

# BIDIRECTIONAL FLOW FIELDS FOR SPARSE INPUT NOVEL VIEW SYNTHESIS OF DYNAMIC SCENES

*Kapil Choudhary, Nagabhushan Somraj, and Rajiv Soundararajan*

Electrical Communication Engineering, Indian Institute of Science, Bengaluru, India.

## ABSTRACT

Novel view synthesis involves generating unseen perspectives of a scene based on videos captured from limited viewpoints. Learning scene representations of such videos containing dynamic scene elements introduces several challenges in modeling the motion. Existing models typically require dense viewpoint coverage to produce high-quality renderings, and their performance degrades significantly when the number of viewpoints is reduced. In this work, we explore the challenges associated with volumetric motion modeling for synthesizing novel views of dynamic scenes from only a few fixed viewpoints. Specifically, we address the limitations of unidirectional motion models, which often result in a many-to-one mapping of scene points. We enforce cyclic flow consistency with the help of bidirectional motion fields to achieve superior reconstruction of novel views of dynamic scenes. Moreover, the bi-directional motion field design allows us to track object motion in the synthesized views.

**Index Terms**— Dynamic novel view synthesis, motion modeling, dynamic radiance fields, volume rendering

## 1. INTRODUCTION

The task of generating novel views of a scene from images or videos captured from different viewpoints has a variety of applications, including virtual and augmented reality, sports streaming, and autonomous driving. This is particularly challenging for dynamic scenes which contain moving objects, even when the scene is captured using multiple cameras. Recent work on dynamic view synthesis addresses this problem by modeling the radiance fields using neural networks [1], low-rank approximations of 4D grids [2, 3] and 3D Gaussian splatting (3DGS) [4, 5, 6]. However, these models require the scene to be captured from a large number of camera viewpoints to train these models for the given scene. In this work, we focus on learning dynamic scene radiance fields using only a sparse set of input cameras.

One of the major challenges with existing grid-based models is the lack of explicit motion modeling. Although 3D Gaussian splatting methods allow for motion modeling,

they rely on rich initialization, which is hard to obtain in the few-shot setting. To constrain the radiance field model in the sparse set-up, RF-DeRF [7] builds on K-planes [2] and introduces explicit motion fields. Such fields are constrained using sparse and dense motion priors to obtain superior reconstructions. However, there are two main drawbacks to this method. Firstly, the motion field maps 3D points at a time instant  $t$  to points in a canonical radiance field space to model motion. However, such a model leads to a many-to-one mapping, where adjacent points with small variations in intensities can get mapped to the same point. This can cause a loss of fine-grained variations of the details. Secondly, unidirectional motion modeling to a canonical space does not allow the visualization of the learned flow across frames. The visualization of such flow is an important feature to enable motion editing-based applications such as [8].

To address these shortcomings, we introduce a pair of bidirectional motion fields: a forward motion field that maps 3D points at any time instant to the canonical volume/space and a backward motion field that maps points in the canonical volume to a particular 3D point given the time instant. We enforce a cyclic flow consistency loss that allows a moving point to return to itself when subjected to both these fields. Such learning of bidirectional motion fields not only helps improve the motion modeling and novel view synthesis, but also allows one to visualize how points move in 3D from one time instant to another. The backward motion field is only used during the training stage and only the forward motion field is used at inference time. We refer to our model as bidirectional flow enabled deformable radiance fields (BF-DeRF).

The main contributions of the paper are as follows:

- We introduce bidirectional motion fields to more effectively model the 3D motion across different time instances.
- We impose a cyclic flow consistency loss to train the pair of motion fields.
- We show superior performance of our model when compared to existing models on three popular dynamic scene novel view synthesis datasets.
- Our method enables the visualization of scene flow between any pair of frames in the novel views.

This work was supported in part by Kotak IISc AI/ML Center (KIAC).  
Project page: <https://kapil-4944.github.io/publications/bfderf.html>

## 2. RELATED WORK

Dynamic radiance fields can be broadly classified into two categories based on how the temporal modeling is handled. A simple approach is to model the dynamic radiance field as a 6D function of position, time, and viewing direction [9]. K-Planes [2] and HexPlane [3] extend TensorRF [10] to a 4D model that maps the position and time to a latent feature, which is then decoded by a tiny MLP. The lack of an explicit motion model in these approaches makes it incompatible to impose motion priors when learning with sparse input viewpoints.

