DynMF: Neural Motion Factorization for Real-time Dynamic View Synthesis with 3D Gaussian Splatting

Agelos Kratimenos [1], Jiahui Lei [1], and Kostas Daniilidis [1,2]

[1] University of Pennsylvania, Philadelphia, PA, USA
[2] Archimedes, Athena RC

Abstract. Accurately and efficiently modeling dynamic scenes and motions is considered so challenging a task due to temporal dynamics and motion complexity. To address these challenges, we propose DynMF, a compact and efficient representation that decomposes a dynamic scene into a few neural trajectories. We argue that the per-point motions of a dynamic scene can be decomposed into a small set of explicit or learned trajectories. Our carefully designed neural framework consisting of a tiny set of learned basis queried only in time allows for rendering speed similar to 3D Gaussian Splatting, surpassing 120 FPS, while at the same time, requiring only double the storage compared to static scenes. Our neural representation adequately constrains the inherently underconstrained motion field of a dynamic scene leading to effective and fast optimization. This is done by biding each point to motion coefficients that enforce the per-point sharing of basis trajectories. By carefully applying a sparsity loss to the motion coefficients, we are able to disentangle the motions that comprise the scene, independently control them, and generate novel motion combinations that have never been seen before. We can reach state-of-the-art render quality within just 5 minutes of training and in less than half an hour, we can synthesize novel views of dynamic scenes with superior photorealistic quality. Our representation is interpretable, efficient, and expressive enough to offer real-time view synthesis of complex dynamic scene motions, in monocular and multi-view scenarios.

Keywords: gaussian splatting · dynamic rendering · motion decomposition

1 Introduction

The generation of a photorealistic representation, from an unseen point of view, also known as Novel View Synthesis, poses a long-standing challenge to the graphics and computer vision community. While there exist early methods that tackle this problem, both for static [22, 38] and dynamic scenes [10, 23], neural rendering has received widespread attention only after the pioneering work of NeRF [29]. Following the latter, numerous methods have contributed to the static neural rendering problem, by tackling aliasing problems [4, 5], generalizing

to larger scales [6, 42, 56], and to new scenes [48] and increasing efficiency and rendering speed [8, 9, 16, 30, 40].

Fig. 1: Our framework can efficiently model all motions in a 3D dynamic scene (Figure 1a) through its neural motion decomposition scheme. We can synthesize photorealistic novel views, both for synthetic and real scenes in real-time, surpassing 120FPS for 1K resolution (Figure 1b). Our simple representation allows for novel motion synthesis by disentangling motions in the scene (Figure 1c). Notice how only the left hand or the green ball are moving in the 'Jumpingjacks' and 'Bouncingballs' D-NeRF scenes respectively.

While the advance in novel view synthesis of static scenes has been rapid, the high-quality efficient photorealistic rendering of dynamic scenes still poses a vivid challenge to the aforementioned communities. The numerous applications, including but not limited to augmented reality/virtual reality (AR/VR), robotics, self-driving, content creation, and controllable 3D primitives for use in video games or the meta-verse, have kept the interest in expressive dynamic novel view synthesis to high levels. From recent pioneers of neural dynamic rendering [24, 34] to the latest [28], numerous efforts have been made to drastically improve this field and bring it one step closer to real applications. Altering the scene representation from MLPs [24, 34] to grids [46], or other decompositions [3, 7, 15, 36] for efficiency and faster training, proposing expressive deformation representations [1, 31, 32, 39, 52] or using template guided methods [50, 59] are just a few of the approaches taken the last few years.

Despite all these approaches, dynamic novel view synthesis is still a challenging problem, with quick training, real-time rendering, and highly accurate and visually appealing novel views yet to be achieved. Dynamic scenes require disambiguating displacements, and disentangling overlapping motions, especially due to non-rigid objects and monocular captured scenes that make this even more challenging. We propose a compact, simple, and expressive neural representation that can adequately tackle the previous challenges. Namely, we propose a sparse trajectory decomposition of all the motions in the dynamic scene, to accurately represent displacements, overlapping and non-rigid motions. Our framework is strategically designed to offer extremely fast training and real-time rendering of

a dynamic scene. Specifically, we argue that a scene consists of only a few basis trajectories (even less than 10) and every point/motion shall be adequately mapped to one of them or a linear combination of them. The basis motions are learned through a tiny MLP queried only in time leading to superfast convergence during training (less than 5 minutes for photorealistic results, and half an hour for superior to the state-of-the-art) and superfast rendering during inference (higher than 120 FPS for resolutions of 1K). By strategically constraining an inherently underconstrained problem in this way, we are able to successfully optimize for cases with very few correspondences (e.g. monocular videos). Simultaneously, being able to efficiently factorize all the motions of a dynamic scene into a few basis trajectories, allows us in turn, to control them, enable or disable them, opening a road to applications like video editing, scene controllability, interactive and real-time motion control, efficient animation and so on.

