MOTION-MOE-GS: DECOMPOSING LOCAL MOTION LAWS FOR DYNAMIC 3D GAUSSIAN RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex object motion is locally coherent but globally heterogeneous: units subject to the same structural constraints, part relations, or interactions share a local motion law, while different regions can obey different laws. Existing dynamic 3D Gaussian methods commonly decode all primitive updates with one shared time-conditioned function. Conditioning this function on location, appearance, or a global dynamic state changes its input, but does not allocate different motion laws to different primitives. We propose Motion-MoE-GS, which casts dynamic 3D Gaussian reconstruction as conditional decomposition and allocation of local motion laws. The method evolves a continuous scene-level dynamic state with a Neural ODE and combines it with each Gaussian's static state to form a local motion descriptor. A motion-conditioned router sparsely selects two of four deformation experts. Each expert represents a learned local motion law and predicts position, rotation, and scale updates. Per-scene evaluations on NVFI-Obj and DynMultiParts support conditional motion-law decomposition as an effective inductive bias for heterogeneous 3D Gaussian dynamics.

1 INTRODUCTION

Dynamic novel-view synthesis must preserve scene geometry and appearance while predicting how a scene changes across time. Neural radiance fields (Mildenhall et al., 2020) established a powerful implicit representation for static view synthesis, while 3D Gaussian Splatting (Kerbl et al., 2023) provides an efficient explicit alternative in which each anisotropic Gaussian carries position, appearance, rotation, scale, and opacity. Dynamic 3DGS methods must therefore determine a time-dependent geometric update for every primitive. This problem is more demanding than reconstructing isolated frames: a model must maintain consistent three-dimensional correspondences and preserve local structure while motion progresses beyond the observed temporal range.

Video pixels provide only indirect evidence about this evolution. Two-dimensional image changes conflate camera projection, occlusion, appearance, and three-dimensional motion, making image space a poor place to directly define reusable motion rules. Explicit Gaussian primitives provide a more suitable carrier: their changing positions, rotations, and scales induce a sequence of geometric states over time. We call this sequence a *Gaussian trajectory*. It is a learned geometric explanation of the video rather than a claim that a Gaussian is a physical particle with ground-truth correspondence.

This paper starts from a representation-level observation: *complex object motion is locally coherent but globally heterogeneous*. Units coupled by the same structural constraints, part relations, or interactions share a motion law locally, whereas rigid components, deformable components, and transition regions need not follow the same law. Complex motion should therefore be represented as a small collection of reusable local motion laws that are conditionally assigned to Gaussian primitives. These laws are not semantic parts, material labels, or predefined physical formulas; they are learned state-update patterns for the Gaussian trajectories that explain a scene over time.

Existing dynamic neural fields and dynamic Gaussian methods predominantly decode all primitive updates through one shared, time-conditioned deformation function. Neural scene flow fields (Li et al., 2021), D-NeRF (Pumarola et al., 2021), Nerfies (Park et al., 2021a), Deformable 3D Gaussians (Yang et al., 2024), and 4D Gaussian Splatting (Wu et al., 2024) enrich conditioning or improve

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

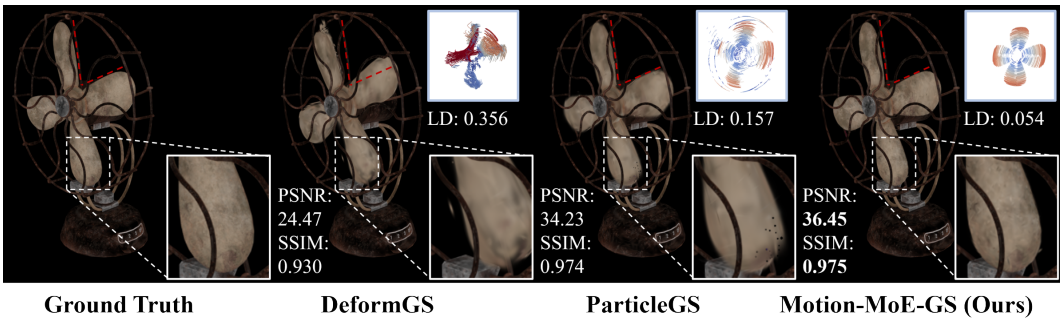


Figure 1: Qualitative extrapolation on the *Fan* scene. The small panels visualize projected Gaussian primitive trajectories. At a late unseen time step, Motion-MoE-GS produces a more coherent global trajectory pattern and preserves blade appearance and boundary details more faithfully than DeformGS and ParticleGS. Insets show the highlighted blade region.

efficiency, but retain this shared-decoder assumption. Input conditioning is not motion-law allocation: one function must still represent incompatible local state transitions in a common parameter space. ParticleGS (Quan et al., 2025) supplies a continuous scene-level dynamics latent space, but its final Gaussian-space decoder is likewise shared across primitives. This entanglement is particularly harmful for extrapolation, where small local transition errors accumulate into phase drift, blurred boundaries, or inconsistent geometry. As Figure 1 illustrates, a shared deformation model can yield irregular primitive trajectories during extrapolation; our decomposition preserves a more coherent global trajectory pattern and sharper geometry at the same unseen time.

