathcal{L}_{\mathrm{Flow}}(t) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \sum_{i=1}^{N_r} w_{\mathbf{r},i} \left| \left(\tilde{\mathbf{P}}_{\mathbf{r},i}^{t-1} - \mathbf{P}_{\mathbf{r},i}^t \right) - \mathbf{f}_{\mathbf{P}_{\mathbf{r},i}^t} \right| , \tag{6}$$

where $w_{\mathbf{r},i} = T_i (1 - \exp(-\sigma_i \delta_i))$ is the ray termination weights from Eq. (3), and $\mathbf{f}_{\mathbf{P}_{\mathbf{r},i}^t}$ is the estimated 2D backward optical flow at pixel $\mathbf{P}_{\mathbf{r},i}^t$.

Total Variation Regularization. We additionally employ a total variation prior [32] when training the 4D voxel deformation field to enforce the motion smoothness between neighboring voxels. At time step t,

$$\mathcal{L}_{\mathrm{TV}}(t) = \frac{1}{2\bar{N}} \sum_{i=1}^{\bar{N}} \sum_{d \in C} \left(\Delta_x^2(\mathbf{v}_i(t), d) + \Delta_y^2(\mathbf{v}_i(t), d) + \Delta_z^2(\mathbf{v}_i(t), d) \right) , \tag{7}$$

where $\Delta_{x,y,z}^2$ is the squared difference of motion vectors between voxel $\mathbf{v}_i$ and its neighbors along x, y, z axes. $\bar{N} = N_x \times N_y \times N_z$ denotes the number of voxels.

Training Objective. The overall training objective of DeVRF is the combination of per-pixel re-rendering loss $\mathcal{L}_{\mathrm{Render}}$, cycle consistency loss $\mathcal{L}_{\mathrm{Cycle}}$, optical flow loss $\mathcal{L}_{\mathrm{Flow}}$, and total variation regularization $\mathcal{L}_{\mathrm{TV}}$:

$$\mathcal{L} = \omega_{\mathrm{Render}} \cdot \mathcal{L}_{\mathrm{Render}} + \omega_{\mathrm{Cycle}} \cdot \mathcal{L}_{\mathrm{Cycle}} + \omega_{\mathrm{Flow}} \cdot \mathcal{L}_{\mathrm{Flow}} + \omega_{\mathrm{TV}} \mathcal{L}_{\mathrm{TV}} , \tag{8}$$

where ω_{Render}, ω_{Cycle}, ω_{Flow}, ω_{TV} are weights for corresponding losses.

Table 2: Averaged quantitative evaluation on inward-facing synthetic and real-world scenes against baselines and ablations of our method. We color code each cell as **best**, **second best**, and **third best**.

	Synthetic inward-facing					Real-world inward-facing				
	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓
Neural Volumes [19]	9.620	0.532	0.5520	19.4	22.4hrs	17.29	0.608	0.3440	19.2	22.0hrs
D-NeRF [29]	31.83	0.960	0.0355	10.0	18.4hrs	29.15	0.946	0.0643	12.4	22.1hrs
D-NeRF [29]-2 stage	28.29	0.945	0.0528	9.7	18.4hrs	27.21	0.936	0.0706	13.2	22.2hrs
D-NeRF [29]-dynamic	17.59	0.839	0.2058	9.8	21.9hrs	21.74	0.911	0.0906	13.5	22.3hrs
Nerfies [25]	33.09	0.989	0.0432	21.8	18.7hrs	29.58	0.980	0.0576	22.5	19.1hrs
Nerfies [25]-2 stage	32.37	0.991	0.0322	22.0	15.8hrs	23.93	0.920	0.0878	22.0	19.7hrs
Nerfies [25]-dynamic	19.45	0.794	0.1674	22.0	21.3hrs	20.70	0.910	0.1080	22.0	19.6hrs
HyperNeRF [26]	33.73	0.965	0.0335	22.5	20.5hrs	28.50	0.944	0.0692	22.0	20.5hrs
HyperNeRF [26]-2 stage	29.16	0.953	0.0555	22.5	19.2hrs	26.53	0.935	0.0802	22.0	19.3hrs
HyperNeRF [26]-dynamic	18.00	0.786	0.2173	22.4	20.6hrs	10.39	0.734	0.3990	22.0	20.5hrs
NSFF [17]	27.06	0.936	0.0800	21.4	12.8hrs	28.44	0.939	0.0714	22.7	15.3hrs
NSFF [17]-dynamic	18.18	0.858	0.1929	15.0	15.5hrs	19.90	0.909	0.0944	22.7	16.2hrs
Ours (base)	22.44	0.887	0.1173	4.4	8mins	24.56	0.917	0.0844	6.6	10mins
Ours w/ c2f	31.97	0.975	0.0185	4.4	7mins	27.83	0.956	0.0465	6.6	10mins
Ours w/ c2f, tv	32.73	0.963	0.0172	4.4	7mins	29.35	0.959	0.0434	6.6	10mins
Ours w/ c2f, tv, cycle	33.97	0.981	0.0142	4.4	8mins	31.56	0.971	0.0292	6.6	11mins
Ours w/ c2f, tv, cycle, flow	34.29	0.982	0.0137	4.4	8mins	31.68	0.972	0.0289	6.6	11mins

4 Experiments

We extensively evaluate the DeVRF on various types of datasets, including five synthetic[3] 360° inward-facing dynamic scenes, two real-world forward-facing dynamic scenes, and one real-world 360° inward-facing dynamic scene. We run all experiments on a single NVIDIA GeForce RTX3090 GPU. During training, we set $\omega_{\text{Render}} = 1$, $\omega_{\text{Cycle}} = 100$, $\omega_{\text{Flow}} = 0.005$, and $\omega_{\text{TV}} = 1$ for all scenes.

