MoBa-GS: Learning a Spatially-Varying Motion Basis over a Dynamic Canonical Space for 4D Reconstruction

Guan Yuan Tan, Arghya Pal, Sailaja Rajanala, Raphaël C.-W. Phan, and Chee-Ming Ting

School of Information Technology, Monash University, Malaysia
{guan.tan, arghya.pal, sailaja.rajanala, raphael.phan, cheeming.ting}@monash.edu

Abstract. Faithfully capturing the intricate relationship between motion and geometry in dynamic scenes is essential for 4D reconstruction. Recent state-of-the-art methods rely on monolithic deformation networks or global time-basis factorizations that struggle to represent complex, non-rigid topological changes. We propose MoBa-GS, a framework that resolves this entanglement by introducing a structural inversion: a spatially-factorized motion field coupled with adaptive geometric optimization. First, our model learns a low-frequency dynamic canonical space to represent coarse scene motion. Next, it decomposes complex, non-rigid motion into a Spatially-Varying Motion Basis of local kinematics, predicted from the canonical geometry, which is then linearly combined using dynamic blending weights. This formulation directly acts as an implicit neural scaffold, recovering both geometry and motion from random initialization, thereby removing the dependency on SfM point cloud priors. This design is further augmented with motion-guided densification and positional annealing to reduce geometry overfitting. Extensive experiments show that our framework surpasses prior state-of-the-art methods in reconstruction fidelity. Enabled by time-invariant caching, MoBa-GS requires an order-of-magnitude shorter training time ($<$18 minutes), a compact storage ($\sim$ 11 MB), and achieves real-time rendering speeds ($>$163 FPS). Our work establishes a new foundation for high-fidelity, efficient 4D representations without relying on explicit geometric priors. Code is available at https://github.com/tgy1221/MoBa-GS.

1 Introduction

Novel View Synthesis (NVS) for dynamic scenes is fundamental to a range of immersive applications, from virtual and augmented reality to advanced sports analysis. While explicit representations such as 3D Gaussian Splatting [18] have transformed the field of static scene reconstruction with photorealistic quality at real-time rendering speeds, extending this paradigm into the four-dimensional spatiotemporal domain introduces significant architectural and optimization complexities.

Recent dynamic 3DGS methods typically adopt one of two paradigms, both of which exhibit critical limitations in modeling complex scenes. The first paradigm relies on monolithic deformation networks (i.e., a single Multilayer Perceptron) [16, 23, 50] to predict the temporal evolution of canonical Gaussians. This approach inherently entangles the underlying 3D geometry of the scenes with the object motion. This entanglement leads to optimization instability, often termed the "moving target" problem, in which the canonical geometry and the deformation field interfere during joint optimization, resulting in severe geometric overfitting [4, 43].

To alleviate this, a second paradigm employs motion factorization, typically using a global "Time-Basis" (e.g., globally shared neural trajectories) [19]. While this approach is more structured, it implicitly assumes the scene consists of rigid components riding on a finite set of predefined global paths. This mathematically restricts the solution space: if a scene contains highly heterogeneous, independent movements, such as wrinkling fabric or fluid dynamics, a global Time-Basis fundamentally cannot span these divergent kinematics without a catastrophic explosion in the rank of the basis. Furthermore, both paradigms suffer from a severe operational vulnerability, as they rely heavily on pre-computed Structure-from-Motion (SfM) point clouds for initialization. Because SfM systematically fails to find consistent feature matches on moving subjects, these methods often collapse or produce severe artifacts when forced to operate without highly calibrated geometric point priors.

In this work, we introduce **MoBa-GS**, a novel framework that resolves these entanglements through a holistic co-design of a disentangled representation and synergistic training strategies. Our primary architectural design principle is a *Structural Inversion* from a global Time-Basis to a *Spatially-Varying Motion Basis* (Space-Basis). First, our model learns a low-frequency dynamic canonical space that serves as a robust, coarse motion prior. Next, it factorizes the complex, high-frequency residual motion into a localized "physics dictionary" of motion primitives specific to each spatial coordinate, which are then linearly combined using a set of dynamic blending weights (see Fig. 1). By intrinsically linking kinematic potential to spatial location, MoBa-GS grants every point independent kinematic freedom, enabling high-fidelity representation of complex non-rigid motions.

Crucially, this spatially-varying basis inherently restricts the solution space to a locally low-dimensional manifold of natural motion. This constraint acts as an *Implicit Neural Scaffold*, enabling MoBa-GS to recover high-fidelity geometry and motion entirely from random point cloud initialization, successfully removing the dependency on sparse SfM priors. Furthermore, because our Basis Predictor depends strictly on static spatial coordinates, it enables *Time-Invariant Caching*. Once the geometry stabilizes, the basis vectors are pre-computed and cached, bypassing the MLP overhead and reducing runtime deformation to a lightweight dot product.

To augment the potential of the current design, we introduce two training strategies. First, Motion-Guided Densification provides an explicit dynamic

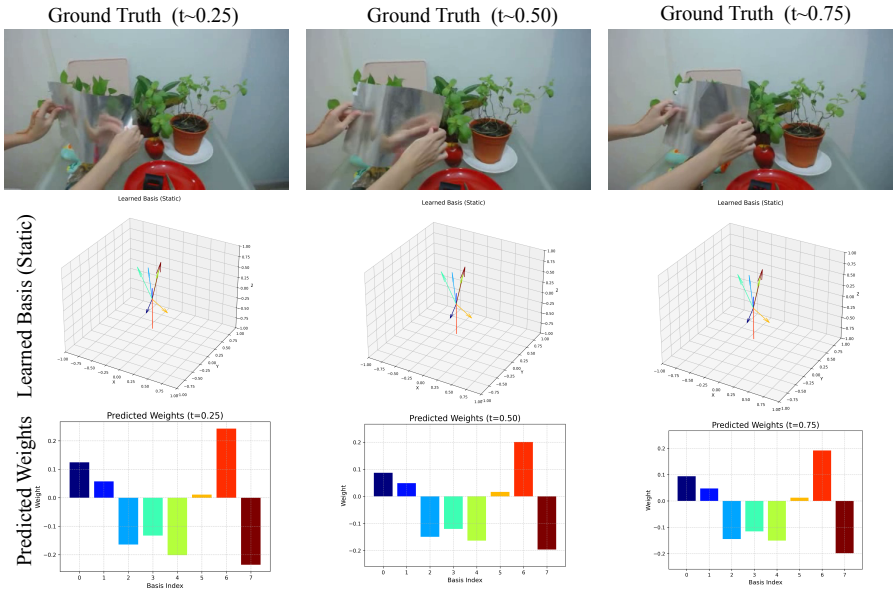

Fig. 1: Spatially-Varying Motion Basis. Our model learns a static, time-invariant "toolkit" of local motion primitives, called **Learned Basis** (middle row), which is consistent across all time steps. The final deformation is then synthesized by a sparse set of **Predicted Weights** (bottom row) that change dynamically to match the motion in the **Ground Truth** frames (top row). This shows that our model is not a black box; it has learned an interpretable, disentangled, and dynamic representation of motion.