We introduce **Motion-MoE-GS**, which represents dynamic Gaussian motion through jointly learned local motion-law experts. A set encoder summarizes the canonical Gaussian configuration, and a Neural ODE evolves a continuous scene-level dynamic context. Each Gaussian combines this context with its static state, and a motion-conditioned router selects a sparse Top-2 mixture among four homogeneous deformation experts. The selected experts predict position, rotation, and scale updates, so routing and experts express local responses to the common evolving scene state. This design lies between two undesirable extremes: a single universal decoder that entangles all responses and a fully independent decoder for every Gaussian that cannot share recurring motion structure.

Our contributions are threefold:

- We formulate dynamic 3D Gaussian reconstruction as the conditional decomposition and allocation of reusable local motion laws, expressed through the trajectories of explicit Gaussian primitives rather than inferred directly in image space.
- We propose Motion-MoE-GS, which combines continuous scene-level dynamics with motion-conditioned sparse routing over jointly trained, homogeneous deformation experts.
- We demonstrate improved reconstruction and extrapolation on NVFI-Obj and DynMultiParts, with per-scene comparisons, controlled ablations, and qualitative analyses that expose more coherent extrapolated Gaussian trajectories and geometry.

2 Related Work

2.1 Dynamic Neural Fields and Canonical Deformation

Neural radiance fields (Mildenhall et al., 2020) represent a scene with a continuous coordinate-conditioned function. For dynamic scenes, Neural Scene Flow Fields (Li et al., 2021) jointly model geometry, appearance, and scene flow. D-NeRF (Pumarola et al., 2021) introduces a canonical-space deformation field, while Nerfies (Park et al., 2021a) and HyperNeRF (Park et al., 2021b) extend canonical deformation to elastic and topologically varying scenes. These methods demonstrate that time-varying rendering benefits from a canonical representation, but their deformation is generally decoded through a shared function.

Factorized dynamic representations improve the efficiency of this paradigm. TiNeuVox (Fang et al., 2022) uses time-aware neural voxels, and HexPlane (Cao & Johnson, 2023) factorizes the dynamic volume into plane features. Such designs reduce the cost of representing spatiotemporal signals, but are not designed to explicitly decompose local Gaussian state-update rules. Motion-MoE-GS retains the canonical-deformation perspective while changing the representation of motion: different local responses are conditionally assigned to a small set of learned deformation experts.

2.2 Dynamic Gaussian Representations and Motion Extrapolation

Dynamic 3D Gaussians (Luiten et al., 2024) associates Gaussian primitives across frames using tracking and rendering objectives. Deformable 3D Gaussians (Yang et al., 2024) and 4D Gaussian Splatting (Wu et al., 2024) apply canonical deformation to explicit Gaussian primitives, and Grid4D (Xu et al., 2024) improves dynamic Gaussian rendering through decomposed hash encoding. These methods primarily pursue accurate dynamic reconstruction by learning an efficient time-conditioned deformation representation.

Several works target motion-aware extrapolation. GaussianPrediction (Zhao et al., 2024) predicts future Gaussian configurations, whereas NVFi (Li et al., 2023), TRACE (Li et al., 2025a), and FreeGaVe (Li et al., 2025b) introduce velocity- or physics-aware representations. ParticleGS (Quan et al., 2025) learns a scene-level dynamics latent space with a Neural ODE and decodes Gaussian updates from that state. Our method shares the goal of stable extrapolation, but addresses a separate representation issue: a continuous global state alone does not determine how every local Gaussian should respond. We use it to condition the allocation of local motion laws rather than decode all updates with one shared Gaussian-space function.

2.3 Continuous Dynamics and Mixture-of-Experts

Neural ODEs (Chen et al., 2018) parameterize continuous-time state evolution, and latent ODEs (Rubanova et al., 2019) show how such evolution can be learned in a latent state space from observations. We use this machinery only for the shared scene-level dynamic context; it does not by itself separate local motion responses.

Sparse mixture-of-experts models (Shazeer et al., 2017; Fedus et al., 2022) increase conditional capacity by routing each input to a small subset of functions. Recent work also studies mixture-of-experts designs for dynamic Gaussian splatting (Jin et al., 2026), including the combination of heterogeneous deformation priors and independently trained dynamic-Gaussian experts. In contrast, Motion-MoE-GS jointly trains a homogeneous bank of primitive-level deformation experts within one scene. Its router is conditioned on both local Gaussian structure and continuously evolved global dynamics, with the explicit purpose of representing reusable local motion laws rather than selecting among pre-existing deformation models.

3 Method

Figure 2 summarizes Motion-MoE-GS. Rather than extracting motion laws directly from image pixels, we express motion through the time-varying geometric states of explicit Gaussian primitives. The rendering objective supervises these latent Gaussian trajectories; local motion laws are then represented as reusable functions that generate their state updates.

