# MVFusion-GS: Motion-Variance Guided Temporal Attention for High-Quality Dynamic Gaussian Splatting

Jianwei Hu[1*], Tingxuan Huang[1*], Hengyu Zhou[1], Ningna Wang[2], Xiaohu Guo[2], Jinshan Lai[3], and Bin Wang[1†]

[1] Tsinghua University, Beijing, China
[2] The University of Texas at Dallas, Richardson, TX, USA
[3] University of Electronic Science and Technology of China, Chengdu, China

hjw17@tsinghua.org.cn, {htx25,zhouhy22}@mails.tsinghua.edu.cn,
{ningna.wang,xguo}@utdallas.edu, laijinshan574@gmail.com, wangbins@tsinghua.edu.cn

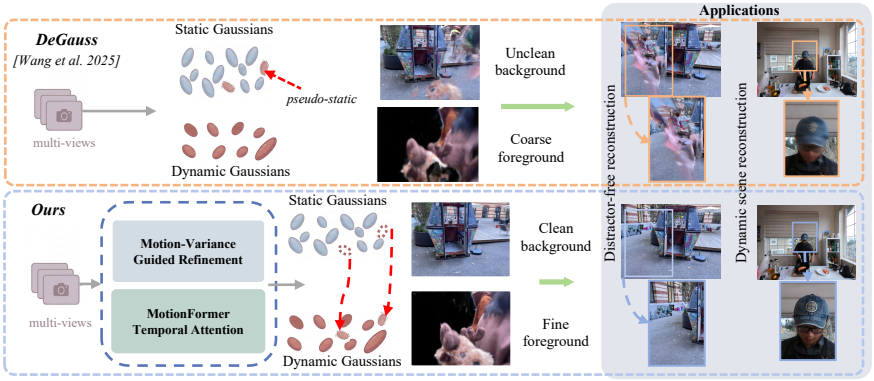


**Fig. 1:** Standard dynamic-static Gaussian decomposition leaves pseudo-static foreground residuals in the static branch (top), causing background contamination. Our method enhances the motion awareness of the deformation network, correctly re-attributing pseudo-static Gaussians to the dynamic branch (bottom). This yields dual benefits: cleaner backgrounds for distractor-free reconstruction and finer foreground details for dynamic scene reconstruction (right).

**Abstract.** 3D Gaussian Splatting (3DGS) enables real-time novel view synthesis for static scenes. Extending it to dynamic scenes via deformation fields has recently attracted significant attention, particularly for *dynamic scene reconstruction* and *distractor-free reconstruction*. However, existing deformation networks lack explicit motion awareness: they neither capture long-term motion intensity nor exploit short-term temporal coherence, leading to inaccurate foreground deformation and pseudo-static residuals in the background. We present **MVFusion-GS**, a method

* Equal contribution.
† Corresponding author.

that enhances deformation networks with two complementary motion-aware mechanisms. The **Motion-Variance Guided Refinement** aggregates per-Gaussian deformation statistics across time to estimate motion variance and uses it to guide dynamic-static separation during deformation prediction. The **MotionFormer Temporal Attention** module applies Transformer self-attention over neighboring timesteps to model local motion dependencies and improve temporal consistency. Extensive experiments on both dynamic scene reconstruction and distractor-free reconstruction benchmarks demonstrate state-of-the-art performance, showing that explicit motion awareness improves both foreground motion modeling and static background reconstruction.

**Keywords:** 3D Gaussian Splatting · Distractor-free · Dynamic Scene Reconstruction

## 1 Introduction

Extending 3D Gaussian Splatting (3DGS) [8] to dynamic scenes via deformation fields is a central research topic, where two representative tasks are *dynamic scene reconstruction* and *distractor-free reconstruction*. The former aims to faithfully reconstruct foreground motion with high deformation fidelity [6, 31, 33], while the latter seeks to recover a clean static background by separating out transient distractors (e.g., pedestrians, vehicles) [9, 20]. Both tasks rely on deformation fields that map Gaussian attributes from a canonical space to each observation timestep, yet their optimization objectives diverge: one pursues high-fidelity foreground fitting, the other clean background separation. Simultaneously addressing both within a single framework remains challenging.

DeGauss [28] tackles this with a decoupled dynamic-static Gaussian framework, decomposing the scene into foreground dynamic Gaussians modeled by a HexPlane-based spatio-temporal deformation module and background static Gaussians that directly fit the scene structure, composing them via a learnable probabilistic mask. While effective for distractor-free reconstruction, its deformation network lacks motion-aware cues. Fig. 1 shows that when foreground objects exhibit low-amplitude or transient motion, the network fails to produce accurate deformation predictions, leaving pseudo-static foreground residuals in the background branch and degrading static reconstruction quality.

The root cause is that standard deformation networks encode latent features through spatio-temporal grids without explicitly modeling the motion pattern each Gaussian undergoes or its temporal context across adjacent frames. To address this, we propose two complementary mechanisms:

**Motion-Variance Guided Refinement.** We periodically sample each Gaussian's deformation across multiple timesteps and compute statistics over its positional, rotational, and scale variations, yielding a 13D global trajectory signature. Injected into the deformation network's latent space, this signature provides an explicit motion-intensity prior for foreground-background discrimination.

**MotionFormer Temporal Attention.** Complementing the global trajectory signature, we introduce a Transformer-based temporal attention module

that aggregates neighboring timesteps through query-centered cross-attention, enabling the network to leverage local temporal context for temporally consistent deformation prediction.

The two mechanisms work synergistically: the global trajectory signature provides long-term motion priors for coarse dynamic-static separation, while temporal attention refines instantaneous deformation through short-term temporal context. This yields dual benefits — more accurate foreground motion capture in dynamic scene reconstruction, and cleaner background purity in distractor-free reconstruction by correctly re-attributing pseudo-static Gaussians to the dynamic branch.

Our main contributions are:

- A dual-task plug-in module that enhances deformation network motion awareness, achieving consistent improvements on both dynamic scene reconstruction and distractor-free reconstruction benchmarks.
- Motion-Variance Guided Refinement, which aggregates per-Gaussian deformation statistics across the global time span into an explicit motion intensity prior for dynamic–static separation.
- MotionFormer Temporal Attention, which leverages cross-attention to model cross-temporal motion dependencies, improving deformation accuracy and temporal consistency.
  Code is available at: `https://github.com/toseeai-com/MVFusion-GS`.

## 2 Related Work

### 2.1 Novel View Synthesis for Static Scenes

Neural Radiance Fields (NeRF) [15] introduced continuous volumetric scene representations for novel view synthesis, with later works improving efficiency [2,16] and geometric accuracy [17,34]. More recently, 3D Gaussian Splatting (3DGS) [8] represents scenes as anisotropic Gaussian primitives and enables real-time rendering via differentiable rasterization. Follow-up works such as Scaffold-GS [13] and 2DGS [4] further improve scalability and rendering quality. Our method builds upon the 3DGS representation.