4.1 Comparisons with SOTA Approaches

To demonstrate the performance of DeVRF, we compare DeVRF to various types of SOTA approaches, including a volumetric method Neural Volumes [19], NeRF-based methods D-NeRF [29], Nerfies [25], HyperNeRF [26], and a time-modulated method NSFF [17]. For a fair comparison, since DeVRF follows a static $\rightarrow$ dynamic learning paradigm, we additionally implement 2-stage versions of D-NeRF, Nerfies, and HyperNeRF to learn a canonical space prior in the first stage and then optimize a deformation network in the second stage. To show the effectiveness of our low-cost capture strategy for dynamic scenes, we also train these baselines using only a few-view dynamic sequences and observe a significant performance drop compared to those trained with both static and dynamic data. For quantitative comparison, peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [42], and Learned Perceptual Image Patch Similarity (LPIPS) [49] with VGG [35] are employed as evaluation metrics [4].

Evaluation on inward-facing synthetic and real-world deformable scenes. We selected five synthetic dynamic scenes with various types of deformations and motions, and rendered synthetic images in 400×400 pixels under the 360° inward-facing setup. For each scene, we use 100-view static images and 4-view dynamic sequences with 50 frames (*i.e.*, time steps) as training data for all approaches, and randomly select another 2 views at each time step for test. In addition, we collected one 360° inward-facing real-world deformable scene in 540×960 pixels. With our data capture setup, only 4 cameras are required to capture dynamic scenes, and we choose 3 views of them as training data and the other view as test data.

We report the metrics of the real-world scene as well as the average metrics of five synthetic scenes for all approaches in Tab. 2 and leave the per-scene metrics to supplementary material. As shown in Tab. 2, for synthetic and real-world scenes, DeVRF achieves the best performance in terms of PSNR and LPIPS, and the second- or third-best in terms of SSIM among all approaches. Most importantly, our per-scene optimization only takes less than 10mins with 4.4GB to 6.6GB GPU memory on a

[3]The *Lego* scene is shared by NeRF [21], licensed under the Creative Commons Attribution 3.0 License: https://creativecommons.org/licenses/by/3.0/. Other scenes are purchased from TurboSquid, licensed under the TurboSquid 3D Model License: https://blog.turbosquid.com/turbosquid-3d-model-license/.

[4]Although improving test-time rendering speed is not the focus of our paper, DeVRF achieves $16\times \sim 32\times$ test-time rendering speedup compared with other approaches, averaging 0.875 seconds per 540×960 image.