3.1 Gaussian State Trajectories and Continuous Dynamics

Let $\mathcal{G}^0 = \{g_i^0\}_{i=1}^N$ be the canonical Gaussian set. Each Gaussian has static state $f_i = [x_i, c_i, q_i, s_i, a_i]$, containing position, DC color coefficient, rotation quaternion, scale, and opacity. At time t, its geometric state is $\theta_{i,t} = [x_{i,t}, q_{i,t}, s_{i,t}]$. The sequence $t \mapsto \theta_{i,t}$ is a latent Gaussian trajectory: it is learned to explain the observed images, rather than assumed to be a ground-truth physical particle trajectory.

As shown in Figure 2(a), a local encoder produces $l_i = \phi_{\text{local}}(f_i)$ for every Gaussian. We aggregate local KNN groups using cross-attention and self-attention to obtain a scene-level initial state z_0. A

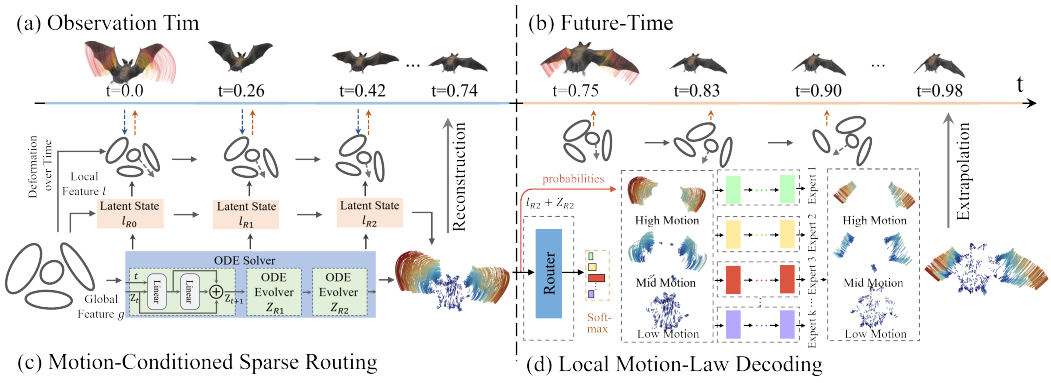


Figure 2: Overview of Motion-MoE-GS. (a) At observation times, local Gaussian descriptors and a scene-level feature initialize continuous latent dynamics. (b) The evolved state is evaluated at future times for extrapolation. (c) A motion-conditioned router uses the local descriptor and evolved global state to sparsely assign deformation experts. (d) Homogeneous experts decode local motion-law responses; their weighted aggregation updates Gaussian geometry for future-time rendering. Colors illustrate emergent motion responses rather than manually specified semantic groups.

Neural ODE evolves this state continuously,

$$\frac{dz(\tau)}{d\tau} = F_{\text{ODE}}(z(\tau), \tau), \qquad z_t = \text{ODESolve}(z_0, -0.01, t). \tag{1}$$

The ODE state summarizes the global dynamic phase shared by the scene. It can be evaluated at both observation and unseen future times, as illustrated in Figure 2(a–b), without discretizing motion into a fixed set of frames.

3.2 Motion-Conditioned Sparse Routing

The global state alone does not determine how every primitive responds: different local structures can undergo different geometric changes at the same dynamic phase. We therefore form a primitive-wise motion descriptor $m_{i,t} = [l_i, z_t]$ and use it to assign motion laws. Specifically, the router produces an expert distribution

$$p_{i,t} = \text{softmax}\left(W_r[l_i, z_t]/T\right), \qquad \alpha_{i,t} = \text{Top2Normalize}(p_{i,t}), \tag{2}$$

where T is a temperature. Top2Normalize retains only the two highest probabilities and renormalizes them. This implements the sparse routing in Figure 2(c): the allocation is conditioned on both the Gaussian's local structure and the current scene dynamics, not on image-space pixels or predefined part labels.

3.3 Local Motion-Law Decoding and Rendering

We use $E = 4$ deformation experts with identical decoder architectures but independent parameters. Expert D_e maps the same motion descriptor to a candidate Gaussian update. The final update is a sparse mixture,

$$\Delta\theta_{i,t} = \sum_{e=1}^{E} \alpha_{i,t}^{(e)} D_e(m_{i,t}), \qquad \|\alpha_{i,t}\|_0 \le 2, \tag{3}$$

and $\theta_{i,t} = \theta_i^0 \oplus \Delta\theta_{i,t}$ applies the predicted translation, rotation, and scale changes to the canonical Gaussian. As depicted in Figure 2(d), experts are jointly optimized and specialize through routing; they are not pretrained deformation models or fixed semantic components. Their weighted output generates the future Gaussian configuration in Figure 2(b), which is rendered using differentiable Gaussian rasterization.