In this work, we present **DynMF** (**DYN**amic neural **M**otion **F**actorization). The contributions of this paper can be summarized as follows:

- We manage to expressively model scene element deformation of complex dynamics, through a simple and interpretable framework. Our method enables robust per-point tracking, overcoming displacement ambiguities, overlappings, and non-rigid complex motions. It also enables the synthesis of novel motion scenes, that have never been seen before through disentangling motion and carefully enforcing sparsity.
- DynMFis extremely efficient. The carefully designed time-only queried MLP allows for training in less than 30 minutes while the rendering speed remains comparable to 3D Gaussian Splatting [20]. This allows for real-time rendering of static and dynamic scenes.
- We propose a neural representation that can effectively constrain the ill-posed nature of dynamic 3D motion reconstruction and requires no prior knowledge of the scene. Our method achieves high-quality real-time dynamic rendering, while significantly surpassing the state-of-the-art benchmarks on the D-NeRF dataset and achieving comparable results on the real-scene datasets.

2 Related Work

Dynamic Neural Rendering: The recent works on NeRF [29] and D-NeRF [34] both proposing a coordinate-based neural network, the former for querying points on a ray to produce color and density for static scenes and the latter to model the deformation field and create an expressive mapping between a canonical space and a dynamic scene, have provoked an explosion of interest in the field of static and dynamic novel view synthesis. On the dynamic side, we can distinguish three main approaches: (i) methods that construct a deformation field to warp a canonical space to every timestep [31, 32, 34, 47, 52] (ii) methods that try to model the dynamic scene as a whole, implicitly or explicitly [3, 7, 15, 24, 46] and (iii) point-based methods [1, 28, 57] that due to dense correspondence over time

between frames and due to their Lagrangian representation, can achieve consistent and expressive dynamic scene modeling. We relax that latter restriction by proposing a point-based method that can construct an efficient and expressive deformation field, explicit or learned, without the need for dense correspondences through multiple frames or multiple views.

Motion Factorization: Decomposing a complex motion field into a linear combination of basis trajectories has been a well-established approach from Mathematics and Physics to Signal Processing and Computer Vision. In the latter field, [2] has proposed an explicit trajectory-based factorization for non-rigid structure for motion in which every point in space can be linearly expressed by a combination of DCT trajectories. Clustering trajectories into a small number of subspaces has been argued by [21] for the same task, while [60] has explored trajectories' convolutional structure. Following [2, 44], [45] has also exploited DCT trajectory basis for the task of dynamic novel view synthesis, by predicting a per-point trajectory using an MLP to acquire the DCT coefficients. Our method argues that a per-point trajectory is redundant and can only work given a lot of correspondences or strong regularization. In fact, [45] leverages optical flow in their optimization scheme.

Dynamic 3D Gaussians and Concurrent work: During the past few months, numerous works have emerged to utilize 3D Gaussian Splatting [20] for Dynamic Neural Rendering. First, Dynamic 3D Gaussians [28] uses online training and rigidity losses to synthesize novel views timestamp per timestamp. For their method to work, though, they require multiple cameras and depth information to reconstruct the first frame, while at the same time, their method cannot handle the appearance of new objects in the scene. DynMF works on scenes captured by a single monocular camera and with random initialization incorporating information from all frames simultaneously. [54] uses a coordinate-based MLP similarly to D-NeRF [34] as a deformation field. This approach may be slow and inefficient because it requires querying MLP for each Gaussian. Our method is extremely time-efficient and does not significantly depend on the number of Gaussians. [53] is optimizing 4D primitives to treat the spacetime as an entirety, while [51] is connecting adjacent Gaussians via a HexPlane to model position and shape deformations. Our neural representation is simple and interpretable by decomposing all the complex motions of a dynamic scene into very few neural trajectories. Other concurrent works include [19] that uses Fourier approximation and flow estimation, [55] which decomposes the scene's objects by designing a control-masking pipeline, while our method inherently enables the disentangling motion and controllability of the scene, and [18] which uses local points that act as local bases to edit the deformable Gaussians. Other notable mentions are [11–13, 25–27, 35, 37, 41].

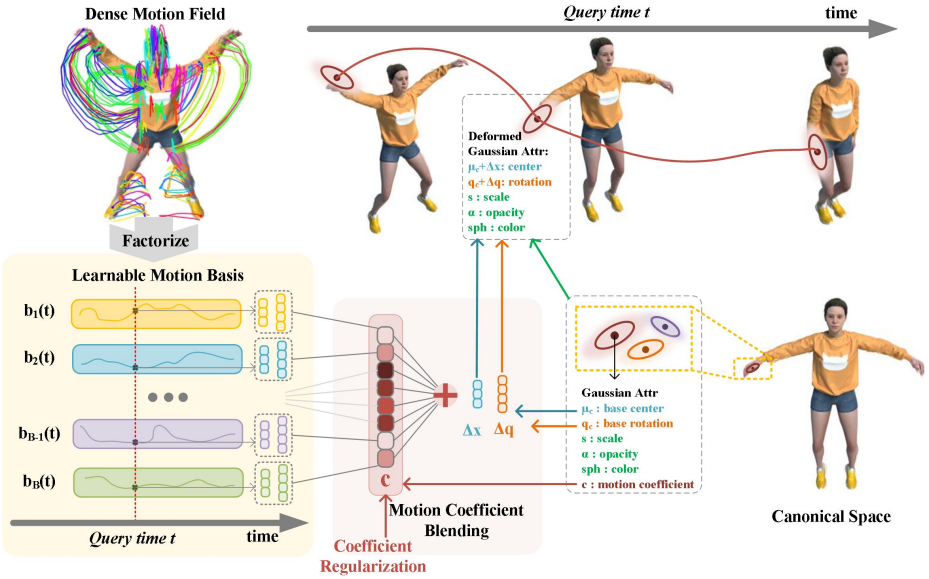

Fig. 2: Overview of DynMF. The underlying dense motion field (left-top) of a dynamic scene is factorized into a set of globally shared learnable motion basis (left-bottom) and their motion coefficients stored on each Gaussian (right-bottom). Given a query time t, the deformation can be efficiently computed via a single global forward of the motion basis and the motion coefficient blending (middle-bottom) to recover the deformed scene (top-right).