signal to the geometry refinement process, preferentially allocating geometric capacity to regions that exhibit both high view-space error and large temporal motion. Second, a Positional Annealing schedule freezes the canonical space progressively during late-stage training. This is introduced to solve the "moving target" problem, preventing geometric overfitting and forcing the model to converge to a clean, more structured foundation.

Our contributions are summarized as follows:

1. We propose a Structural Inversion from global temporal trajectories to a Spatially-Varying Motion Basis, enabling high-fidelity, non-rigid 4D reconstruction.
2. We introduce an ultra-lightweight architecture using time-invariant caching and an Implicit Neural Scaffold, eliminating the dependency on COLMAP initialization.
3. We introduce motion-guided densification and positional annealing to mitigate geometric overfitting.

2 Related Works

Dynamic 3D Reconstruction. Neural Radiance Field (NeRF) [26] served as a frontier of static 3D reconstruction. Subsequent works extended this implicit representation and improved its quality and efficiency [2, 3, 27, 34, 51, 52]. They also inspired several works to extend into dynamic NeRF [13, 14, 22, 37, 44]. These dynamic NeRF models map and deform the static, canonical representation by learning a deformation field. Popular dynamic NeRF models such as D-NeRF [30], Nerfies [28], and HyperNeRF [29] have demonstrated the feasibility of a monolithic MLP for generating the deformation fields to capture non-rigid motion. However, these works suffered from long training time and slow rendering speed, limiting their practicality in real-world applications [25, 30, 32, 40, 41, 47]. To improve efficiency, alternatives to the large MLP such as explicit voxel grids [10, 38, 45] or plane-based factorizations such as [12, 31] and HexPlane [4] offer faster convergence and rendering speed. Although these prior works improved efficiency, the core deformation field is still handled by a single holistic function.

Dynamic 3D Gaussian Splatting. The introduction of 3D Gaussian Splatting (3DGS) [18] offered a new path to real-time rendering, and also inspired many works to extend to dynamic scenes [6, 9, 20, 21, 43, 48, 49]. Most of these works rely on a global MLP-based deformation field paradigm from dynamic NeRFs. These works [1, 16, 23, 35, 50] employ an MLP to predict the transformation of each Gaussian (translation, rotation, scale) at each time step. While this monolithic approach is effective, it inherits the same shortcomings as its NeRF-based predecessors: a single, monolithic network is tasked with learning a complex, entangled 4D field, which can be inefficient and prone to overfitting.

To establish structural constraints, recent works [7, 17, 21, 24] have explored various forms of motion factorization, decomposing motion into simpler, structured components. Plane-based factorization [12] represents the 4D field as low-dimensional planes, inspiring works like [9, 43] to use it as a backbone. Moreover, [19] factorizes motion into a set of globally shared, time-varying basis trajectories. Similarly, [42] utilizes global $SE(3)$ bases, while DynaSplat [8] uses hard static-dynamic classification and hierarchical neighbor averaging. However, global temporal bases implicitly enforce rigid trajectories that struggle to span heterogeneous, localized non-rigid warping without catastrophic rank explosion. Furthermore, methods such as Shape of Motion [42] rely on external data-driven priors (e.g., depth and 2D tracking). In contrast, our model introduces a structural inversion to a *Spatially-Varying Motion Basis*, fully self-supervised from RGB inputs only. Unlike DynMF [19], which uses a global dictionary for all points, our framework learns a localized kinematic subspace for every distinct spatial point, allowing our model to distinguish between different material properties (e.g., rigid vs elastic) within the same scene.

Scaffolded and Feed-Forward 4D Representations. To manage memory overhead, MoDec-GS [20] employs Temporal Interval Adjustment to slice videos into discrete temporal segments, while 4D Scaffold [6] utilizes sparse 4D grid-aligned Eulerian anchors. However, discrete temporal partitioning inherently risks boundary discontinuities, and grid-based tracking is prone to aliasing

over long sequences. In contrast, MoBa-GS is a purely continuous Lagrangian formulation (at minimum C^1 continuous in time), tracking points smoothly without grid constraints while maintaining a highly compact memory footprint ($\sim$ 11 MB vs 100+ MB). Finally, feed-forward models [5, 11, 33, 53] and VGGT-based models [15, 39, 55] offer robust, amortized priors via massive pre-training. While they perform well at generalized correspondence, their one-shot paradigm inherently smooths scene-specific, high-frequency details. By utilizing our SpaceBasis, MoBa-GS serves as an ideal, high-fidelity refinement backend to elevate raw feed-forward outputs to photorealistic ceiling, achieving test-time optimization quality from scratch while time-invariant caching guarantees real-time rendering speeds.

