# FS-GSTREAM: DISTRIBUTION-ADAPTIVE MOTION MASKING AND OPTIMIZATION FOR FLICKER-SUPPRESSED 3D GAUSSIAN STREAMING

**Anonymous authors**
Paper under double-blind review

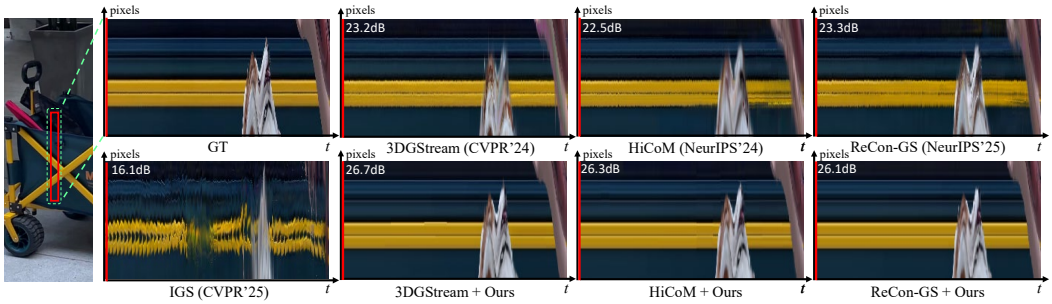


Figure 1: **Visualization of Spatiotemporal consistency.** We concatenate a fixed vertical scanline over time to reveal flickering artifacts. Existing online reconstruction methods Sun et al. (2024); Yan et al. (2025); Gao et al. (2024a); Fu et al. (2025) show noisy and unstable patterns, while our method substantially reduces flickering when applied to 3DGStream Sun et al. (2024), HiCoM Gao et al. (2024a) and ReCon-GS Fu et al. (2025).

## ABSTRACT

The online reconstruction of dynamic scenes from multi-view video streams enables fast free-viewpoint video synthesis, but existing streaming 3DGS methods often suffer from temporal flickering with geometry degradation. Temporal flickering partly arises from the difficulty of distinguishing true scene motion from static background during per-frame optimization, leading to unnecessary changes in static scene representations even when they do not change. In this paper, we propose FS-GStream, distribution-adaptive motion masking and optimization for flicker-suppressed 3D Gaussian streaming, which can be easily integrated as a generalizable plug-and-play module with two-stage streaming 3D Gaussian reconstruction models. Our FS-GStream *significantly* improves temporal consistency through two main components: Distribution-Adaptive Motion Masking (DAMM) and Mask-Guided Optimization (MGO). The DAMM first estimates a frame-wise motion threshold from the estimated distribution of log-transformed Gaussian translation magnitudes, which stably separates static and dynamic Gaussians in a statistical manner without manually setting a fixed displacement threshold. Guided by the masks that distinguish static and dynamic Gaussians from the DAMM, the MGO jointly refines the geometry and appearance of dynamic Gaussians while preserving static geometry and selectively adapting static appearance, and restricts densification and pruning to dynamic Gaussians. Extensive experiments on state-of-the-art two-stage streaming baselines demonstrate the generalizability of our plug-and-play design, which *consistently alleviates temporal flickering* and *significantly improves qualitative and quantitative performance in highly dynamic scenes*.

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.

# 1 INTRODUCTION

Reconstructing dynamic 3D scenes from multi-view videos is an important task for immersive applications such as VR, AR, and free-viewpoint sports broadcasting. Recent advances in 4D reconstruction Luiten et al. (2024); Yang et al. (2024; 2023); Huang et al. (2024); Wu et al. (2024); Lei et al. (2025); Park et al. (2025); Lee et al. (2024); Wu et al. (2025b) have achieved photorealistic rendering by optimizing over complete video sequences, but their offline nature and long training time make them difficult to deploy in real-time scenarios. To address this limitation, streaming 3D Gaussian Splatting (3DGS) methods Yan et al. (2025); Gao et al. (2024a); Fu et al. (2025); Sun et al. (2024); Liu et al. (2024); Bao et al. (2025; 2026); Girish et al. (2024); Hu et al. (2025); Chen et al. (2025a); Yun et al. (2025) have emerged, enabling online per-frame reconstruction from newly arrived observations under a limited optimization budget. However, learning from only consecutive frames makes streaming reconstruction highly sensitive to temporal inconsistency, resulting in flickering artifacts. In recent streaming 3DGS methods Sun et al. (2024); Gao et al. (2024a); Fu et al. (2025), static Gaussians can undergo unnecessary updates even when the underlying geometry remains unchanged. Under limited temporal supervision, motion optimization can assign displacement updates to static Gaussians, while residual photometric errors can repeatedly trigger densification in the static background. Global opacity-based pruning can further remove well-optimized static Gaussians, progressively degrading the persistent background representation. To mitigate these issues, recent studies Yan et al. (2025); Liu et al. (2024); Bao et al. (2025; 2026) attempt to decouple dynamic and static Gaussians. Despite their effectiveness, these methods often rely on pre-trained optical flow models, manual annotations, or fixed displacement thresholds to define motion masks. Such dependencies limit their generalizability, especially in fully automated streaming settings where motion distributions and noise levels vary over time.

To address these issues, we propose FS-GStream, a generalizable plug-and-play module for two-stage streaming 3D Gaussian reconstruction. FS-GStream consists of two main components: Distribution-Adaptive Motion Masking (DAMM) and Mask-Guided Optimization (MGO). First, the DAMM estimates a frame-wise motion threshold from the distribution of Gaussian translation magnitudes. Instead of manually tuning a fixed displacement threshold, we estimate the per-frame density of log-transformed translation magnitudes using Kernel Density Estimation and select the motion threshold from its density. This allows DAMM to separate true scene motion from the static background with a threshold adapted to the motion distribution of each frame. Second, guided by the masks that distinguish static and dynamic Gaussians from the DAMM, the MGO jointly refines the geometry and appearance of dynamic Gaussians to capture residual motion and appearance changes, while preserving the geometry of static Gaussians and selectively adapting their appearance based on photometric residuals. The MGO restricts densification and pruning to dynamic and newly added Gaussians, limiting unnecessary changes to the persistent static representation. Because FS-GStream derives its motion masks solely from per-Gaussian translation magnitudes, it does not depend on a specific parameterization of the backbone's motion model. Therefore, our FS-GStream can be integrated into existing two-stage streaming 3DGS pipelines Gao et al. (2024a); Fu et al. (2025); Sun et al. (2024) having motion alignment and continual Gaussian refinement with minimal modification. By adding our DAMM and MGO to these backbones, FS-GStream consistently improves temporal stability and rendering quality without relying on pre-trained motion estimators or fixed displacement thresholds. Our main contributions are as follows:

- We propose Distribution-Adaptive Motion Masking (DAMM), a non-parametric thresholding strategy that estimates a frame-wise motion threshold from the Kernel Density Estimation (KDE)-based distribution of log-transformed Gaussian translation magnitudes, without manually tuning a fixed displacement threshold or relying on pre-trained flow models.
- We propose Mask-Guided Optimization (MGO) to jointly refine dynamic geometry and appearance while preserving static geometry and selectively adapting static appearance. MGO further restricts densification and pruning to dynamic Gaussians to reduce unnecessary changes to the persistent representation, which helps reducing flickering artifacts.
- We demonstrate that, as a generalizable plug-and-play module, our FS-GStream can be seamlessly integrated into two-stage SOTA streaming baselines with motion alignment and continual Gaussian refinement, such as 3DGStream, HiCoM and ReCon-GS, consistently improving their temporal stability, e.g., reducing $E_{\mathrm{warp}}$ by 37.3% on average, and improving rendering quality by an average of +1.52 dB PSNR across dynamic datasets.