3 Method

We propose a compact representation that is shared across all points and models the dynamics of a scene. We argue that every scene is comprised of a limited fixed number of trajectories. In other words, we can model a 3D dynamic scene by a few universal trajectories that are shared between the 3D world points. In this section, we first formally present the representation of our motion field, DynMF, which is able to model the 3D dynamic scene and synthesize novel views (Section 3.1). In Section 3.2 we review the framework in which we bundle our motion field representation to efficiently learn the 3D world, namely 3D Gaussian Splatting, while in Section 3.3 we present the optimization and regularization details.

3.1 Motion Representation Fields

Let a dynamic scene be represented by a point cloud of N points $\boldsymbol{x} \in \mathbb{R}^{N\times 3}$. We assume a sequence of frames in the interval $t \in [0, T]$, taken from one or multiple viewpoints. Assuming a canonical space $\boldsymbol{x_c} \in \mathbb{R}^{N\times 3}$ we define B trajectories $\boldsymbol{b}_j(t) : [0, T] \to \mathbb{R}^3$ such that for every $\boldsymbol{x} \in \mathbb{R}^{1\times 3}$ we have:

$$\boldsymbol{x}(t) = \boldsymbol{x_c} + \sum_{j=1}^{B} c_j \boldsymbol{b}_j(t), \tag{1}$$

where $c_j \in \mathbb{R}^{N\times 1}$ are the motion coefficients for each point, and $\boldsymbol{x}(t)$ are the points' motion trajectories across the entire sequence. We let the basis be shared across all points arguing that the dynamic motions of a scene can be explicitly represented by a linear combination of B trajectories. This factorization problem becomes an overconstrained low-rank factorization as soon as the number of points N exceeds the number of basis trajectories B. This is easy to see if we discretize the time into T timestamps and write only the x-coordinates as matrices $\boldsymbol{X}, \boldsymbol{X_c} \in \mathbb{R}^{N\times T}$

$$\boldsymbol{X} = \boldsymbol{X}_c + \boldsymbol{C} \cdot \boldsymbol{B}, \tag{2}$$

with $\boldsymbol{C} \in \mathbb{R}^{N\times B}$ and $\boldsymbol{B} \in \mathbb{R}^{B\times T}$ and B much lower than N or T.

We reason that given enough correspondences and expressive basis functions, we can jointly learn the canonical space and each point's motion by reconstructing its trajectory across the whole sequence. Sharing the basis across all points enables the representation to produce physically plausible motions that stay consistent across all frames.

If we let

$$\boldsymbol{b}_j(t) = \begin{cases} cos\left(\frac{2\pi j}{T}t\right), \text{ if j is odd} \\ sin\left(\frac{2\pi j}{T}t\right), \text{ if j is even} \end{cases} \tag{3}$$

the factorization is equivalent to modeling the dynamic motion trajectories of a scene with Fourier series. While Fourier bases are global approximators for any trajectories and have been shown to be a compact representation for dynamic scenes with deformations [49,58], we argue that a learned basis will be smoother and more expressive for this task (see Section 4.5 for ablations). Thus, we let:

$$\mathbf{b}_j(t) = \mathbf{MLP}\left(\frac{t}{T}; w_j\right), \tag{4}$$

be a small multilayer perceptron queried only over time. While the learned functions are not strictly a basis, we empirically observe that they span a large enough space that covers the variation of deformations in the scene. Overall, we propose a representation of the dynamic scene in which a small MLP predicts along all timestamps a few compact basis trajectories, and all the points in the scene are bound to linearly choose between these to model their motion across time.

3.2 3D Gaussian Splatting

We choose to bind the aforementioned representation with 3D Gaussian Splatting [20]. The latter is an explicit 3D scene representation used for efficient and expressive neural rendering.

Static 3D Gaussians: The 3D space is described in the form of a point cloud by a set of 3D Gaussians which are defined by a mean value $\boldsymbol{\mu}$ and a 3D covariance matrix $\boldsymbol{\Sigma}$ in the world space. The influence of one Gaussian with parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ on a given spatial position $\boldsymbol{x} \in \mathbb{R}^3$ is defined by:

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \tag{5}$$

The anisotropic full 3D covariance matrix is decomposed into a rotation matrix $\boldsymbol{R}$ and a scaling matrix $\boldsymbol{S}$:

$$\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{R}^T. \tag{6}$$

To render the 3D Gaussians in a differentiable manner [20] Gaussians are splatted on the image plane by approximating the 2D means and covariances from the 3D ones. The center of the Gaussian is splatted using the standard point rendering formula:

$$\boldsymbol{\mu}^{2D} = \boldsymbol{K}\frac{\boldsymbol{E}\boldsymbol{\mu}}{\boldsymbol{E}\boldsymbol{\mu}_z}, \tag{7}$$

where K, E are the intrinsic projection matrix and the world-to-camera extrinsic matrix, respectively. The 3D covariance matrix is splatted into 2D using [61]:

$$\boldsymbol{\Sigma}^{2D} = \boldsymbol{J}\boldsymbol{E}\boldsymbol{\Sigma}\boldsymbol{E}^T\boldsymbol{J}^T, \tag{8}$$

where $\boldsymbol{J}$ is the Jacobian of the point projection formula in 7. Finally, the color of a specific pixel p can be computed as:

$$C_p = \sum_{i=1}^{N} c_i\alpha_i f\left(p|\boldsymbol{\mu}^{2D}, \boldsymbol{\Sigma}^{2D}\right)\prod_{j=1}^{i-1}\left(1-\alpha_j f\left(p|\boldsymbol{\mu}^{2D}, \boldsymbol{\Sigma}^{2D}\right)\right), \tag{9}$$

where c_i, α_i are the color and opacity, bound with each 3D Gaussian. Overall, each Gaussian is characterized by the following attributes: i. a center mean $\boldsymbol{\mu} \in \mathbb{R}^3$, ii. a scale factor $\boldsymbol{s} \in \mathbb{R}^3$ and a rotation $\boldsymbol{R} \in SO(3)$ that we parametrize with a unit quaternion in $\mathbb{S}^3$, iii. an opacity logit $\alpha \in \mathbb{R}$ and iv. each color defined by spherical harmonics coefficients $\boldsymbol{c}_{R,G,B} \in \mathbb{R}^{(L+1)^2}$ where L is the maximum degree of spherical harmonics.

Dynamic 3D Gaussians: The previous framework is capable of modeling efficiently and expressively a 3D static scene. To use it for representing a 3D dynamic scene we propose the following:

1. Substituting the mean $\boldsymbol{\mu}$ with $\boldsymbol{\mu}(t)$. The mean's trajectory $\boldsymbol{\mu}(t)$ will be decomposed according to Equations 1 and 2:

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu}_c + \sum_{j=1}^{B} c_j \boldsymbol{b}_j^{\mu}(t), \tag{10}$$

   where $\boldsymbol{\mu}_c \in \mathbb{R}^3$ is the mean center of the Gaussian in the canonical space and $\boldsymbol{c} = [c_j]_{j=1,\ldots,B}$ the motion coefficients, both stored as learned parameters per Gaussian.
2. Substituting the rotation quaternion $\boldsymbol{q}$ with $\boldsymbol{q}(t)$. The rotation quaternion will be decomposed similarly:

$$\boldsymbol{q}(t) = \boldsymbol{q}_c + \sum_{j=1}^{B} c_j \boldsymbol{b}_j^{R}(t), \tag{11}$$

where $\boldsymbol{q}_c \in \mathbb{R}^4$ is the rotation quaternion of the Gaussian in the canonical space stored as learned parameter per Gaussian. Note that the motion coefficients are shared for both the translational and rotational decompositions.
3. Keeping density and spherical harmonic coefficients of each Gaussian fixed along time. The basis functions $\{\boldsymbol{b}_j^{\mu}(t)\}$ and $\{\boldsymbol{b}_j^{R}(t)\}$ are chosen according to Equations 3 and 4.

We argue that the proposed framework is a compact and expressive representation of 3D dynamic scenes that consist of multiple decoupled and complicated motions. By sharing the basis functions between Gaussians, we constrain neighbor Gaussians and rigid areas to choose similar trajectories, softly enforcing locality and rigidity a priori. At the same time, sharing the motion coefficients and keeping the number of basis trajectories low, dramatically softens the restrains of this ill-posed optimization problem. The efficient design of the basis functions retains the rendering speed at the same levels as with the original Gaussian Splatting, enabling real-time rendering not only for static scenes but for dynamic as well.

3.3 Optimization Framework

Regularization: The optimization process will naturally converge to a linear combination of all basis functions per Gaussian. Furthermore, Gaussians tend to move, even if they are part of the background as long as they do not harm the photometric accuracy with their movement (e.g. Gaussians moving on a white wall). While the former is desirable and increases expressivity, the latter harms the representation's interpretability since the basis functions are forced to learn trajectories for unnecessary movements. To solve this problem, we add an L1 regularization loss on the motion coefficients to penalize unnecessary choices of basis functions and thus, apparent but nonexistent movements:

$$\mathcal{L}_1 = \frac{1}{NB} \sum_{i=1}^{N} \sum_{j=1}^{B} |c_{ij}|. \tag{12}$$

Sparsity: Enforcing sparsity between the trajectories is more challenging. While $\mathcal{L}_1$ loss is supposed to encourage sparsity by increasing competition between the weights, we observed that the moving Gaussians kept choosing a linear combination of the trajectories with just smaller weights. To solve this we added a new stronger loss that explicitly forces each Gaussian to choose only very few trajectories:

$$\mathcal{L}_s = \frac{1}{N} \sum_{i=1}^{N} \frac{\|c_i\|_1}{\|c_i\|_\infty}. \tag{13}$$

With a strong enough loss weight, this quantity enforces strict sparsity which can lead to dynamic motion decoupling and novel motion synthesis.

