

MoRE-GS: Motion-Regime Experts for Future Gaussian Splatting

Shima Salehi Joshua Peebles

Department of Electrical and Computer Engineering

Texas A&M University

College Station, Texas, USA

shima.salehi@tamu.edu

jpeeples@tamu.edu

Abstract

Future dynamic novel-view synthesis requires extrapolating multiple motion regimes within the same scene. A single dynamics model may not simultaneously represent stationary regions, coordinated motion, local interactions, and articulated trajectories. We introduce Motion Regime Expert Routing for Gaussian Splatting (MoRE-GS), a causal, render-aware framework for forecasting dynamic 3D Gaussian scenes. MoRE-GS freezes a Gaussian representation reconstructed exclusively from the observed temporal prefix and forecasts its control-keypoint motion using four complementary experts: persistence, a coefficient-space Transformer, a graph convolutional network, and a structured Transformer with rigid and non-rigid motion constraints. A compact routing table assigns expert mixtures across soft motion groups and forecast-horizon bins, while a persistence gate suppresses unreliable extrapolation. Across D-NeRF and HyperNeRF scenes, MoRE-GS achieves improved performance against other methods and these results demonstrate that renderer-calibrated composition of complementary motion predictors improves causal dynamic Gaussian forecasting across heterogeneous motion regimes.

1. Introduction

Forecasting future 3D scene states is important for predictive perception and planning in autonomous driving, closed-loop evaluation of safety-critical autonomy systems, and model-based robotic navigation, where an agent must anticipate scene evolution before future observations become available [2, 5, 21, 24]. This problem is challenging because motion in dynamic scenes is both spatially heterogeneous and temporally varying. Background regions may remain stationary, objects may follow smooth or periodic trajectories, and articulated or deformable components may exhibit lo-

cally coupled motion [14, 16]. Prediction reliability also changes with the forecast horizon, defined as the temporal distance from the last observed frame: recent motion can support accurate near-term extrapolation, whereas trajectory, phase, and multimodal uncertainty accumulate farther into the future [2]. Accurate future synthesis must therefore adapt the motion model to both spatial region and forecast horizon.

3D Gaussian Splatting (3DGS) [6] provides an explicit, differentiable, and efficiently rendered representation for novel-view synthesis. Dynamic extensions have achieved high-quality reconstruction by optimizing Gaussian geometry, appearance, and deformation over observed timestamps [11, 12, 20, 22]. Future novel-view synthesis presents a fundamentally different problem: the model must extrapolate open-loop to timestamps beyond the observation interval while rendering from novel camera views. Moreover, the complete future suffix must remain unavailable during scene reconstruction, motion estimation, model selection, checkpoint selection, and early stopping.

Existing forecasting approaches model motion using velocity fields, keypoint graphs, continuous latent dynamics, or coherent motion groups [8, 10, 19, 23]. Each formulation provides a useful inductive bias, but existing methods typically rely on a single forecasting family across the scene or apply the same predictor family to all motion groups. A predictor that is well suited to smooth or periodic trajectories may be less appropriate for articulated, interacting, or nearly static regions. Moreover, the most appropriate predictor may change with the forecasting horizon. This motivates routing among complementary motion models at both the motion-group and forecast-horizon levels.

We propose Motion Regime Expert Routing for Gaussian Splatting (MoRE-GS), a causal and render-aware framework for forecasting dynamic 3D Gaussian scenes. Given a dynamic 3DGS reconstructed from the observed prefix, MoRE-GS freezes the Gaussian repre-

sensation and predicts the future translation and rotation of its control keypoints. A compact router combines complementary motion hypotheses separately for each motion group and forecast-horizon bin. The router is supervised through the frozen Gaussian renderer and includes an explicit persistence gate that suppresses unreliable motion extrapolation. It is calibrated exclusively from chronologically partitioned observations within the available prefix and frozen before future evaluation.

Our contributions are:

- A heterogeneous motion-expert framework for forecasting a frozen dynamic 3D Gaussian representation by combining persistence, an orthonormal coefficient-space predictor, graph-based interactions, and rigidity-aware structured motion prediction.
- A render-aware, group- and horizon-conditioned router that separately controls persistence and combines complementary learned motion predictors through image-space supervision from the frozen Gaussian renderer.
- An observed-only temporal fit-calibrate-refit protocol that calibrates the router without exposing reconstruction, expert training, routing, checkpoint selection, or early stopping to the withheld future suffix.

2. Related Work

2.1. Static and dynamic scene representations

Neural Radiance Fields (NeRF) [13] represent scenes using continuous neural functions, whereas 3DGS uses explicit Gaussian primitives and differentiable splatting for efficient rendering. Both representations can become poorly constrained when observations are limited. Sparse-view methods address this issue using geometric, appearance, or frequency-domain prior [17, 25]. Typical NeRF and 3DGS methods assume static scenes, but prior work has extended these methods to account for dynamic changes [11, 12, 15, 16, 20, 22]. Previous methods primarily target reconstruction or interpolation over observed timestamps rather than strict causal extrapolation to a withheld future suffix. The proposed MoRE-GS follows the broader motivation of combining complementary inductive biases, but applies it to causal temporal forecasting through distinct motion predictors rather than sparse-view regularization.