# 2 Related Work

## 2.1 Offline Dynamic Gaussian Splatting

The advent of 3D Gaussian Splatting (3DGS) Kerbl et al. (2023) has encouraged high-fidelity scene reconstruction by offering explicit point-based representation and high-speed differentiable rendering. To extend this paradigm to dynamic environments, various offline methods Luiten et al. (2024); Yang et al. (2024; 2023); Huang et al. (2024); Wu et al. (2024); Lei et al. (2025); Park et al. (2025); Lee et al. (2024); Wu et al. (2025b); Li et al. (2025b; 2023); Wang et al. (2025); Kwak et al. (2025b;a); Luo et al. (2025); Oh et al. (2025); Gao et al. (2024b); Wu et al. (2025a); Shen et al. (2025); Yin et al. (2025); Chen et al. (2025b); Li et al. (2025a) have been developed. One line of research directly expands the attributes of Gaussian primitives to model dynamics by adding a temporal dimension to formulate 4D Gaussians Yang et al. (2023); Duan et al. (2024) or explicitly incorporating per-frame positions and rotations into the primitives Luiten et al. (2024). Another approach Yang et al. (2024); Wu et al. (2024) implicitly models scene dynamics by decomposing them into a base representation in canonical space coupled with a time-varying motion field, typically employing Multi-Layer Perceptrons (MLPs) to regress per-frame Gaussian residuals. Alternatively, some methods Lei et al. (2025); Park et al. (2025); Lee et al. (2024) utilize explicit motion modeling, employing mathematical structures such as motion scaffolds or adaptive splines to directly parameterize Gaussian trajectories. Furthermore, Swift4D Wu et al. (2025b) introduces an adaptive divide-and-conquer approach by masking to efficiently optimize static and dynamic regions separately. Despite diverse modeling strategies and high-fidelity results, all offline methods require access to the complete video sequence for training.

## 2.2 Online Streamable Dynamic Gaussian Splatting

While offline dynamic 3DGS methods achieve high-fidelity novel view synthesis, their strict requirement for complete video sequences makes them fundamentally unsuitable for real-time applications. To circumvent this, online streaming frameworks operate under a strict constraint, processing frames sequentially on the fly. One line of research in this domain focused on tracking scene dynamics through motion modeling. For instance, 3DGStream Sun et al. (2024) introduced a Neural Transformation Cache (NTC) to model per-frame transformations, while subsequent works like HiCoM Gao et al. (2024a) and ReCon-GS Fu et al. (2025) proposed hierarchical coherent motions and anchor-based graphs, respectively, to capture topological deformations. A concurrent direction Yan et al. (2025); Liu et al. (2024); Bao et al. (2025; 2026) is to utilize masking techniques that attempt to explicitly isolate dynamic Gaussians. IGS Yan et al. (2025) leverages a pre-trained optical flow model and uses dynamic bounding boxes during the streaming phase, which require additional external information. Similarly, DASS Liu et al. (2024) and MGStream Bao et al. (2025) utilize optical flows combined with heuristic thresholding to guide the optimization of dynamic Gaussians. However, integrating heavy external networks imposes a severe computational burden and makes their performance depend on the quality of the estimated 2D priors, such as optical flow. Alternatively, CPOStream Bao et al. (2026) attempted to achieve flicker-free rendering without relying on external networks by collaborating prediction and observation steps. However, to classify dynamic Gaussians and control refinement, densification, and pruning, it still relies on predefined, manually set thresholds. In summary, existing streaming 3DGS methods either rely on manual, fixed thresholds to identify dynamic Gaussians or utilize pre-trained optical flow models, which limits their generalization across varying motion distributions. To address these limitations without relying on external networks or predefined fixed motion thresholds, we propose Distribution-Adaptive Motion Masking (DAMM).

# 3 Methodology

## 3.1 Preliminaries

In the standard streaming 3DGS setting, the process begins with a base Gaussian set trained on the initial frame ($t = 0$), and each subsequent frame is learned online only using its previous reconstruction and the newly arrived multi-view observations in the current frame. To achieve real-time per-frame reconstruction under this highly constrained condition, recent SOTA streaming 3DGS methods Gao et al. (2024a); Fu et al. (2025); Sun et al. (2024) commonly adopt a two-stage optimization paradigm. Since our FS-GStream is based on existing two-stage streaming methods

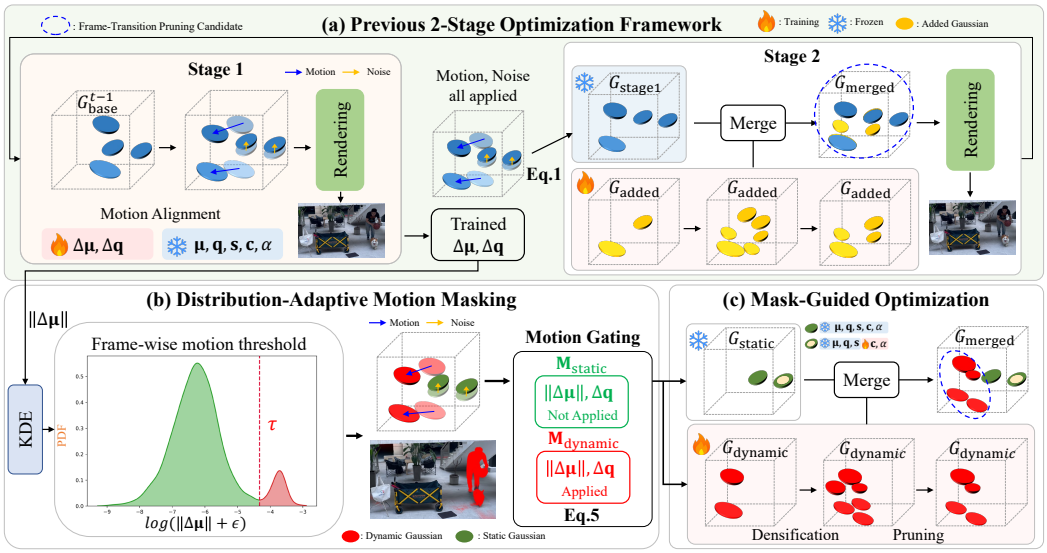


Figure 2: **Overview of FS-GStream.** (a) A two-stage streaming 3DGS pipeline with motion alignment in Stage 1 and continual refinement in Stage 2. (b) Distribution-Adaptive Motion Masking (DAMM) estimates a frame-wise motion threshold from the KDE density of translation magnitudes and separates dynamic and static Gaussians through motion masking. (c) Mask-Guided Optimization (MGO) jointly refines dynamic geometry and appearance, preserves static geometry with selective appearance updates, and restricts densification and pruning to dynamic Gaussians.

with motion alignment and continual refinement, we briefly describe the motion alignment in Stage 1 and the continual refinement in Stage 2.

**Stage 1: Motion Alignment.** Given a base Gaussian set $G_{\text{base}}^{t-1}$ from the previous frame at time $t-1$, the model predicts per-Gaussian motion updates in the current frame at time $t$, typically represented by translation ($\Delta\boldsymbol{\mu}$) and rotation ($\Delta\mathbf{q}$), to align $G_{\text{base}}^{t-1}$ with the current frame at time $t$. In this stage, the Gaussian attributes, i.e., position $\boldsymbol{\mu}$, rotation $\mathbf{q}$, scale $\mathbf{s}$, SH coefficients $\mathbf{c}$, and opacity $\alpha$, of $G_{\text{base}}^{t-1}$ are frozen, and only the motion parameters ($\Delta\boldsymbol{\mu}$, $\Delta\mathbf{q}$) are optimized by minimizing the photometric reconstruction loss $\mathcal{L}_{\text{photo}}$. Then, $\boldsymbol{\mu}^t$ and $\mathbf{q}^t$ are updated as:

$$\boldsymbol{\mu}^t = \boldsymbol{\mu}^{t-1} + \Delta\boldsymbol{\mu}, \quad \mathbf{q}^t = norm(\mathbf{q}^{t-1}) \times norm(\Delta\mathbf{q}), \tag{1}$$

where $\times$ denotes quaternion multiplication and $norm$ denotes normalization. The set of updated Gaussians with $\boldsymbol{\mu}^t$ and $\mathbf{q}^t$ at time $t$ in Stage 1 is denoted as $G_{\text{stage1}}$.

**Stage 2: Continual Refinement.** In this stage, the motion-applied Gaussian set $G_{\text{stage1}}$ is frozen. To address the remaining discrepancies between $G_{\text{stage1}}$ and the current frame at $t$, and to capture newly emerging details in the current frame, a set of new Gaussians $G_{\text{added}}$ is introduced through densification in the regions with high accumulated gradients. Here, only $G_{\text{added}}$ is optimized, and the final scene representation is formed by merging both Gaussian sets as $G_{\text{merged}} = G_{\text{stage1}} \cup G_{\text{added}}$. To maintain a compact representation, $G_{\text{merged}}$ is pruned before being carried to the next frame, a process we refer to as *frame-transition pruning*. In the original count-based formulation, this step removes $|G_{\text{added}}|$ Gaussians with the lowest opacity from the entire $G_{\text{merged}}$, matching the number removed to the number added. The resulting set becomes $G_{\text{base}}^t$ for the next frame.