Rigidity: Our representation naturally encourages rigidity between the Gaussians. Restraining the number of trajectories drives nearby points to choose the same trajectory. Furthermore, when a Gaussian is split or densified into a new one, we choose to copy the motion coefficients to the new Gaussian, to naturally enforce rigidity to the scene. We experiment with a simple rigid loss:

$$\mathcal{L}_r = \frac{1}{Nk} \sum_{i=1}^{N} \sum_{j \in knn_{i;k}} w_{ij} \|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2, \tag{14}$$

where the set of j Gaussians choose the k-nearest neighbours of i and weight the loss, similar to [28], by an isotropic Gaussian weighting factor per Gaussian pair $w_{ij} = exp\left(-\lambda_w \|\boldsymbol{\mu}_i(t^*) - \boldsymbol{\mu}_j(t^*)\|_2^2\right)$. We demonstrate that $\mathcal{L}_r$ does not further increase the rendering quality of the scene, solidifying our intuition that our motion representation is inherently rigid, given a small number of trajectories. For ablations on the optimization scheme, please refer to the Supplementary material.

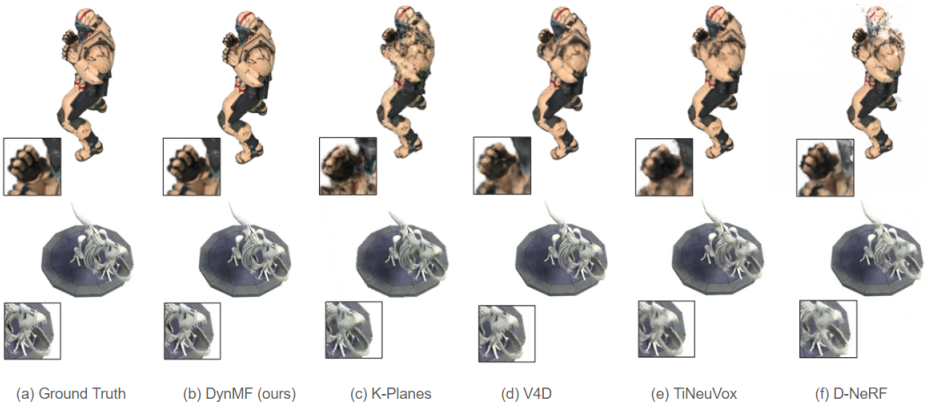

Fig. 3: Qualitative comparison on the D-NeRF dataset between our method and the state-of-the-art.

4 Experiments

4.1 Experimental Setup

In this section, we present the results of our method for three different scenarios; synthetic-scene monocular, real-scene monocular and real-scene multi-view.

D-NeRF [34] is a monocular synthetic dataset with a total of 8 rigid and non-rigid scenes. Each scene contains dynamic frames, ranging from 50 to 200 in number and on a unique timestamp. The camera pose for each timestamp is also different. **HyperNeRF** [32] is a monocular dataset on real-scenes. It consists of

Table 1: Quantitative results on synthetic monocular D-NeRF dataset. The rendering resolution is 400×400. [1] Concurrent work that utilizes Gaussian-splatting. [2] Rendering resolution is 800×800.

Model	PSNR(dB) ↑	SSIM ↑	LPIPS ↓	Time ↓	FPS ↑
D-NeRF [34]	29.67	0.953	0.06	15.9hours	0.1
K-Planes [15]	31.61	0.970	–	52mins	0.1
TiNeuVox [14]	32.67	0.970	0.04	21m	1.5
V4D [17]	33.72	0.980	–	4.9hours	2
[19][1]	32.07	0.96	–	8mins	–
[12][1]	34.26	0.97	0.03	**5mins**	1257
Gaufre [26][1]	34.80	0.982	0.02	25mins	50
4DGS [51][1,2]	34.05	0.98	0.02	20mins	82
DynMF (Ours)	**36.89**	**0.983**	**0.02**	20mins	**300**

many scenes of different real objects and actions with one or two cameras that are moving over time. We choose one rigid and three highly non-rigid scenes to test the method's efficiency in challenging cases like these. The four scenes are the '3d-printer', 'chicken', 'broom', and 'peel-banana'. **N3DV dataset** [24] dataset is a real-world dataset consisting of 6 very challenging dynamic scenes. The scenes are captured by 15-20 static cameras in total that share a common time. The scenes consist of long highly non-rigid motions and dynamic opacity (e.g. flame or smoke).

4.2 Implementation Details

The motion representation field is implemented in PyTorch [33], while we keep the differentiable Gaussian rasterization implemented by 3D-GS [20]. The MLP consists of 8 linear layers of 256-width with 2 final layers producing $(B, 3)$ and $(B, 4)$ numbers that constitute the displacement and quaternion correction per trajectory for a specific time. For the number of trajectories B, we choose 10 for D-NeRF and 32 for HyperNeRF and N3DV dataset. We apply positional encoding to the input of the MLP, which has been shown to be fundamental for good performance [29, 43]. Since the network is not coordinate-based and is solely based on time, a high-frequency input is needed for producing expressive trajectories for the dynamic scenes. We choose 32 frequencies to encode time. See Section 4.5 for ablations on the number of trajectories and position encoding of time. For training, we conduct training for a total of 30k iterations, while enabling the training of the basis functions and motion coefficients only after the first 3k iterations. All the parameters are kept the same with respect to 3D-GS while the learning rate of the basis is exponentially decaying from 8e-4 to 8e-6 and the learning rate of the coefficients is kept at 8e-3. 2k Gaussians' means are randomly initialized around the center of the scene for all datasets, solidifying the stability and generalizability of our representation.