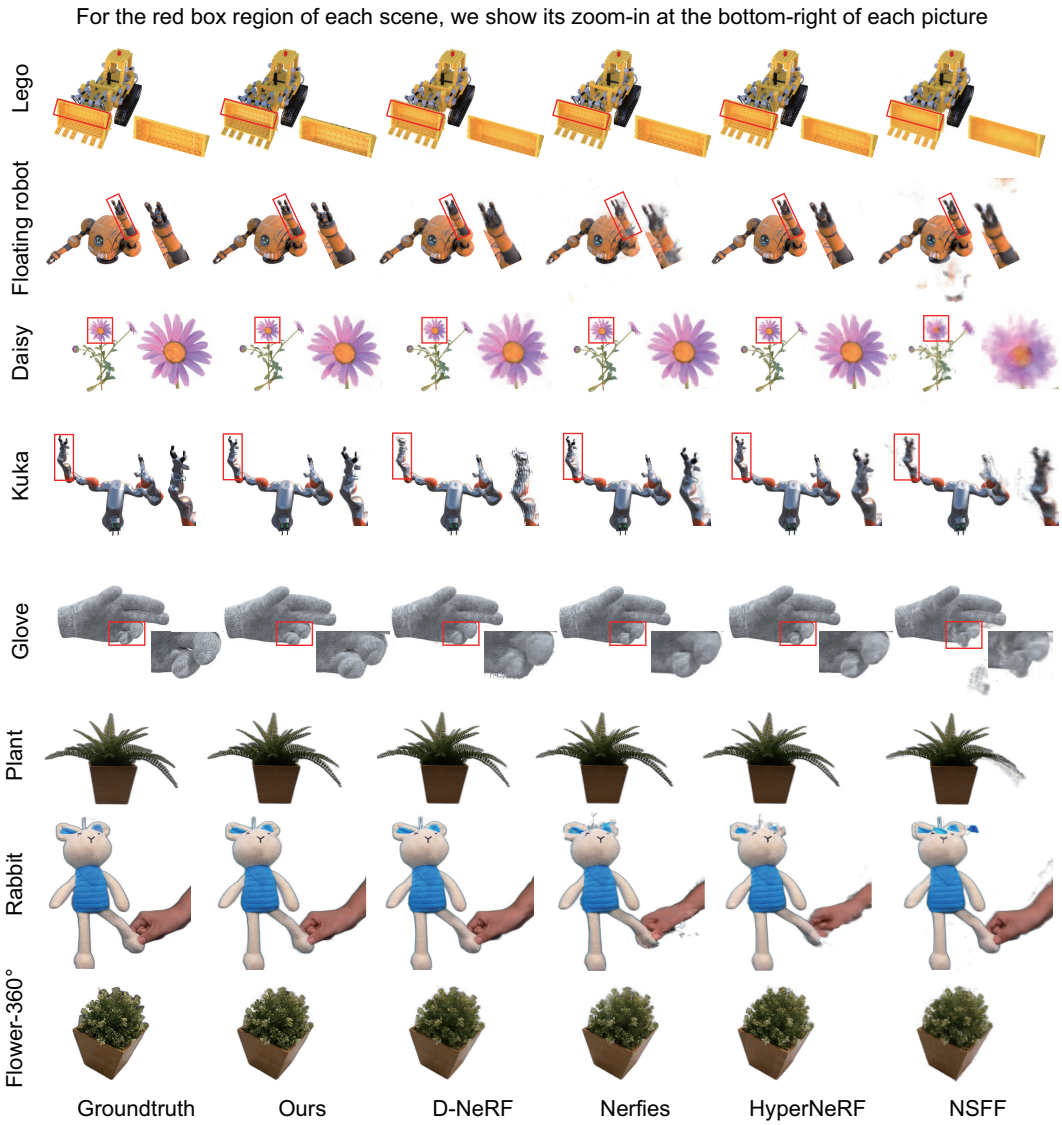

Figure 3: Qualitative comparisons of baselines and DeVRF on synthetic and real-world scenes.

single NVIDIA GeForce RTX3090 GPU, which is about two orders of magnitude faster than other approaches. The above quantitative comparison demonstrates the efficiency and effectiveness of DeVRF. Besides, the qualitative results of DeVRF and baselines on synthetic and real-world scenes are illustrated in Fig. 3, where DeVRF achieves on-par high-fidelity in comparison to SOTA methods. Please see the supplementary video for more results.

For a fair comparison, we additionally report the results of the 2-stage versions for D-NeRF [29], Nerfies [25], and HyperNeRF [26] in Tab. 2. Since these approaches are not designed to separately learn a canonical space and a deformation field, there is no significant difference in the results between their 2-stage versions and 1-stage versions using the training dataset. Furthermore, we also report the results of these baselines trained with dynamic data only (baseline-dynamic) in Tab. 2. Their performances drop significantly compared to the results trained with both static and dynamic data. In addition, since Neural Volumes [19] requires dozens of dynamic sequences as input, its performance is poor with our few-view dynamic sequences. The experimental results not only show the effectiveness of our low-cost data capture process and the proposed DeVRF model; but also validate our observation that few-view dynamic sequences alone fail to provide complete information about the dynamic scene, while static multi-view data can favorably serve as a supplement.

Table 3: Quantitative evaluation on forward-facing real-world scenes against baselines and ablations of our system. We color code each cell as **best**, **second best**, and **third best**.

	PLANT					RABBIT				
	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓
D-NeRF [29]	31.94	0.979	0.0251	11.4	21.5hrs	33.51	0.974	0.0384	11.4	21.7hrs
Nerfies [25]	31.36	0.991	0.0309	21.9	18.3hrs	24.83	0.952	0.0795	21.9	19.8hrs
HyperNeRF [26]	32.08	0.978	0.0331	22.0	20.0hrs	24.97	0.942	0.0849	22.0	20.7hrs
NSFF [17]	29.45	0.966	0.0526	20.2	14.5hrs	27.68	0.945	0.0854	20.2	14.5hrs
Ours (base)	26.13	0.946	0.0722	8.1	8mins	25.58	0.910	0.1300	8.9	6mins
Ours w/ c2f	31.85	0.980	0.0275	8.1	8mins	26.79	0.938	0.0946	8.9	6mins
Ours w/ c2f, tv	31.89	0.980	0.0263	8.1	8mins	29.28	0.951	0.0655	8.9	6mins
Ours w/ c2f, tv, cycle	31.99	0.981	0.0235	8.1	9mins	31.05	0.963	0.0543	8.9	7mins
Ours w/ c2f, tv, cycle, flow	32.01	0.981	0.0236	8.1	10mins	32.05	0.966	0.0492	8.9	7mins

Evaluation on forward-facing real-world deformable scenes. We collected two forward-facing real-world deformable scenes in 540×960 pixels using 4 cameras, and we chose 3 views of them as training data and the other view as test data. To handle forward-facing scenes, we adapt DeVRF to use normalized device coordinates (NDC) and multi-plane images (MPI) as in DVGO [37]. As shown in Tab. 3, DeVRF achieves the best result in the *plant* scene in terms of LPIPS metric and the second-best result in the *rabbit* scene in terms of all metrics. Fig. 3 also demonstrates qualitative comparisons on these two scenes.

4.2 Ablation Study

Ablation study of DeVRF components. We carry out ablation studies on both synthetic and real-world scenes to evaluate the effectiveness of each proposed component in DeVRF. We progressively ablate each component from optical flow, cycle consistency, total variation, to coarse-to-fine strategy. As shown in Tab. 2 and 3, the performance of DeVRF progressively drops with the disabling of each component, where disabling the coarse-to-fine training strategy causes the most significant performance drop. This is as expected since the coarse-to-fine training strategy is critical to reducing local minimums during optimization.