**Limitation.** Two-stage methods can introduce unnecessary changes to static scene representations even when the underlying geometry remains unchanged. These changes can destabilize the reconstruction over time in three ways. First, since Stage 1 is supervised only by consecutive-frame observations, static Gaussians receive small displacement updates, and this optimization-induced displacement noise is baked into $G_{\text{stage1}}$ as jitter. Second, regions poorly reconstructed at the initial frame keep accumulating large gradients, so Stage 2 repeatedly densifies the static background with Gaussians fitted only to the current frame. Third, the global opacity ranking used for frame-transition pruning can remove well-optimized static Gaussians to accommodate newly added ones, progressively eroding the background. These effects manifest as temporal flickering and background degradation.

### 3.2 Overview of Our FS-GStream

Fig. 2 illustrates the overview of our proposed FS-GStream. Our primary objective is to develop a generalizable plug-and-play module for two-stage streaming 3D Gaussian reconstruction that suppresses temporal flickering by separating dynamic and static Gaussians. To address the limitations of existing two-stage streaming pipelines discussed in Sec. 3.1, FS-GStream is seamlessly integrated into such pipelines Sun et al. (2024); Gao et al. (2024a); Fu et al. (2025) by adding a distribution-adaptive masking and optimization process, as illustrated in Fig. 2. Specifically, FS-GStream consists of two core components: (i) First, Distribution-Adaptive Motion Masking (DAMM) estimates a frame-wise motion threshold from the motion distribution of each frame and classifies Gaussians into dynamic and static subsets, as described in Sec. 3.3; (ii) Second, guided by the estimated masks, Mask-Guided Optimization (MGO) jointly refines the geometry and appearance of dynamic Gaussians, selectively adapts static appearance while preserving static geometry, and restricts densification and pruning to dynamic Gaussians, as detailed in Sec. 3.4. This unified design supports accurate reconstruction of dynamic content while reducing temporal flickering and preserving the persistent static representation.

### 3.3 Distribution-Adaptive Motion Masking (DAMM)

Our DAMM is a non-parametric masking strategy that adapts the static-dynamic separation to the frame-wise distribution of Gaussians' translation magnitudes (displacement) $\|\Delta\boldsymbol{\mu}\|_1$.

**Motion Alignment and the static noise.** To capture temporal dynamics, we first perform the motion alignment of Stage 1 in Sec. 3.1. Although the regularization loss $\mathcal{L}_{\text{reg}} = \|\Delta\boldsymbol{\mu}\|_1$ encourages sparse motion, photometric optimization can still induce non-zero displacement estimates for Gaussians representing static scene regions. We refer to these spurious displacements as *static displacement noise (SDN)*. Since the magnitude and spread of SDN vary across frames and motion distributions, a fixed threshold can lead to misclassification between static and dynamic Gaussians. We therefore utilize the distribution of the translation magnitudes ($\|\Delta\boldsymbol{\mu}\|_1$) obtained in Stage 1 to adaptively estimate frame-wise motion thresholds.

**Distribution Reshaping by Log-Transformation.** The raw translation magnitudes $\|\Delta\boldsymbol{\mu}\|_1$ are typically concentrated around a near-zero SDN, with a smaller population spread over larger values associated with scene motion. To make these scales easier to separate, we map the raw translation magnitudes to log-space motion scores as $a_i = \log(\|\Delta\boldsymbol{\mu}_i\|_1 + \epsilon)$, where $\epsilon$ is a small constant. This transformation expands the near-zero ranges of the distribution and compresses the large magnitude range, making the density easier to be estimated by using the Kernel Density Estimation (KDE) technique Chen (2017), so that the separation between static noise and true scene motion becomes easier by thresholding from the distribution.

**Distribution-Adaptive Threshold Selection.** For each frame $t$, we estimate a continuous density over its $N$ motion scores $\{a_i\}_{i=1}^{N}$ using KDE:

$$\hat{f}_t(a) = \frac{1}{N h_t} \sum_{i=1}^{N} K\left(\frac{a - a_i}{h_t}\right), \tag{2}$$

where $K$ is the kernel function and the bandwidth $h_t$ is selected independently for each frame using Scott's rule Scott (2011). The distribution of SDN magnitudes is expected to form a peak mode at a low displacement and gets decayed beyond it. In case that dynamic motion exists, the second mode tends to appear (See Appendix C). The appearance of such peak and valley modes becomes more prominent in the log scale of the displacement magnitude axis. We therefore identify the leftmost prominent density peak as the SDN mode and use the first subsequent valley as the boundary between SDN and true scene motion. However, small density fluctuations can introduce spurious local maxima (peaks). To exclude such maxima from SDN mode selection, we apply the relative topographic prominence criterion Virtanen et al. (2020) to peaks. For a local maximum $x$ of a function $g$, $\text{Prom}_g(x)$ measures how far the peak rises above its surrounding baseline. Details are provided in Appendix B.1.

We first retain density peaks whose prominence is at least a fraction $\rho_{\text{prom}}$ of the maximum density:

$$\mathcal{P}_t = \left\{ p \in \text{LocMax}(\hat{f}_t) \;\middle|\; \text{Prom}_{\hat{f}_t}(p) \geq \rho_{\text{prom}} \max_a \hat{f}_t(a) \right\}, \tag{3}$$

where $\mathrm{LocMax}$ denotes the set of local maxima, and $\rho_{\mathrm{prom}}$ controls the minimum required prominence relative to the maximum density and is set to 0.05 for all experiments. When $\mathcal{P}_t$ is nonempty, we select its leftmost peak, $p_t = \min \mathcal{P}_t$, as an estimate of the SDN mode. We then set $\tau_{\mathrm{frame}}$ to the first local minimum to the right of $p_t$, i.e., $\tau_{\mathrm{frame}} = \min\{v \in \mathrm{LocMin}(\hat{f}_t) \mid v > p_t\}$. If no peak or qualifying valley is detected, we use the Exponential Moving Average (EMA) of thresholds from previous frames with valid detections, updating the EMA only when a valid detection is available.

**Motion Gating and Masking.** $\tau_{\mathrm{frame}}$ is used for frame-wise motion gating to separate static and dynamic Gaussians. For each Gaussian $\mathrm{g}_i \in G^{t-1}_{\mathrm{base}}$, $i = 1, \ldots, N$, a binary dynamic mask to distinguish the dynamic Gaussians from the static ones is defined as:

$$\mathcal{M}^i_{\mathrm{dynamic}} = \mathbb{I}\left[\log(\|\Delta\boldsymbol{\mu}_i\|_1 + \epsilon) > \tau_{\mathrm{frame}}\right], \tag{4}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function that outputs 1 if the condition is met and 0 otherwise. The static mask is then defined as $\mathcal{M}^i_{\mathrm{static}} = 1 - \mathcal{M}^i_{\mathrm{dynamic}}$.

We apply the motion updates $(\Delta\boldsymbol{\mu}_i, \Delta\mathbf{q}_i)$ estimated in Stage 1 only to the Gaussians classified as dynamic by $\mathcal{M}_{\mathrm{dynamic}}$. For each Gaussian $\mathrm{g}_i$, the mask-gated motion update is computed as:

$$(\boldsymbol{\mu}^t_i, \mathbf{q}^t_i) = \begin{cases} \left(\boldsymbol{\mu}^{t-1}_i + \Delta\boldsymbol{\mu}_i,\, \mathrm{norm}(\mathbf{q}^{t-1}_i) \times \mathrm{norm}(\Delta\mathbf{q}_i)\right), & \mathcal{M}^i_{\mathrm{dynamic}} = 1, \\ \left(\boldsymbol{\mu}^{t-1}_i,\, \mathbf{q}^{t-1}_i\right), & \mathcal{M}^i_{\mathrm{static}} = 1. \end{cases} \tag{5}$$

This update preserves the positions and rotations of static Gaussians. All remaining attributes, i.e., $\mathbf{s}_i$, $\mathbf{c}_i$, and $\alpha_i$, retain their values from $G^{t-1}_{\mathrm{base}}$ in this step (Stage 1). The resulting motion-gated Gaussian set and the corresponding masks are used for the subsequent Mask-Guided Optimization.

### 3.4 Mask-Guided Optimization (MGO)

After motion alignment, dynamic Gaussians require further refinement of both geometry and appearance to capture residual motion and appearance changes. Static Gaussians may also require appearance adaptation while retaining their geometry. MGO uses $\mathcal{M}_{\mathrm{dynamic}}$ and $\mathcal{M}_{\mathrm{static}}$ to control parameter updates, densification, and pruning according to these different refinement requirements.