3 Methodology

3.1 Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting represents the scene with a set of explicit 3D Gaussians. Unlike point clouds, each Gaussian has unique attributes that control its influence over the scene. For a position $x \in \mathbb{R}^{3\times1}$, the 3D Gaussian can be formulated as:

$$G(x) = o \cdot e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} \tag{1}$$

where opacity $o \in [0, 1]$ is used to determine the importance of the Gaussian during rendering. Moreover, each Gaussian is parameterized as the center location of the Gaussian μ, and the covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$. The covariance matrix can be decomposed as $\Sigma = RSS^TR^T$, where the rotation matrix R is represented by quaternion $q = [r_w, r_x, r_y, r_z]$, and a scaling factor, represented by $s \in \mathbb{R}^3$. Each 3D Gaussian has its view-dependent color c modeled with spherical harmonics coefficients sh. During rendering, 3D Gaussians are splatted onto the 2D camera plane via differential projection. Next, the projected 2D Gaussians are sorted with a tile-based rasterizer, and the α-blending is employed. To project a 3D Gaussian into 2D, a viewing transform matrix W and the Jacobian matrix J of the affine approximation of the projective transformation are used to obtain the 2D covariance matrix Σ_{2D} through: $\Sigma_{2D} = JW\Sigma W^TJ^T$, and the 2D center position $\mu_{2D} = JW\mu$. Given a 2D pixel p, the rendering contribution of a 3D Gaussian on the viewpoint W is computed through a 2D version of (1). Finally, for each pixel, the color $C(p)$ can be rendered by α-blending:

$$C(p) = \sum_{i}^{N} SH(sh_i, v_i)\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where SH is the spherical harmonic function, v_i is the view direction, and α_i represents the density computed from the i-th 3D Gaussian.

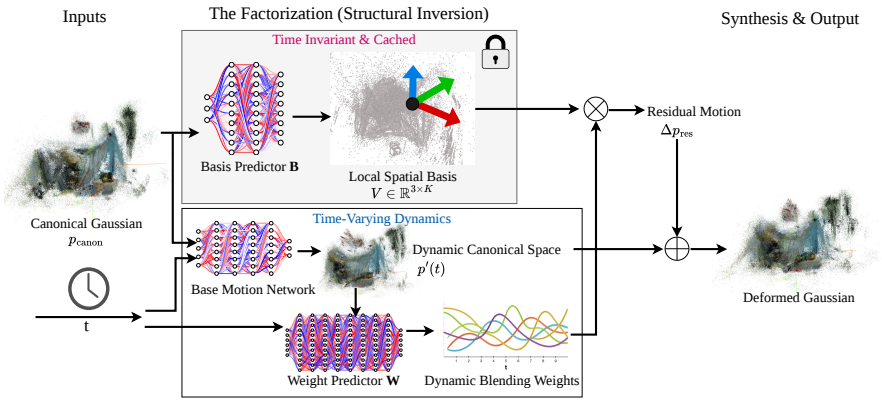

Fig. 2: An overview of our framework. Our architecture consists of two main stages. **First**, a Base Motion Network learns a low-frequency motion prior ($\Delta\mathbf{p}_{\text{base}}$) to create a Dynamic Canonical Space ($\mathbf{p}'$). **Second**, the residual motion is factorized. A Basis Predictor uses the *static* $\mathbf{p}_{\text{canon}}$ to learn a Local Motion Basis ($\mathbf{B}(\mathbf{p}_{\text{canon}})$). A Weight Predictor uses the *dynamic* state $(\mathbf{p}', t)$ to learn a sparse set of blending weights ($\mathbf{w}(\mathbf{p}', t)$). The final residual motion ($\Delta\mathbf{p}_{\text{res}}$) is synthesized via a weighted sum of the basis vectors. This hierarchical, factorized approach disentangles the motion representation, leading to higher fidelity and more stable training.

3.2 Overview of MoBa-GS

We aim to design a pipeline that models dynamic 4D scenes while mitigating concerns such as *fixed origin*, *moving target*, and the *geometric overfitting*, as discussed in Sec. 1. Our methodology provides a holistic co-design of a disentangled representation and synergistic training strategies to tackle the above challenges by introducing a spatially-varying motion basis, motion-guided densification, and positional annealing, which will be discussed in the following sections.

3.3 Learning a Spatially-Varying Motion Basis

Directly regressing the 4D deformation field $\Delta(\mathbf{p}, t)$ using a single *tiny* neural network [4, 43], which learns a mapping from the low-dimensional input $(\mathbf{p}, t) \in \mathbb{R}^4$ to high-dimensional deformation outputs (i.e., 3D displacements, rotations, and scales), imposes significant limitations. To address this, we reformulate the problem through the lens of *motion field-aware factorization* [4, 12], which decomposes the deformation field into a coarse, low-frequency base motion and a high-frequency residual field that is locally low-rank, as visualized in Fig. 2 and Fig. 3.

Dynamic Canonical Space. To solve the "fixed origin" problem, our model first learns a coarse, low-frequency motion prior. The Base Motion Network,

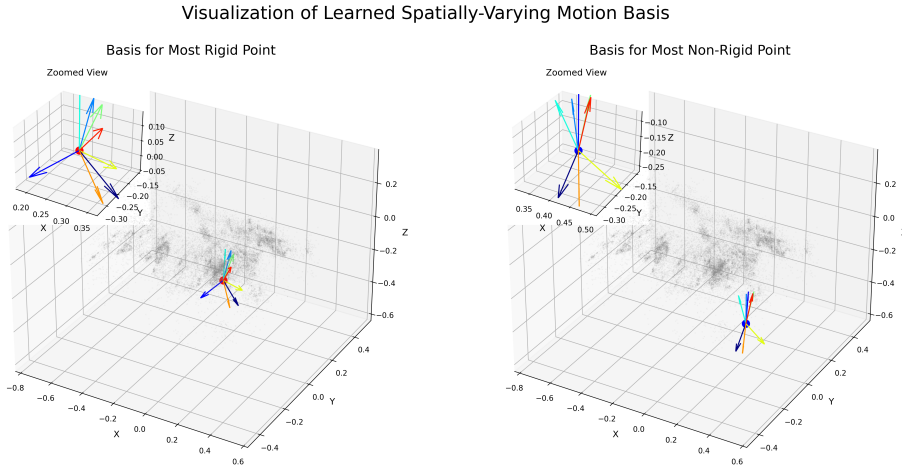

Fig. 3: Visualization of our Learned Spatially-Varying Motion Basis. We visualize the learned basis vectors for the most rigid point (left) and the most non-rigid point (right) in the canonical space of the 'As' scene. The basis for the rigid point is well distributed, indicating a capacity for general transformations. In contrast, the basis for the non-rigid point is more directionally aligned, learning a specialized and efficient representation for complex, localized deformation. This shows that our model learns a meaningful, spatially-varying dictionary of motion primitives.