### 2.2 Dynamic Gaussian Scene Reconstruction

Recent works extend 3DGS to dynamic scenes by learning deformation fields that map Gaussians from a canonical space to each timestep. Deformable-GS [33] uses an MLP-based deformation network, while 4DGS [31] adopts a HexPlane encoder with MLP decoders for time-dependent attribute prediction. SC-GS [6] introduces sparse control nodes to guide Gaussian deformation. These approaches implicitly learn motion through deformation but lack explicit awareness of the motion patterns associated with each Gaussian. Other recent works improve dynamic Gaussian representations from complementary perspectives. Some parameterize Gaussian motion with explicit trajectories, such as Shape-of-Motion [27],

SplineGS [18], and Gaussian Marbles [24], which are primarily designed for monocular dynamic reconstruction. SpaceTimeGS [12] and ST-4DGS [10] improve spatial-temporal consistency through spacetime Gaussians and motion-aware regularization, and FreeTimeGS [29] allows Gaussian primitives to appear at arbitrary times and locations for complex motion. In contrast, our method injects motion-aware signals into deformation-based Gaussian pipelines by incorporating global trajectory statistics and local temporal attention into the deformation feature space.

### 2.3 Distractor Handling in Dynamic Scene Reconstruction

Dynamic scene reconstruction from casually captured imagery is often affected by transient distractors. Some approaches suppress their influence without explicitly modeling them. RobustNeRF [21] down-weights high-residual pixels during optimization. NeRF On-the-go [19] predicts per-pixel dynamic masks using DINOv2 features, later extended to Gaussian Splatting in WildGaussians [9]. SpotlessSplats [20] further leverages clustered diffusion features to identify transient regions.

Another line of work explicitly models transient content. NeRF-W [14] decomposes scenes into static and transient neural fields, while $D^2$NeRF [32] improves decomposition stability through assignment regularization. DeSplat [30] extends this idea to Gaussian Splatting using view-specific Gaussian primitives to represent transient occluders. DeGauss [28] later proposes a dual-branch Gaussian Splatting framework with a learnable foreground–background mask. However, lacking motion-aware cues, its deformation network often misclassifies subtle or transient motion as static. Our method introduces motion trajectory priors and temporal attention to improve separation and reconstruction.

## 3 Preliminary

### 3.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [8] represents a scene as a set of anisotropic Gaussian primitives, each defined by a mean position $\mu$, opacity $\alpha$, color $c$, and covariance $\Sigma$. Instead of storing $\Sigma$ directly, 3DGS parameterizes it using a rotation–scale decomposition:

$$\Sigma = RSS^\top R^\top, \tag{1}$$

where $R \in \mathrm{SO}(3)$ encodes orientation, and $S = \mathrm{diag}(s_x, s_y, s_z)$ controls the anisotropic scale along local axes. This formulation guarantees positive semi-definiteness and enables independent optimization of shape and orientation.

When rendering, each 3D Gaussian is projected to the image plane through the camera transformation $W$ and the projection Jacobian $J$. The resulting 2D covariance is computed as

$$\Sigma_{2D} = JW\Sigma W^\top J^\top, \tag{2}$$

which determines the ellipse footprint of the projected Gaussian. The splats are then alpha-composited in front-to-back order to produce the final image.

### 3.2 Dynamic 3DGS

Dynamic 3D Gaussian Splatting extends static 3DGS to videos by introducing a time-conditioned deformation model that maps Gaussians from a canonical space to observation space at each timestamp. Early methods (e.g., 4DGS [31], SC-GS [6]) typically use a single deformation network for all Gaussians, which can lead to unstable optimization: dynamic residuals may leak into static structures, while background regions may become over-deformed.

**Decoupled Dynamic–Static Representation and Composition.** To alleviate this issue, DeGauss [28] adopts a decoupled formulation with two Gaussian sets: a static background set $\mathcal{G}_s$ and a dynamic foreground set $\mathcal{G}_d$ whose attributes are modulated by a deformation network.

At time $t$, only the dynamic branch is deformed:

$$\mathcal{G}_d(t) = \mathcal{G}_d + \Delta\mathcal{G}_d(t), \qquad \Delta\mathcal{G}_d(t) = \Phi(h_t), \tag{3}$$

where $h_t = f(\mathbf{x}, t)$ denotes the spatiotemporal feature extracted from Gaussian coordinates $\mathbf{x}$ and timestamp $t$, and $\Phi(\cdot)$ predicts Gaussian attribute offsets.

Both branches are rendered independently to produce dynamic and static images $\hat{I}_d(t)$ and $\hat{I}_s(t)$. A per-pixel composition mask $M(t) \in [0, 1]$ predicted from the dynamic branch blends the two renders:

$$\hat{I}(t) = M(t) \odot \hat{I}_d(t) + \big(1 - M(t)\big) \odot \hat{I}_s(t). \tag{4}$$

This decoupled formulation encourages motion to be explained by $\mathcal{G}_d$ while keeping the background $\mathcal{G}_s$ stable.

## 4 Method

### 4.1 Overview

**Motivation.** In a decoupled dynamic–static reconstruction pipeline, the dynamic branch is responsible for modeling scene motion while keeping the static branch stable. However, standard spatiotemporal deformation networks often underfit subtle or transient motions, causing dynamic residuals to leak into the static branch and degrade the reconstruction quality.

To address this issue, we treat the baseline deformation network as a coarse predictor and introduce a lightweight motion-aware refinement. The refinement leverages motion statistics and short-term temporal context to enhance the deformation feature, enabling more accurate modeling of subtle dynamic behaviors.

Fig. 2 illustrates the overall pipeline of the proposed method. The refined deformation is formulated as

$$\Delta\mathcal{G}_d(t) = \Delta\mathcal{G}_d^{\text{base}}(t) + \Delta\mathcal{G}_d^{\text{MR}}(t), \tag{5}$$

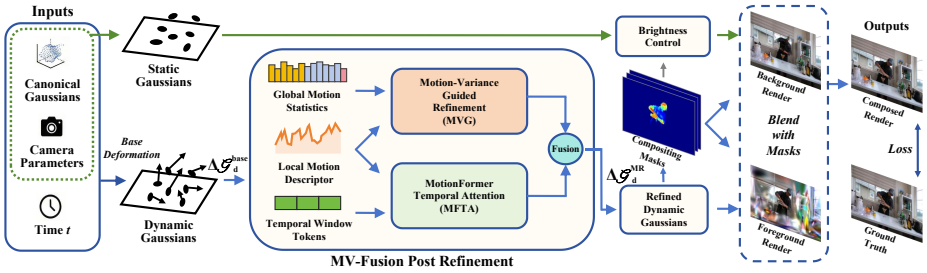


**Fig. 2:** Overview of the proposed motion-aware refinement framework. The baseline deformation network first predicts a coarse deformation for dynamic Gaussians. Two motion-aware modules are then applied: MVG extracts global trajectory signatures and local variance cues, while MFTA aggregates short-term temporal context. The fused motion-aware feature produces refined Gaussian updates, followed by foreground–background rendering and mask-based composition.

where the motion-aware refinement term is predicted from fused features:

$$\Delta\mathcal{G}_d^{\mathrm{MR}}(t) = \Phi\big(h_{\mathrm{base}}(t) + h_{\mathrm{MVG}}(t) + h_{\mathrm{MFTA}}(t)\big), \tag{6}$$

Here $h_{\mathrm{base}}$ denotes the baseline deformation feature, and $\Phi(\cdot)$ denotes the deformation heads applied on the fused feature. In other words, MVG and MFTA operate purely in the feature space: they are fused with the baseline deformation feature and decoded by the unchanged deformation heads $\Phi$ to predict refined Gaussian attribute updates. This plug-in design requires no modification to the original deformation heads and thus integrates seamlessly into existing deformation-based Gaussian pipelines.