**Mask-Guided Parameter Updates.** Dynamic and newly added Gaussians are optimized in both geometric and appearance parameters to model residual motion and appearance changes following Bao et al. (2025; 2026). Updating static geometry at every frame can introduce drift caused by residual photometric errors. However, freezing all attributes also prevents static Gaussians from adapting to appearance changes. We therefore fix their geometric parameters $\theta^{\mathrm{geo}}_i = (\mu_i, q_i, s_i)$ while selectively updating appearance parameters $\theta^{\mathrm{app}}_i = (c_i, \alpha_i)$, comprising SH coefficients and opacity. Let $w^{(k)}_i$ denote a binary mask selecting static Gaussians for appearance updates at Stage 2 iteration $k$. The mask is initially zero. After initial refinement, we select static Gaussians using the multi-view error score of FastGS Ren et al. (2025), which averages high-error pixel counts within projected Gaussian footprints across views. The masked gradients of the photometric loss $\mathcal{L}_{\mathrm{photo}}$ are

$$\widetilde{\nabla}_{\theta^{\mathrm{geo}}_i}\mathcal{L}_{\mathrm{photo}} = \mathcal{M}^i_{\mathrm{dynamic}}\nabla_{\theta^{\mathrm{geo}}_i}\mathcal{L}_{\mathrm{photo}}, \qquad \widetilde{\nabla}_{\theta^{\mathrm{app}}_i}\mathcal{L}_{\mathrm{photo}} = (\mathcal{M}^i_{\mathrm{dynamic}} \vee w^{(k)}_i)\nabla_{\theta^{\mathrm{app}}_i}\mathcal{L}_{\mathrm{photo}}. \tag{6}$$

where $\widetilde{\nabla}$ denotes the masked gradient used for parameter updates. Static Gaussians retain their positions, rotations, and scales, while those selected by $w^{(k)}_i$ can update their spherical harmonics coefficients and opacity. The mask construction and update schedules are detailed in **Appendix**.

**Mask-Guided Densification and Pruning.** Motion gating suppresses displacement updates to static Gaussians, but does not by itself prevent densification in static regions. Persistent photometric residuals, including those caused by imperfect initial reconstruction, can produce large view-space gradients and repeatedly trigger densification. The resulting changes to the static Gaussian set can introduce temporal flickering and increase the resulting representation size. We therefore restrict densification candidates to dynamic Gaussians with $\mathcal{M}^i_{\mathrm{dynamic}} = 1$ and confine Stage 2 densification and pruning to the dynamic and newly added Gaussian sets. The static Gaussians selected for appearance updates remain excluded from these operations.

**Mask-Guided Frame-Transition Pruning.** The original count-based frame-transition pruning rule can remove well-optimized static Gaussians to accommodate newly added ones as described in

Table 1: **Quantitative comparison on all datasets.** Our FS-GStream is applied to 3DGStream, HiCoM, and ReCon-GS, while IGS is included as a stand-alone streaming model.

| Datasets | Methods | Publications | $E_{\text{warp}}\downarrow$ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Payload (MB) ↓ | Train (sec) ↓ |
|---|---|---|---|---|---|---|---|---|
| SelfCap | IGS Yan et al. (2025) | CVPR'25 | 0.034 | 22.64 | 0.797 | 0.278 | 34.79 | 2.6 |
| | 3DGStream Sun et al. (2024) | CVPR'24 | 0.022 | 24.40 | 0.813 | 0.277 | 4.41 | 10.5 |
| | 3DGStream + FS-GStream | – | **0.014** | **25.36** | **0.833** | **0.252** | 6.00 | **4.4** |
| | HiCoM Gao et al. (2024a) | NeurIPS'24 | 0.030 | 23.73 | 0.789 | 0.313 | 16.40 | 6.4 |
| | HiCoM + FS-GStream | – | **0.020** | **25.34** | **0.839** | **0.243** | **7.07** | **5.2** |
| | ReCon-GS Fu et al. (2025) | NeurIPS'25 | 0.032 | 23.73 | 0.794 | 0.311 | 11.39 | 6.0 |
| | ReCon-GS + FS-GStream | – | **0.014** | **25.37** | **0.837** | **0.244** | **6.44** | **4.1** |
| VRU | IGS Yan et al. (2025) | CVPR'25 | 0.044 | 23.40 | 0.827 | 0.245 | 41.45 | 1.9 |
| | 3DGStream Sun et al. (2024) | CVPR'24 | 0.034 | 26.06 | 0.871 | 0.204 | 4.54 | 7.5 |
| | 3DGStream + FS-GStream | – | **0.026** | **27.21** | **0.886** | **0.184** | 8.20 | **2.7** |
| | HiCoM Gao et al. (2024a) | NeurIPS'24 | 0.055 | 23.76 | 0.797 | 0.332 | 20.26 | 3.9 |
| | HiCoM + FS-GStream | – | **0.028** | **26.84** | **0.884** | **0.189** | **5.23** | **3.5** |
| | ReCon-GS Fu et al. (2025) | NeurIPS'25 | 0.044 | 25.32 | 0.843 | 0.259 | 12.76 | 3.5 |
| | ReCon-GS + FS-GStream | – | **0.025** | **27.22** | **0.889** | **0.181** | **6.42** | **2.3** |
| Meet Room | IGS Yan et al. (2025) | CVPR'25 | 0.015 | 29.35 | 0.938 | 0.189 | 4.89 | 1.2 |
| | 3DGStream Sun et al. (2024) | CVPR'24 | 0.011 | 31.36 | 0.956 | 0.174 | 4.05 | 3.7 |
| | 3DGStream + FS-GStream | – | **0.007** | **31.62** | 0.955 | **0.172** | 4.33 | **2.8** |
| | HiCoM Gao et al. (2024a) | NeurIPS'24 | 0.012 | 29.09 | 0.936 | 0.208 | 2.10 | 2.9 |
| | HiCoM + FS-GStream | – | **0.008** | **32.36** | **0.962** | **0.161** | **2.04** | **2.8** |
| | ReCon-GS Fu et al. (2025) | NeurIPS'25 | 0.009 | 30.51 | 0.949 | 0.184 | 1.56 | 2.7 |
| | ReCon-GS + FS-GStream | – | **0.006** | **32.51** | **0.963** | **0.161** | **1.51** | **2.1** |
| N3DV | IGS Yan et al. (2025) | CVPR'25 | 0.016 | 31.10 | 0.943 | 0.133 | 10.22 | 2.0 |
| | 3DGStream Sun et al. (2024) | CVPR'24 | 0.014 | 30.81 | 0.946 | 0.133 | 8.01 | 8.1 |
| | 3DGStream + FS-GStream | – | **0.007** | **31.50** | 0.944 | 0.137 | **4.41** | **6.4** |
| | HiCoM Gao et al. (2024a) | NeurIPS'24 | 0.015 | 30.91 | 0.941 | 0.143 | 5.75 | 5.8 |
| | HiCoM + FS-GStream | – | **0.010** | **31.83** | **0.948** | **0.131** | **4.20** | **5.1** |
| | ReCon-GS Fu et al. (2025) | NeurIPS'25 | 0.010 | 31.10 | 0.945 | 0.138 | 4.03 | 5.7 |
| | ReCon-GS + FS-GStream | – | **0.008** | **31.83** | **0.947** | **0.133** | **2.64** | **3.2** |

Sec. 3.1. We instead preserve all static Gaussians and remove dynamic and newly added Gaussians only when their opacity falls below a fixed threshold $\alpha_{\text{th}}$. Let $G_{\text{static}}$ and $G_{\text{dynamic}}$ denote the static and dynamic subsets of the existing Gaussians after Stage 2, respectively. The Gaussian set carried to the next frame is defined as:

$$G^t_{\text{base}} = G_{\text{static}} \cup \{g_i \in G_{\text{dynamic}} \cup G_{\text{added}} \mid \alpha_i \geq \alpha_{\text{th}}\}. \tag{7}$$

where $\alpha_{\text{th}}$ is empirically set to $0.1$ for all experiments. This mask-guided frame-transition pruning can preserve the static Gaussian set during frame transitions, regardless of its opacity values.

# 4 Experiments

## 4.1 Datasets and Metrics

We evaluate FS-GStream on four multi-view dynamic scene datasets with different motion scales and scene complexities. Detailed preprocessing for training and testing is provided in the **Appendix**.

**Basketball Court dataset (VRU)** Wu et al. (2025b). The VRU dataset is captured using a multi-view system with 34 cameras, recording dynamic scenes at a resolution of $1920 \times 1080$ and 25 FPS. Each sequence lasts 10 seconds, resulting in 250 frames. It contains a large-scale basketball scene with fast player motions, bouncing balls, severe occlusions, and transient objects, making it highly challenging.