2.2. Future dynamic scene forecasting

Neural Velocity Fields for 3D Physics Learning from Dynamic Videos (NVFi) [10] combines a radiance representation with a learned velocity field to support

temporal extrapolation. GaussianPrediction [23] introduces lifecycle-aware canonical Gaussians and concentric motion distillation, reducing dense Gaussian deformation to a compact keypoint graph. Future keypoint translations and rotations are then predicted using a graph convolutional network. MoRE-GS retains this Gaussian representation and keypoint-based motion interface as a controlled foundation, while extending the forecasting stage with complementary motion experts and scene-calibrated group-horizon routing.

Latent Ordinary Differential Equations for Dynamic Scene Extrapolation with 3D Gaussian Splatting (ODE-GS) [19] encodes observed Gaussian trajectories using a Transformer and evolves the resulting latent state through a neural ordinary differential equation, providing a continuous formulation for temporal extrapolation. Motion Group-aware Gaussian Forecasting (MoGaF) [8] constructs motion-aware Gaussian groups and applies different optimization constraints to approximately rigid and non-rigid regions, encouraging spatially coherent future motion. These methods demonstrate the value of continuous dynamics and structured motion groups, but they do not explicitly route among heterogeneous predictor families as a function of both motion regime and forecast horizon. MoRE-GS instead combines several forecasting hypotheses within a shared frozen Gaussian scene and learns their relative contributions for each motion group and forecast-horizon bin.

2.3. Motion structure and expert integration

Motion factorization provides compact and spatially coherent scene dynamics. Neural Motion Factorization (DynMF) [7] uses shared neural trajectories, whereas Motion Scaffold (MoSca) [9] controls deformation through a sparse 4D scaffold. These representations motivate our use of compact motion variables but do not combine predictors with complementary temporal, relational, and geometric assumptions. Mixture of Experts for Dynamic Gaussian Splatting (MoE-GS) [3] combines deformation experts using a volume-aware pixel router for reconstruction at observed timestamps, with pruning and distillation for efficient inference. In contrast, MoRE-GS forecasts a withheld future suffix, routes control-keypoint motion over 3D motion groups and forecast-horizon bins, and calibrates its router on a disjoint interval within the observed prefix. Thus, MoRE-GS targets causal future extrapolation rather than pixel-level reconstruction of observed timestamps.

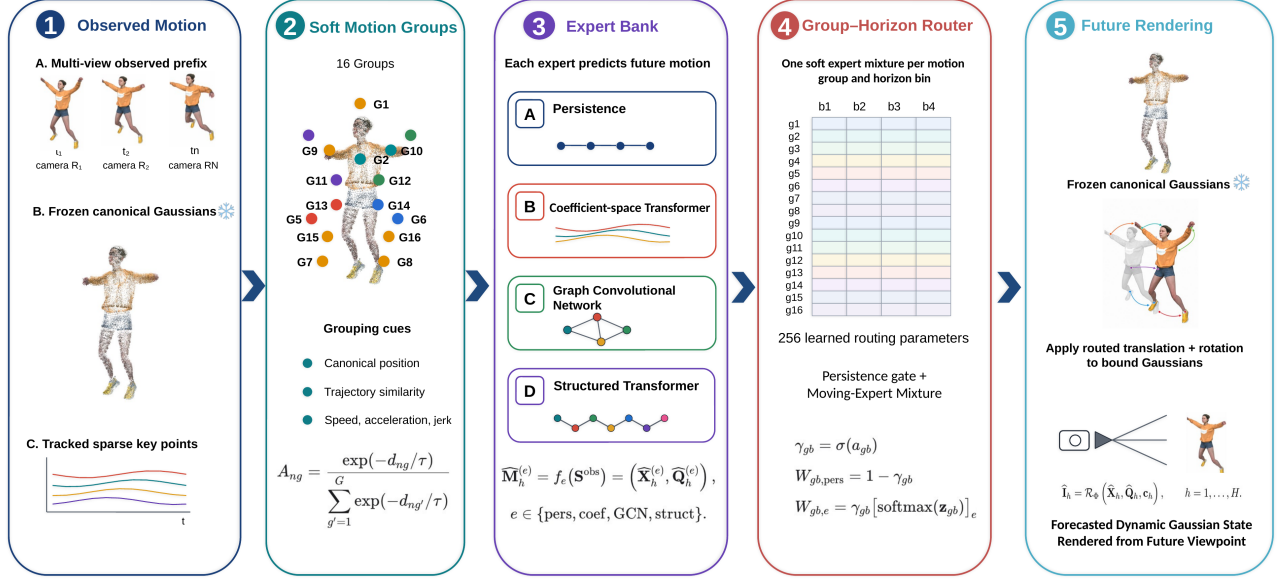


Figure 1. MoRE-GS forecasts control-keypoint motion using four complementary experts and an observed-only group-horizon router.

3. Method

3.1. Problem Setup and Overview

We use the dynamic 3D Gaussian representation and renderer released with GaussianPrediction [23] as the reconstruction backbone. The reconstruction stage is trained using only the observed temporal interval. We then freeze the Gaussian geometry, appearance, scale, opacity, deformation representation, control keypoints, keypoint-to-Gaussian motion bindings, and renderer. MoRE-GS forecasts motion without reoptimizing this reconstructed Gaussian representation.