**Decoupled rendering.** The refined deformation $\Delta\mathcal{G}_d(t)$ is applied to update the dynamic Gaussians. The decoupled Gaussian sets are rendered independently and composited using a learned soft mask $M(t)$ derived from deformation $\mathrm{SH}_{\mathrm{mask}}$ channels, which represent static, dynamic, and brightness-control components.

### 4.2 Motion-Variance Guided Refinement (MVG)

**Motivation.** In a decoupled dynamic–static pipeline, the deformation branch must determine which Gaussians belong to the moving foreground and how they evolve over time. However, standard spatiotemporal grids only encode local $(\mathbf{x}, t)$ context and provide no explicit cue about the *global motion behavior* of each Gaussian. Consequently, low-amplitude or transient motions are often under-modeled, producing "pseudo-static" residuals that leak into the background branch.

To address this issue, we periodically extract motion descriptors from simulated deformation trajectories and use them as motion priors. MVG contains two complementary signals: (i) a *global trajectory signature* that captures long-term motion statistics, represented as $\mathbf{V} \in \mathbb{R}^{N\times 13}$; and (ii) a cached *local variance*

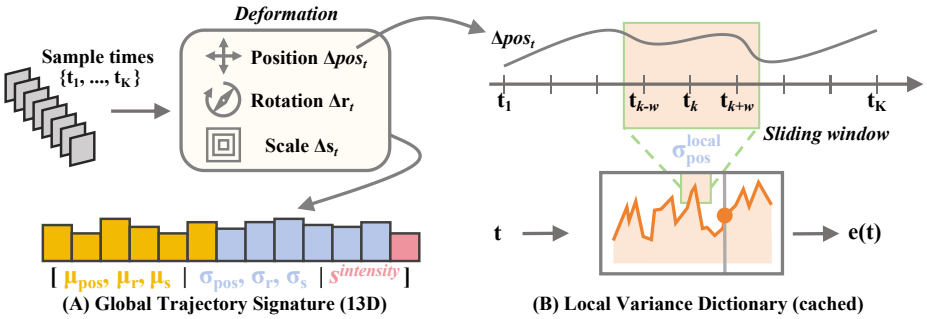


**Fig. 3:** Motion descriptor extraction in MVG. Given sampled timestamps $\{t_k\}_{k=1}^{K}$, we evaluate each Gaussian's deformation trajectory and compute cross-frame variations in position, rotation, and scale. (A) The full trajectory is aggregated into a 13D global trajectory signature that captures long-term motion statistics. (B) Local motion variance is cached as a time-indexed dictionary: given a query timestamp $t$, the dictionary maps it to $e(t)$ by retrieving or interpolating the local variance computed within a sliding temporal window.

*dictionary* that maps a query timestamp $t$ to an instantaneous motion intensity $e(t) \in \mathbb{R}^{N\times 1}$. Both signals are computed without gradient propagation and cached as per-Gaussian attributes for efficient reuse during training.

**Trajectory Sampling.** Fig. 3 summarizes how we extract the global trajectory signature and the local variance dictionary from simulated deformation trajectories. Every $E$ training iterations, we sample $K$ timestamps $\{t_k\}_{k=1}^{K}$ from the training cameras and evaluate the deformation network for all $N$ dynamic Gaussians:

$$\Delta\mathcal{G}_i(t_k) = \big(\Delta\mathbf{pos}_{i,k},\, \Delta\mathbf{r}_{i,k},\, \Delta\mathbf{s}_{i,k},\, \ldots\big), \quad k = 1, \ldots, K. \tag{7}$$

We then derive trajectory variations from the predicted deformation residuals. In particular, we use the position displacement $\Delta\mathbf{pos}_{i,k}$, a scalar rotation magnitude $\theta_{i,k} = \|\Delta\mathbf{r}_{i,k}\|_2$, and the log-scale variation $\Delta\boldsymbol{\ell}_{i,k} = \log\big(\mathbf{s}_i + \Delta\mathbf{s}_{i,k}\big) - \log(\mathbf{s}_i)$. The log-scale variation $\Delta\boldsymbol{\ell}_{i,k}$ is further decomposed into isotropic and anisotropic components, which are later summarized as $\ell^{\text{iso}}$ and $\ell^{\text{aniso}}$ in the global trajectory signature.

**Global Motion Trajectory Signature.** We summarize the sampled trajectories into a compact global trajectory signature that captures the overall motion behavior of each Gaussian. Specifically, we compute the mean and variance of position, rotation, and scale variations, forming a 13-dimensional signature:

$$\mathbf{v}_i = \Big[\underbrace{\Delta\bar{\mathbf{pos}}_i,\, \sigma_{\Delta\mathbf{pos}_i}}_{\text{position (6d)}},\ \underbrace{\bar{\theta}_i,\, \sigma_{\theta_i}}_{\text{rotation (2d)}},\ \underbrace{\bar{\ell}_i^{\text{iso}},\, \sigma_i^{\text{iso}}}_{\text{scale-iso (2d)}},\ \underbrace{\bar{\ell}_i^{\text{aniso}},\, \sigma_i^{\text{aniso}}}_{\text{scale-aniso (2d)}},\ \underbrace{s_i^{\text{intensity}}}_{\text{motion intensity (1d)}}\Big]^{\top}, \tag{8}$$

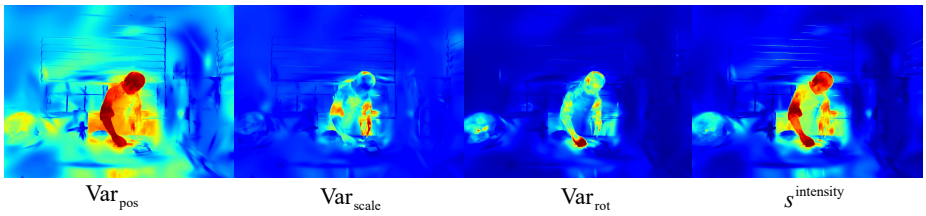


**Fig. 4:** Visualization of per-Gaussian motion variance heatmaps for individual attributes and the fused motion intensity score.