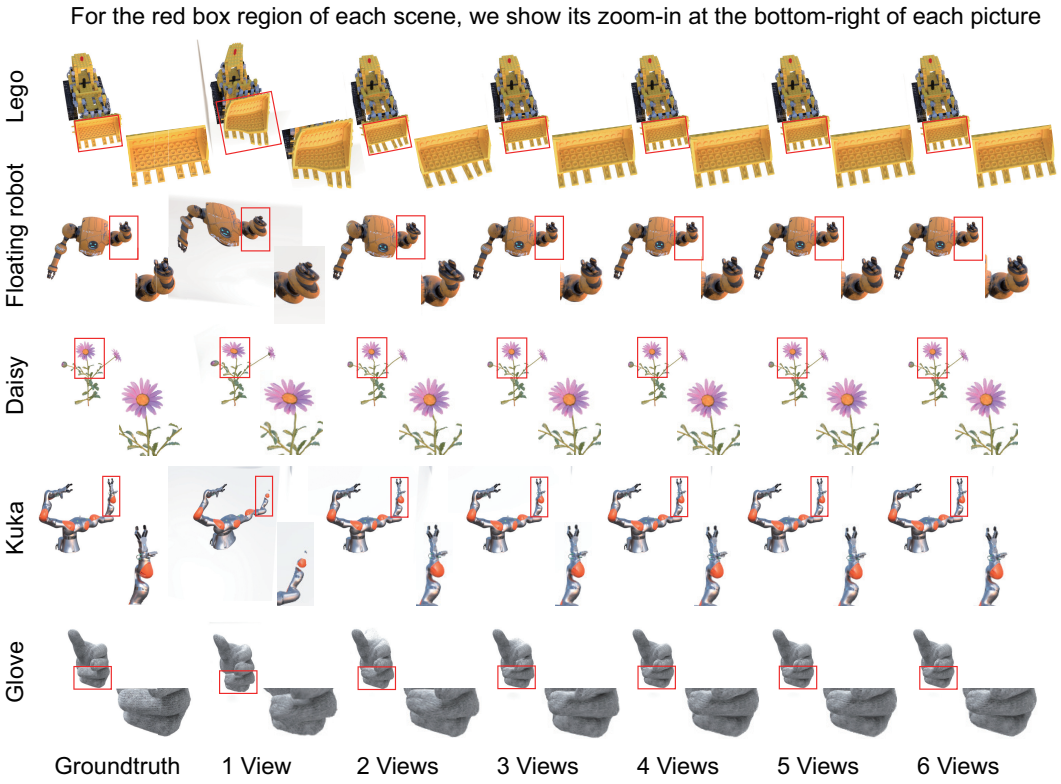

Figure 4: Qualitative results of DeVRF with different numbers of dynamic training views.

Ablation study of dynamic training views on synthetic dataset. Our static → dynamic learning paradigm is based on a low-cost yet effective capture setup: the multi-view static images provide complete 3D geometry and appearance information of the scene, while few-view dynamic sequences show how the scene deforms in 3D space over time. The entire capture process only requires a few cameras that are convenient to deploy in practice. To evaluate the influence of the number of dynamic views, we conduct additional ablations for DeVRF on five inward-facing synthetic dynamic scenes and report the per-scene metrics as well as the average metrics with respect to the number of dynamic training views. As shown in Fig. 5, given the same multi-view static images, the performance of DeVRF largely improves with the increment of dynamic training views and almost saturates at six dynamic views, and the four dynamic training views used in our paper can yield comparable results compared to six dynamic views. We additionally visualize the qualitative results of DeVRF with different numbers of dynamic training views in Fig. 4. Therefore, in our *static* → *dynamic* learning paradigm, with static multi-view data as a supplement, only a few (e.g., four) dynamic views are required to significantly boost the performance of dynamic neural radiance fields reconstruction. This further demonstrates the effectiveness of our low-cost data capture process and the DeVRF model.

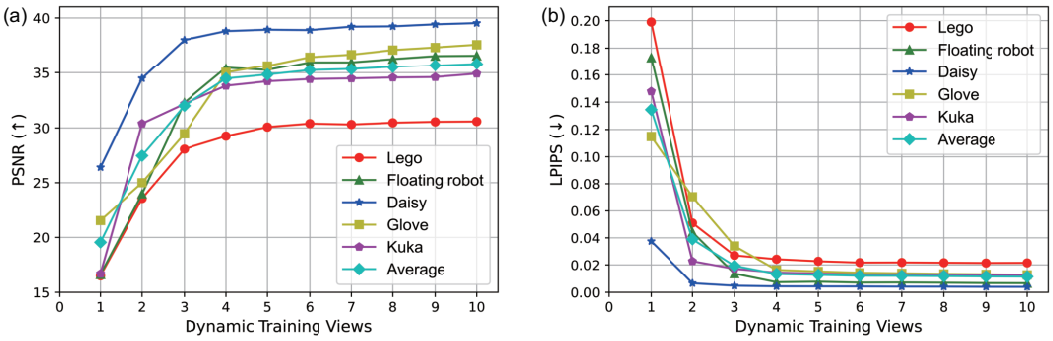

Figure 5: Ablation evaluation on the number of dynamic training views: (a) PSNR, (b) LPIPS.

5 Conclusion

We introduced DeVRF, a novel approach to tackle the challenging task of fast non-rigid radiance field reconstruction by modeling both the 3D canonical space and 4D deformation field of a dynamic scene with voxel-based representations. The DeVRF can be efficiently optimized in two major steps. We first proposed a *static* → *dynamic* learning paradigm to pinpoint that the 3D volumetric canonical prior can be effectively transferred into the 4D voxel deformation field. Second, based on this learning paradigm, we developed a series of optimization strategies, including coarse-to-fine learning, deformation cycle consistency, optical flow supervisions, and total variation priors. Such DeVRF finally produced a 100× faster training efficiency with on-par high-fidelity results in comparison to SOTA approaches. We believe our DeVRF can provide a complement to existing literature and new insights into the view synthesis community.