The second set of models employs a motion or deformation field that maps the 3D points from a given time instant to a canonical space [1, 11, 12]. TiNeuVox [13] and SWAGS [14] replace the scene representation MLP in D-NeRF [1] with a low-rank tensor approximation, but use MLPs to model the motion field. Recently, 3D Gaussian splatting models have also been extended for dynamic scenes. CoGS [15], 4DGS [5], Ex4DGS [16] and STGS [6] model a canonical space with 3D Gaussians. A deformation field is learned to model the displacement of these Gaussians over time. While CoGS uses MLPs to model the displacement, 4DGS uses low-rank tensor approximations, and Ex4DGS and STGS use polynomials.

Most existing methods require a dense set of input viewpoints, and their performance drops sharply with sparse input viewpoints. RF-DeRF [7] employs motion modeling using factorized fields to explore the reliability of priors to constrain the models in case of sparse input views for dynamic scenes. However, the motion modeling is unidirectional, which causes poor learning of the canonical space. Further, it is very difficult to visualize the learned flow in existing models. We seek to address these challenges in our work.

## 3. METHOD

### 3.1. DeRF

We briefly review the deformable radiance field (DeRF) model [7] here before discussing our contributions. DeRF introduces motion modeling using factorized motion fields that maps any 4D point  $(x, y, z, t)$  in a scene at time  $t$  to a static canonical space. Further, the static canonical radiance field is learned to obtain color and occupancy. The deformation or flow field is represented by six spatio-temporal planes:  $\{\mathbf{S}_{xy}, \mathbf{S}_{yz}, \mathbf{S}_{xz}, \mathbf{S}_{xt}, \mathbf{S}_{yt}, \mathbf{S}_{zt}\}$ , defined at varying resolutions at multiple scales. These planes, combined with a small MLP, compute the flow of a 4D point to the canonical space. The Hadamard product of the features obtained from these multi-resolution planes is fed to a tiny MLP to obtain the scene flow. The static canonical radiance field is modeled by three spatial planes.

Given multiview video sequences of length  $T$  frames, we

sample pixels randomly from these frames. Our objective is to learn the pixel color in view  $\mathbf{v}$  at time  $t$  using the supervision of ground truth color. To render a pixel,  $N$  points are sampled along the ray that originates from the camera center and passes through the pixel at depths  $\{z_i\}_{i=1}^N$ , producing  $\{\mathbf{p}_i\}_{i=1}^N$ . For every 3D point  $\mathbf{p}_i$ , we first obtain the corresponding 3D point  $\mathbf{p}'_i$  at canonical space by computing the scene flow from time  $t$  to canonical space using forward flow field  $\mathcal{F}_f$  as

$$\mathbf{p}'_i = \mathcal{F}_f(\mathbf{p}_i, t) + \mathbf{p}_i. \quad (1)$$

We then query the static radiance field  $\mathbf{G}_s$  at  $\mathbf{p}'_i$  to obtain the volume density  $\sigma_i$  and a feature  $\mathbf{h}'_i$  corresponding to  $\mathbf{p}_i$  as

$$\sigma_i, \mathbf{h}'_i = \mathbf{G}_s(\mathbf{p}'_i). \quad (2)$$

A tiny MLP  $\mathbf{M}_s$  maps  $\mathbf{h}'_i$ , encoded viewing direction  $\mathbf{v}$  and encoded time  $t$  to the color  $\mathbf{c}_i$  of  $\mathbf{p}_i$  as

$$\mathbf{c}_i = \mathbf{M}_s(\mathbf{h}'_i, \gamma(\mathbf{v}), \gamma(t)), \quad (3)$$

where  $\gamma$  denotes the encoding of the viewing direction and the time. Note that  $\mathbf{M}_s$  is time and view-dependent, capturing time-varying color changes due to object motion. The color  $\mathbf{c}$  at a pixel is then obtained by volume rendering using  $\mathbf{c}_i$  obtained from  $\mathbf{M}_s$  as