We optimize the rendered images with a reconstruction loss and add a load-balancing term to discourage persistent expert collapse:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}}. \tag{4}$$

The rendering term supervises the final Gaussian configuration through the available images; it does not provide expert identities, object-part masks, or particle trajectories. Consequently, an expert is useful only when its state-update response improves rendered reconstruction across the primitives routed to it. The balancing term acts on router utilization rather than prescribing a semantic partition. This is important for the proposed interpretation: the learned motion laws are reusable geometric update patterns discovered from rendering supervision, not manually named object parts or externally supplied physical rules.

4 EXPERIMENTS

4.1 DATASETS AND PROTOCOL

We evaluate on NVFI-Obj and DynMultiParts using the same dataset splits and reconstruction/extrapolation protocol as the published comparison (Quan et al., 2025). For NVFI-Obj, frames with $\mathrm{fid} < 0.75$ are reconstruction frames and frames with $\mathrm{fid} \geq 0.75$ are extrapolation frames. For DynMultiParts, reconstruction uses 552 training frames and extrapolation uses 210 test frames. We report PSNR, SSIM, and LPIPS, which are available for all compared methods.

4.2 IMPLEMENTATION AND COMPARISON CONTROLS

For our model and all controlled variants, we fix the canonical Gaussian initialization, data split, rendering resolution, optimizer, and densification schedule. Each variant is trained for 50k iterations, with the same warm-up schedule and densification interval (iterations 500–20k); the only difference is the component named in Table 3. The router and each expert decoder use hidden width 128, except for the parameter-matched single-expert control. We use four homogeneous experts and Top-2 routing in the complete model. For the main comparisons, we retain the exact dataset protocols and reported baseline scores from the published ParticleGS evaluation (Quan et al., 2025); thus the tables compare methods under common scene definitions and reconstruction/extrapolation splits rather than mixing incompatible evaluation settings.

4.3 BASELINES

We compare with two dynamic radiance-field baselines, HexPlane (Cao & Johnson, 2023) and TiNeuVox (Fang et al., 2022); four dynamic Gaussian reconstruction methods, DeformGS (Yang et al., 2024), Grid4D (Xu et al., 2024), GaussianPrediction (Zhao et al., 2024), and ParticleGS (Quan et al., 2025); and three dynamics-aware methods, NVFi (Li et al., 2023), TRACE (Li et al., 2025a), and Free-GaVe (Li et al., 2025b). These are precisely the methods reported in the source ParticleGS comparison under the same datasets and evaluation protocol.

4.4 QUANTITATIVE RESULTS

Table 1 consolidates all six NVFI-Obj scenes, and Table 2 consolidates all four DynMultiParts scenes. Each table is divided into scene-pair panels but contains no scene average. We retain the per-scene protocol and all baselines reported by ParticleGS (Quan et al., 2025). Bold, underline, and italic text denote first, second, and third place, respectively, for each scene and metric.

The breakdown makes performance differences visible rather than hiding them in an aggregate. On NVFI-Obj, Motion-MoE-GS achieves the best extrapolation PSNR and SSIM on *bat*, *telescope*, *shark*, and *whale*, and the best reconstruction PSNR and SSIM on *fan*. The difficult *fallingball* scene remains a failure case relative to velocity-field methods. On DynMultiParts, Motion-MoE-GS is first in reconstruction PSNR on all scenes and in extrapolation PSNR on three scenes; it is second on *hyperbolic*. It attains the best reconstruction SSIM on every scene. The LPIPS columns reveal complementary perceptual trade-offs and include the same per-scene ranking protocol. We omit L_1 because published baselines report LPIPS instead.

The per-scene presentation is deliberate. A single average would obscure that gains are strongest where the scene contains repeated but distinct local responses, such as the articulated *bat*, the rotating *fan*, and the deformable *shark*. Conversely, *fallingball* exposes a boundary of the method: its fast, localized ball-and-hoop interaction is not uniformly improved by the proposed representation.

We therefore do not interpret the method as a replacement for explicit velocity or physical priors. Instead, the results support the narrower claim that conditional decomposition of Gaussian state updates is beneficial when a shared decoder must explain heterogeneous local changes.

The metric pattern also matters. PSNR and SSIM quantify image-level fidelity and structural agreement, while LPIPS captures a complementary perceptual distance; no single metric alone establishes superior dynamics. We report all three scores per scene and retain cases where another method has a better LPIPS value. This makes the evidence auditable: our principal result is improved reconstruction and, especially, future-time geometric consistency on most scenes, rather than an unsupported claim of dominance on every scene and metric.

4.5 Qualitative Analysis

Figure 3 compares late-time extrapolation on *Bat*. HexPlane produces a severely blurred silhouette, while DeformGS and ParticleGS retain the global shape but lose fine wing and hand structures. Motion-MoE-GS most closely follows the ground truth in both highlighted regions: it preserves the elongated wing boundary and the articulated hand geometry. Figure 4 provides a complementary case on *Shark*, where the target motion requires preserving a continuous deformable body surface. Our method maintains a sharper body contour and more faithful surface and jaw details than the compared methods. Together with the projected trajectories in Figure 1, these comparisons indicate that modeling local motion laws improves both the global regularity of extrapolated Gaussian motion and the local fidelity of rendered geometry.