Limitations and Future Work. Although DeVRF achieves fast deformable radiance field reconstruction, the model size is large due to its large number of parameters. In addition, DeVRF currently does not synchronously optimize the 3D canonical space prior during the second stage, and thus may not be able to model drastic deformations. We consider these limitations as faithful future directions.

6 Acknowledgements

This project is supported by the National Research Foundation, Singapore under its NRFF Award NRF-NRFF13-2021-0008, Mike Zheng Shou's Start-Up Grant from NUS, and National Research Foundation, Singapore and A*STAR, under its RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) grant call (Grant No. I2001E0059). The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore. Jia-Wei Liu is supported by NUS IDS-ISEP scholarship.

References

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

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

[4] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001.

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

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996.

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

[8] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. *arXiv preprint arXiv:2112.01554*, 2021.

[9] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Trans. Graph.*, 35(6), nov 2016.

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

[11] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379. Springer, 2016.

[12] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.

[13] Hanqing Jiang, Haomin Liu, Ping Tan, Guofeng Zhang, and Hujun Bao. 3d reconstruction of dynamic scenes with multiple handheld cameras. In *European Conference on Computer Vision*, pages 601–615. Springer, 2012.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

[16] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. *arXiv preprint arXiv:2103.02597*, 2022.

[17] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 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*, pages 6498–6508, 2021.

[18] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.

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

[20] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[22] Xin Min, Wenqiao Zhang, Shouqian Sun, Nan Zhao, Siliang Tang, and Yueting Zhuang. Vpmodel: High-fidelity product simulation in a virtual-physical environment. *IEEE transactions on visualization and computer graphics*, 25(11):3083–3093, 2019.

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

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

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

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

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

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

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

[31] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. *arXiv preprint arXiv:2111.14643*, 2021.

[32] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st international conference on image processing*, volume 1, pages 31–35. IEEE, 1994.

[33] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[34] Ruizhi Shao, Hongwen Zhang, He Zhang, Mingjia Chen, Yanpei Cao, Tao Yu, and Yebin Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In *CVPR*, 2022.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Haoxuan Song, Jiahui Huang, Yan-Pei Cao, and Tai-Jiang Mu. Hdr-net-fusion: Real-time 3d dynamic scene reconstruction with a hierarchical deep reinforcement network. *Computational Visual Media*, 7(4):419–435, 2021.

[37] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

[38] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. *arXiv preprint arXiv:2202.05263*, 2022.

[39] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

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

[41] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Lan Xu, and Jingyi Yu. Fourier plenoctrees for dynamic radiance field rendering in real-time. *arXiv preprint arXiv:2202.08614*, 2022.

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[43] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. *arXiv preprint arXiv:2201.04127*, 2022.

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

[45] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Citynerf: Building nerf at city scale. *arXiv preprint arXiv:2112.05504*, 2021.

[46] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 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*, pages 5336–5345, 2020.

[47] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

[48] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[50] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[51] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We believe our method will not bring negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and dataset are released in `https://github.com/showlab/DeVRF`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Supplementary Material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] In our experiments, our model converged well, and random seeds had little effect on results.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See references and the footnote in Section 4.
   (b) Did you mention the license of the assets? [Yes] See the footnote in Section 4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We discussed this in the footnote in Section 4.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We believe the data we are using does not contain personally identifiable information and offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

This supplementary material is organized as follows:

- Section A provides more implementation details of the proposed DeVRF.
- Section B presents additional results of the per-scene evaluation as well as evaluations on an additional forward-facing real-world deformable scene.
- Section C conducts additional ablations to further verify the effectiveness of our low-cost data capture process and the proposed DeVRF model.

In addition to this supplementary material, it is worth noting that we also provide a **supplementary video** to better visualize and compare our results to other SOTA approaches on all synthetic and real-world deformable scenes.

A Implementation Details

We use the PyTorch [27] deep learning framework to conduct all our experiments on a single NVIDIA GeForce RTX3090 GPU.

3D canonical space optimization. During training, we set the voxel resolution of 3D canonical space, *i.e.*, density grid $\mathbf{V}_{\text{density}}$ and color grid $\mathbf{V}_{\text{color}}$, to $160 \times 160 \times 160$ for inward-facing scenes and $256 \times 256 \times 128$ for forward-facing scenes, and we use a shallow MLP with 2 hidden layers (128 channels for inward-facing scenes, and 64 channels for forward-facing scenes). The 3D canonical space is optimized using a standard Adam optimizer [14] for 20k with a batch size of 8192 rays for inward-facing scenes and 4096 rays for forward-facing scenes. The learning rate of $\mathbf{V}_{\text{density}}$, $\mathbf{V}_{\text{color}}$, and the designed MLP are set to 10^{-1}, 10^{-1}, and 10^{-3}, respectively.

4D voxel deformation field optimization. The dense 4D voxel deformation field is modeled in $N_t \times C \times N_x \times N_y \times N_z$ resolution, which corresponds to $50 \times 3 \times 160 \times 160 \times 160$. In our proposed coarse-to-fine optimization, we progressively upscale the $(x\text{-}y\text{-}z)$ resolution of the 4D voxel deformation field $\mathbf{V}_{\text{motion}}$ as $(10 \times 10 \times 10) \rightarrow (20 \times 20 \times 20) \rightarrow (40 \times 40 \times 40) \rightarrow (80 \times 80 \times 80) \rightarrow (160 \times 160 \times 160)$. Such an optimization strategy can estimate the fine-grained voxels motion from a cascaded learning sequence. The base learning rate of the 4D voxel deformation field is 10^{-3}, which is progressively decayed to 10^{-4} during coarse-to-fine optimization. For loss weights, we set $\omega_{\text{Render}} = 1$, $\omega_{\text{Cycle}} = 100$, $\omega_{\text{Flow}} = 0.005$, and $\omega_{\text{TV}} = 1$ across all scenes. The 4D voxel deformation field is optimized using Adam optimizer [14] for 25k iterations with a batch size of 8192 rays.