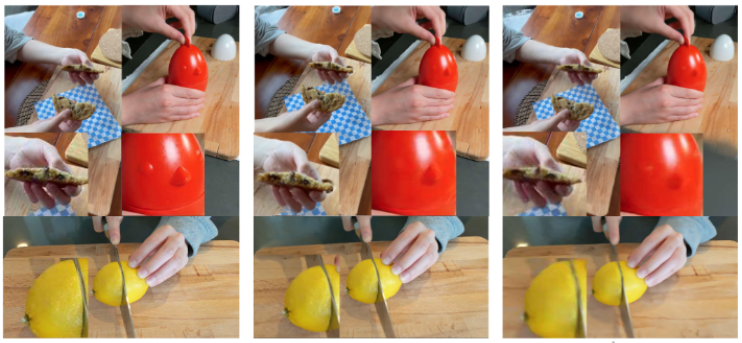

Fig. 4: Qualitative comparison on the HyperNeRF dataset.

Table 2: Per-scene quantitative comparisons on HyperNeRF monocular real dynamic scenes. [1] Concurrent work that utilizes Gaussian-splatting.

Method	3D Printer		Chicken		Broom		Peel Banana	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Nerfies [31]	20.6	0.83	26.7	0.94	19.2	0.56	22.4	0.87
HyperNeRF [32]	20.0	0.59	26.9	0.94	19.3	0.59	23.3	0.90
TiNeuVox [14]	**22.8**	**0.84**	28.3	**0.95**	21.5	0.69	24.4	0.87
3D-GS [20][1]	18.3	0.60	19.7	0.70	20.6	0.63	20.4	0.80
4DGS [7][1]	22.1	0.81	28.7	0.93	22.0	**0.70**	28.0	**0.94**
[12][1]	21.1	0.75	25.8	0.86	20.4	0.66	26.6	0.92
DynMF (ours)	22.4	0.70	**28.5**	0.81	**22.1**	0.61	**28.1**	0.89

4.3 Results

Results on Synthetic data: A quantitative evaluation of the D-NeRF synthetic Dataset is presented in Table 1. Our method surpasses all state-of-the-art methods by a significant gap in all metrics. This confirms that DynMF is an adequate representation of rigid, decoupled, and non-rigid/deformable motion as well. Regarding speed, our method achieves state-of-the-art results in only 10 minutes of training, while after half an hour it synthesizes most of the dynamic scenes almost perfectly. The FPS rate for 400×400 images reaches more than 300FPS confirming the carefully designed representation in terms of efficiency, achieving real-time dynamic novel view synthesis. Qualitative comparisons can be seen in Figure 3.

Results on Real data: DynMF seems to achieve state-of-the-art results on the HyperNeRF and N3DV datasets as well, as shown in Tables 2 and 3. These datasets consist of very challenging non-rigid motions and semi-transparent areas that the representation of motion factorization can adequately and efficiently express. Again, time-wise our method allows for real-time dynamic rendering even on scenes of high resolution like N3DV (i.e. 1280×1080). For a per-scene evaluation of all three datasets and for more qualitative results, please refer to the supplementary material.

Fig. 5: Novel view synthesis of DynMF. The figure shows selected frames from different time frames for the scenes 'flame-steak' and 'sear-steak'.

Fig. 6: Qualitative comparison on the N3DV dataset between our method and the state-of-the-art.

4.4 Motion Decomposition

A key component of our proposed representation is its ability to explicitly decompose each dynamic scene to its core independent motions. Specifically, by applying the sparsity loss described in Section 3.3, we enforce each Gaussian to choose only one of the few trajectories available. This design combined with the inherent rigid property of our representation drives all nearby Gaussians to choose the same and only trajectory. This strongly increases the controllability of the dynamic scene, by disentangling motions, allowing for novel scene creation, interactively choosing which part of the scene is moving, and so on. Figure 7 shows two examples of motion control in two synthetic scenes. Please, refer to the supplementary video for a deeper understanding of the motion decomposition possibilities that our representation enables.

4.5 Ablation Studies

Time encoding and Number of trajectories: The number of trajectories, B is the most important hyperparameter of DynMF, for it is the one that decides in what detail will the scene be modeled. In parallel, the positional encoding on

Table 3: Quantitative results on real-scene multi-view N3DV dataset. The rendering resolution is 1386×1014. [1] Concurrent work that utilizes Gaussian-splatting.

Model	PSNR(dB) ↑	SSIM ↑	LPIPS ↓	Time ↓	FPS ↑
DyNeRF [24]	29.58	-	**0.08**	1344hours	0.015
MixVoxels-L [46]	30.80	0.96	-	1.3hours	16.7
K-Planes [15]	31.63	0.964	-	1.8hours	0.15
Hexplane [7]	**32.22**	**0.98**	0.09	12hours	0.09
[19][1]	28.89	0.945	-	1hours	118
[12][1]	31.62	0.94	0.14	1hours	**277**
4DGS [53][1]	32.01	-	**0.06**	-	114
DynMF (Ours)	31.70	0.946	0.18	**40mins**	135

Fig. 7: Demonstration of the decomposition capabilities of our representation. Independent movement of the right foot and the left hand of the 'Jumpingjacks' and 'Mutant' scenes respectively.