$\mathcal{D}_{\text{base}}$, is explicitly designed to act as a low-pass filter in the temporal domain. It is a *lightweight* MLP, enabled with a time-varying positional embedding, γ_t, configured with a low maximum frequency ($L = 4$). This network produces a base displacement $\Delta\mathbf{p}_{\text{base}}$ which deforms the canonical positions $\mathbf{p}_{\text{canon}}$ into a time-varying dynamic canonical space $\mathbf{p}'(t)$, which serves as a robust motion prior, i.e.,

$$\mathbf{p}'(t) = \mathbf{p}_{\text{canon}} + \Delta\mathbf{p}_{\text{base}}(\mathbf{p}_{\text{canon}}, \gamma_{t,L=4}(t)) \tag{3}$$

This creates a dynamic reference frame. By digesting the shared, coarse trajectories (e.g., global translation or remaining camera ego-motion), $\mathcal{D}_{\text{base}}$ prevents the high-frequency residual fields from being contaminated by low-frequency signals. This simplifies the overall learning problem by constraining the subsequent residual network to model only the localized, non-linear deviation.

Factorizing Residual Motion with a Spatially-Varying Basis. Our key insight is that, while the high-frequency residual motion is complex globally, it is locally low-dimensional. The possible deformations of a point on a rigid object are fundamentally different from those of a point on a deforming cloth. We model this by factorizing the residual motion into two decoupled components: a time-invariant basis field representing the intrinsic motion capabilities of the geometry, and a time-varying coefficient field representing the dynamic activation of those capabilities.

1. The Basis Field: Low-Rank Local Subspaces. We learn a **Basis Predictor Network** $\mathcal{B}$ that maps a point's canonical location to a low-rank subspace of possible motions. The Basis Predictor network, conditioned solely on the static position $\mathbf{p}_{\text{canon}}$, produces a set of K basis vectors. This projection matrix $\mathbf{V}(\mathbf{p}_{\text{canon}}) \in \mathbb{R}^{3\times K}$ represents the time-invariant, intrinsic kinematic priors of the geometry, forming a compact and efficient, local motion dictionary for each point that is far more expressive than a global basis.

$$\mathbf{V}(\mathbf{p}_{\text{canon}}) = \{\mathbf{b}_1, \ldots, \mathbf{b}_K\} = \mathcal{B}(\gamma_{\text{pos}}(\mathbf{p}_{\text{canon}})) \tag{4}$$

Time-Invariant Caching: As the Basis Predictor depends on the static canonical position p_{canon}, it enables superior computational efficiency. Once the geometry stabilizes during late-stage training, these basis vectors are pre-computed and cached. During rendering, the MLP's runtime overhead is bypassed, enabling real-time performance.

2. The Coefficient Field: Spatiotemporal Dynamics. A separate, higher-capacity **Weight Predictor** network, $\mathcal{W}$, learns the time-varying activation of the local basis. It is conditioned on the dynamic state—the dynamic canonical position $\mathbf{p}'(t)$ and a high-frequency time embedding ($L = 10$)—to predict a set of K scalar blending weights, $\mathbf{w}$. These weights represent the time-varying coordinates within the local motion subspace defined by the basis.

$$\mathbf{w}(\mathbf{p}', t) = \{w_1, \ldots, w_K\} = \mathcal{W}(\mathbf{p}'(t), \gamma_{t,L=10}(t)) \tag{5}$$

3. Motion Synthesis. The final residual displacement, $\Delta\mathbf{p}_{\text{res}}$, is synthesized via a linear combination of the basis vectors. This can be viewed as projecting the dynamic temporal signal (encoded in the weights) onto the learned spatial basis.

$$\Delta\mathbf{p}_{\text{res}} = \mathbf{V}(\mathbf{p}_{\text{canon}})\mathbf{w}(\mathbf{p}', t) = \sum_{k=1}^{K} w_k(\mathbf{p}', t) \cdot \mathbf{b}_k(\mathbf{p}_{\text{canon}}) \tag{6}$$

The final deformed position is $\mathbf{p}(t) = \mathbf{p}'(t) + \Delta\mathbf{p}_{\text{res}}$. Separate heads on the Weight Predictor network also output residual transformations for rotation ($\Delta\mathbf{q}$) and scale ($\Delta\mathbf{s}$).

4. Intrinsic Structural Prior. Regressing 3D displacement directly offers no meaningful constraints on the motion field. By forcing the 4D trajectory into a sparse combination of learned spatial bases, our solution is inherently constrained to lie on a locally low-dimensional manifold of natural motion. This acts as an Implicit Neural Scaffold, eliminating the need for engineered smoothness penalties and allowing high-fidelity recovery from random initialization, without relying on SfM point clouds. We further guide this manifold via a dynamic acceleration penalty, which exponentially decays for moving points, ensuring static regions remain rigid while allowing dynamic points to have full kinematic freedom.

3.4 Motion-Guided Densification

Standard densification logic is driven solely by view-space projection errors, making it spatially-biased and time-agnostic. It cannot distinguish between high-frequency texture on a static surface and true dynamic complexity, leading

to sub-optimal allocation of geometric capacity. To address this, we propose a spatiotemporally-aware strategy that treats densification as a form of importance sampling.

Our **Motion-Guided Densification** introduces a feedback loop that allocates geometric capacity preferentially to regions that are both complex in appearance and dynamic in motion. A Gaussian G_i is only eligible for densification if its view-space gradient magnitude exceeds a threshold τ_{grad} *and* its predicted motion magnitude exceeds a threshold τ_{motion}.

$$\text{Densify}(G_i) \quad \text{if} \quad \|\nabla_{\mathbf{xyz}}(G_i)\| > \tau_{\text{grad}} \quad \wedge \quad \|\Delta\mathbf{p}_i(t)\| > \tau_{\text{motion}} \tag{7}$$

where $\|\Delta\mathbf{p}_i(t)\|$ is computed from our deformation network, this feedback loop safely ignores static backgrounds. It focuses the model's finite capacity on the dynamic regions of highest complexity, improving efficiency and final fidelity.