B Additional Results

B.1 Per-scene Evaluation on Inward-facing Synthetic Deformable Scenes.

For quantitative comparison, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [42], and Learned Perceptual Image Patch Similarity (LPIPS) [49] with VGG [35] are employed as evaluation metrics. PSNR and SSIM are simple and shallow functions, while LPIPS measures the perceptual similarity of deep visual representations and is more representative of visual quality. We report the per-scene comparisons on five inward-facing synthetic dynamic scenes - Lego, Floating robot, Daisy, Glove, Kuka - in Tab. 4 and Tab. 5. DeVRF achieves the best performance in terms of LPIPS in five scenes, and almost the second- or third-best in terms of PSNR and SSIM among all approaches. For the floating robot, daisy, and kuka, DeVRF achieves the best performance in terms of both the PSNR and LPIPS. Most importantly, our per-scene optimization only takes less than 10mins with less than 5.0GB GPU memory on a single NVIDIA GeForce RTX3090 GPU, which is about two orders of magnitude faster than other approaches.

B.2 Per-scene Video Comparisons on Synthetic and Real-world Deformable Scenes.

We also provide a supplementary video to better visualize and compare our results to SOTA approaches on all five synthetic and three real-world deformable scenes. As can be seen from the video,

Table 4: Per-scene quantitative evaluation on inward-facing synthetic scenes (Lego, Floating robot, and Daisy) against baselines and ablations of our method. We color code each cell as **best**, **second best**, and **third best**.

	LEGO					FLOATING ROBOT					DAISY				
	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓
Neural Volumes [19]	5.958	0.369	0.8314	19.4	22.4hrs	6.403	0.405	0.7127	19.4	22.4hrs	13.47	0.679	0.4429	19.4	22.4hrs
D-NeRF [29]	28.41	0.935	0.0582	10.3	18.8hrs	31.98	0.978	0.0251	10.3	18.4hrs	33.51	0.990	0.0137	9.8	17.9hrs
D-NeRF [29]-2 stage	24.34	0.885	0.1020	9.7	18.5hrs	28.79	0.973	0.0289	9.7	18.5hrs	31.40	0.985	0.0225	9.7	17.8hrs
D-NeRF [29]-dynamic	20.29	0.852	0.1360	10.0	22.0hrs	14.22	0.821	0.2720	10.0	21.6hrs	22.76	0.947	0.0873	9.5	21.7hrs
Nerfies [25]	30.34	0.986	0.0303	22.5	19.3hrs	27.07	0.973	0.0773	22.5	18.4hrs	38.26	0.998	0.0056	22.5	18.5hrs
Nerfies [25]-2 stage	29.27	0.984	0.0449	22.4	15.8hrs	30.05	0.991	0.0286	22.4	15.8hrs	35.81	0.997	0.0100	22.4	15.8hrs
Nerfies [25]-dynamic	21.15	0.911	0.1410	22.5	18.7hrs	19.84	0.859	0.1400	22.5	19.3hrs	23.71	0.864	0.1450	22.4	19.4hrs
HyperNeRF [26]	30.99	0.963	0.0360	22.5	21.3hrs	33.15	0.930	0.0511	22.5	19.8hrs	36.31	0.994	0.0080	22.5	20.8hrs
HyperNeRF [26]-2stage	27.28	0.933	0.0758	22.5	21.3hrs	28.28	0.955	0.0554	22.3	18.9hrs	31.63	0.983	0.0238	22.3	18.5hrs
HyperNeRF [26]-dynamic	14.41	0.774	0.2910	22.4	19.8hrs	14.88	0.835	0.2700	22.4	21.0hrs	22.73	0.946	0.0957	22.4	21.0hrs
NSFF [17]	25.44	0.912	0.0941	23.6	12.7hrs	25.27	0.935	0.0944	20.9	13.2hrs	28.71	0.967	0.0493	20.9	12.8hrs
NSFF [17]-dynamic	15.14	0.762	0.2732	12.5	14.8hrs	16.66	0.878	0.1769	15.7	16.0hrs	24.02	0.937	0.1069	15.7	16.0hrs
Ours (base)	17.83	0.799	0.2060	4.1	8mins	21.74	0.907	0.1040	4.5	8mins	25.91	0.950	0.0637	4.2	7mins
Ours w/ c2f	27.55	0.952	0.0342	4.1	7mins	32.04	0.983	0.0108	4.5	7mins	37.89	0.996	0.0055	4.2	6mins
Ours w/ c2f, tv	28.44	0.958	0.0301	4.1	8mins	32.78	0.985	0.0101	4.5	7mins	38.55	0.950	0.0048	4.2	6mins
Ours w/ c2f, tv, cycle	29.11	0.963	0.0254	4.1	8mins	34.12	0.988	0.0084	4.5	8mins	39.00	0.996	0.0044	4.2	7mins
Ours w/ c2f, tv, cycle, flow	29.25	0.964	0.0250	4.1	9mins	35.20	0.989	0.0074	4.5	9mins	38.39	0.996	0.0046	4.2	7mins

Table 5: Per-scene quantitative evaluation on inward-facing synthetic scenes (Glove and Kuka) against baselines and ablations of our method. We color code each cell as **best**, **second best**, and **third best**.