Let frames $1, \dots, T_{\text{obs}}$ form the observed prefix and the following H frames form the withheld future suffix. GaussianPrediction represents deformation using N control keypoints with translation and rotation trajectories. Given the final $L = 10$ observed keypoint poses, MoRE-GS predicts their transformations for the next H steps. The frozen bindings transfer these transformations to the Gaussians, and Eq. 1 renders the corresponding future view:

$$\hat{\mathbf{I}}_h = \mathcal{R}_\Phi(\hat{\mathbf{X}}_h, \hat{\mathbf{Q}}_h, \mathbf{c}_h), \quad h = 1, \dots, H. \quad (1)$$

Here, $\hat{\mathbf{I}}_h$ is the predicted future image at forecast step h , $\hat{\mathbf{X}}_h \in \mathbb{R}^{N \times 3}$ and $\hat{\mathbf{Q}}_h \in \mathbb{R}^{N \times 4}$ contain the predicted translations and quaternion rotations of the control keypoints, $\mathbf{c}_h$ contains the evaluation-camera parameters, Φ denotes the frozen Gaussian scene and motion bindings, and $\mathcal{R}_\Phi$ is the frozen renderer.

As illustrated in Fig. 1, MoRE-GS combines four heterogeneous predictors using a compact, scene-specific routing table whose weights vary across soft motion groups and forecast-horizon bins. A persistence gate suppresses unreliable extrapolation. The router is calibrated through the frozen renderer on a chronologically held-out part of the observed prefix and is frozen before future evaluation.

3.2. Motion Experts

MoRE-GS uses four complementary experts: persistence, a coefficient-space Transformer, a graph convolutional network (GCN), and a structured Transformer. Each expert receives observed control-keypoint trajectories and predicts future translation and rotation. All learned experts use only observed trajectories and follow fixed training and output conventions across routing comparisons.

Persistence. The persistence expert copies the final observed translation and rotation to every future step. It has no trainable parameters and provides a conservative static-motion hypothesis for stationary regions or when learned extrapolation is unreliable.

Coefficient Transformer. Let $\tilde{\mathbf{S}}_n \in \mathbb{R}^{L \times 7}$ denote the channel-standardized motion history of keypoint n , containing three translation coordinates and four quaternion components at each observed step. Quaternions are first unit-normalized and sign-aligned over

time; the seven channels are then standardized independently. Equation 2 reparameterizes this history using a fixed, seeded random orthonormal basis:

$$\mathbf{Z}_n = \mathbf{B}\tilde{\mathbf{S}}_n, \quad \mathbf{B} \in \mathbb{R}^{L \times L}, \quad \mathbf{B}^\top \mathbf{B} = \mathbf{I}, \quad (2)$$

where $\mathbf{Z}_n \in \mathbb{R}^{L \times 7}$ contains the transformed coefficients, $\mathbf{B}$ is the temporal basis, and $\mathbf{I}$ is the $L \times L$ identity matrix. Orthonormality preserves trajectory norms and pairwise distances: $\|\mathbf{B}\tilde{\mathbf{S}}_n\|_F = \|\tilde{\mathbf{S}}_n\|_F$. Because $\mathbf{B}$ acts only along time, it never mixes translation and quaternion channels. Its exact inverse $\mathbf{B}^\top$ and unit condition number make this a lossless, well-conditioned reparameterization when all coefficients are retained.

The coefficients are partitioned into three blocks and combined with canonical-position and forecast-horizon embeddings. A shared one-layer Transformer encoder [18] and gated decoder predict translation and rotation residuals relative to persistence for all future steps in parallel. Seven-dimensional coefficients are projected to width 32, augmented with coefficient, block, and canonical-position embeddings, and pooled within three blocks. The pooled contexts and a normalized-horizon embedding feed the gated decoder, whose $N \times H \times 7$ output is rescaled, added to persistence, and quaternion-normalized. Training combines pose, velocity, acceleration, and gate penalties; weights are listed in the supplement. The random basis does not impose a predetermined frequency ordering. It outperforms the frequency-ordered discrete cosine transform type II (DCT-II) [1] in the supplementary matched ablation; therefore, we assign no physical frequency interpretation to its coefficient blocks.

Graph expert. The GCN uses the forecasting architecture released with GaussianPrediction [23]. Separate translation and rotation branches learn affinities among control keypoints and propagate coordinated motion across related regions. It provides a relational motion hypothesis that complements the parallel temporal prediction of the coefficient Transformer. The GCN predicts one future step at a time and is rolled out autoregressively over the forecast interval.

Structured expert and hard groups. The structured Transformer captures group-level dynamics within coherent keypoint groups. Its motion-aware grouping is motivated by MoGaF [8], while its architecture and supervision are specific to MoRE-GS. Each keypoint descriptor concatenates its canonical position with the first six principal component analysis (PCA) coefficients of its observed translation trajectory. We use the same empirically selected rank of six for every scene. In an observed-only analysis, these components retain

99.87% of the displacement variance on average across the eight D-NeRF scenes, supporting a compact representation with negligible loss of observed motion information.