3.5 Positional Annealing

We identified a critical failure mode in late-stage training: **geometric overfitting** [4,43], where the canonical positions $\mathbf{p}_{\text{canon}}$ become hyperspecialized to the training views, degrading generalization. We propose **Positional Annealing**, a curriculum learning strategy that stabilizes the optimization landscape. The learning rate for the canonical positions, $\eta_{\mathbf{p}}(i)$, follows a two-phase schedule:

$$\eta_{\mathbf{p}}(i) = \begin{cases} \eta_{\text{main}}(i) & \text{if } i < i_{\text{anneal}} \\ \eta_{\text{anneal}}(i) & \text{if } i \geq i_{\text{anneal}} \end{cases} \tag{8}$$

where i_{anneal} is set as a ratio of total training time to ensure robustness across different training durations. This curriculum constrains the model to a more generalizable solution manifold by freezing the canonical geometry and forcing the motion fields to learn a more robust representation for the final fine-tuning stage.

3.6 Optimization

Decoupled Alternating Optimization We observed that joint optimization of the canonical geometry and the complex motion networks can be unstable, as each component serves as a moving target for the other. To stabilize the training process, particularly during the refinement phase after densification, we introduce and analyze a Decoupled Alternating Optimization schedule. For a defined range of iterations, we alternate between:

1. **Phase 1 (Deformation Focus):** Freezing the canonical Gaussian parameters $(\mu, q, s, ...)$ and training only the deformation networks $(\mathcal{D}_{\text{base}}, \mathcal{B}, \mathcal{W})$ for $N_{\text{def}} = 2$ consecutive iterations. This forces the motion field to learn the optimal temporal mappings grounded in a stable geometric foundation.

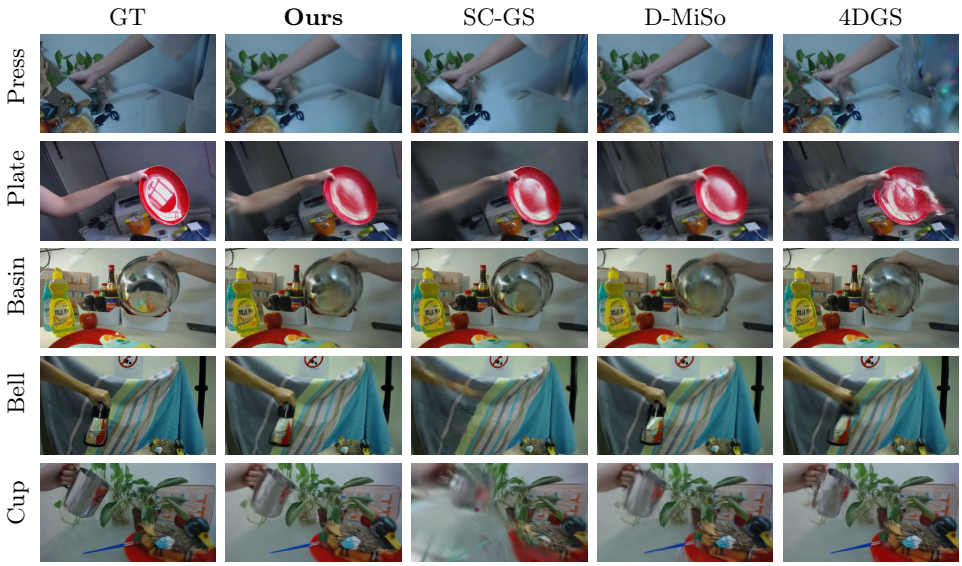

Fig. 4: Qualitative comparisons between baseline methods and our approach on the NeRF-DS real-world dataset. Results show that our method delivers superior rendering quality in the case of complex scene dynamics. Our method is capable of capturing finer details, preserving complex structure, and handling the dynamic scene elements more effectively than baseline methods such as SC-GS [16], D-MiSo [36], and 4DGS [43].

2. **Phase 2 (Geometric Focus):** Freezing the weights of the deformation networks and training only the canonical Gaussian parameters for $N_{\text{geo}} = 1$ iteration. The geometry refines its physical placement based on a predictable, unchanging temporal motion field.
3. **Phase 3 (Positional Annealing):** Upon reaching the defined threshold i_{anneal}, we execute the Positional Annealing schedule (Eq. 8) to freeze the canonical space progressively, and force the motion network to absorb all remaining spatiotemporal variance.

This strategy breaks the unstable feedback loop, allowing each component to converge on a more stable target.

Loss Function Our model is trained end-to-end with a composite loss function that combines an L1 loss and the D-SSIM loss. The full loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = (1 - \lambda_{\text{dssim}}) \cdot \mathcal{L}_1 + \lambda_{\text{dssim}} \cdot \mathcal{L}_{\text{dssim}} \tag{9}$$

4 Experiments

We evaluate our framework on monocular real-world datasets from NeRF-DS [46] and HyperNeRF [29]. We assess rendering quality using standard evaluation

Table 1: Quantitative comparison on NeRF-DS [46] dataset. We evaluate our method against state-of-the-art (SOTA) methods across all seven scenes. Our method achieves the **highest mean PSNR and SSIM**, demonstrating superior overall performance and robustness. Best results are highlighted in red.

Method	As			Basin			Bell			Cup		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3D-GS [18]	22.69	0.802	0.299	18.42	0.717	0.315	21.01	0.789	0.250	21.71	0.830	0.255
Deformable-GS [50]	26.22	0.881	0.184	19.65	0.791	0.189	25.08	0.842	0.162	24.66	0.887	0.157
SC-GS [16]	24.47	0.847	0.193	19.30	0.806	0.145	21.98	0.801	0.223	20.08	0.740	0.336
D-MiSo [36]	26.06	0.881	0.153	19.79	0.784	0.177	24.75	0.851	0.147	24.36	0.885	0.124
4DGS [43]	25.68	0.885	0.195	19.26	0.817	0.215	23.22	0.853	0.148	24.04	0.893	0.171
Motion-GS [56]	25.97	0.868	0.223	19.80	0.779	0.250	23.99	0.809	0.263	24.36	0.860	0.208
Ours	26.48	0.885	0.205	19.80	0.798	0.220	25.23	0.847	0.172	24.79	0.894	0.162