**SelfCap dataset** Xu et al. (2024). The SelfCap dataset provides long multi-view dynamic videos captured at 60 FPS with fixed multi-camera systems. We use the Corgi and Bike scenes, captured with 24 and 22 cameras, respectively, and evaluate 300 consecutive frames from each scene.

**Meet Room dataset** Li et al. (2022a). The Meet Room dataset is captured using a multi-view system with 13 cameras, recording indoor dynamic scenes at a resolution of $1280 \times 720$ and 30 FPS.

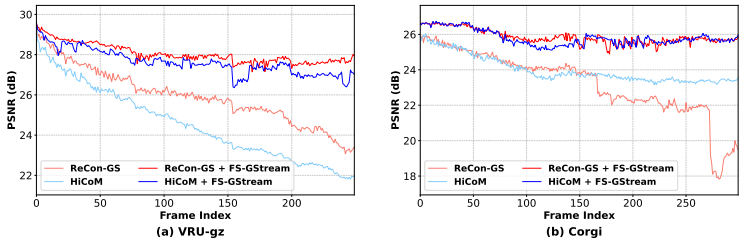


Figure 3: **PSNR trends over streaming frames.** We compare HiCoM and ReCon-GS with and without FS-GStream on the VRU-gz and Corgi scenes.

**Neural 3D Video dataset (N3DV)** Li et al. (2022b). The N3DV dataset is captured using a multi-view system with 18–21 cameras, recording dynamic scenes at a resolution of $2704 \times 2028$ and 30 FPS.

**Metrics.** To comprehensively evaluate rendering quality, temporal consistency, and efficiency, we report metrics averaged over all evaluated frames. For rendering quality, we compute the average PSNR, SSIM, and LPIPS on the test view. To assess temporal flickering, we use the warping error $E_{\text{warp}}$, a reference-free temporal consistency metric widely used in video de-flickering studies Lei et al. (2020; 2023); Qiu et al. (2024), using the same optical flow model Xu et al. (2023) as in prior streaming works Bao et al. (2025; 2026).

### 4.2 Implementation Details

We initialize the first-frame reconstruction using FastGS Ren et al. (2025). The maximum degree of spherical harmonics (SH) is set to 1, and the initial Gaussian representation is trained for 20,000 to 30,000 steps depending on the dataset. For rasterization, densification, and pruning, we follow the implementation of FastGS Ren et al. (2025). During online per-frame optimization, we optimize the motion alignment in Stage 1 for 100 steps and perform the continual refinement in Stage 2 for 100 steps. All experiments are conducted on a single NVIDIA RTX 4090 GPU. For a fair comparison, we reproduce all compared streaming baselines Yan et al. (2025); Sun et al. (2024); Gao et al. (2024a); Fu et al. (2025) using their official codebases under the same experimental environment, including the same datasets, preprocessing protocol, hardware, and evaluation metrics. Since FastGS initialization and low-level Gaussian operations could affect performance, we further provide an identical-setup ReCon-GS experiment in **Appendix A.4**.

### 4.3 Experimental Results

**Quantitative comparisons.** Table 1 shows the quantitative comparisons among 'IGS' Yan et al. (2025), '3DGStream' Sun et al. (2024)'HiCoM Gao et al. (2024a)', 'HiCoM+Ours(FS-GStream)', 'ReCon-GS Fu et al. (2025)', and 'ReCon-GS + Ours(FS-GStream)'. When FS-GStream is integrated into the two-stage streaming baseline, it consistently reduces $E_{\text{warp}}$ across all datasets and backbones, demonstrating improved temporal consistency. In terms of rendering quality, FS-GStream also provides consistent overall improvements in PSNR, SSIM, and LPIPS across most datasets and backbones, with only minor degradations for 3DGStream on Meet Room and N3DV. Fig. 3 further shows the per-frame PSNR trends on representative dynamic scenes. For both HiCoM and ReCon-GS baselines, applying our method leads to overall PSNR improvements throughout all the frames. More importantly, the baselines tend to suffer from gradual quality degradation as the frame index increases, whereas their FS-GStream-applied variants maintain higher and more stable PSNR in later frames. This indicates that DAMM and MGO not only improve per-frame reconstruction quality but also maintain the reconstruction quality throughout the subsequent frames.

**Qualitative comparisons.** Fig. 1 and Fig. 4 provide qualitative comparisons in terms of temporal consistency and rendering quality, respectively. After applying our FS-GStream, 3DGStream, HiCoM and ReCon-GS show more stable temporal patterns, with substantially reduced flickering artifacts compared to their original baselines. Fig. 4 further presents magnified dynamic regions. Compared with the baselines, the FS-GStream-enhanced methods reconstruct dynamic scene details more precisely, preserving sharper structures around moving objects. More results are provided in **Appendix E** and videos in supplementary material.

Table 2: **Ablation on DAMM and MGO.** All variants are evaluated using ReCon-GS Fu et al. (2025) under the identical FastGS-based setup. VRU Wu et al. (2025b) and SelfCap Xu et al. (2024) results are reported. The best and second-best results are highlighted in **bold** and underlined, respectively.

| Variant | DAMM | | MGO | VRU | | | | | SelfCap | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Masking | Gating | | $E_{\text{warp}}\downarrow$ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Payload ↓ | $E_{\text{warp}}\downarrow$ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Payload ↓ |
| Baseline | ✗ | ✗ | ✗ | 0.066 | 22.62 | 0.770 | 0.387 | 10.34 | 0.041 | 23.68 | 0.773 | 0.367 | 14.79 |
| Only MGO | ✗ | ✗ | ✓ | 0.036 | **27.71** | **0.902** | **0.163** | 24.59 | 0.031 | 24.78 | 0.837 | **0.237** | 42.33 |
| Masking+MGO | ✓ | ✗ | ✓ | 0.031 | 27.04 | 0.885 | 0.187 | **5.92** | 0.020 | 25.26 | 0.830 | 0.255 | **4.66** |
| Ours (full) | ✓ | ✓ | ✓ | **0.025** | 27.22 | 0.889 | 0.181 | 6.42 | **0.014** | **25.37** | **0.837** | 0.244 | 6.44 |

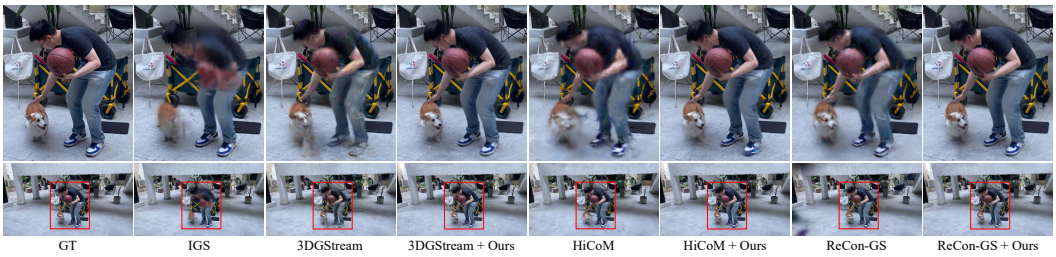


Figure 4: **Qualitative results on Corgi scene in SelfCap Xu et al. (2024) dataset.**

### 4.4 Ablation Study

We analyze the effectiveness of two core components of our FS-GStream through ablation studies: (i) Distribution-Adaptive Motion Masking (DAMM) and (ii) Mask-Guided Optimization (MGO). As shown in Table 2, the '*Only MGO*' variant improves per-frame reconstruction quality because MGO is applied with an all-one dynamic mask, allowing all Gaussians to be optimized for each frame. However, this unconstrained optimization substantially increases the payload and yields degraded temporal consistency with higher $E_{\text{warp}}$, since static Gaussians can also be modified by optimization-induced displacement noise. By adding motion masking, the '*Masking+MGO*' variant restricts geometric updates, densification, and pruning to dynamic Gaussians, which greatly reduces the payload and improves temporal consistency. Finally, '*Ours (full)*' further applies motion gating, filtering out Stage-1 motion updates assigned to static Gaussians, and achieves the lowest $E_{\text{warp}}$ (best temporal consistency) on both VRU and SelfCap datasets. These results show that DAMM and MGO are complementary: MGO improves dynamic refinement, while DAMM preserves static Gaussians from unnecessary motion and optimization, leading to better temporal stability. Additional ablation on components of MGO and the evaluation of DAMM mask are provided in **Appendix A**.