The descriptors are divided into eight hard groups. Within each group, a shared one-layer Transformer projects each standardized $L \times 7$ keypoint history to width 32 and applies temporal self-attention within that keypoint. A contiguous masked span is annealed during training; a linear head predicts the next pose, and autoregressive rollout produces the future sequence. A normalized Kabsch alignment residual [4] classifies each group as approximately rigid or non-rigid. We align its first observed keypoint configuration with sampled subsequent configurations and normalize the root-mean-square alignment residual by the root-mean-square group radius. A group is classified as rigid when the temporal-median normalized residual is at most 0.10 and as non-rigid otherwise. Rigid groups receive a radius-normalized Kabsch alignment loss, whereas non-rigid groups receive a radius-normalized local displacement-consistency loss over four nearest-neighbor edges. These physical terms are combined with pose and acceleration losses; their weights are reported in the supplement. These hard groups are internal to the structured expert and are distinct from the soft router groups defined below.

3.3. Soft Groups and Group-Horizon Router

Soft router groups. We construct $G = 16$ soft groups before router calibration. Canonical positions come from the frozen reconstruction, while trajectory-derived descriptor terms use only the expert-fitting interval. Each keypoint descriptor contains its canonical position, up to eight uniformly sampled scene-normalized displacements, and observed speed, acceleration, and jerk statistics. After standardization, we apply soft k -means with deterministic farthest-point initialization: the first center is the descriptor with the largest squared norm, subsequent centers maximize distance to the nearest existing center, and the centers are updated for 30 fixed iterations without random sampling.

The resulting assignment matrix $\mathbf{A} \in [0, 1]^{N \times G}$ is defined by Eq. 3:

$$A_{ng} = \frac{\exp(-d_{ng}/\tau)}{\sum_{g'=1}^G \exp(-d_{ng'}/\tau)}, \quad (3)$$

where A_{ng} is the membership of keypoint n in group g , d_{ng} is its normalized squared distance to group center g , and τ controls assignment softness. The assignment matrix is fixed before router calibration, allow-

ing boundary keypoints to share routing weights across compatible groups.

Routing weights. We divide the forecast interval into $B = 4$ contiguous bins and let $b(h)$ denote the bin containing forecast step h . Bins use relative progress, $b(h) = 1 + \min\{B - 1, \lfloor (h - 1)B/H \rfloor\}$, rather than absolute frame indices. The moving-expert set and complete expert set are

$$\mathcal{E}_{\text{mov}} = \{\text{coef}, \text{GCN}, \text{struct}\}, \quad \mathcal{E} = \mathcal{E}_{\text{mov}} \cup \{\text{pers}\}.$$

For every group g and horizon bin b , Eq. 4 defines a persistence gate and the moving-expert weights:

$$\gamma_{gb} = \sigma(a_{gb}),$$

$$W_{gb,e} = \begin{cases} 1 - \gamma_{gb}, & e = \text{pers}, \\ \gamma_{gb} [\text{softmax}(\mathbf{z}_{gb})]_e, & e \in \mathcal{E}_{\text{mov}}. \end{cases} \quad (4)$$

Here, a_{gb} is the scalar persistence-gate logit, σ is the sigmoid function, $\gamma_{gb} \in [0, 1]$ is the total probability assigned to moving experts, $\mathbf{z}_{gb} \in \mathbb{R}^3$ contains their logits, and $W_{gb,e}$ is the weight assigned to expert e . The gate determines how much motion is used, while the softmax distributes this probability among the three learned experts. With 16 groups and 4 bins, the router contains $16 \times 4 \times 4 = 256$ trainable parameters and is a scene-specific calibrated table rather than an input-conditioned network.

Equation 5 converts group-level weights into the expert weight for keypoint n at horizon h :

$$\pi_{nhe} = \sum_{g=1}^G A_{ng} W_{g,b(h),e}, \quad (5)$$

where π_{nhe} is the final weight assigned to expert e .

The routed translation is defined in Eq. 6:

$$\hat{\mathbf{x}}_{n,h} = \sum_{e \in \mathcal{E}} \pi_{nhe} \hat{\mathbf{x}}_{n,h}^e, \quad (6)$$

where $\hat{\mathbf{x}}_{n,h}^e \in \mathbb{R}^3$ is expert e 's candidate translation and $\hat{\mathbf{x}}_{n,h}$ is the routed result.

Because a quaternion and its negative represent the same rotation, Eq. 7 aligns each candidate with the persistence quaternion before computing a normalized weighted average:

$$\tilde{\mathbf{q}}_{n,h}^e = \text{sign} \left(\left\langle \hat{\mathbf{q}}_{n,h}^e, \hat{\mathbf{q}}_{n,h}^{\text{pers}} \right\rangle \right) \hat{\mathbf{q}}_{n,h}^e,$$

$$\hat{\mathbf{q}}_{n,h} = \frac{\sum_{e \in \mathcal{E}} \pi_{nhe} \tilde{\mathbf{q}}_{n,h}^e}{\left\| \sum_{e \in \mathcal{E}} \pi_{nhe} \tilde{\mathbf{q}}_{n,h}^e \right\|_2}. \quad (7)$$

Here, $\hat{\mathbf{q}}_{n,h}^e$ is expert e 's candidate quaternion, $\hat{\mathbf{q}}_{n,h}^{\text{pers}}$ is the persistence quaternion, $\tilde{\mathbf{q}}_{n,h}^e$ is its sign-aligned form, and $\hat{\mathbf{q}}_{n,h}$ is the final unit quaternion.