4.6 Ablation Studies

We evaluate each design choice on both NVFI-Obj and DynMultiParts. Table 3 reports equally weighted means over the six NVFI-Obj scenes and the four DynMultiParts scenes, respectively. All variants use the same split, initial Gaussians, optimizer, densification schedule, training budget, and evaluation protocol; only the named component is changed. We report PSNR, SSIM, and LPIPS for reconstruction and extrapolation.

Expert capacity. The full four-expert model provides the strongest extrapolation PSNR on both datasets. On DynMultiParts, it exceeds $E = 2$ and $E = 8$ by 0.2434 dB and 0.8002 dB, respectively; on NVFI-Obj, the corresponding gains are 3.3815 dB and 1.8837 dB. Although $E = 2$ is marginally higher in DynMultiParts reconstruction PSNR (47.2057 versus 47.1838), it is less effective for future motion. These results show that the gain is not monotonic in expert count and identify four experts as an effective shared capacity across both datasets.

Sparse routing. Top-2 routing is consistently preferable to both hard single-expert routing and dense averaging for extrapolation. It improves PSNR over Top-1 by 3.9678 dB on DynMultiParts and 3.0550 dB on NVFI-Obj; relative to dense routing, the gains are 1.5373 dB and 5.4769 dB. The accompanying SSIM and LPIPS trends are consistent. Thus, one expert is too restrictive for complex local changes, whereas evaluating all experts and averaging their responses weakens motion specialization.

Local and global motion conditions. Removing either condition consistently harms extrapolation. The complete model improves over Local-only and Global-only by 1.0427 dB and 1.8893 dB on DynMultiParts, and by 2.4847 dB and 3.8630 dB on NVFI-Obj. It also achieves the strongest extrapolation SSIM and LPIPS in both dataset means. This supports complementary roles for the two conditions: the global dynamic state captures the temporal evolution of the scene, while local Gaussian features distinguish the local deformation to be produced under that state.

Continuous evolution and balancing. Replacing the Neural ODE with a time-conditioned MLP decreases extrapolation PSNR by 1.0258 dB on DynMultiParts and 2.2547 dB on NVFI-Obj, while worsening SSIM and LPIPS on both means. Removing the balance loss likewise reduces extrapolation PSNR by 0.4107 dB and 3.2910 dB, respectively, and degrades the remaining extrapolation metrics. Continuous global evolution and balanced expert utilization are therefore both important for stable future prediction.

Table 1: Full per-scene comparison on NVFI-Obj at the source resolution. Each panel reports a different scene pair; no scene average is used. Ours denotes Motion-MoE-GS. Bold and underline indicate the best and second-best result, respectively, for each scene and metric. LPIPS↓ is lower-is-better.

	Falling Ball						Bat					
	Reconstruction			Extrapolation			Reconstruction			Extrapolation		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	19.754	0.816	0.215	20.166	0.906	0.130	22.099	0.953	0.080	22.422	0.954	0.061
TiNeuVox	24.323	0.894	0.167	20.474	0.940	0.098	17.611	0.946	0.076	17.139	0.947	0.070
DeformGS	17.736	0.830	0.156	26.384	0.945	0.060	47.034	0.996	0.004	27.109	0.944	0.034
Grid4D	19.119	0.821	0.153	24.926	0.925	0.069	**47.443**	**0.997**	0.004	29.551	0.953	0.290
NVFi	34.900	0.969	0.068	27.013	0.966	0.074	24.648	0.970	0.044	25.529	0.973	0.040
GaussianPred.	24.549	0.910	0.154	16.790	0.830	0.238	43.290	**0.997**	**0.002**	23.495	0.952	0.075
ParticleGS	38.275	**0.991**	**0.010**	38.067	**0.990**	**0.013**	43.232	**0.997**	0.003	36.348	0.984	**0.011**
Ours	**38.470**	0.983	0.020	**38.788**	0.979	0.024	46.430	**0.997**	0.008	**38.270**	**0.986**	0.016

	Fan						Telescope					
	Reconstruction			Extrapolation			Reconstruction			Extrapolation		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	18.424	0.842	0.190	19.998	0.887	0.106	21.814	0.917	0.149	23.667	0.931	0.082
TiNeuVox	15.775	0.855	0.189	21.307	0.919	0.081	23.088	0.949	0.084	20.801	0.925	0.071
DeformGS	38.083	0.973	0.026	24.473	0.930	0.047	38.435	0.992	0.005	23.186	0.934	0.037
Grid4D	37.031	0.967	0.030	26.720	0.941	0.044	38.488	0.992	0.006	25.696	0.942	0.033
NVFi	27.748	0.953	0.059	27.347	0.957	0.054	26.889	0.955	0.051	26.861	0.955	0.053
GaussianPred.	34.864	0.976	**0.020**	20.747	0.935	0.054	36.191	0.992	**0.004**	22.018	0.939	0.043
ParticleGS	38.852	0.979	0.021	34.235	0.974	**0.023**	**38.770**	**0.993**	**0.004**	38.294	0.983	**0.004**
Ours	**39.339**	**0.980**	0.033	**36.454**	**0.975**	0.029	38.740	**0.993**	0.007	**41.997**	**0.996**	**0.004**