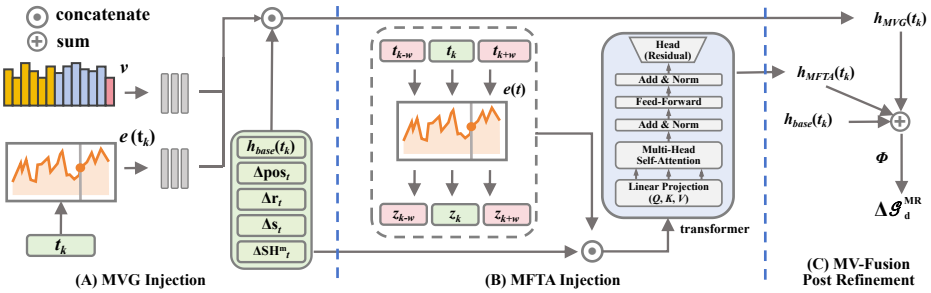


**Fig. 5:** Feature-space motion injection and fusion. (A) MVG encodes the cached global signature $\mathbf{v}_i$ and local intensity $e_i(t)$ into a residual feature $h_{\mathrm{MVG}}$. (B) MFTA forms temporal tokens $\mathbf{z}_{i,t}$ over a short window and aggregates them via cross-attention to obtain $h_{\mathrm{MFTA}}$. (C) The fused feature is decoded by the unchanged deformation heads $\Phi$ to predict refined deformation $\Delta\mathcal{G}_d^{\mathrm{MR}}$.

where $\bar{\cdot}$ and $\sigma_\cdot$ denote temporal mean and variance over the trajectory; the rotation term is 2D because we summarize the scalar rotation magnitude $\theta_{i,k} = \|\Delta\mathbf{r}_{i,k}\|_2$ by its mean and variance. The isotropic component $\ell^{\mathrm{iso}}$ measures global scale change, while $\ell^{\mathrm{aniso}}$ captures anisotropic deformation, yielding a more stable representation than directly modeling per-axis scale statistics. To further summarize motion magnitude, we introduce a scalar motion-intensity score:

$$s_i^{\mathrm{intensity}} = 0.7\,\sigma_{\mathrm{pos}}(i) + 0.2\,\sigma_{\mathrm{scale}}(i) + 0.1\,\sigma_{\mathrm{rot}}(i). \tag{9}$$

This score combines position, scale, and rotation variances with empirically chosen weights. Position deviation captures large displacements, scale variance reflects depth-dependent size changes, and rotation variance indicates local articulated motion. The fused intensity score integrates these complementary cues into a unified motion descriptor with clearer foreground–background contrast, serving as an effective motion prior for deformation feature extraction. Fig. 4 visualizes the attribute-wise variance maps and the fused score, showing its clearer foreground–background separation.

**Local Variance Dictionary.** While the global signature $\mathbf{v}_i$ captures long-term motion statistics, it does not describe short-term variations. We therefore introduce a lightweight local variance dictionary to provide instantaneous motion intensity at arbitrary query times. The dictionary stores local variance values at sparsely sampled timestamps and supports interpolation for unseen timestamps.

Given the displacement trajectory $\Delta\mathbf{pos}_{i,k}$, we define a local motion variance

$$e_i(t_k) = \sigma\left(\{\Delta\mathbf{pos}_{i,j}\}_{j=k-w}^{k+w}\right), \tag{10}$$

which measures the local motion variance within a short temporal window around time $t_k$. To reduce storage overhead, the local variance values are stored only at sparsely sampled timestamps. The value $e_i(t)$ at an arbitrary time is obtained by linear interpolation. For example, in a 300-frame sequence, storing only 10 samples per Gaussian reduces storage from 300 to 10 values (about 30× compression) while preserving the temporal structure of the motion signal.

**MVG Injection.** Fig. 5 (A) illustrates the MVG feature injection. The cached global trajectory signature and queried local motion intensity are encoded and injected into the deformation feature space. Given a Gaussian queried at time $t$, the deformation network produces a hidden feature $h_{i,t}$ and a coarse deformation prediction $\Phi(h_{i,t})$. We map the global trajectory signature $\mathbf{v}_i$ together with the local motion intensity $e_i(t)$ to the deformation feature dimension using a lightweight MLP, producing a motion feature $\mathbf{h}_{\text{MVG}} \in \mathbb{R}^{N\times H}$. This feature is then fused with the baseline deformation feature before the refinement heads.

Intuitively, MVG introduces trajectory-level motion priors into the deformation network, helping it distinguish static Gaussians from true movers and better model subtle or transient motion.

### 4.3 MotionFormer Temporal Attention (MFTA)

**Motivation.** While MVG provides trajectory-level motion statistics, it does not explicitly capture short-term temporal interactions around the current frame. Such local temporal context is crucial for resolving ambiguous motion patterns. We therefore introduce a lightweight temporal attention module, termed MotionFormer Temporal Attention (MFTA), to aggregate motion cues within a short temporal window as shown in Fig. 5 (B).

**Temporal Feature Construction.** Inspired by Timeformer [7], for a Gaussian queried at time $t_i$, we construct a temporal neighborhood $\mathcal{T}(t_i) = \{t_{i-w}, t_i, t_{i+w}\}$. For each timestamp $t \in \mathcal{T}(t_i)$, we use the corresponding deformation feature $h_{i,t}$, the predicted deformation value $\Phi(h_{i,t})$, and the local motion intensity $e_i(t)$ to form temporal tokens

$$\mathbf{z}_{i,t} = [\, h_{i,t},\, \Phi(h_{i,t}),\, e_i(t)\,]. \tag{11}$$

Here, $e_i(t)$ provides an instantaneous cue of motion intensity, helping the attention module emphasize informative temporal neighbors.

**MotionFormer Attention.** To aggregate temporal information, we perform cross-attention [25] where the current token attends to its temporal neighbors:

$$\mathbf{a}_{i,t_i} = \text{Attn}\big(\mathbf{z}_{i,t_i}, \mathbf{z}_{i,t_{i-w}}, \mathbf{z}_{i,t_{i+w}}\big), \tag{12}$$

where $\text{Attn}(Q, K, V)$ denotes a standard cross-attention operator with $Q = \mathbf{z}_{i,t_i}$.

**MFTA Injection.** The attention output is mapped by a lightweight MLP to produce a temporal motion feature $\mathbf{h}_{\text{MFTA}} \in \mathbb{R}^{N\times H}$, which is fused with the baseline deformation feature to form the final motion-aware representation.

### 4.4 Training Strategy

Training proceeds in three stages with progressively activated deformation components.

**Stage 1: Base Geometry.** In the first stage, only the Gaussian geometry is optimized while the deformation network is disabled. Aggressive densification and opacity reset are used to establish the initial scene structure.

**Stage 2: Base Deformation.** Once the geometry stabilizes, the baseline deformation network is activated to learn the fundamental spatiotemporal transformations. Near the end of this stage, the temporal branch is enabled to begin collecting motion statistics.