3.4. Observed-Only Router Calibration

Calibrating the router on the future suffix would introduce evaluation information into model selection. We therefore use an observed-only temporal fit—calibrate—refit procedure with a fixed chronological 75/25 division of the observed prefix. The first 75% is used to fit temporary experts, while the remaining 25% forms a contiguous calibration interval excluded from temporary-expert fitting. It is included in the frozen reconstruction, so “out-of-fit” applies only to the temporary motion experts, not to representation learning. Under the primary D-NeRF 80/20 protocol, this calibration interval nominally matches the duration of the sealed future suffix, up to rounding. The division is fixed across scenes and is not selected using future performance.

The temporary experts forecast the held-out 25% of the observed prefix, and the router is calibrated from these out-of-fit forecasts through the frozen renderer using the objective in Sec. 3.5. The temporary experts are then discarded. Independently initialized deployment experts with the same architectures, preprocessing, objectives, and output conventions are trained on the complete observed prefix and paired with the frozen router. No future evaluation frame, motion state, image, or metric is used for reconstruction, grouping, expert training, router calibration, checkpoint selection, or early stopping. Relative-progress bins permit reuse across interval lengths. Applying the router after refitting experts is an empirical matched-interface assumption supported by final forecasting results, not a theoretical invariance or isolated transfer ablation.

3.5. Render-Aware Router Objective

Keypoint-space error weights visible and occluded motion equally, although their effects on the rendered image may differ substantially. We therefore calibrate the router using image-space error from the frozen renderer. For pixel p in calibration image $\mathbf{I}_h$, Eq. 8 defines foreground and dynamic-change scores:

$$\delta_h^{\text{fg}}(p) = \frac{1}{3} \|\mathbf{I}_h(p) - \mathbf{I}_{\text{bg}}\|_1,$$

$$\delta_h^{\text{dyn}}(p) = \frac{1}{3} \left\| \mathbf{I}_h(p) - \hat{\mathbf{I}}_h^{\text{pers}}(p) \right\|_1. \quad (8)$$

Here, $\delta_h^{\text{fg}}(p)$ measures mean absolute RGB difference from the background, $\delta_h^{\text{dyn}}(p)$ measures mean absolute RGB difference from persistence, $\mathbf{I}_{\text{bg}}$ is the background color, and $\hat{\mathbf{I}}_h^{\text{pers}}$ is the persistence rendering.

Equation 9 defines the important-pixel mask:

$$\mathcal{M}_h = \left\{ p : \delta_h^{\text{fg}}(p) > \tau_{\text{fg}} \vee \delta_h^{\text{dyn}}(p) > \tau_{\text{dyn}} \right\}, \quad (9)$$

where τ_{fg} and τ_{dyn} are fixed foreground and dynamic-change thresholds. This mask emphasizes visible content and regions not explained by persistence.

For calibration frame h , Eq. 10 defines the rendering loss:

$$\ell_{\text{render}}^{(h)} = \text{MSE}(\hat{\mathbf{I}}_h, \mathbf{I}_h) + \text{MSE}_{p \in \mathcal{M}_h}(\hat{\mathbf{I}}_h, \mathbf{I}_h), \quad (10)$$

where the first mean-squared error (MSE) term is evaluated over the complete image and the second over pixels in $\mathcal{M}_h$. The aggregate rendering loss is

$$\mathcal{L}_{\text{render}} = \frac{1}{T_{\text{cal}}} \sum_{h=1}^{T_{\text{cal}}} \ell_{\text{render}}^{(h)}. \quad (11)$$

The router uses three regularizers:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{G(B-1)|\mathcal{E}|} \sum_{g=1}^G \sum_{b=1}^{B-1} \sum_{e \in \mathcal{E}} (W_{g,b+1,e} - W_{g,b,e})^2, \quad (12)$$

$$\mathcal{L}_{\text{move}} = \frac{1}{GB} \sum_{g=1}^G \sum_{b=1}^B \sum_{e \in \mathcal{E}_{\text{mov}}} W_{g,b,e} = \frac{1}{GB} \sum_{g=1}^G \sum_{b=1}^B \gamma_{gb}, \quad (13)$$

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{GB} \sum_{g=1}^G \sum_{b=1}^B \sum_{e \in \mathcal{E}_{\text{mov}}} W_{g,b,e} \log(\max(W_{g,b,e}, \epsilon)). \quad (14)$$

Here, $|\mathcal{E}|$ is the number of experts and $\epsilon = 10^{-8}$ prevents numerical instability. The smoothness term discourages abrupt routing changes between adjacent horizon bins, the moving-mass term favors persistence unless motion reduces the rendered error, and the entropy term encourages concentrated moving-expert weights.

The complete router-calibration objective is

$$\mathcal{L}_{\text{router}} = \mathcal{L}_{\text{render}} + \lambda_s \mathcal{L}_{\text{smooth}} + \lambda_m \mathcal{L}_{\text{move}} + \lambda_e \mathcal{L}_{\text{entropy}}, \quad (15)$$

where λ_s , λ_m , and λ_e are the nonnegative regularization weights. Exact mask thresholds, regularization weights, and optimization hyperparameters are provided in the supplementary material.