	Shark						Whale					
	Reconstruction			Extrapolation			Reconstruction			Extrapolation		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	22.736	0.956	0.064	24.180	0.962	0.041	24.684	0.976	0.049	24.075	0.964	0.049
TiNeuVox	19.592	0.951	0.058	16.683	0.933	0.096	21.281	0.969	0.052	20.671	0.945	0.063
DeformGS	40.492	**0.995**	0.008	29.159	0.966	0.022	42.475	**0.996**	**0.004**	26.666	0.959	0.027
Grid4D	**40.752**	**0.995**	**0.007**	30.904	0.966	0.021	**42.666**	**0.996**	**0.004**	28.246	0.967	0.020
NVFi	31.374	0.982	0.035	28.649	0.979	0.033	30.778	0.984	0.025	25.529	0.977	0.029
GaussianPred.	40.015	**0.995**	0.010	30.280	0.971	0.032	39.540	**0.996**	0.005	25.596	0.965	0.046
ParticleGS	40.322	**0.995**	0.008	37.475	0.983	**0.010**	39.241	0.995	0.005	34.236	0.983	**0.013**
Ours	40.208	**0.995**	0.011	**40.878**	**0.992**	**0.010**	39.674	0.994	0.010	**34.544**	**0.986**	0.021

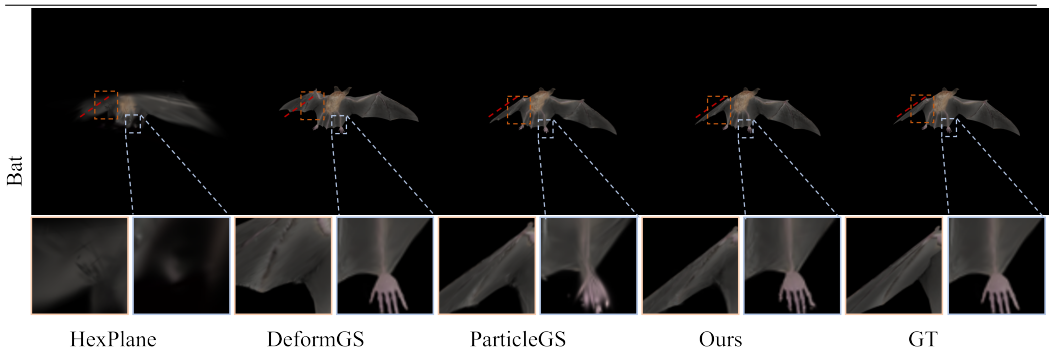


Figure 3: Qualitative extrapolation on the *Bat* scene. The dashed boxes and enlarged insets highlight a wing boundary and articulated hand region at a late unseen time. Motion-MoE-GS best preserves the fine structures and their alignment with the ground truth.

Taken together, the ablations test the proposed representation in a sequence rather than treating MoE as an isolated architectural addition. The capacity controls test whether multiple reusable update

Table 2: Full per-scene comparison on DynMultiParts at the source resolution. Each panel reports a different scene pair; no scene average is used. Ours denotes Motion-MoE-GS. Bold and underline indicate the best and second-best result, respectively, for each scene and metric. LPIPS↓ is lower-is-better.

	Folding Chair						Satellite					
	Reconstruction			Extrapolation			Reconstruction			Extrapolation		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DeformGS	41.562	0.994	0.009	18.194	0.847	0.123	35.884	0.988	0.010	29.454	0.980	0.014
TRACE	38.286	0.993	0.011	28.724	0.974	0.023	36.647	0.990	0.010	33.612	0.983	0.009
FreeGave	39.086	0.994	0.010	26.043	0.951	0.037	37.155	0.991	0.010	34.476	**0.989**	**0.007**
ParticleGS	41.620	0.994	0.009	31.512	**0.977**	**0.015**	38.697	0.991	0.010	34.562	0.981	0.017
Ours	**42.258**	**0.996**	**0.006**	**32.646**	0.972	0.029	**45.788**	**0.998**	**0.003**	**36.704**	0.988	0.013

	Stove						Hyperbolic					
	Reconstruction			Extrapolation			Reconstruction			Extrapolation		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DeformGS	40.169	0.988	0.027	30.694	0.953	0.030	36.105	0.983	0.031	33.614	0.987	0.015
TRACE	36.201	0.990	0.028	33.869	0.989	0.016	33.942	0.985	0.036	37.628	0.990	0.012
FreeGave	35.578	0.989	0.028	32.291	0.989	0.021	32.747	0.985	0.034	**41.294**	**0.995**	**0.011**
ParticleGS	39.115	0.989	0.024	40.069	**0.991**	**0.013**	34.998	0.984	0.032	38.399	0.990	0.012
Ours	**49.121**	**0.997**	**0.014**	**40.672**	0.984	0.033	**51.568**	**0.999**	**0.002**	39.879	0.992	0.014