	GLOVE					KUKA				
	PSNR↑	SSIM↑	LPIPS↓	GPU(GB)↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	GPU(GB)↓	Time↓
Neural Volumes [19]	6.371	0.449	0.6101	19.4	22.4hrs	15.92	0.757	0.1645	19.4	22.4hrs
D-NeRF [29]	34.24	0.927	0.0455	9.8	18.5hrs	31.03	0.975	0.0349	9.8	18.4hrs
D-NeRF [29]-2 stage	30.86	0.922	0.0494	9.7	18.3hrs	26.05	0.959	0.0614	9.6	18.8hrs
D-NeRF [29]-dynamic	15.71	0.801	0.2660	10.3	22.1hrs	14.97	0.777	0.2679	9.5	22.4hrs
Nerfies [25]	36.37	0.993	0.0328	18.9	18.9hrs	33.40	0.996	0.0193	22.5	18.5hrs
Nerfies [25]-2 stage	34.96	0.991	0.0549	22.4	15.9hrs	31.79	0.994	0.0227	22.4	15.8hrs
Nerfies [25]-dynamic	15.67	0.636	0.1740	22.5	18.5hrs	16.86	0.698	0.2372	22.5	19.3hrs
HyperNeRF [26]	35.33	0.956	0.0471	22.5	20.0hrs	32.88	0.983	0.0255	22.5	20.8hrs
HyperNeRF [26]-2stage	31.47	0.935	0.0694	22.3	18.8hrs	27.12	0.959	0.0532	22.3	18.6hrs
HyperNeRF [26]-dynamic	22.04	0.850	0.1870	22.4	20.6hrs	15.97	0.856	0.2429	22.4	20.8hrs
NSFF [17]	27.66	0.902	0.1050	20.7	12.7hrs	28.24	0.962	0.0574	20.9	12.7hrs
NSFF [17]-dynamic	16.51	0.846	0.2091	15.7	16.0hrs	18.59	0.865	0.1986	15.7	14.8hrs
Ours (base)	25.14	0.867	0.1080	4.6	7mins	21.59	0.914	0.1050	4.5	8mins
Ours w/ c2f	30.69	0.959	0.0274	4.6	7mins	31.70	0.985	0.0147	4.5	7mins
Ours w/ c2f, tv	31.38	0.963	0.0234	4.6	8mins	32.48	0.987	0.0176	4.5	8mins
Ours w/ c2f, tv, cycle	33.67	0.970	0.0183	4.6	8mins	33.95	0.989	0.0147	4.5	8mins
Ours w/ c2f, tv, cycle, flow	34.67	0.973	0.0168	4.6	8mins	33.96	0.989	0.0147	4.5	9mins

our DeVRF achieves on-par high-fidelity dynamic novel view synthesis results on all scenes and synthesizes the cleanest depth maps compared to other approaches.

Significant quality enhancements of our DeVRF can be observed in the video examples for floating robot, kuka, flower-360°, plant, and rabbit. Notably, clear differences can be observed in the plant and rabbit scenes, where D-NeRF [29] and NSFF [17] generate intermittent motions. In contrast, the quadruple interpolation of the 4D voxel deformation field in our DeVRF allows us to synthesize smooth motions at novel time steps.

In addition, the quadruple interpolation of the 4D voxel deformation field in DeVRF allows us to conveniently and efficiently synthesize novel views at novel time steps, while existing approaches (i.e., Nerfies [25] and HyperNeRF [26]) cannot synthesize the views at novel time steps that have not been seen during model training [25, 26]. Thus, when rendering video examples, we generate results for Nerfies [25] and HyperNeRF [26] only on the training and testing time steps. This makes their videos' duration shorter than ours.

B.3 Additional Forward-facing Real-world Deformable Scene Evaluation.

We additionally collected another forward-facing real-world deformable scene where a human rotated and squeezed a toy pig in 540×960 pixels using 4 cameras, and we chose 3 views of them as training data and the other view as test data. As shown in Tab. 6, DeVRF achieves two orders of magnitude speedup compared to other approaches, and the second-best result in terms of PSNR and LPIPS metrics and the third-best result in terms of SSIM metric. Fig. 6 visualizes qualitative comparisons of DeVRF and baselines on this scene.

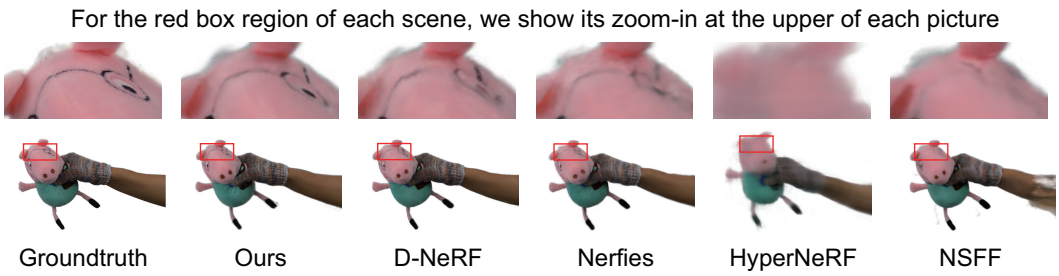

Figure 6: Qualitative comparisons of baselines and DeVRF on the additional real-world scene.

Table 6: Additional quantitative evaluation on a forward-facing real-world scene against baselines and ablations of our system. We color code each cell as **best**, **second best**, and **third best**.