Table 4: Ablation study on the number B of the motion trajectories and the frequencies F of the positional encoding applied on time. PSNR is used as the metric on the 'Mutant' scene from the D-NeRF dataset and the 'Flame-steak' scene from the N3DV dataset.

Model (B, F)	Mutant	Steak	Model (B, F)	Mutant	Steak
DynMF (1,32)	29.75	28.85	DynMF (10,1)	39.08	29.32
DynMF (4,32)	39.98	30.34	DynMF (10,6)	39.61	29.59
DynMF (10,32)	**40.79**	**31.57**	DynMF (10,16)	40.50	31.29
DynMF (20,32)	40.76	31.03	DynMF (10,32)	**40.79**	**31.57**
DynMF (50,32)	40.53	30.99	DynMF (10,60)	40.53	29.80

time proved to be important as well, with notoriously high frequencies, F, needed for state-of-the-art results. Table 4 shows experiments on multiple combinations of (F, B). We can see that both synthetic and real scenes can be adequately modeled by very few trajectories while adding more only subtly increases the performance. At the same time, while encoding time is not vital for satisfactory rendering, mapping it to higher frequencies provides the mean for more expressive trajectories by the small MLP. Please, refer to the Supplementary material for qualitative comparisons for different values of B and F.

Fourier Series: Table 5 shows results on two different scenes (mutant from D-NeRF and flame-steak from DyNeRF) in the case of a non-learned motion and rotation base, in this case the Fourier Series. We conducted a few experiments with the basis functions following Equation 3 and we varied the number of frequencies the basis could incorporate. We observe that the Fourier series can satisfactorily express the motion field of a dynamic scene but is not expressive and smooth enough to decode the details of complicated scenes. To the best of our knowledge, this is the first totally explicit dynamic scene representation, comprised of Gaussians and Fourier series. In supplementary material, we further investigate the reasons why the Fourier basis is not expressive enough.

Table 5: Ablation study on the basis function. PSNR is used as the metric on the 'Mutant' scene from the D-NeRF dataset and the 'Flame-steak' scene from the N3DV dataset.

Model	Mutant	Steak
DynMF (Learned Basis)	**40.79**	**31.57**
DynMF (Fourier 10)	33.17	28.95
DynMF (Fourier 20)	35.21	27.18
DynMF (Fourier 50)	33.87	23.42
DynMF (Fourier 100)	21.12	19.15

5 Conclusion & Future Work

In this work, we have introduced, DynMF, a motion factorization framework that enables real-time rendering of dynamic scenes. The carefully designed representation of a small MLP queried only by time allows for training convergence in half an hour and for a rendering speed of more than 120 FPS. At the same time, the shared motion basis scheme supports inherently the locality and rigidity of the motions and allows for state-of-the-art rendering quality from novel views. Last but not least, the motion decomposition properties allow us to dive deeper into the structure of the scene and create novel instances by independently enabling and disabling motions on the scene. For future work, we believe that DynMFis potentially a good representation for efficient and accurate tracking as well. While the motion factorization framework along with the 3D Gaussian splatting gives the opportunity for detailed and expressive novel view synthesis of dynamic scenes, it is by itself constructed in a way to allow tracking of all the points on the scene at every time.

References

1. Abou-Chakra, J., Dayoub, F., Sünderhauf, N.: Particlenerf: Particle based encoding for online neural radiance fields. arXiv preprint arXiv:2211.04041 (2022)
2. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: A dual representation for nonrigid structure from motion. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(7), 1442–1456 (2011). https://doi.org/10.1109/TPAMI.2010.201
3. Attal, B., Huang, J.B., Richardt, C., Zollhoefer, M., Kopf, J., O'Toole, M., Kim, C.: HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
4. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. ICCV (2021)
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)
6. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. ICCV (2023)
7. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. CVPR (2023)
8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision (ECCV) (2022)
9. Chen, A., Xu, Z., Wei, X., Tang, S., Su, H., Geiger, A.: Factor fields: A unified framework for neural fields and beyond. arXiv preprint arXiv:2302.01226 (2023)
10. Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., Sullivan, S.: High-quality streamable free-viewpoint video. ACM Trans. Graph. **34**(4) (jul 2015)
11. Das, D., Wewer, C., Yunus, R., Ilg, E., Lenssen, J.E.: Neural parametric gaussians for monocular non-rigid object reconstruction (2023)
12. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes (2024)
13. Duisterhof, B.P., Mandi, Z., Yao, Y., Liu, J.W., Shou, M.Z., Song, S., Ichnowski, J.: Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes (2023)
14. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., Tian, Q.: Fast dynamic radiance fields with time-aware neural voxels. In: SIGGRAPH Asia 2022 Conference Papers (2022)
15. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: CVPR (2023)
16. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
17. Gan, W., Xu, H., Huang, Y., Chen, S., Yokoya, N.: V4d: Voxel for 4d novel view synthesis (2022)
18. Huang, Y.H., Sun, Y.T., Yang, Z., Lyu, X., Cao, Y.P., Qi, X.: Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. arXiv preprint arXiv:2312.14937 (2023)
19. Katsumata, K., Vo, D.M., Nakayama, H.: An efficient 3d gaussian representation for monocular/multi-view dynamic scenes (2023)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