Method	Plate			Press			Sieve			Mean		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3D-GS [18]	16.14	0.697	0.409	22.89	0.816	0.290	23.16	0.820	0.225	20.29	0.782	0.292
Deformable-GS [50]	20.29	0.808	0.222	25.35	0.858	0.196	25.08	0.869	0.150	23.76	0.848	0.180
SC-GS [16]	19.92	0.814	0.222	24.89	0.870	0.166	25.12	0.896	0.136	22.25	0.824	0.203
D-MiSo [36]	20.50	0.805	0.204	25.71	0.864	0.141	26.12	0.885	0.113	23.90	0.851	0.151
4DGS [43]	18.55	0.760	0.288	24.13	0.822	0.251	22.88	0.830	0.213	22.54	0.837	0.212
Motion-GS [56]	20.41	0.793	0.276	25.86	0.861	0.243	25.62	0.847	0.219	23.71	0.831	0.240
Ours	20.86	0.823	0.221	25.96	0.877	0.219	25.86	0.878	0.177	24.14	0.857	0.197

metrics including PSNR, SSIM, and LPIPS. Our framework is implemented in PyTorch. The Base Motion Network is a 4-layer MLP with a hidden dimension of 128, conditioned on a low-frequency time embedding with L=4. The Residual Deformation network consists of a 2-layer Basis Predictor and an 8-layer Main Trunk with a hidden dimension of 256. We set the number of motion bases K=8. All the experiments are performed on a single NVIDIA A100 GPU. We compare MoBa-GS with 3DGS [18] and various state-of-the-art dynamic 3DGS methods, including Deformable-GS [50], 4DGS [43], SC-GS [16], D-MiSo [36], and Motion-GS [56]. We run the official public code using the default configuration provided to ensure fair evaluation. We are unable to compare with the motion factorization work, DynMF [19], as the code is not publicly released.

4.1 NeRF-DS Results

As shown in Tab. 1, MoBa-GS achieves a state-of-the-art result on the NeRF-DS dataset. Our method achieves the highest mean PSNR and SSIM across all seven scenes, demonstrating robustness and generalizability. While D-MiSo achieves a lower LPIPS score, recent literature [54] notes that VGG-based perceptual metrics reward over-smoothed artifacts on motion-blurred scenes. As evidenced by our higher PSNR and SSIM, MoBa-GS preserves accurate geometry and sharp boundaries rather than smoothed approximations. Our visual results (Fig. 4) confirm MoBa-GS avoids the motion blur present in D-MiSo.

From Fig. 4, our method reconstructs sharper details and suffers from fewer floating artifacts than prior work. For example, our model captures the fine

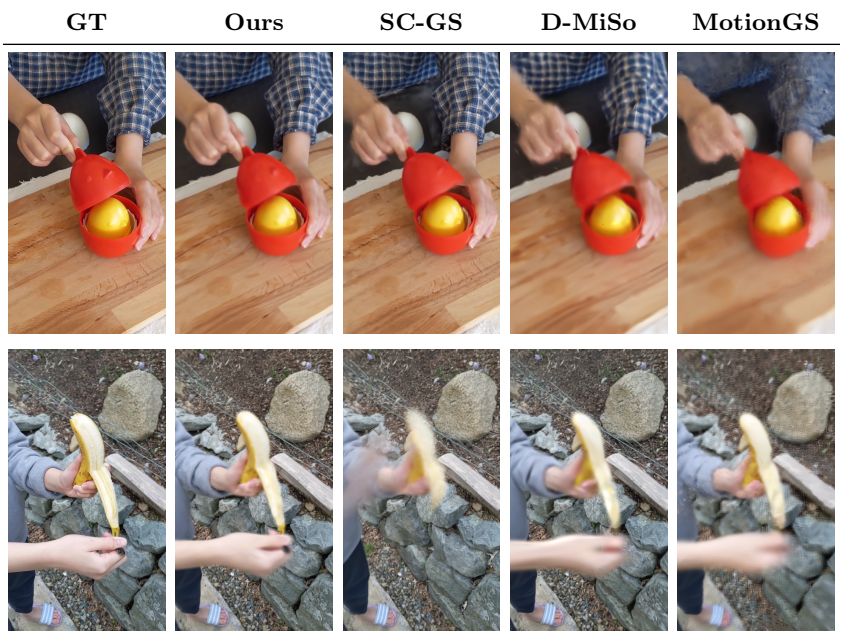

Fig. 5: Qualitative comparison on the HyperNeRF dataset [29]. Top: 'Chicken' scene, **Bottom:** 'Banana' scene. Our method consistently reconstructs finer details and sharper object boundaries in scenes with complex non-rigid motion.

details of the moving hand and objects in the 'Bell' and 'Cup' scenes, whereas the other methods produce noticeable motion blur.

4.2 HyperNeRF Results

We further validate our approach on the HyperNeRF dataset, which contains more extreme deformation. We follow the convention of [56] to omit LPIPS evaluation, as the dramatic camera motion present in this dataset can make the LPIPS metric less stable and harder to interpret consistently across methods. MoBa-GS again performed the best in the HyperNeRF dataset quantitatively, as shown in Tab. 2. Our method obtains the highest mean PSNR and SSIM in all testing scenes in HyperNeRF. The performance gains are more noticeable in 'Broom' and 'Banana' scenes, which contain thin structures and complex motion that are challenging for monolithic deformation models to represent accurately. In contrast, MoBa-GS preserves more details than the other baselines. Fig. 5 also shows superior results, preserving a higher level of detail and capturing the fast-moving objects more accurately than the other baseline methods. This further demonstrates the effectiveness of each module in our framework.

Table 2: Quantitative comparison on HyperNeRF [29]. MoBa-GS achieves SOTA performance (highest mean PSNR/SSIM), with significant gains in complex scenes like 'Banana'.