$$\mathbf{c} = \sum_{i=1}^N w_i \mathbf{c}_i. \quad (4)$$

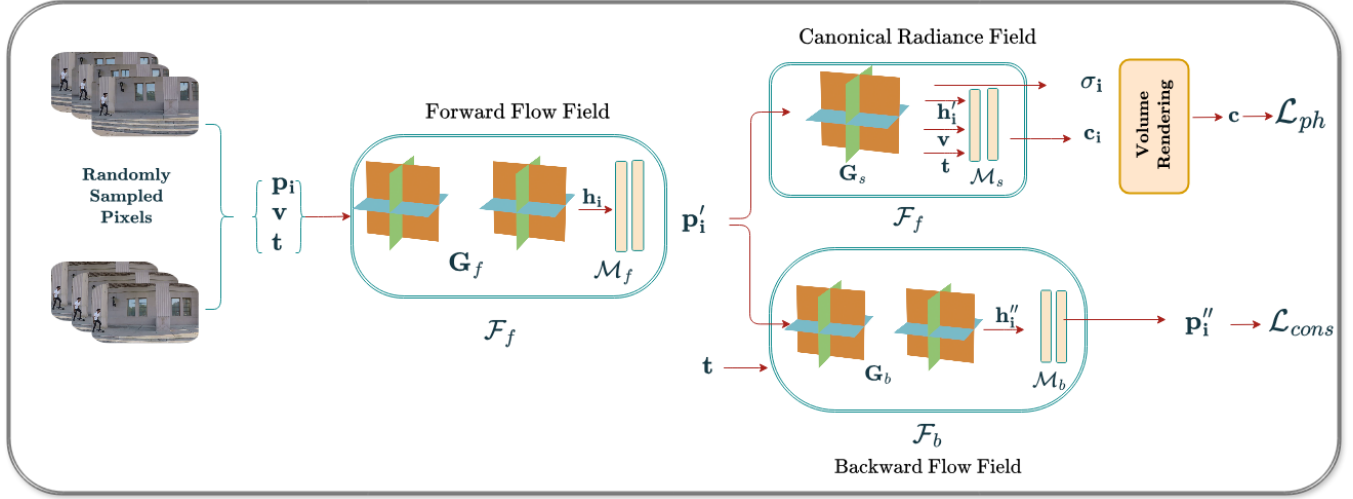
We use the occupancy to obtain the weight contributing to each point as

$$w_i = (\prod_{j=1}^{i-1} \exp(-\delta_j \sigma_j)) (1 - \exp(-\delta_i \sigma_i)). \quad (5)$$

We refer to the photometric loss that measures the squared error between the predicted color and the ground truth color at a pixel as  $\mathcal{L}_{ph}$ . While the DeRF model introduces a motion field, it is constrained using sparse and dense flow priors in RF-DeRF [7]. We refer to the sparse and dense flow prior losses introduced in RF-DeRF as  $\mathcal{L}_{sf}$  and  $\mathcal{L}_{df}$ .

### 3.2. BF-DeRF

Our main contribution is the introduction of bi-directional motion modeling of dynamic scenes through a backward flow field in addition to the forward flow field. The backward flow field is similar in architecture to the forward flow field and consists of six spatio-temporal planes similar to DeRF. While the forward flow field  $\mathcal{F}_f$  predicts the flow of 3D points at time  $t$  to the canonical space, the backward flow field  $\mathcal{F}_b$  maps canonical points to any desired time instance  $t''$ . The bi-directional flow of 3D points discourages many-to-one mapping of points in canonical space caused by the unidirectional



**Fig. 1. Model architecture:** We model the 4D dynamic radiance field as a combination of bi-directional 4D scene flow fields ( $\mathcal{F}_f, \mathcal{F}_b$ ) and 3D static canonical radiance field  $\mathcal{F}_s$ . Here,  $\mathcal{F}_f$  maps any point  $(\mathbf{p}_i, t)$  to the canonical space.  $\mathcal{F}_b$  maps canonical points to any desired time  $t''$  allowing for scene flow visualization. Both scene flow fields have the same structure consisting of six spatio-temporal planes and a small MLP  $\mathcal{M}$ . Forward-backward flow fields enable the cyclic motion of 4D points. Static canonical radiance field  $\mathcal{F}_s$  models the color and occupancy of the canonical space.