## 5 Conclusion

In this paper, we propose FS-GStream, a generalizable plug-and-play module for two-stage streaming 3D Gaussian reconstruction. We identify that temporal flickering and degradation of the persistent static representation mainly arise from (i) the difficulty of separating true scene motion from optimization-induced displacement noise and (ii) unnecessary updates, densification, and pruning of static Gaussians. To address these issues, we introduce Distribution-Adaptive Motion Masking (DAMM), which estimates a frame-wise motion threshold from the KDE-based distribution of log-transformed Gaussian translation magnitudes, and Mask-Guided Optimization (MGO), which refines dynamic geometry and appearance while preserving static geometry, selectively adapting static appearance, and restricting densification and pruning to dynamic and newly added Gaussians. Across existing two-stage streaming backbones, FS-GStream consistently improves temporal stability and provides overall improvements in rendering quality without relying on pre-trained motion estimators or fixed displacement thresholds.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We did not use generative AI tools to interpret results, implement methods, or design or provide feedback on research methodology or experiments, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript to improve its readability. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES


Zhenyu Bao, Qing Li, Guibiao Liao, Zhongyuan Zhao, and Kanglin Liu. Mgstream: Motion-aware 3d gaussian for streamable dynamic scene reconstruction. *arXiv preprint arXiv:2505.13839*, 2025.

Zhenyu Bao, Qing Li, Jinhan Xie, and Kanglin Liu. Cpostream: Collaborating prediction and observation for flicker-free streamable free-viewpoint video with 3dgs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2426–2434, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Jiacong Chen, Qingyu Mao, Youneng Bao, Xiandong Meng, Fanyang Meng, Ronggang Wang, and Yongsheng Liang. Motion matters: Compact gaussian streaming for free-viewpoint video reconstruction. *arXiv preprint arXiv:2505.16533*, 2025a.

Jianing Chen, Zehao Li, Yujun Cai, Hao Jiang, Chengxuan Qian, Juyuan Kang, Shuqin Gao, Honglong Zhao, Tianlu Mao, and Yucheng Zhang. Haif-gs: Hierarchical and induced flow-guided gaussian splatting for dynamic scene. *arXiv preprint arXiv:2506.09518*, 2025b.

Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1): 161–187, 2017.

Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Jiaye Fu, Qiankun Gao, Chengxiang Wen, Yanmin Wu, Siwei Ma, Jiaqi Zhang, and Jian Zhang. Recon-gs: Continuum-preserved gaussian streaming for fast and compact reconstruction of dynamic scenes. *arXiv preprint arXiv:2509.24325*, 2025.

Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:80609–80633, 2024a.

Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024b.

Sharath Girish, Tianye Li, Amrita Mazumdar, Abhinav Shrivastava, David Luebke, and Shalini De Mello. Queen: Quantized efficient encoding of dynamic gaussians for streaming free-viewpoint videos. *Advances in Neural Information Processing Systems*, 37:43435–43467, 2024.

Qiang Hu, Zihan Zheng, Houqiang Zhong, Sihua Fu, Li Song, Xiaoyun Zhang, Guangtao Zhai, and Yanfeng Wang. 4dgc: Rate-aware 4d gaussian compression for efficient streamable free-viewpoint video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 875–885, 2025.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4220–4230, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11338–11348, 2025a.

Sangwoon Kwak, Weeyoung Kwon, Jun Young Jeong, Geonho Kim, Won-Sik Cheong, and Jihyong Oh. Morel: Long-range flicker-free 4d motion modeling via anchor relay-based bidirectional blending with hierarchical densification. *arXiv preprint arXiv:2512.09270*, 2025b.

Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. *Advances in Neural Information Processing Systems*, 37:5384–5409, 2024.

Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. *Advances in neural information processing systems*, 33:1083–1093, 2020.

Chenyang Lei, Xuanchi Ren, Zhaoxiang Zhang, and Qifeng Chen. Blind video deflickering by neural filtering with a flawed atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10439–10448, 2023.

Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6165–6177, 2025.

Hao Li, Sicheng Li, Xiang Gao, Abudouaihati Batuer, Lu Yu, and Yiyi Liao. Gifstream: 4d gaussian-based immersive video with feature stream. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21761–21770, 2025a.

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

Jinxi Li, Ziyang Song, Siyuan Zhou, and Bo Yang. Freegave: 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.

Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35:13485–13498, 2022a.

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5521–5531, 2022b.

Zhening Liu, Yingdong Hu, Xinjie Zhang, Rui Song, Jiawei Shao, Zehong Lin, and Jun Zhang. Dynamics-aware gaussian splatting streaming towards fast on-the-fly 4d reconstruction. *arXiv preprint arXiv:2411.14847*, 2024.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Zhanpeng Luo, Haoxi Ran, and Li Lu. Instant4d: 4d gaussian splatting in minutes. *arXiv preprint arXiv:2510.01119*, 2025.

Seungjun Oh, Younggeun Lee, Hyejin Jeon, and Eunbyung Park. Hybrid 3d-4d gaussian splatting for fast dynamic scene representation. *arXiv preprint arXiv:2505.13215*, 2025.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26866–26875, 2025.

Xinmin Qiu, Congying Han, Zicheng Zhang, Bonan Li, Tiande Guo, Pingyu Wang, and Xuecheng Nie. Blazebvd: Make scale-time equalization great again for blind video deflickering. In *European Conference on Computer Vision*, pp. 37–53. Springer, 2024.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. Fastgs: Training 3d gaussian splatting in 100 seconds. *arXiv preprint arXiv:2511.04283*, 2025.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

David W Scott. Multivariate density estimation and visualization. In *Handbook of computational statistics: Concepts and methods*, pp. 549–569. Springer, 2011.

Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Treesplat: mergeable tree for deformable gaussian splatting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20675–20685, 2024.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21750–21760, 2025.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20310–20320, 2024.

Jiahao Wu, Rui Peng, Jianbo Jiao, Jiayu Yang, Luyang Tang, Kaiqiang Xiong, Jie Liang, Jinbo Yan, Runling Liu, and Ronggang Wang. Localdygs: Multi-view global dynamic scene modeling via adaptive local implicit feature decoupling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9519–9529, 2025a.

Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, and Ronggang Wang. Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. *arXiv preprint arXiv:2503.12307*, 2025b.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 13941–13958, 2023.

Zhen Xu, Yinghao Xu, Zhiyuan Yu, Sida Peng, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Representing long volumetric video with temporal gaussian hierarchy. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.

Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16520–16531, 2025.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20331–20341, 2024.

Daheng Yin, Isaac Ding, Yili Jin, Jianxin Shi, and Jiangchuan Liu. Trackersplat: Exploiting point tracking for fast and robust dynamic 3d gaussians reconstruction. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Youngsik Yun, Jeongmin Bae, Hyunseung Son, Seoha Kim, Hahyun Lee, Gun Bang, and Youngjung Uh. Compensating spatiotemporally inconsistent observations for online dynamic 3d gaussian splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–9, 2025.

Table 3: **Evaluation of motion masking.** We evaluate dynamic masks against SAM3-derived dynamic-object masks. "Fixed" applies a constant threshold to Gaussian translation magnitudes, whereas DAMM uses the same adaptive procedure for all datasets. "UniMatch threshold" obtains motion masks using a pretrained optical-flow model with a one-pixel threshold. Average results are computed across SelfCap and VRU. The best and second-best results are highlighted in **bold** and underlined, respectively.

| Method | SelfCap | | VRU | | Average | |
|---|---|---|---|---|---|---|
| | IoU ↑ | F1 ↑ | IoU ↑ | F1 ↑ | IoU ↑ | F1 ↑ |
| Fixed 0.001 | 0.3457 | 0.4551 | 0.3643 | 0.5143 | 0.3550 | 0.4847 |
| Fixed 0.002 | 0.2880 | 0.3864 | 0.5334 | 0.6912 | 0.4107 | 0.5388 |
| Fixed 0.003 | 0.1967 | 0.2756 | 0.4768 | 0.6402 | 0.3368 | 0.4579 |
| DAMM (Ours) | 0.4166 | 0.5297 | 0.5397 | 0.6943 | 0.4782 | 0.6120 |
| UniMatch Xu et al. (2023) threshold | **0.4787** | **0.5926** | **0.5940** | **0.7356** | **0.5364** | **0.6641** |

Table 4: **Sensitivity to fixed displacement thresholds.** Changes are measured relative to DAMM (Ours), averaged over two scenes for each dataset. Positive ΔPSNR indicates higher reconstruction quality, whereas positive $\Delta E_{\mathrm{warp}}$ indicates increased temporal warping error and thus worse temporal consistency.