Method	3D Printer		Chicken		Broom		Banana		**Mean**	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SC-GS [16]	19.33	0.60	22.46	0.63	20.26	0.49	21.63	0.80	20.92	0.63
D-MiSo [36]	20.13	0.66	23.66	0.70	20.86	0.31	25.22	0.80	22.47	0.62
Deformable-GS [50]	20.18	0.64	22.68	0.60	20.54	0.34	24.85	0.78	22.06	0.59
MotionGS [56]	20.47	0.61	22.23	0.61	19.64	0.25	21.51	0.51	20.96	0.50
Ours	20.51	0.67	23.77	0.70	21.03	0.32	25.62	0.80	22.73	0.63

4.3 Computational Cost Analysis

Beyond increasing the reconstruction fidelity, another objective of our work is to create a representation that is easy to train and practical for real-time applications. Our factorized architecture and the training curricula synergistically achieve this objective, as shown in Tab. 3. MoBa-GS significantly reduces the training time compared to the baselines. Our model takes an average of 13.3 minutes to converge, leading to a 16× speedup over the leading baseline, D-MiSo [36]. This rapid convergence lowers the barrier for research and applications in dynamic 3D scenes. Furthermore, MoBa-GS extends this to fast inference and rendering speed, with an average of 163 FPS, up to 237 FPS for the 'Sieve' scene, as shown in Tab. 4. In addition, our model only uses an average of 39K Gaussians per scene, with a total framework taking up 11.3MB, making it a compact and robust solution.

4.4 Ablation Studies

We perform ablation studies on the full NeRF-DS dataset to evaluate the impact of each contribution. We remove each key component: Positional Annealing, Motion-Guided Densification, and the Spatially-Varying Motion Basis. Tab. 5 summarizes the mean results across all seven scenes.

Analysis of Positional Annealing. We first examine our complete model without Positional Annealing. The results show a performance drop in every metric. This shows that allowing the canonical geometry to be trained for the full duration would lead to overfitting on training views. Our annealing curriculum mitigates the geometric overfitting by stabilizing the canonical space in the later stage of training, achieving a better generalization.

Analysis of Motion-Guided Densification. Next, we ablate our densification strategy. This also results in a 0.31 dB PSNR drop. This confirms our hypothesis that the common densification strategies from previous works are time-agnostic and sub-optimal. By introducing a motion-aware feedback loop, our method allocates geometric capacity to the regions with significant motion that are difficult to reconstruct.

Table 3: Training Time Comparison. We compare the total training time of our framework against a leading baseline, D-MiSo [36], on the NeRF-DS dataset. Our method achieves an **order-of-magnitude speedup (over 16x faster on average)**, a direct result of our efficient, factorized architecture and training curricula. Time is recorded on a single NVIDIA A100 GPU.

Scene	D-MiSo [36]	**Ours (MoBa-GS)**
As	2h 47m	**13m**
Basin	2h 38m	**15m**
Bell	3h 14m	**18m**
Cup	4h 0m	**11m**
Plate	2h 47m	**9m**
Press	1h 54m	**12m**
Sieve	3h 52m	**15m**
Mean	∼3h 27m	∼**13m**

Table 4: Inference Performance and Model Compactness. MoBa-GS achieves real-time rendering speed (>163 FPS) and is noticeably lightweight (only ∼ 39k Gaussians per scene). Reported on a single NVIDIA A100 GPU.

Scene	FPS↑	#Gaussians (K)↓	Storage (MB)↓
As	181	28	8.73
Basin	146	41	11.96
Bell	103	61	16.63
Cup	150	37	10.92
Plate	149	35	10.42
Press	174	31	9.56
Sieve	237	37	10.88
Mean	163	39	11.3

Analysis of Spatially-Varying Motion Basis. Finally, we analyze our core architectural contribution. We replace our factorized motion basis model with "Base + Residual", a simple MLP structure. This ablation has the most significant performance degradation, with a massive drop of 0.66dB in PSNR. This further proves that simply decomposing the motion into coarse and fine is insufficient. The key to the SOTA performance of our model is that it provides a more structured representation that is fundamentally more efficient and expressive for complex non-rigid motion. These studies show that all three building blocks are essential and work in synergy to achieve the final result.

Sensitivity to Number of Motion Bases (K). The number of basis vectors, K, is a crucial hyperparameter that controls the rank of the learned local motion subspace and thus the expressive power of our residual deformation model. As shown in Tab. 5, extending our evaluation across $K \in \{2, 4, 8, 16, 32\}$ reveals a symmetric U-shaped capacity curve that peaks at $K = 8$. Lower ranks (e.g., $K = 2$) underfit the complex residual non-rigid motion, resulting in a 0.60

Table 5: Ablation Studies on NeRF-DS. Top: Removing core components significantly degrades performance, proving their synergistic importance. **Bottom:** Varying the basis count (K) reveals a symmetric U-shaped capacity curve, validating $K = 8$ as the optimal structural regularizer.

Model Configuration	PSNR↑	SSIM↑	LPIPS↓
MoBa-GS (Full, $K = 8$)	**24.14**	**0.857**	**0.197**
w/o Positional Annealing	23.87	0.850	0.210
w/o Motion Densification	23.83	0.849	0.207
w/o Motion Basis	23.48	0.841	0.221
Basis Count: $K = 2$	23.54	0.840	0.222
Basis Count: $K = 4$	23.62	0.847	0.216
Basis Count: $K = 16$	23.54	0.845	0.216
Basis Count: $K = 32$	23.44	0.841	0.218

dB PSNR drop. In contrast, higher ranks ($K \geq 16$) provide excess capacity that begins to overfit the training views, confirming symmetric capacity degradation. The model is structurally robust, with no basis collapse observed anywhere in $K \in [2, 32]$, and proving $K = 8$ is the empirical sweet spot. Further detailed analysis is provided in the Suppl. Mat.

5 Conclusion

In this work, we present MoBa-GS, a new paradigm for 4D reconstruction through a structural inversion of standard motion factorization. We shift from a global Time-Basis to a Spatially-Varying Motion-Basis, resolving the rank limitations of prior methods and enabling the robust representation of complex, spatially heterogeneous non-rigid dynamics. This architecture functions as an Implicit Neural Scaffold with the intrinsic constraints of the local basis, allowing high-fidelity recovery from random initialization, and thus eliminating the reliance on fragile SfM priors. Augmented by motion-guided densification and positional annealing, MoBa-GS achieves state-of-the-art performance across all monocular video benchmarks while remaining computationally lightweight. Enabled by time-invariant caching, our framework delivers real-time inference speed (>163 FPS), and significantly shorter training time (<18 minutes) than prior SOTA methods, providing a principled foundation for future work in 4D representation learning.