**Stage 3: Motion-Aware Refinement.** In the final stage, the full motion-aware refinement is activated. The zero-initialized MVG and MFTA branches are fused with the baseline deformation feature and trained end-to-end to produce refined deformation updates.

**Inference.** During training, the MVG motion descriptors are computed without back-propagation and cached for reuse. At inference time, the cached MVG descriptors are directly reused, and the temporal module can optionally be disabled for additional efficiency.

**Loss Function.** The training objective combines photometric and regularization losses:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{rgb}} + \lambda_2 \mathcal{L}_{\text{ssim}} + \lambda_3 \mathcal{L}_{\text{reg}}, \tag{13}$$

Here $\mathcal{L}_{\text{rgb}}$ is the L1 photometric loss, $\mathcal{L}_{\text{ssim}}$ encourages structural similarity, and $\mathcal{L}_{\text{reg}}$ includes standard Gaussian regularization terms (e.g., scale and aspect-ratio constraints).

**Implementation Details.** Our implementation is built upon the DeGauss framework in PyTorch. Training runs for 30k iterations using Adam on a single NVIDIA RTX 3090 GPU. Motion statistics are updated every 2000 iterations by sampling $K$=64 timestamps to estimate trajectory variance. For the temporal module, a short temporal window of $w$=5 neighboring frames is used to aggregate temporal context. All other settings follow the default configuration of the DeGauss framework.

## 5 Experiments

### 5.1 Experimental Settings

We evaluate our method on dynamic scene reconstruction and distractor-free reconstruction benchmarks. Additional implementation details and sensitivity analyses are provided in the supplementary material.

**NeRF On-the-Go** [19] contains casually captured real-world scenes with transient distractors such as pedestrians, cyclists, and vehicles. Each scene contains roughly one hundred images with varying occlusion levels, along with a small set of clean hold-out views for evaluation. Following DeGauss, we use seven scenes and train on the occluded images while evaluating novel view synthesis on the clean views.

**RobustNeRF** [21] contains static scenes with manually placed distractors that are repositioned or removed across captures, providing a complementary benchmark for distractor-free reconstruction under cross-view inconsistencies.

**Neu3D** [11] is a multi-view dynamic video dataset captured by around 20 synchronized static cameras over approximately 300 frames. Following the standard protocol, camera view 0 is used for testing and the remaining views for training. We report PSNR, SSIM, and LPIPS for image quality assessment.

### 5.2 Distractor-Free Scene Reconstruction

**Table 1:** Distractor-free scene reconstruction on NeRF On-the-go Dataset [19]. The best, second best, and third best are highlighted. Our method shows generally superior performance over state-of-the-art methods.

| | Mountain | | | Fountain | | | Corner | | | Patio | | | Spot | | | Patio-High | | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| RobustNeRF [21] | 17.54 | 0.496 | 0.383 | 15.65 | 0.318 | 0.576 | 23.04 | 0.764 | 0.244 | 20.39 | 0.718 | 0.251 | 20.65 | 0.625 | 0.391 | 20.54 | 0.578 | 0.366 | 19.64 | 0.583 | 0.369 |
| NeRF On-the-go [19] | 20.15 | 0.644 | 0.259 | 20.11 | 0.609 | 0.314 | 24.22 | 0.806 | 0.190 | 20.78 | 0.754 | 0.219 | 23.33 | 0.787 | 0.189 | 21.41 | 0.718 | 0.235 | 21.67 | 0.720 | 0.234 |
| 3DGS [8] | 19.40 | 0.638 | 0.213 | 19.96 | 0.659 | 0.185 | 20.90 | 0.713 | 0.241 | 17.48 | 0.704 | 0.199 | 20.77 | 0.693 | 0.316 | 17.29 | 0.604 | 0.363 | 19.30 | 0.668 | 0.253 |
| WildGaussians [9] | 20.43 | 0.653 | 0.255 | 20.81 | 0.662 | 0.215 | 24.16 | 0.822 | 0.139 | 21.44 | 0.800 | 0.138 | 23.82 | 0.816 | 0.138 | 22.23 | 0.725 | 0.206 | 22.16 | 0.746 | 0.182 |
| DeSplat [30] | 19.59 | 0.715 | 0.175 | 20.27 | 0.685 | 0.175 | 26.05 | 0.885 | 0.095 | 20.89 | 0.815 | 0.115 | 26.07 | 0.905 | 0.095 | 22.59 | 0.845 | 0.125 | 22.58 | 0.813 | 0.130 |
| SpotlessSplats [20] | 21.64 | 0.725 | 0.195 | 22.38 | 0.768 | 0.166 | 25.77 | 0.877 | 0.117 | 22.40 | 0.833 | 0.108 | 25.35 | 0.866 | 0.127 | 22.98 | 0.808 | 0.155 | 23.42 | 0.813 | 0.145 |
| DeGauss [28] | 22.31 | 0.746 | 0.163 | 22.40 | 0.764 | 0.139 | 25.94 | 0.869 | 0.078 | 22.88 | 0.850 | 0.087 | 26.59 | 0.886 | 0.089 | 23.35 | 0.799 | 0.124 | 23.91 | 0.819 | 0.113 |
| Ours | 22.44 | 0.755 | 0.127 | 22.33 | 0.764 | 0.119 | 25.58 | 0.868 | 0.061 | 22.77 | 0.870 | 0.047 | 27.31 | 0.891 | 0.057 | 23.20 | 0.805 | 0.097 | 23.94 | 0.826 | 0.085 |

Table 1 presents the quantitative comparison on the NeRF On-the-go benchmark for distractor-free static scene reconstruction. Our method ranks first across all three averaged metrics and achieves the lowest LPIPS on all six scenes, indicating consistently superior perceptual quality. Even on scenes where our PSNR is marginally below DeGauss, we obtain substantially better LPIPS, confirming that enhanced motion awareness leads to cleaner and more artifact-free background reconstruction.

We further report quantitative results for dynamic foreground regions in Table 2. Our method consistently outperforms DeGauss across all scenes and metrics, demonstrating that the proposed motion-aware refinement not only improves background stability but also enhances the reconstruction of dynamic objects.