| Dataset | Fixed threshold | ΔPSNR (dB) ↑ | $\Delta E_{\mathrm{warp}}$ (%) ↓ |
|---|---|---|---|
| SelfCap | 0.001 | +0.072 | +4.68 |
| | 0.002 | -0.284 | -17.28 |
| | 0.003 | -0.615 | -19.10 |
| VRU | 0.001 | +0.452 | +9.35 |
| | 0.002 | -0.118 | +1.31 |
| | 0.003 | -0.747 | -0.19 |

# APPENDIX

## A ADDITIONAL EXPERIMENTAL RESULTS

### A.1 EVALUATION OF MOTION MASKING

We evaluate whether the distribution-adaptive thresholding in DAMM provides a reliable alternative to manually selected displacement thresholds. We compare DAMM with fixed thresholds of $0.001$, $0.002$, and $0.003$, and with a UniMatch Xu et al. (2023)-based alternative obtained by thresholding optical flow at one pixel. Dynamic masks are evaluated against SAM3 Carion et al. (2026)-derived pseudo ground truth, which is used only for evaluation. As shown in Table 3, the best fixed threshold differs across datasets: $0.001$ performs best on SelfCap, whereas $0.002$ performs best on VRU. In contrast, DAMM uses the same adaptive procedure for both datasets and outperforms all evaluated fixed thresholds in IoU and F1, both per dataset and on average. Although UniMatch achieves higher mask accuracy, it requires an additional pretrained optical-flow network and an explicit flow threshold, whereas DAMM uses only the per-Gaussian motion distribution already obtained in Stage 1. We also examine the downstream effect of fixed thresholds. Table 4 shows that the preferred operating point is dataset dependent. For example, a threshold of $0.001$ improves PSNR by $0.072$ dB on SelfCap and $0.452$ dB on VRU, but increases $E_{\mathrm{warp}}$ by $4.68\%$ and $9.35\%$, respectively. Larger thresholds can improve temporal consistency, but at the cost of rendering quality. Overall, DAMM provides a balanced operating point without dataset-specific threshold tuning by adapting the static–dynamic separation to the frame-wise motion distribution.

### A.2 ABLATION OF MASK-GUIDED OPTIMIZATION (MGO)

We ablate the individual design choices within MGO; mask-guided densification, selective static appearance updates, and mask-guided frame-transition pruning. All variants retain DAMM and modify only the corresponding MGO component. As shown in Table 5, removing mask-guided densification substantially increases the streaming payload while degrading both rendering quality and temporal consistency. This confirms that restricting densification to dynamic regions is important for preventing unnecessary growth of the persistent representation. Selective static appearance

Table 5: **Ablation of individual MGO components.** All variants retain DAMM and differ from the full model only in the specified MGO component. "w/o Mask-Guided Densification" removes the dynamic-mask restriction on densification, "w/o Selective Static Appearance Updates" disables appearance updates for static Gaussians, and "Count-Based Frame-Transition Pruning" replaces our mask-guided frame-transition pruning with the original count-based frame-transition pruning. Results are averaged over two scenes for each dataset. The best and second-best results are highlighted in **bold** and underlined, respectively.

| Dataset | Variant | PSNR ↑ | SSIM ↑ | LPIPS ↓ | $E_{\mathrm{warp}}$ ↓ | Payload (MB) ↓ |
|---|---|---|---|---|---|---|
| SelfCap | Ours (full) | <u>25.37</u> | **0.837** | **0.244** | <u>0.014</u> | 6.44 |
| | w/o Mask-Guided Densification | 25.24 | 0.835 | 0.249 | 0.020 | 17.89 |
| | w/o Selective Static Appearance Updates | **25.38** | <u>0.836</u> | <u>0.245</u> | **0.012** | <u>5.14</u> |
| | Count-Based Frame-Transition Pruning | 25.28 | 0.830 | 0.260 | 0.017 | **4.27** |
| VRU | Ours (full) | **27.22** | **0.889** | **0.181** | <u>0.025</u> | 6.42 |
| | w/o Mask-Guided Densification | 26.29 | 0.876 | 0.197 | 0.029 | 16.74 |
| | w/o Selective Static Appearance Updates | <u>26.84</u> | <u>0.886</u> | <u>0.184</u> | **0.024** | <u>6.25</u> |
| | Count-Based Frame-Transition Pruning | 26.22 | 0.857 | 0.255 | 0.035 | **4.31** |

Table 6: **Ablation of the prominence criterion in DAMM.** "Ours" uses $\rho_{\mathrm{prom}} = 0.05$, while "w/o prominence" sets $\rho_{\mathrm{prom}} = 0$, allowing all local density peaks to participate in threshold selection. Results are averaged over two scenes for each dataset.

| **Backbone** | **Dataset** | **Setting** | **PSNR ↑** | **SSIM ↑** | **LPIPS ↓** | $E_{\mathrm{warp}}$ ↓ |
|---|---|---|---|---|---|---|
| HiCoM | SelfCap | Ours | 25.34 | 0.839 | 0.243 | 0.020 |
| | | w/o prominence | 25.14 | 0.840 | 0.238 | 0.029 |
| | VRU | Ours | 26.84 | 0.884 | 0.189 | 0.028 |
| | | w/o prominence | 27.45 | 0.899 | 0.163 | 0.034 |
| 3DGStream | SelfCap | Ours | 25.36 | 0.833 | 0.252 | 0.014 |
| | | w/o prominence | 25.38 | 0.843 | 0.233 | 0.028 |
| | VRU | Ours | 27.21 | 0.886 | 0.184 | 0.026 |
| | | w/o prominence | 27.54 | 0.897 | 0.169 | 0.032 |

updates exhibit a different trade-off. Disabling them slightly reduces $E_{\mathrm{warp}}$, but generally degrades rendering quality, particularly on VRU. This indicates that selectively updating the SH coefficients and opacity of static Gaussians helps accommodate residual appearance changes while keeping their geometry fixed. Finally, replacing our mask-guided frame-transition pruning with the original count-based frame-transition pruning reduces the payload, but substantially degrades rendering quality and temporal consistency. These results support our design of preserving static Gaussians during frame transitions rather than globally removing Gaussians according to the count-based rule. Overall, the three MGO operations play complementary roles in balancing rendering quality, temporal stability, and streaming efficiency.

### A.3 Effect of the Prominence Criterion

We analyze the role of the prominence criterion used in DAMM. As described in Sec. 3.3, DAMM identifies the dominant low-motion mode from the KDE of log-transformed Gaussian translation magnitudes and selects the first subsequent valley as the motion threshold. To prevent small density fluctuations from being incorrectly selected as motion modes, we retain only peaks whose prominence exceeds $\rho_{\mathrm{prom}}$ of the maximum density. We use $\rho_{\mathrm{prom}} = 0.05$ in all main experiments. Table 6 compares the default setting with $\rho_{\mathrm{prom}} = 0$, which removes prominence-based peak filtering and therefore allows all local maxima to participate in the threshold selection. We evaluate this ablation on both HiCoM and 3DGStream over the SelfCap and VRU datasets.

Table 7: **Identical-Setup ReCon-GS experiment.** "ReCon-GS (FastGS applied)" uses the same initial Gaussian set and identical rasterization, densification, and pruning procedures as our implementation. "ReCon-GS + FS-GStream" further applies DAMM and MGO on top of this identical setup. The best and second-best results are highlighted in **bold** and underlined, respectively.

| Dataset | Variant | $E_{\text{warp}} \downarrow$ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Payload (MB) ↓ | Train (sec) ↓ |
|---|---|---|---|---|---|---|---|
| VRU | ReCon-GS | 0.044 | 25.32 | 0.843 | 0.259 | 12.76 | 3.5 |
| | ReCon-GS (FastGS applied) | 0.066 | 22.62 | 0.770 | 0.387 | 10.34 | **2.0** |
| | ReCon-GS + FS-GStream | **0.025** | **27.22** | **0.889** | **0.181** | **6.42** | 2.3 |
| SelfCap | ReCon-GS | 0.032 | 23.73 | 0.794 | 0.311 | 11.39 | 6.0 |
| | ReCon-GS (FastGS applied) | 0.041 | 23.68 | 0.773 | 0.367 | 14.79 | **3.4** |
| | ReCon-GS + FS-GStream | **0.014** | **25.37** | **0.837** | **0.244** | **6.44** | 4.1 |
| Meet Room | ReCon-GS | 0.009 | 30.51 | 0.949 | 0.184 | **1.56** | 2.7 |
| | ReCon-GS (FastGS applied) | 0.038 | 23.40 | 0.872 | 0.304 | 2.28 | **1.8** |
| | ReCon-GS + FS-GStream | **0.006** | **32.51** | **0.963** | **0.161** | 1.51 | 2.1 |
| N3DV | ReCon-GS | 0.010 | 31.10 | 0.945 | 0.138 | 4.03 | 5.7 |
| | ReCon-GS (FastGS applied) | 0.019 | 30.20 | 0.924 | 0.179 | 4.99 | **3.1** |
| | ReCon-GS + FS-GStream | **0.008** | **31.83** | **0.947** | **0.133** | **2.64** | 3.2 |