4. Experiments

4.1. Experimental Setup

We evaluate on all eight D-NeRF scenes [16] at 800×800 resolution. Each training-view sequence is ordered chronologically: the first 80% is observed, and the final 20% is reserved exclusively for forecasting evaluation. The Gaussian representation is reconstructed for 60K iterations using only the observed prefix. HyperNeRF [15] provides a secondary real-capture stress test.

We retain only the official training-camera sequence, exclude the held-out validation cameras, and apply a chronological 90/10 observed/future split at the native scene resolution using the released scene-specific reconstruction configuration. Thus, future images are never used for representation fitting, grouping, expert training, router calibration, checkpoint selection, or early stopping.

Forecasting configuration. The observed-only Gaussian representation is frozen before forecasting. Each expert receives the final ten observed keypoint poses. The coefficient and structured Transformers use one layer, width 32, eight attention heads, feed-forward width 64, and dropout 0.2. They are optimized for 2,001 updates with a learning rate decayed from 10^{-3} to 10^{-5} . The GCN follows the released GaussianPrediction objective and autoregressive rollout. The proposed MoRE-GS configuration uses 16 soft motion groups and four forecast-horizon bins. For causal calibration, temporary experts are trained on the first 75% of the observed prefix and evaluated through the frozen renderer on the remaining 25%. The router is optimized for 200 AdamW updates with a learning rate of 0.05. The temporary experts are then discarded, deployment experts are independently trained on the complete observed prefix, and the calibrated router is frozen before future evaluation.

We report peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS). We first average over evaluation views within each scene and then weight all scenes equally. Every reported checkpoint is cold-reloaded before evaluation. GaussianPrediction and ODE-GS are local reproductions with the same split and evaluation identities but method-specific representations and budgets; MoGaF values are paper-reported. These are system-level comparisons, whereas Table 3 contains matched controls.

4.2. D-NeRF Forecasting

Table 1 reports 27.47 dB PSNR, 0.9480 SSIM, and 0.0446 LPIPS, improving PSNR by 2.13, 1.60, and 0.82 dB over GaussianPrediction, MoGaF, and ODE-GS. MoRE-GS leads 16/24 per-scene metrics. ODE-GS has higher PSNR on T-Rex, Jumping Jacks, and Hook, but MoRE-GS has higher SSIM and lower LPIPS. Hell Warrior is a failure case: MoRE-GS trails ODE-GS in PSNR/SSIM and MoGaF in all metrics.

Figure 2 provides qualitative evidence at the longest forecast horizon. On Mutant and Stand Up, MoRE-GS better preserves global pose and reduces foreground residuals. On the real-capture Chicken scene, it sub-

Table 1. D-NeRF 80/20 future forecasting. Best values, determined from the unrounded scores, are shown in bold.

Scene	GaussianPrediction			MoGaF			ODE-GS			MoRE-GS		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
T-Rex	21.08	0.940	0.046	20.60	0.941	0.050	21.51	0.944	0.054	21.13	0.944	0.048
Mutant	27.55	0.953	0.027	28.72	0.955	0.022	31.01	0.967	0.026	33.55	0.976	0.015
Jumping Jacks	20.00	0.912	0.084	20.12	0.922	0.073	22.23	0.920	0.080	21.18	0.925	0.069
Stand Up	27.44	0.948	0.044	30.38	0.954	0.034	27.23	0.944	0.054	30.68	0.958	0.034
Bouncing Balls	27.09	0.972	0.030	26.82	0.973	0.037	25.76	0.964	0.053	28.40	0.977	0.030
Hook	24.21	0.914	0.054	24.13	0.908	0.056	26.99	0.934	0.054	26.47	0.937	0.041
Hell Warrior	31.03	0.929	0.073	31.29	0.932	0.068	31.66	0.931	0.083	30.22	0.918	0.081
Lego	24.30	0.919	0.063	24.88	0.901	0.053	26.75	0.935	0.053	28.10	0.949	0.039
Mean	25.34	0.936	0.053	25.87	0.936	0.049	26.64	0.942	0.057	27.47	0.948	0.045

Table 2. HyperNeRF future forecasting under the chronological 90/10 protocol. MoRE-GS uses 16 motion groups. Best values are shown in bold.

Scene	ODE-GS			MoRE-GS		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Chicken	18.49	0.594	0.467	19.52	0.581	0.368
3D Printer	16.73	0.543	0.485	16.45	0.519	0.423
Cut Lemon	20.46	0.617	0.379	20.97	0.598	0.345
Split Cookie	19.65	0.630	0.264	19.95	0.638	0.259
Mean	18.83	0.596	0.399	19.22	0.584	0.349

stantially reduces the motion distortion produced by ODE-GS.

Across D-NeRF, MoRE-GS improves mean PSNR over persistence by 3.124, 3.169, 3.021, and 1.728 dB from the nearest to the farthest future quartile. The 1.728 dB improvement at the farthest horizon confirms that its advantage is not limited to near-term prediction. Complete quartile-level results are provided in the supplementary material.