scene flow of RF-DeRF. The backward flow field maps the canonical points  $\mathbf{p}'_i$  to points  $\mathbf{p}''_i$  at any time  $t''$ . We obtain the flow of 3D points  $\{\mathbf{p}'_i\}_{i=1}^N$  from canonical space to time  $t''$  using  $\mathcal{F}_b$  as

$$\mathbf{p}'' = \mathcal{F}_b(\mathbf{p}', t'') + \mathbf{p}'. \quad (6)$$

Employing bi-directional flow fields requires constraining both the forward and backward flow fields ( $\mathcal{F}_f, \mathcal{F}_b$ ). Note that the static radiance field  $\mathcal{F}_s$  is also trained along with the flow fields. To constrain the bi-directional fields, the canonical point  $\mathbf{p}'$  is mapped to  $\mathbf{p}''$  at time  $t'' = t$ . We regularize the backward flow field  $\mathcal{F}_b$  by employing a cyclic flow consistency loss  $\mathcal{L}_{\text{cons}}$  defined as

$$\mathcal{L}_{\text{cons}} = \sum_{i=0}^N w_i \|(\mathbf{p}_i - \mathbf{p}''_i)\|^2, \quad (7)$$

where  $w_i$  are computed according to Eq. (5). Our cyclic flow consistency loss  $\mathcal{L}_{\text{cons}}$  ensures each canonical point to the initial 3D point. Note that we do not take the expected 3D point loss instead, our loss encourages each point  $\mathbf{p}''$  at time  $t'' = t$  to map again to the initial starting point  $\mathbf{p}_i$  at time  $t$ .

We train the two flow fields and the static field in our model with  $\mathcal{L}_{\text{ph}}$  and  $\mathcal{L}_{\text{cons}}$  and the flow losses. The overall loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{ph}} + \lambda_{\text{sf}} \mathcal{L}_{\text{sf}} + \lambda_{\text{df}} \mathcal{L}_{\text{df}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}}, \quad (8)$$

where  $\lambda_{\text{sf}}$ ,  $\lambda_{\text{df}}$  and  $\lambda_{\text{cons}}$  are hyperparameters. Note that the flow losses only constrain the forward field, while the cyclic

flow consistency loss constrains both the forward and backward flow fields. Further, we do not employ stop-gradient on either flow field in the consistency loss  $\mathcal{L}_{\text{cons}}$ . Thus, we do not treat either flow field as superior to another.

We note that the backward flow field is mainly used to constrain the learning of the forward flow field. This prevents the many-to-one mapping, leading to improved learning of the motion field. Further, the backward field is not used during the inference time. Once both the motion fields and the static radiance field are learned during training, at inference time, only the forward flow field and the static radiance field are used to render a pixel from a given choice of view.

### 3.3. Visualizing Scene Flow

Visualizing object motion in existing volumetric dynamic scene models [2, 3] is challenging because these methods often lack explicit motion modeling. Thus, the motion of objects over time is not readily available for visualization in new perspectives. One of the benefits of our bidirectional flow field model is that it allows us to visualize the 3D motion learned across time and across cameras.

To visualize the motion of a pixel  $(x, y)$  at time  $t$  at another time instance  $t''$ , we use BF-DeRF in a two-step mapping process as follows. First, the 3D points  $\mathbf{p}_i$ 's along the ray corresponding to the pixel  $(x, y)$  are mapped into the canonical space using the forward flow field  $\mathcal{F}_f$ . From the canonical space, the respective 3D points are subsequently mapped to the desired time instance  $t''$  using backward flow field  $\mathcal{F}_b$ . This transformation yields the 3D points  $\mathbf{p}''$ 's



**Fig. 2. Flow Visualization :** During the transition between frames 75 to 125 and 125 to 230, the pixels on the spatula, moving hand, and body accurately follow the motion. The corresponding pixel locations are marked by colored circles.

at time  $t''$ , which correspond to the original points  $\mathbf{p}_i$ 's in canonical space. To obtain the final visualization, the points  $\mathbf{p}''$  are alpha-composited to determine the expected 3D point, which is then projected onto the image plane. The resulting projected point represents the pixel location corresponding of  $(x, y, t)$  at time  $t''$ . We show an example of such flow visualization in Fig. 2.