Fig.s 6 and 7 visualize the background reconstruction quality on the NeRF On-the-Go benchmark. As highlighted by the red boxes, DeGauss retains pseudo-static foreground residuals in the background branch that are not captured by the deformation network, including a blurred pedestrian silhouette in Patio-High and diffuse smearing in Corner and Fountain. In contrast, MVFusion-GS substantially removes these artifacts, producing cleaner backgrounds with sharper edges and textures. The full composed predictions in Fig. 7 further show that

**Table 2:** Quantitative comparison of foreground dynamic region reconstruction on the NeRF On-the-go benchmark. Our method outperforms DeGauss across all scenes and all metrics.

| | Mountain | | | Fountain | | | Corner | | | Patio | | | Spot | | | Patio-High | | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| DeGauss | 21.81 | 0.803 | 0.126 | 19.79 | 0.694 | 0.175 | 21.70 | 0.851 | 0.123 | 25.56 | 0.903 | 0.076 | 27.11 | 0.832 | 0.189 | 21.63 | 0.689 | 0.323 | 22.93 | 0.795 | 0.169 |
| Ours | **25.62** | **0.851** | **0.092** | **22.52** | **0.808** | **0.091** | **28.23** | **0.919** | **0.053** | **31.57** | **0.941** | **0.039** | **28.92** | **0.880** | **0.129** | **28.73** | **0.869** | **0.149** | **27.60** | **0.878** | **0.092** |

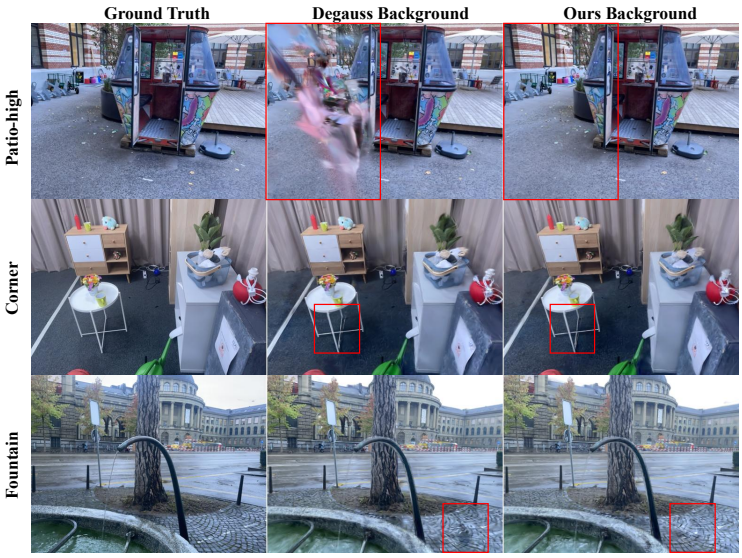


**Fig. 6:** Qualitative comparison of static background reconstruction on held-out test views from the NeRF On-the-go benchmark. Red boxes highlight regions where DeGauss retains visible pseudo-static foreground residuals (e.g., ghostly pedestrian silhouettes in Patio-High), while our method produces substantially cleaner backgrounds.

the improved dynamic-static separation preserves overall rendering quality, while the held-out test views in Fig. 6 demonstrate generalization beyond the training views.

Table 3 and Fig. 8 further show that MVFusion-GS generalizes to Robust-NeRF, where manually repositioned distractors create cross-view inconsistencies rather than continuous motion. The gains indicate that our motion-aware refinement also handles distractors that appear or disappear across views.

### 5.3 Dynamic Scene Reconstruction Results

Table 4 reports Neu3D results. MVFusion-GS achieves the best mean PSNR and LPIPS, and its unified variant improves over the single-branch baseline, validating the plug-in use of MVG/MFTA. MVFusion-GS also obtains the best dynamic-region PSNR of 31.78, outperforming SpaceTimeGS (30.85) and ST-4DGS (31.35).

Fig. 9 shows qualitative comparisons from the Neu3D dataset. In highly dynamic regions such as flames and rapidly moving objects, competing methods

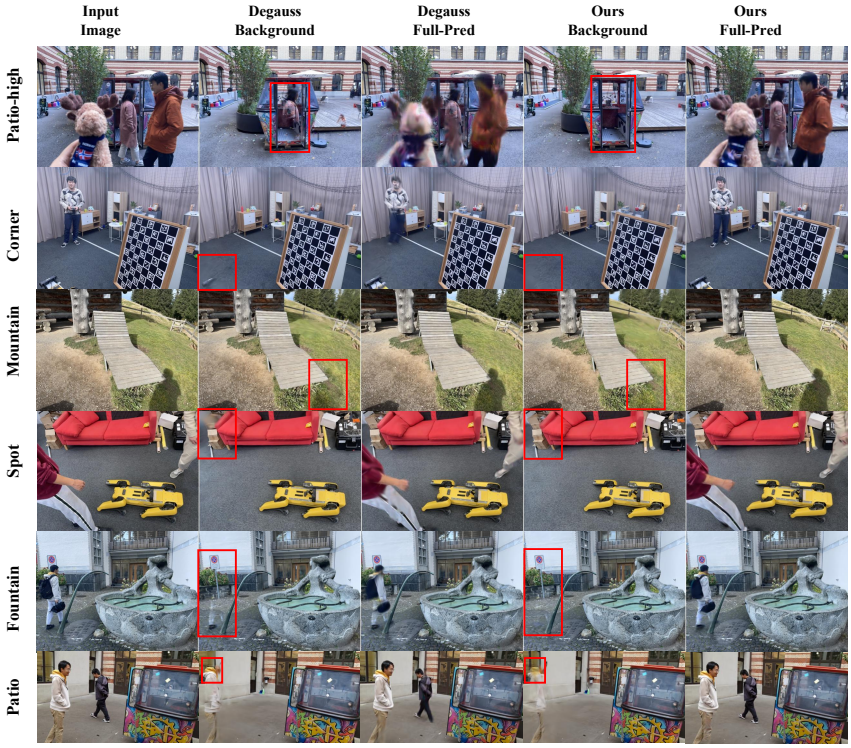


**Fig. 7:** Qualitative comparison of background-only and full composed renderings on NeRF On-the-go benchmark. Red boxes reveal that DeGauss backgrounds retain ghostly foreground residuals, while our method achieves cleaner backgrounds and sharper full-scene compositions.

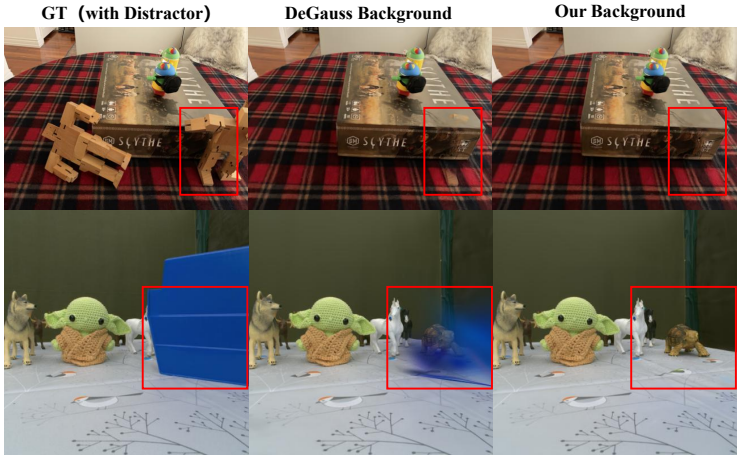


**Fig. 8:** Qualitative comparison on RobustNeRF. MVFusion-GS produces cleaner backgrounds than DeGauss.

**Table 3:** Quantitative comparison on the RobustNeRF [21] benchmark. The best, second best, and third best results are highlighted.