4.3. HyperNeRF Results

HyperNeRF provides a challenging real-capture evaluation involving handheld cameras, occlusion, non-rigid interactions, and appearance variation. Table 2 reports results under the causal chronological 90/10 observed/future protocol using 16 motion groups. MoRE-GS improves mean PSNR by 0.389 dB and reduces mean LPIPS by 0.0496 relative to ODE-GS reproduction. It achieves higher PSNR on three of four scenes and lower LPIPS on all four scenes. ODE-GS retains a 0.0121 advantage in mean SSIM, indicating that motion-only expert routing does not fully preserve fine appearance and visibility structure under real capture.

Additional near-, middle-, and long-horizon examples are provided in the supplementary material.

Table 3. Causal D-NeRF ablations averaged over all eight scenes. Best overall values, determined from unrounded results, are bold; the selected configuration is underlined.

Variant	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Individual experts			
Persistence	24.70	0.936	0.053
Coefficient Transformer	26.51	0.941	0.051
GCN	25.22	0.934	0.052
Structured Transformer	25.95	0.932	0.061
Routing design			
Observed-only fixed expert	26.61	0.942	0.051
Uniform mixture	26.66	0.944	0.045
Scene-level routing	27.13	0.945	0.047
Horizon-only routing	27.07	0.946	0.046
Group-only routing	27.53	0.947	0.045
Without persistence gate	26.85	0.944	0.048
Shuffled groups	26.92	0.943	0.050
Group-count sensitivity			
MoRE-GS ($G = 8$)	27.26	0.947	0.046
MoRE-GS ($G = 16$, selected)	<u>27.47</u>	<u>0.948</u>	<u>0.045</u>
MoRE-GS ($G = 32$)	27.66	0.949	0.044
Regularization			
MoRE-GS ($G = 16$), no regularizers	27.44	0.948	0.045

4.4. Causal Ablations

Table 3 compares individual experts with matched routing, grouping, group-count, and regularization controls; group-dependent rows use $G = 16$ unless marked. MoRE-GS gains 0.96 dB over the strongest expert, 0.86 dB over observed-only selection, and 0.81 dB over uniform mixing. Removing persistence costs 0.62 dB, while motion-aware groups outperform shuffled groups by 0.55 dB.

In the matched $G = 16$ comparison, group-horizon routing is 0.06 dB below group-only routing in PSNR but improves SSIM by 0.0012 and LPIPS by 0.0008. Supplementary quartiles show that its structural/perceptual gains persist beyond the nearest interval. Horizon conditioning is therefore a modest secondary component; motion grouping provides the main specialization.

Increasing G from 8 to 16 improves PSNR by 0.20 dB, while $G = 32$ provides a further 0.20 dB improvement. We retain the cross-dataset setting $G = 16$;

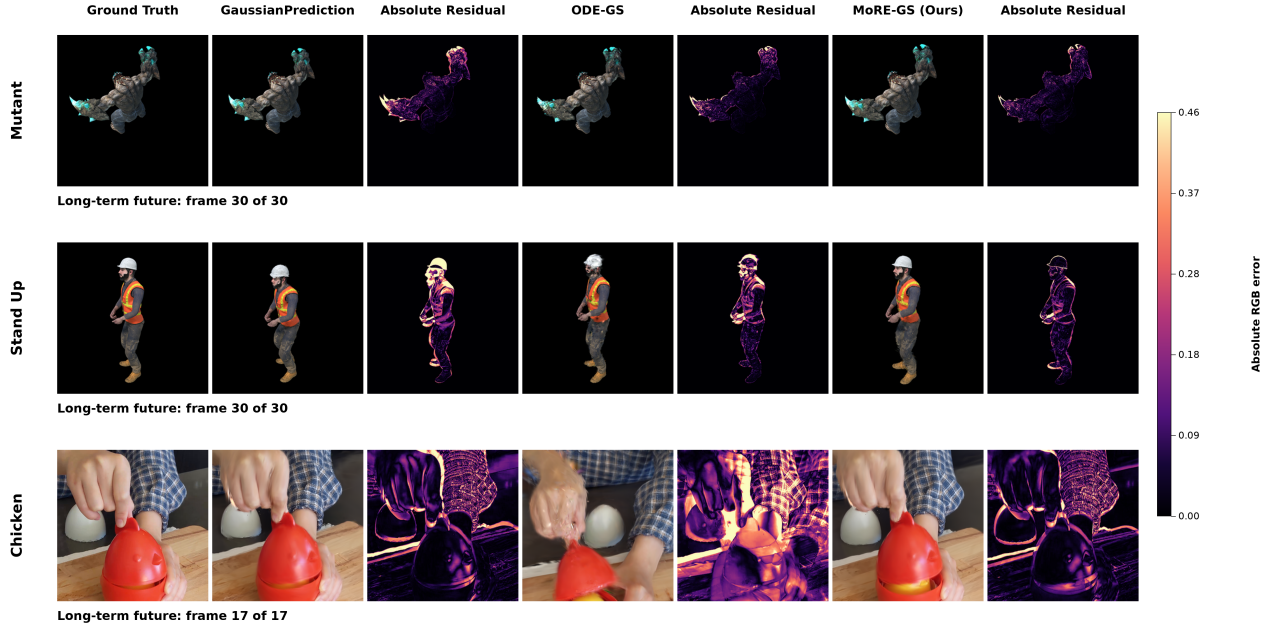


Figure 2. Long-horizon forecasts and absolute RGB residuals on Mutant, Stand Up, and Chicken. All residual maps use the same color scale.