## 4. EXPERIMENTS AND RESULTS

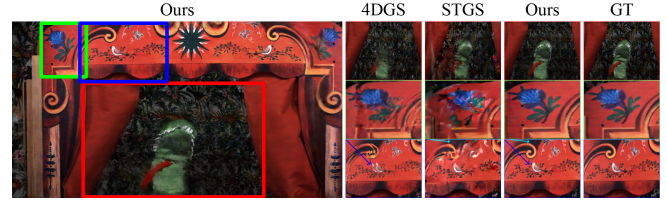
### 4.1. Datasets and Implementation Details

We evaluate our model on three popular multi-view dynamic scene datasets, N3DV [17], InterDigital [18] and Nvidia [19] with two and three input views. Similar to prior work [2, 7], all the videos are spatially downsampled by factor of two. The **N3DV dataset** contains 6 scenes with 17 to 21 multiview cameras arranged in 2 rows with a spatial resolution of  $1352 \times 1014$ . Each video sequence has 300 frames. The **InterDigital dataset** contains multi-view videos from a 16 camera rig. We select 5 scenes as suggested in [7] with a spatial resolution of  $1024 \times 544$  having 300 frames per video. The **Nvidia dataset** contains a multi-view camera setup of 12 cameras. We choose longer-duration videos which involve more than 90 frames. The resolution of the videos is  $960 \times 540$ . All the datasets have frame rate of 30fps. The test camera in all datasets is selected as the center of the arranged cameras. For training, we select cameras diagonally with respect to the test camera and add nearby cameras as we increase training views.

We compare the performance of our model against RF-DeRF [7], 4DGS [5] and STGS [6]. We use the official code release of RF-DeRF and modify it to implement our model. For the BF-DeRF model, we set  $\lambda_{\text{cons}} = 0.01$ . We evaluate all the models using PSNR, SSIM and LPIPS.

### 4.2. Results

We show the quantitative performance of our model in Tab. 1 and Tab. 2, where our BF-DeRF model remains consistently good across all datasets. On the other hand, dynamic 3D Gaussian splatting methods, such as 4DGS [5] and STGS [6] fail to generalize across datasets. We observe that 4DGS performs much worse on the InterDigital dataset as the dataset



**Fig. 3. Qualitative example on InterDigital dataset with 3 input views:** We see that the details are clearly reconstructed by BF-DeRF and better than the other models in all the three highlighted regions.



**Fig. 4. Qualitative example on Nvidia dataset with 2 views:** we observe that 4DGS renders a headless moving person while STGS is suffers from loss of details and color distortions. BF-DeRF does not suffer from any of these artifacts.

involves complex motion with objects entering and leaving the scene. It is unable to learn such motions, whereas our model performs best on this dataset. In the Nvidia dataset, we observe that the BF-DeRF model is very close to the 4DGS model, and the STGS model falls significantly short. The poor performance of STGS here could be due to a very high dependency on hyperparameters. Further, our model performs consistently well for both 2 and 3 input cases.

From Fig. 3 and Fig. 4, we observe that BF-DeRF better reconstructs finer details throughout the frame as opposed to just a few regions for 4DGS and STGS. 4DGS performs much worse in moving regions where objects enter or leave the scene or have complex motions. STGS performs better than 4DGS in InterDigital due to the time-varying opacity and color, but it fails drastically on the Nvidia dataset for the same reason. In particular, having the ability to change the color and opacity of the Gaussians over time increases the model capacity, which leads to the model being sensitive to hyperparameters. In particular, we observe that STGS is highly dependent on hyper-parameters, which lead to color degradations in the rendered frames as shown in Fig. 4.