| | Android | | | Crab2 | | | Statue | | | Yoda | | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| 3DGS [8] | 23.32 | 0.794 | 0.159 | 31.76 | 0.925 | 0.172 | 20.83 | 0.830 | 0.148 | 28.92 | 0.905 | 0.192 | 26.21 | 0.863 | 0.168 |
| WildGaussians [9] | 24.67 | 0.828 | 0.151 | 30.52 | 0.909 | 0.213 | 22.54 | 0.863 | 0.129 | 30.55 | 0.905 | 0.202 | 27.07 | 0.876 | 0.174 |
| SpotlessSplats [20] | 24.20 | 0.810 | 0.159 | 33.90 | 0.933 | 0.169 | 21.97 | 0.821 | 0.163 | 34.24 | 0.938 | 0.156 | 28.58 | 0.875 | 0.162 |
| DeGauss [28] | 24.54 | 0.813 | 0.083 | 34.48 | 0.952 | 0.076 | 23.08 | 0.861 | 0.097 | 33.48 | 0.947 | 0.082 | 28.89 | 0.893 | 0.085 |
| MVFusion-GS (Ours) | 24.58 | 0.814 | 0.068 | 34.87 | 0.955 | 0.065 | 23.07 | 0.863 | 0.070 | 35.03 | 0.955 | 0.074 | 29.39 | 0.897 | 0.069 |

**Table 4:** Quantitative comparison on the Neu3D dataset. *MVFusion-GS (4DGS plug-in)* applies the proposed MVG/MFTA refinement to the single-branch 4DGS deformation pipeline, while *MVFusion-GS (Ours)* denotes the full decoupled model.

| | Cut Beef | | | Cook Spinach | | | Sear Steak | | | Flame Steak | | | Flame Salmon | | | Coffee Martini | | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| NeRFPlayer [23] | 31.83 | 0.928 | 0.119 | 32.06 | 0.930 | 0.116 | 32.31 | 0.940 | 0.111 | 27.36 | 0.867 | 0.215 | 26.14 | 0.849 | 0.233 | 32.05 | 0.938 | 0.111 | 30.29 | 0.909 | 0.151 |
| HexPlane [1] | 30.83 | 0.927 | 0.115 | 31.05 | 0.928 | 0.114 | 30.00 | 0.939 | 0.105 | 30.42 | 0.939 | 0.104 | 29.23 | 0.905 | 0.088 | 28.45 | 0.891 | 0.149 | 30.00 | 0.922 | 0.113 |
| K-Planes [3] | 31.82 | 0.966 | 0.114 | 32.60 | 0.966 | 0.114 | 32.52 | 0.974 | 0.104 | 32.39 | 0.970 | 0.102 | 30.44 | 0.953 | 0.132 | 29.99 | 0.953 | 0.134 | 31.63 | 0.964 | 0.117 |
| MixVoxels [26] | 31.30 | 0.965 | 0.111 | 31.65 | 0.965 | 0.113 | 31.43 | 0.971 | 0.103 | 31.21 | 0.970 | 0.108 | 29.92 | 0.945 | 0.163 | 29.36 | 0.946 | 0.147 | 30.81 | 0.960 | 0.124 |
| SWinGS [22] | 31.84 | 0.945 | 0.099 | 31.96 | 0.946 | 0.094 | 32.21 | 0.950 | 0.092 | 32.18 | 0.953 | 0.087 | 29.25 | 0.925 | 0.100 | 29.25 | 0.925 | 0.100 | 31.12 | 0.941 | 0.095 |
| 4DGS [31] | 32.66 | 0.946 | 0.053 | 32.46 | 0.949 | 0.052 | 32.49 | 0.957 | 0.041 | 32.75 | 0.954 | 0.040 | 29.00 | 0.912 | 0.081 | 27.34 | 0.905 | 0.083 | 31.12 | 0.937 | 0.058 |
| MangoGS [5] | 33.28 | 0.949 | 0.043 | 32.83 | 0.947 | 0.042 | 33.09 | 0.953 | 0.043 | 34.11 | 0.958 | 0.035 | 29.10 | 0.918 | 0.074 | 28.95 | 0.916 | 0.073 | 31.89 | 0.940 | 0.052 |
| DeGauss [28] | 32.56 | 0.957 | 0.042 | 32.61 | 0.950 | 0.041 | 33.20 | 0.956 | 0.035 | 32.75 | 0.955 | 0.034 | 29.23 | 0.916 | 0.068 | 28.80 | 0.916 | 0.062 | 31.52 | 0.942 | 0.047 |
| MVFusion-GS (4DGS plug-in) | 33.31 | 0.952 | 0.052 | 33.01 | 0.954 | 0.051 | 33.04 | 0.960 | 0.040 | 33.45 | 0.957 | 0.039 | 29.25 | 0.918 | 0.080 | 27.90 | 0.911 | 0.080 | 31.66 | 0.942 | 0.057 |
| MVFusion-GS (Ours) | 33.62 | 0.960 | 0.040 | 33.41 | 0.954 | 0.039 | 33.73 | 0.957 | 0.033 | 34.13 | 0.959 | 0.032 | 28.89 | 0.915 | 0.070 | 28.63 | 0.916 | 0.065 | 32.07 | 0.943 | 0.046 |

exhibit motion blur or artifacts, while our approach preserves clearer structural details and more consistent motion patterns. Meanwhile, the improvements are not limited to dynamic regions. Our method also produces cleaner backgrounds with fewer residual artifacts, leading to more coherent full-scene renderings.

### 5.4 Ablation Study

Table 5 evaluates the contribution of each component. Starting from the base deformation model without MVG and MFTA, adding the refinement branch without trajectory-level MVG statistics improves Neu3D PSNR from 31.52 to 31.64, while using only positional variance further improves it to 31.93. This shows that motion statistics are useful and that the full global trajectory signature is more effective than position-only motion cues.

Removing the temporal module (w/o MFTA) reduces the performance to 31.78 PSNR, indicating that temporal context aggregation helps capture dynamic motion patterns. Replacing the proposed cross-attention with a simpler self-attention design also lowers the performance to 32.01 PSNR, showing that cross-frame attention is more effective than processing temporal tokens without query-centered aggregation.