Table 4. Post-canonical efficiency for the profiled $G = 8$ configuration, averaged over eight D-NeRF scenes. State excludes reconstructed scenes.

Method	Params	State	Train (s)	FPS	VRAM
GaussianPrediction	14.173M	69.26 MiB	—	155.11	—
ODE-GS	1.093M	13.10 MiB	10,444	39.00	2.43 GiB
MoRE-GS ($G = 8$ profile)	14.280M	69.75 MiB	1,135	84.83	2.15 GiB

$G = 32$ is reported only as a post-hoc D-NeRF sensitivity result. The selected routing table has 256 parameters, excluding the 14.28M forecasting parameters in Table 4. Removing all regularizers reduces PSNR by only 0.03 dB, indicating a modest stabilizing effect.

4.5. Efficiency

Table 4 profiles $G = 8$ over eight D-NeRF scenes on one RTX 4090; $G = 16$ was not reprofiled. Relative to ODE-GS, MoRE-GS trains 9.20 $\times$ faster, raises post-canonical forecasting-and-rendering throughput from 39.00 to 84.83 FPS (2.18 $\times$), and uses 11.8% less peak memory. The measurements include all experts and routing but exclude the 60K reconstruction; matched end-to-end timing is not reported.

Limitations. Evaluation covers eight synthetic D-NeRF scenes and four HyperNeRF captures; driving

and robotics remain motivations, not evaluated deployments. Routers are calibrated per scene, and cross-scene transfer is untested. Dense inference evaluates every expert and costs more than a single expert, though less than ODE-GS. MoRE-GS moves existing Gaussians but does not forecast appearance, illumination, opacity, unseen geometry, or topology.

5. Conclusion

We presented MoRE-GS, a causal, render-aware framework that combines complementary motion experts through group- and horizon-conditioned routing. The router is calibrated entirely within the observed prefix, preventing future information from influencing training or model selection.

MoRE-GS achieves the best average performance in the primary cross-method comparison on D-NeRF, improving PSNR by 0.82 dB over ODE-GS reproduction. The profiled $G = 8$ configuration provides substantially faster post-canonical training and inference. Results on four HyperNeRF captures provide preliminary real-scene evidence. Overall, the results support renderer-calibrated expert composition as an accurate and efficient approach to future dynamic Gaussian forecasting.

References

- [1] Nasir Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. [4](#)
- [2] Anthony Hu, Zak Murez, Nikhil Mohan, Sofia Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance prediction in bird’s-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15253–15262, 2021. [1](#)
- [3] In-Hwan Jin, Hyeongju Mun, Joonsoo Kim, Kugjin Yun, and Kyeongbo Kong. MoE-GS: Mixture of experts for dynamic gaussian splatting. In *International Conference on Learning Representations*, 2026. [arXiv:2510.19210](#). [2](#)
- [4] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. [4](#)
- [5] Baris Kayalibay, Atanas Mirchev, Patrick van der Smagt, and Justin Bayer. Tracking and planning with spatial world models. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference*, pages 124–137. PMLR, 2022. [1](#)
- [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. [1](#)
- [7] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. DynMF: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In *European Conference on Computer Vision*, pages 252–269, 2024. [2](#)
- [8] Junmyeong Lee, Hoseung Choi, and Minsu Cho. Space-time forecasting of dynamic scenes with motion-aware gaussian grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41033–41043, 2026. [1](#), [2](#), [4](#)
- [9] Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6177, 2025. [2](#)
- [10] 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*, 2023. [1](#), [2](#)
- [11] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. [1](#), [2](#)
- [12] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision*, pages 800–809, 2024. [1](#), [2](#)
- [13] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020. [2](#)
- [14] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. [1](#)
- [15] 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. *ACM Transactions on Graphics*, 40(6):238:1–238:12, 2021. [2](#), [6](#)
- [16] 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*, pages 10318–10327, 2021. [1](#), [2](#), [6](#)
- [17] Shima Salehi, Atharva Agashe, Andrew J. McFarland, and Joshua Peeples. LGDWT-GS: Local and global discrete wavelet-regularized 3-d gaussian splatting for sparse-view scene reconstruction. *IEEE Access*, 14: 72746–72756, 2026. [2](#)
- [18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. [4](#)
- [19] Daniel Wang, Patrick Rim, Tian Tian, Dong Lao, Alex Wong, and Ganesh Sundaramoorthi. ODE-GS: Latent ODEs for dynamic scene extrapolation with 3d gaussian splatting. In *International Conference on Learning Representations*, 2026. [1](#), [2](#)
- [20] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. [1](#), [2](#)
- [21] Ze Yang, Yun Chen, Jingkan Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. [1](#)
- [22] 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*, pages 20331–20341, 2024. [1](#), [2](#)
- [23] Bomong 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 Conference Papers, pages 84:1–84:12, 2024. [1](#), [2](#), [3](#), [4](#)
- [24] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D occupancy world model for autonomous driving. In European Conference on Computer Vision, pages 55–72, 2024. [1](#)
- [25] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In Proceedings of the European Conference on Computer Vision, 2024. [2](#)