Model (2-views)	N3DV			InterDigital			Nvidia		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
4DGS	21.46	0.78	0.29	15.9	0.42	0.46	<b>19.0</b>	<b>0.58</b>	<b>0.36</b>
STGS	19.9	<b>0.79</b>	0.28	15.5	0.57	0.41	8.5	0.37	0.69
RF-DeRF	20.3	0.71	0.33	<u>19.1</u>	<u>0.66</u>	<u>0.31</u>	17.4	0.45	0.51
BF-DeRF	<b>21.9</b>	<b>0.79</b>	<b>0.26</b>	<b>19.2</b>	<b>0.70</b>	<b>0.28</b>	<b>18.6</b>	<b>0.57</b>	<b>0.39</b>

**Table 1. Quantitative Results:** we compare our mode on 3 datasets N3DV, InterDigital and Nvidia with 4DGS , STGS and RF-DeRF methods we report PSNR, SSIM, and LPIPS scores for the rendered images for 2 input training views.

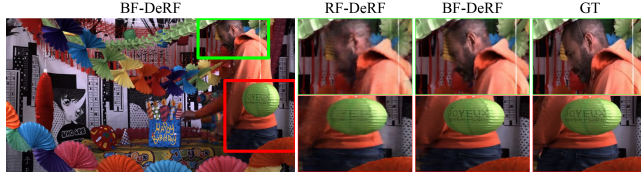


Model (3-views)	N3DV			InterDigital			Nvidia		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
4DGS	26.5	0.90	0.17	19.57	0.60	0.36	20.0	0.66	0.33
STGS	24.4	0.86	0.2	21.3	0.82	0.20	15.8	0.60	0.42
RF-DeRF	26.1	0.90	0.16	22.8	0.80	0.21	19.2	0.60	0.42
BF-DeRF	27.0	0.92	0.12	23.1	0.83	0.19	19.6	0.65	0.36

**Table 2.** Quantitative Results: we compare our mode on 3 datasets N3DV, InterDigital and Nvidia with 4DGS, STGS and RF-DeRF methods we report PSNR, SSIM, and LPIPS scores for the rendered images for 3 input training views.

### 4.3. Need for Bidirectional Flow

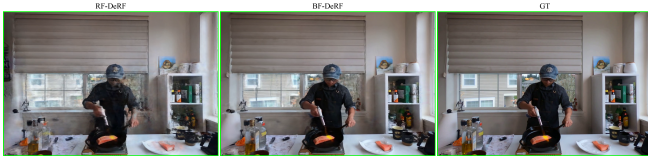
From Fig. 5 and Fig. 6, we conclude that RF-DeRF struggles to capture fine details, while BF-DeRF successfully reconstructs both the structure and fine details of the scene. RF-DeRF, with its uni-directional flow, results in visible object distortions and loss of finer details. The incorporation of bi-directional flow in motion modeling helps reduce object deformation and minimize discontinuities in objects. where the 3D scene points are uniquely mapped to the canonical space, leading to more accurate and consistent reconstructions.



**Fig. 5. Qualitative comparison between RF-DeRF and BF-DeRF on InterDigital dataset:** BF-DeRF performs better on the person’s face, as shown in the top row. RF-DeRF struggles to render the writing on the hanging balloon.

### 4.4. Poor initialization in dynamic Gaussian splatting

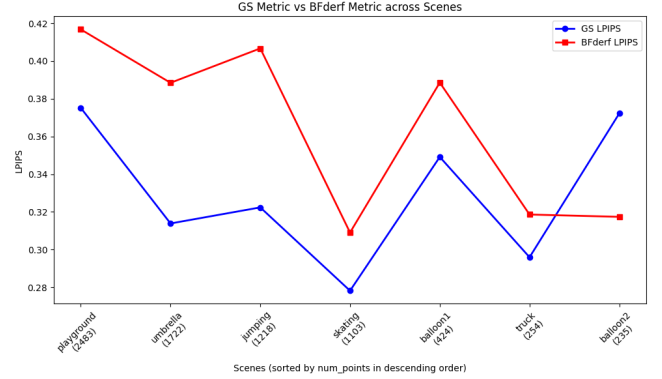
We study the impact of initialization on 4DGS to explore scenarios where our BF-DeRF model can perform better than 4DGS on the Nvidia dataset. We observe that for a small number of initialization points, BF-DeRF performs better than 4DGS, as shown in Fig. 8 for 3 views. The superior performance of 4DGS on the Nvidia dataset may be attributed



**Fig. 6. Qualitative comparison on N3DV between RF-DeRF and BF-DeRF:** BF-DeRF renders features such as the cap, the person’s face, hands, the torch, and in the background in the bottom-left region of the image with bottles and the window better compared to RF-DeRF.