The full model achieves the best result, demonstrating that combining motion statistics with cross-frame temporal reasoning provides complementary benefits for dynamic scene reconstruction. Table 5 also reports ablation results on NeRF On-the-go. Since this benchmark evaluates static background reconstruction, the PSNR margins between variants are relatively small, as background regions are inherently less sensitive to deformation accuracy. However, the perceptual metrics reveal clear trends. Further ablations on motion-score fusion weights,

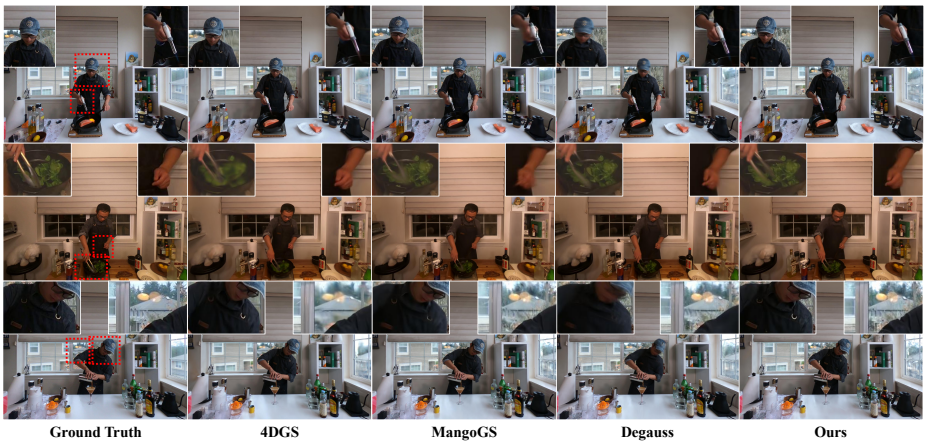


**Fig. 9:** Qualitative comparison on Neu3D. MVFusion-GS reconstructs sharper motion details with fewer artifacts.

**Table 5:** Ablation study of the proposed motion-aware refinement.

| Method | Neu3D | | | NeRF On-the-go | | |
|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| w/o MVG & MFTA (Base deformation) | 31.52 | 0.942 | 0.047 | 23.86 | 0.807 | 0.122 |
| w/o MVG (no motion statistics) | 31.64 | 0.937 | 0.052 | 23.87 | 0.810 | 0.114 |
| w/ MVG (position variance only) | 31.93 | 0.943 | 0.046 | 23.92 | 0.815 | 0.096 |
| w/o MFTA | 31.78 | 0.942 | 0.047 | 23.89 | 0.820 | 0.089 |
| w/o Cross-attention (Self-attention) | 32.01 | 0.943 | 0.046 | 23.93 | 0.822 | 0.090 |
| Full Model | **32.07** | **0.943** | **0.046** | **23.94** | **0.826** | **0.085** |

motion-statistics update interval, temporal window size, and temporal consistency are reported in the supplementary material.

## 6 Conclusion

We introduced MVFusion-GS, a motion-aware refinement framework for dynamic 3D Gaussian Splatting. By combining motion trajectory statistics and temporal attention, the proposed method improves motion modeling and dynamic–static separation. Experiments demonstrate consistent gains in both dynamic scene reconstruction and distractor-free reconstruction.

Despite these improvements, our method still depends on the baseline deformation field to provide reliable motion cues, and the estimated variance may become less discriminative when the initial deformation underfits complex foreground motion. In such challenging cases, especially under heavy occlusion or extremely weak motion evidence, residual artifacts may remain and motivate future work on more robust motion-aware Gaussian reassignment.

## References


1. Cao, A., Johnson, J.: HexPlane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 130–141 (June 2023)
2. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16569–16578 (2023)
3. 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)
4. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 Conference Papers. SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3641519.3657428
5. Huang, T., Zhu, H., Yong, J.H., Pan, H., Wang, B.: Mango-GS: Enhancing spatio-temporal consistency in dynamic scenes reconstruction using multi-frame node-guided 4d gaussian splatting. In: The Fourteenth International Conference on Learning Representations
6. 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 (CVPR). pp. 4220–4230 (2024)
7. Jiang, D., Hou, Z., Ke, Z., Yang, X., Zhou, X., Qiu, T.: Timeformer: Capturing temporal relationships of deformable 3d gaussians for robust reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8721–8732 (2025)
8. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
9. Kulhanek, J., Peng, S., Kukelova, Z., Pollefeys, M., Sattler, T.: WildGaussians: 3D gaussian splatting in the wild. In: Proceedings of the 38th International Conference on Neural Information Processing Systems (2024)
10. Li, D., Huang, S.S., Lu, Z., Duan, X., Huang, H.: ST-4DGS: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering. In: ACM SIGGRAPH 2024 Conference Papers (2024). https://doi.org/10.1145/3641519.3657520
11. Li, T., Slavcheva, M., Zollhöfer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., Lv, Z.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5521–5531 (June 2022)
12. 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 (CVPR). pp. 8508–8520 (June 2024)
13. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)

14. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: CVPR (2021)
15. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (Dec 2021). https://doi.org/10.1145/3503250
16. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics **41**(4), 102:1–102:15 (2022). https://doi.org/10.1145/3528223.3530127
17. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 165–174 (2019)
18. Park, J., Bui, M.Q.V., Bello, J.L.G., Moon, J., Oh, J., Kim, M.: SplineGS: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 26866–26875 (June 2025)
19. Ren, W., Zhu, Z., Sun, B., Chen, J., Pollefeys, M., Peng, S.: NeRF On-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8931–8940 (June 2024)
20. Sabour, S., Goli, L., Kopanas, G., Matthews, M., Lagun, D., Guibas, L., Jacobson, A., Fleet, D., Tagliasacchi, A.: Spotlesssplats: Ignoring distractors in 3d gaussian splatting. ACM Trans. Graph. **44**(2) (Apr 2025). https://doi.org/10.1145/3727143
21. Sabour, S., Vora, S., Duckworth, D., Krasin, I., Fleet, D.J., Tagliasacchi, A.: RobustNeRF: Ignoring distractors with robust losses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20626–20636 (June 2023)
22. Shaw, R., Nazarczuk, M., Song, J., Moreau, A., Catley-Chandar, S., Dhamo, H., Pérez-Pellitero, E.: Swings: sliding windows for dynamic 3d gaussian splatting. In: European Conference on Computer Vision. pp. 37–54. Springer (2024)
23. Song, L., Chen, A., Li, Z., Chen, Z., Chen, L., Yuan, J., Xu, Y., Geiger, A.: Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. IEEE Transactions on Visualization and Computer Graphics **29**(5), 2732–2742 (2023)
24. Stearns, C., Harley, A.W., Uy, M., Dubost, F., Tombari, F., Wetzstein, G., Guibas, L.: Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
26. Wang, F., Tan, S., Li, X., Tian, Z., Song, Y., Liu, H.: Mixed neural voxels for fast multi-view video synthesis pp. 19706–19716 (2023)
27. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4d reconstruction from a single video. In: International Conference on Computer Vision (ICCV) (2025)
28. Wang, R., Lohmeyer, Q., Meboldt, M., Tang, S.: DeGauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6294–6303 (October 2025)

29. Wang, Y., Yang, P., Xu, Z., Sun, J., Zhang, Z., Chen, Y., Bao, H., Peng, S., Zhou, X.: FreeTimeGS: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21750–21760 (2025)
30. Wang, Y., Klasson, M., Turkulainen, M., Wang, S., Kannala, J., Solin, A.: DeSplat: Decomposed gaussian splatting for distractor-free rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 722–732 (June 2025)
31. 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 (CVPR). pp. 20310–20320 (June 2024)
32. Wu, T., Zhong, F., Tagliasacchi, A., Cole, F., Oztireli, C.: $D^2$NeRF: self-supervised decoupling of dynamic and static objects from a monocular video. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
33. 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 (CVPR). pp. 20331–20341 (2024)
34. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)