	Pig Toy				
	PSNR↑	SSIM↑	LPIPS↓	GPU (GB)↓	Time↓
D-NeRF [29]	31.23	0.974	0.0381	12.4	22.1hrs
Nerfies [25]	30.66	0.986	0.0487	22.0	20.5hrs
HyperNeRF [26]	25.92	0.942	0.1037	22.0	22.3hrs
NSFF [17]	28.39	0.957	0.0654	20.5	14.5hrs
Ours (base)	22.84	0.942	0.0739	10.8	8mins
Ours w/ c2f	21.11	0.956	0.0595	10.8	8mins
Ours w/ c2f, tv	21.62	0.957	0.0578	10.8	8mins
Ours w/ c2f, tv, cycle	30.56	0.967	0.0447	10.8	9mins
Ours w/ c2f, tv, cycle, flow	30.85	0.967	0.0447	10.8	10mins

C Additional Ablations

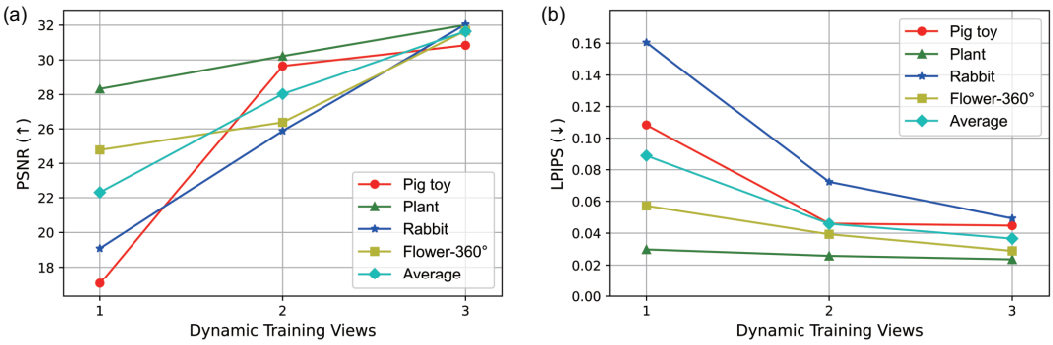

Figure 7: Ablation evaluation on the number of dynamic training views on real-world dataset: (a) PSNR, (b) LPIPS.

To further evaluate the influence of the number of dynamic training views, we conduct additional ablations for DeVRF on four real-world dynamic scenes and report the per-scene metrics as well as the average metrics with respect to the number of dynamic training views. As shown in Fig. 7, given the same multi-view static images, the performance of DeVRF largely improves with the increment of dynamic training views and achieves the best performance at three dynamic views. We additionally visualize the qualitative results of DeVRF with different numbers of dynamic training views in Fig. 8. Therefore, in our *static* $\rightarrow$ *dynamic* learning paradigm, with static multi-view data as a supplement, only a few dynamic views are required to significantly boost the performance of dynamic neural radiance fields reconstruction. This further demonstrates the effectiveness of our low-cost data capture process and the DeVRF model.

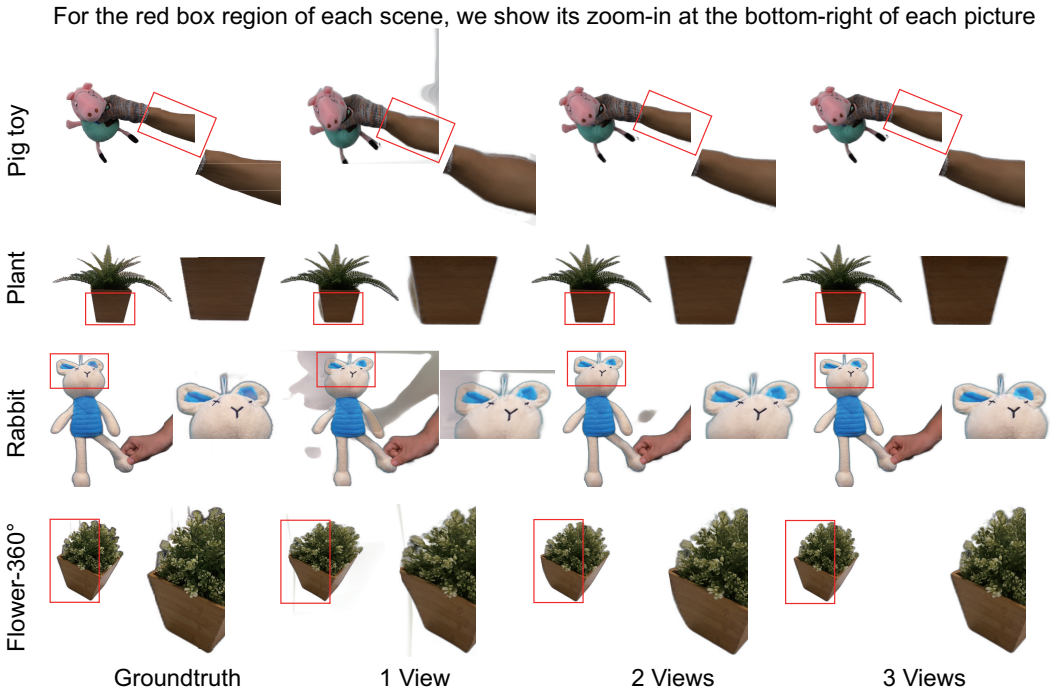

Figure 8: Qualitative results of DeVRF with different numbers of dynamic training views on real-world dataset.