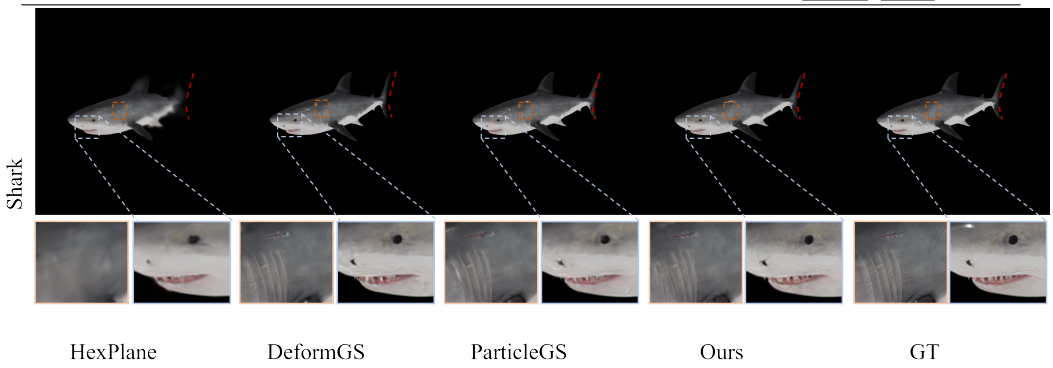


Figure 4: Qualitative extrapolation on the *Shark* scene. The enlarged insets assess body-surface continuity and jaw details at a late unseen time. Motion-MoE-GS most closely matches the ground-truth contour and retains the fine local structures.

functions are needed; Top-1 and dense routing test how these functions should be allocated; Local-only and Global-only test the descriptor used for allocation; and the ODE and balance controls test whether the allocation remains meaningful over time. The strongest and most consistent differences occur in extrapolation, which is the setting where a shared decoder must repeatedly extend local state transitions beyond its observed frames.

5 Conclusion

Motion-MoE-GS is grounded in the observation that complex motion is locally coherent but globally heterogeneous. Shared decoders condition one function on time and global state; this is not equivalent to allocating local motion laws. By evolving a continuous scene state and sparsely allocating learned experts, Motion-MoE-GS produces Gaussian-specific deformation responses. Per-scene results and controlled ablations show that the benefit is most pronounced for future-time prediction, while also exposing scene-dependent and metric-dependent trade-offs. This framing makes the contribution a representation principle for dynamic Gaussian state updates, not merely the insertion of a generic MoE block.

Table 3: Ablation study using equal-scene means on DynMultiParts and NVFI-Obj. Each block averages all four DynMultiParts scenes or all six NVFI-Obj scenes, respectively. Bold and underline indicate the best and second-best mean, respectively, within each dataset, stage, and metric. LPIPS↓ is lower-is-better.

Method	Reconstruction			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DynMultiParts (mean over 4 scenes)						
$E = 1$	44.5615	0.9963	0.0090	35.8274	0.9821	0.0268
$E = 2$, Top-2	**47.2057**	<u>0.9971</u>	0.0073	<u>37.2319</u>	0.9832	<u>0.0233</u>
$E = 8$, Top-2	46.9208	0.9968	0.0077	36.6751	0.9783	0.0297
Top-1	40.8842	0.9919	0.0149	33.5075	0.9730	0.0367
Dense routing	46.6028	0.9963	**0.0063**	35.9380	0.9805	0.0254
Local-only	45.9238	0.9967	<u>0.0065</u>	36.4326	0.9810	0.0255
Global-only	46.0589	<u>0.9971</u>	**0.0063**	35.5860	0.9807	0.0267
No balance	44.1028	0.9962	0.0069	37.0646	<u>0.9833</u>	0.0227
MLP evolver	45.5490	0.9965	**0.0063**	36.4495	0.9830	0.0246
Ours ($E = 4$, **Top-2**)	<u>47.1838</u>	**0.9975**	**0.0063**	**37.4753**	**0.9840**	**0.0223**
NVFI-Obj (mean over 6 scenes)						
$E = 1$	38.9060	0.9879	0.0173	35.5584	0.9821	0.0203
$E = 2$, Top-2	<u>40.3205</u>	0.9895	<u>0.0150</u>	35.1070	0.9792	0.0195
$E = 8$, Top-2	40.1175	0.9892	0.0158	36.6048	0.9842	0.0185
Top-1	38.9686	0.9880	0.0173	35.4335	0.9819	0.0224
Dense routing	40.0488	0.9892	0.0153	33.0116	0.9746	0.0230
Local-only	39.6871	0.9897	0.0151	36.0038	<u>0.9850</u>	<u>0.0176</u>
Global-only	38.8598	0.9883	0.0165	34.6255	0.9797	0.0213
No balance	40.3003	0.9897	0.0151	35.1975	0.9803	0.0190
MLP evolver	40.1161	<u>0.9898</u>	<u>0.0150</u>	<u>36.2338</u>	0.9828	0.0187
Ours ($E = 4$, **Top-2**)	**40.4768**	**0.9903**	**0.0148**	**38.4885**	**0.9857**	**0.0173**

Limitations and Scope

The experts represent learned geometric update patterns, not identified physical laws. They are supervised only through rendered observations and can therefore explain the video with a useful latent decomposition without recovering forces, masses, material parameters, or uniquely defined physical particles. This scope is intentional: the method targets prior-free dynamic reconstruction and extrapolation from videos, rather than simulation under known governing equations. It should consequently be combined with explicit physical constraints when conservation, collision response, or controllable intervention is required.