**Fig. 7.** Qualitative comparison of poor initialization in 4DGS with BF-DeRF : Better reconstructed regions in BF-DeRF are marked by the arrows in the scene.



**Fig. 8. Effect of initialization on 4D Gaussian splatting:** For 3-views on the Nvidia dataset, overall 4DGS does slight better than BF-DeRF. But as the number of initialization points decreases, performance of 4DGS decreases, whereas BF-DeRF performs better.

to the rich initialization from COLMAP while our BF-DeRF model, which is volumetric, is not constrained by such limitations. The problem of obtaining rich initializations becomes even more important in the sparse set-up where there are very few input views. We also show visual examples to elucidate this observation. From Fig. 7, we observe that 4DGS struggles to render the details in the object marked with arrows. 4DGS renders a smooth balloon, whereas our model is able to reconstruct the finer details of the balloon. Please refer to the project page to view the video comparisons.

## 5. CONCLUSION

Our introduction of bi-directional flow in motion modeling reduces the ambiguities introduced by unidirectional motion modeling and uniquely mapping every point in canonical space. The forward-backward motion fields also help visualize the 3D scene flow. While the backward flow field slightly increases the train time when compared to RF-DeRF, there are gains in rendering quality. Although 3D Gaussian splatting methods may be superior occasionally, they suffer from poor initialization in sparse input view cases. However, volumetric models can learn to populate occupancy around most of the objects in the radiance field. Exploring a combined approach of splatting and volumetric modeling of a scene could be an interesting future direction.

## 6. REFERENCES

- [1] 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 (CVPR)*, 2021.
- [2] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa, “K-Planes: Explicit radiance fields in space, time, and appearance,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [3] Ang Cao and Justin Johnson, “HexPlane: A fast representation for dynamic scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, 2023.
- [5] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang, “4d gaussian splatting for real-time dynamic scene rendering,” 2024.
- [6] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu, “Spacetime gaussian feature splatting for real-time dynamic view synthesis,” 2024.
- [7] Nagabhushan Somraj, Kapil Choudhary, Sai Harsha Mupparaju, and Rajiv Soundararajan, “Factorized motion fields for fast sparse input dynamic view synthesis,” in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers ’24*, July 2024, SIGGRAPH ’24, p. 1–12, ACM.
- [8] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan, “Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis,” in *2024 International Conference on 3D Vision (3DV)*, Mar. 2024, p. 800–809, IEEE.
- [9] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang, “Dynamic view synthesis from dynamic monocular video,” *arXiv e-prints*, p. arXiv:2105.06468, May 2021.
- [10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su, “TensorRF: Tensorial radiance fields,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
- [11] Chaoyang Wang, Lachlan Ewen MacDonald, László A. Jeni, and Simon Lucey, “Flow supervision for deformable nerf,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [12] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely, “Tracking everything everywhere all at once,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [13] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian, “Fast dynamic radiance fields with time-aware neural voxels,” in *Proceedings of the SIGGRAPH Asia 2022 Conference Papers*, 2022.
- [14] Richard Shaw, Jifei Song, Arthur Moreau, Michal Nazarczuk, Sibi Catley-Chandar, Helisa Dharmo, and Eduardo Perez-Pellitero, “Swags: Sampling windows adaptively for dynamic 3d gaussian splatting,” *arXiv e-prints*, p. arXiv:2312.13308, 2023.
- [15] Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni, “Cogs: Controllable gaussian splatting,” *arXiv e-prints*, p. arXiv:2312.05664, 2023.
- [16] Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon, “Fully explicit dynamic gaussian splatting,” in *Proceedings of the Neural Information Processing Systems*, 2024.
- [17] Tianye Li, Mira Slavcheva, Michael Zollhöfer, 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,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [18] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, and Valerie Allie, “Dataset and pipeline for multi-view light-field video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2017.
- [19] 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,” 2020.