Acknowledgments

We acknowledge the GRS support from the School of Information Technology, as well as the support and provision of GPU resources from the HPC/APC team and its manager, Dr. Marcus.

References

1. Bae, J., Kim, S., Yun, Y., Lee, H., Bang, G., Uh, Y.: Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. arXiv preprint arXiv:2404.03613 (2024)
2. 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. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
4. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2023)
5. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Easi3r: Estimating disentangled motion from dust3r without training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9158–9168 (2025)
6. Cho, W.O., Cho, I., Kim, S., Bae, J., Uh, Y., Kim, S.J.: 4d scaffold gaussian splatting for memory efficient dynamic scene reconstruction. arXiv preprint arXiv:2411.17044 (2024)
7. Chu, W.H., Ke, L., Fragkiadaki, K.: Dreamscene4d: Dynamic multi-object scene generation from monocular videos. Advances in Neural Information Processing Systems **37**, 96181–96206 (2024)
8. Deng, J., Shi, P., Li, Q., Guo, J.: Dynasplat: Dynamic-static gaussian splatting with hierarchical motion decomposition for scene reconstruction. In: 2025 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2025)
9. 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. arXiv preprint arXiv:2312.00583 (2023)
10. 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. pp. 1–9 (2022)
11. Feng, H., Zhang, J., Wang, Q., Ye, Y., Yu, P., Black, M.J., Darrell, T., Kanazawa, A.: St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8503–8513 (2025)
12. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023)
13. Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5712–5721 (2021)
14. Gao, H., Li, R., Tulsiani, S., Russell, B., Kanazawa, A.: Monocular dynamic view synthesis: A reality check. Advances in Neural Information Processing Systems **35**, 33768–33780 (2022)
15. Hu, Y., Cheng, C., Yu, S., Guo, X., Wang, H.: Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. arXiv preprint arXiv:2511.19971 (2025)

16. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4220–4230 (2024)
17. Katsumata, K., Vo, D.M., Nakayama, H.: An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. arXiv preprint arXiv:2311.12897 (2023)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
19. Kratimenos, A., Lei, J., Daniilidis, K.: Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In: European Conference on Computer Vision. pp. 252–269. Springer (2025)
20. Kwak, S., Kim, J., Jeong, J.Y., Cheong, W.S., Oh, J., Kim, M.: Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting (2025), `https://arxiv.org/abs/2501.03714`, accessed 2025-06-30
21. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8508–8520 (2024)
22. Li, Z., Wang, Q., Cole, F., Tucker, R., Snavely, N.: Dynibar: Neural dynamic image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4273–4284 (2023)
23. 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. arXiv preprint arXiv:2312.11458 (2023)
24. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21136–21145 (2024)
25. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13–23 (2023)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
27. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
28. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021)
29. 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. arXiv preprint arXiv:2106.13228 (2021)
30. 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. pp. 10318–10327 (2021)
31. Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., Liu, Y.: Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16632–16642 (2023)

32. Stephen, L., Simon, T., Schwartz, G., Zollhoefer, M., Sheikh, Y., Saragih, J.: Mixture of volumetric primitives for efficient neural rendering. vol. 40, pp. 1–13. ACM New York, NY, USA (2021)
33. Sucar, E., Lai, Z., Insafutdinov, E., Vedaldi, A.: Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7295–7305 (2025)
34. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022)
35. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20675–20685 (2024)
36. Waczyńska, J., Borycki, P., Kaleta, J., Tadeja, S., Spurek, P.: D-miso: Editing dynamic 3d scenes using multi-gaussians soup. Advances in Neural Information Processing Systems **37**, 107865–107889 (2024)
37. Wang, C., Eckart, B., Lucey, S., Gallo, O.: Neural trajectory fields for dynamic novel view synthesis. arXiv preprint arXiv:2105.05994 (2021)
38. Wang, F., Tan, S., Li, X., Tian, Z., Song, Y., Liu, H.: Mixed neural voxels for fast multi-view video synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19706–19716 (2023)
39. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
40. Wang, L., Hu, Q., He, Q., Wang, Z., Yu, J., Tuytelaars, T., Xu, L., Wu, M.: Neural residual radiance fields for streamably free-viewpoint videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 76–87 (2023)
41. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19795–19806 (2023)
42. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4d reconstruction from a single video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9660–9672 (2025)
43. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20310–20320 (2024)
44. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9421–9431 (2021)
45. Xu, Z., Peng, S., Lin, H., He, G., Sun, J., Shen, Y., Bao, H., Zhou, X.: 4k4d: Real-time 4d view synthesis at 4k resolution. arXiv preprint arXiv:2310.11448 (2023)
46. Yan, Z., Li, C., Lee, G.H.: Nerf-ds: Neural radiance fields for dynamic specular objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8285–8295 (2023)
47. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., Joo, H.: Banmo: Building animatable 3d neural models from many casual videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2863–2873 (2022)

48. Yang, Z., Pan, Z., Zhu, X., Zhang, L., Feng, J., Jiang, Y.G., Torr, P.H.: 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. arXiv preprint arXiv:2412.20720 (2024)
49. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023)
50. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20331–20341 (2024)
51. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
52. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
53. Zhang, S., Ge, Y., Tian, J., Xu, G., Chen, H., Lv, C., Shen, C.: Pomato: Marrying pointmap matching with temporal motions for dynamic 3d reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5680–5689 (2025)
54. Zheng, H., Lin, Z., Lu, J., Cohen, S., Shechtman, E., Barnes, C., Zhang, J., Liu, Q., Amirghodsi, S., Zhou, Y., et al.: Structure-guided image completion with image-level and object-level semantic discriminators. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 7669–7681 (2024)
55. Zhou, K., Wang, Y., Chen, G., Beaudouin, G., Zhan, F., Liang, P.P., Wang, M.: Page-4d: Disentangled pose and geometry estimation for 4d perception. In: The Fourteenth International Conference on Learning Representations (2025)
56. Zhu, R., Liang, Y., Chang, H., Deng, J., Lu, J., Yang, W., Zhang, T., Zhang, Y.: Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. arXiv preprint arXiv:2410.07707 (2024)