The *fallingball* results further delimit the claim. Fast localized interactions and severe visibility changes can require stronger velocity, collision, or contact priors than a scene-level latent dynamics state provides. In addition, a small expert bank is appropriate only when local state updates recur across a scene; highly non-repeating motion may not benefit from reusable experts. The proposed decomposition should therefore be read as an inductive bias for heterogeneous but structured dynamic scenes. Future work can incorporate physical state variables, uncertainty-aware routing, and richer temporal supervision while retaining the same conditional motion-law formulation.

AI use statement

We used generative AI tools for language drafting and editing. We did not use generative AI tools to generate experimental results, fabricate citations, or make unsupported scientific claims. The authors take responsibility for the final content.

Ethics statement

This work studies dynamic scene reconstruction from established research datasets. We use data only under their applicable licenses and terms. The method does not require human-subject intervention or collect new personal data.

REFERENCES

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

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

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

In-Hwan Jin, Hyeongju Mun, Joonsoo Kim, Kugjin Yun, and Kyeongbo Kong. On the design of mixture-of-experts for dynamic gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3712736.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Jinxi Li, Ziyang Song, and Bo Yang. Nvfi: Neural velocity fields for 3d physics learning from dynamic videos. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34723–34751, 2023.

Jinxi Li, Ziyang Song, and Bo Yang. Trace: Learning 3d gaussian physical dynamics from multi-view videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Jinxi Li, Ziyang Song, Siyuan Zhou, and Bo Yang. Free-gave: 3d physics learning from dynamic videos by gaussian velocity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12433–12443, 2025b.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6508, 2021.

Michiel Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking, learning and rendering dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision*, 2021a.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. In *ACM SIGGRAPH Asia*, 2021b.

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

Jinsheng Quan, Qiaowei Miao, Yichao Xu, Zizhuo Lin, Ying Li, Wei Yang, Zhihui Li, and Yawei Luo. Particlegs: Learning neural gaussian particle dynamics from videos for prior-free physical motion extrapolation. *arXiv preprint arXiv:2505.20270*, 2025.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qixing Huang, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 123787–123811, 2024.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Boming Zhao, Yuan Li, Ziyu Sun, Lin Zeng, Yujun Shen, Rui Ma, Yinda Zhang, Hujun Bao, and Zhaopeng Cui. Gaussianprediction: Dynamic 3d gaussian prediction for motion extrapolation and free view synthesis. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024.

Appendix A: Additional Qualitative Results

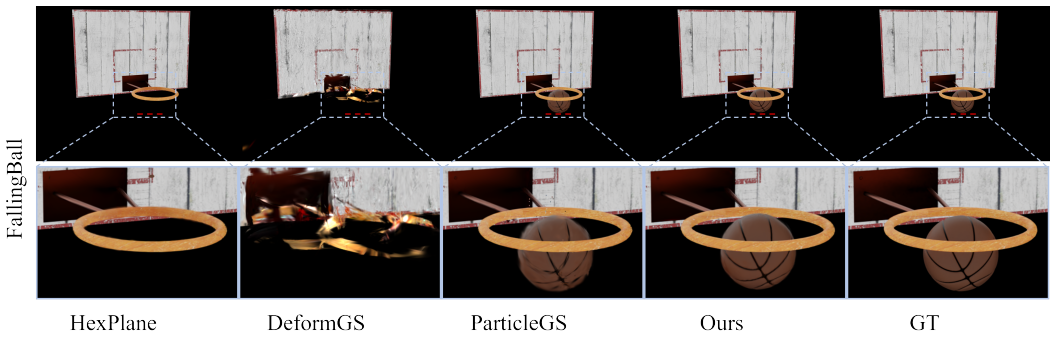


Figure 5: Additional qualitative extrapolation on the challenging *Falling Ball* scene. The enlarged region assesses the basketball geometry and its alignment with the hoop at a late unseen time. HexPlane fails to recover the ball, DeformGS produces severe artifacts, and ParticleGS yields an incomplete ball geometry. Motion-MoE-GS most closely matches the ground truth while preserving the ball contour and hoop alignment.

Reproducibility Details

The manuscript specifies the representation, data splits, metrics, and numerical values used in this draft. The final submission will provide complete implementation details, fixed evaluation scripts, random seeds, hardware configuration, and anonymized supplementary material where permitted.