21. Kumar, S., Dai, Y., Li, H.: Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. Pattern Recognition **71**, 428–443 (2017). https://doi.org/https://doi.org/10.1016/j.patcog.2017.05.014, https://www.sciencedirect.com/science/article/pii/S0031320317302029
22. Levoy, M., Hanrahan, P.: Light field rendering. ACM Transactions on Graphics (SIGGRAPH) (1996)
23. Li, H., Luo, L., Vlasic, D., Peers, P., Popović, J., Pauly, M., Rusinkiewicz, S.: Temporally coherent completion of dynamic shapes. ACM Trans. Graph. **31**(1) (feb 2012)
24. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., Lv, Z.: Neural 3d video synthesis from multi-view video. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5511–5521. IEEE Computer Society, Los Alamitos, CA, USA (jun 2022)
25. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. arXiv preprint arXiv:2312.16812 (2023)
26. Liang, Y., Khan, N., Li, Z., Nguyen-Phuoc, T., Lanman, D., Tompkin, J., Xiao, L.: Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis (2023)
27. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. arXiv:2312.03431 (2023)
28. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 3DV (2024)
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022)
31. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. ICCV (2021)
32. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. ACM Trans. Graph. **40**(6) (dec 2021)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
34. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
35. Shao, R., Sun, J., Peng, C., Zheng, Z., Zhou, B., Zhang, H., Liu, Y.: Control4d: Efficient 4d portrait editing with text. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
36. Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., Liu, Y.: Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering (2023)

37. Shaw, R., Song, J., Moreau, A., Nazarczuk, M., Catley-Chandar, S., Dhamo, H., Perez-Pellitero, E.: Swags: Sampling windows adaptively for dynamic 3d gaussian splatting (2023)
38. Shenchang, E.C., Williams, L.: View interpolation for image synthesis. ACM Transactions on Graphics (SIGGRAPH) (1993)
39. Song, L., Chen, A., Li, Z., Chen, Z., Chen, L., Yuan, J., Xu, Y., Geiger, A.: Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. IEEE Transactions on Visualization and Computer Graphics **29**(5), 2732–2742 (2023). https://doi.org/10.1109/TVCG.2023.3247082
40. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022)
41. Sun, J., Jiao, H., Li, G., Zhang, Z., Zhao, L., Xing, W.: 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In: arXiv preprint arXiv:2403.01444 (2024)
42. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P., Barron, J.T., Kretzschmar, H.: Block-NeRF: Scalable large scene neural view synthesis. arXiv (2022)
43. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020)
44. Valmadre, J., Lucey, S.: General trajectory prior for non-rigid reconstruction. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1394–1401 (2012). https://doi.org/10.1109/CVPR.2012.6247826
45. Wang, C., Eckart, B., Lucey, S., Gallo, O.: Neural trajectory fields for dynamic novel view synthesis. In: ArXiv Preprint (March 2021)
46. Wang, F., Tan, S., Li, X., Tian, Z., Liu, H.: Mixed neural voxels for fast multi-view video synthesis. arXiv preprint arXiv:2212.00190 (2022)
47. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. In: International Conference on Computer Vision (2023)
48. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021)
49. Wei, M., Miaomiao, L., Mathieu, S., Hongdong, L.: Learning trajectory dependencies for human motion prediction. In: ICCV (2019)
50. Weng, C.Y., Curless, B., Srinivasan, P.P., Barron, J.T., Kemelmacher-Shlizerman, I.: HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16210–16220 (June 2022)
51. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Xinggang, W.: 4d gaussian splatting for real-time dynamic scene rendering. arXiv preprint arXiv:2310.08528 (2023)
52. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., Joo, H.: Banmo: Building animatable 3d neural models from many casual videos. In: CVPR (2022)
53. Yang, Z., Yang, H., Pan, Z., Zhu, X., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv 2310.10642 (2023)
54. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. arXiv preprint arXiv:2309.13101 (2023)

55. Yu, H., Julin, J., Milacski, Z.A., Niinuma, K., Jeni, L.A.: Cogs: Controllable gaussian splatting. arXiv (2023)
56. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492 (2020)
57. Zhang, Q., Baek, S.H., Rusinkiewicz, S., Heide, F.: Differentiable point-based radiance fields for efficient view synthesis. In: SIGGRAPH Asia 2022 Conference Papers. SA '22, Association for Computing Machinery, New York, NY, USA (2022)
58. Zhang, Y., Black, M.J., Tang, S.: We are more than our joints: Predicting how 3d bodies move. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3372–3382 (2021)
59. Zhao, F., Yang, W., Zhang, J., Lin, P., Zhang, Y., Yu, J., Xu, L.: Humannerf: Efficiently generated human radiance field from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7743–7753 (June 2022)
60. Zhu, Y., Lucey, S.: Convolutional sparse coding for trajectory reconstruction. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 529–540 (2015). https://doi.org/10.1109/TPAMI.2013.2295311
61. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa splatting. IEEE Transactions on Visualization and Computer Graphics **8**(3), 223–238 (2002)