Removing the prominence criterion consistently weakens temporal stability. Although removing prominence filtering can slightly improve spatial reconstruction metrics, such as PSNR, SSIM, and LPIPS, these gains come at the cost of substantially degraded temporal consistency. This behavior can be explained by the role of prominence filtering in DAMM. With $\rho_{\text{prom}} = 0$, small fluctuations in the KDE are also accepted as candidate peaks. A spurious low-motion peak can therefore be selected as the static mode, yielding an overly permissive dynamic mask and causing more Gaussians to undergo motion updates, optimization, and densification. Consequently, the behavior moves toward the unconstrained regime observed when motion masking is removed in the MGO ablation. The prominence criterion therefore acts as an important regularizer for reliable distribution-adaptive motion separation, preventing DAMM from degenerating toward an excessively permissive masking regime.

### A.4 ReCon-GS Comparison under Identical Setup

To isolate the contributions of our proposed modules, Table 7 compares different variants of ReCon-GS using a consistent pipeline for initial frame setup, rasterization, densification, and pruning, adopted from FastGS Ren et al. (2025). Specifically, the "ReCon-GS (FastGS applied)" baseline employs the exact same initial Gaussian set, rasterization, densification, and pruning as our full framework, but omits the DAMM and MGO components. Building upon this controlled setup, 'ReCon-GS + FS-GStream' integrates both DAMM and MGO.

Note that we also include the original ReCon-GS results reported in Table 1. As observed in Table 7, simply migrating ReCon-GS to the FastGS setup often degrades its baseline rendering quality. By reporting the original ReCon-GS in our main evaluations, we ensure a fair comparison against its optimal configuration, rather than an artificially weakened baseline.

The results demonstrate that adopting the FastGS setup alone is insufficient to account for the overall performance enhancements. Rather, the integration of FS-GStream yields consistent and significant improvements in temporal stability ($E_{\text{warp}}$), rendering quality, and payload efficiency. This confirms that our DAMM and MGO are the primary source of the performance gains.

## B Implementation Details

### B.1 Topographic Prominence

We compute prominence following the procedure implemented in SciPy Virtanen et al. (2020).

For a local maximum $x$ of a one-dimensional function $g$, we extend the search in each direction until encountering a value greater than $g(x)$ or reaching the evaluation boundary. Peaks of equal height do not terminate the search. Let $b_l$ and $b_r$ denote the minimum function values within the resulting left and right intervals, respectively. The prominence is

$$\mathrm{Prom}_g(x) = g(x) - \max\{b_l, b_r\}. \tag{8}$$

Thus, the prominence measures how distinctly a peak rises aboveits surrounding density landscape.

### B.2 Implementation Details of Mask-Guided Optimization

**Appearance Selection from Multi-View Residuals.** To selectively adapt the appearance of static Gaussians, we identify those that consistently contribute to regions with large photometric residuals. We follow the Gaussian importance formulation of FastGS Ren et al. (2025), but use the resulting score only to determine appearance updates for static Gaussians.

After $k_0 = 20$ Stage 2 optimization steps, we render the current representation from a fixed set $\mathcal{C}$ of 10 training views. For each view $c$, we compute the per-pixel mean absolute RGB residual

$$e_c(p) = \frac{1}{3}\left\|\widehat{I}_c(p) - I_c(p)\right\|_1, \tag{9}$$

and normalize it independently within each view as

$$\widetilde{e}_c(p) = \frac{e_c(p) - \min_{p'} e_c(p')}{\max_{p'} e_c(p') - \min_{p'} e_c(p') + \epsilon_r}, \tag{10}$$

where $\epsilon_r > 0$ ensures numerical stability.

Let $a_{i,c,p} \in \{0,1\}$ indicate whether Gaussian $i$ contributes to pixel $p$ in view $c$. We assign each Gaussian a residual contribution score

$$r_i = \left\lfloor \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \sum_{p} a_{i,c,p}\, \mathbf{1}[\widetilde{e}_c(p) > \tau_{\mathrm{pix}}] \right\rfloor, \qquad \tau_{\mathrm{pix}} = 0.3. \tag{11}$$

Thus, $r_i$ measures how frequently Gaussian $i$ contributes to high-residual image regions across the sampled views.

We then enable appearance updates only for static Gaussians whose residual contribution score exceeds the threshold:

$$w_i^{(k)} = M_{\mathrm{static}}^i\, \mathbf{1}[r_i > \tau_{\mathrm{count}}], \qquad \tau_{\mathrm{count}} = 5. \tag{12}$$

Selected static Gaussians update only their SH coefficients and opacity, while their position, rotation, and scale remain fixed.

**Selection Schedule.** Before the residual evaluation, $w_i^{(k)}$ is set to 0. THerefore, Stage 2 first refines only dynamic and newly added Gaussians. The residual-based appearance mask is then computed once per frame and kept fixed for the remaining Stage 2 iterations. This delayed selection allows dynamic refinement to explain the dominant residuals before static appearance adaptation is activated. The appearance mask is reset at the next frame, while the updated SH coefficients and opacity are propagated forward.

## C Visualization of Motion-Score Distributions

Fig. 5 compares the distribution of Gaussian translation magnitudes in the original and log-transformed domains. In the original domain, most Gaussians are concentrated near zero, while Gaussians with larger motions are distributed over a much wider range of displacement magnitudes. The log transformation expands the near-zero displacement range and compresses larger magnitudes, revealing a clearer density separation between the dominant low-motion population and the higher-motion population. Accordingly, DAMM estimates the KDE and selects the frame-wise motion threshold from the valley in the log-transformed domain.

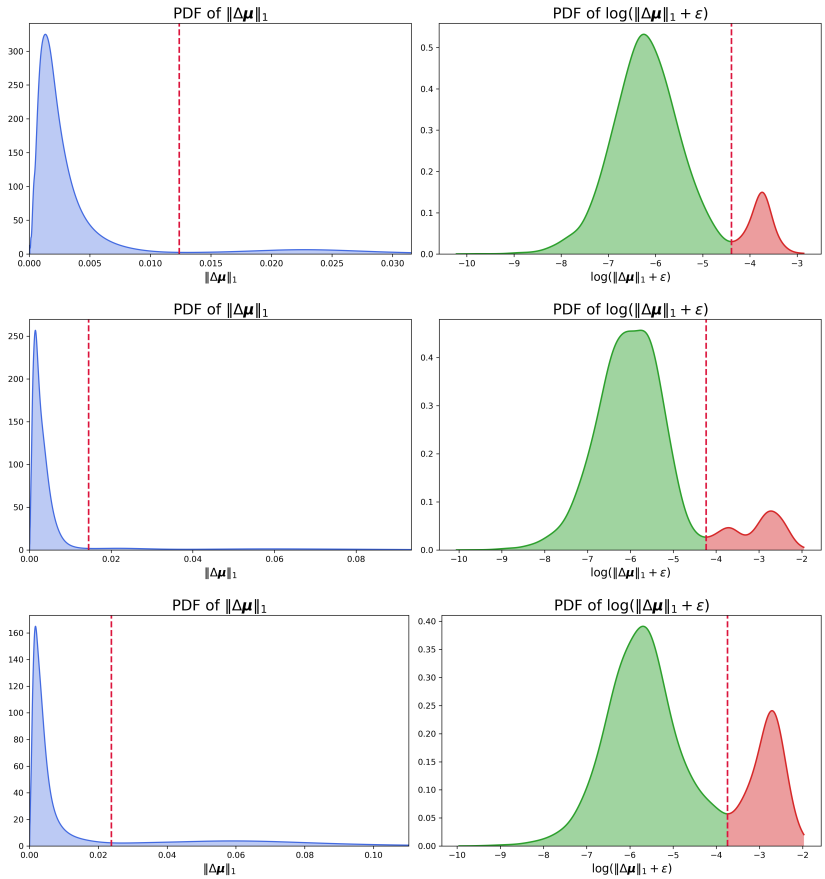


Figure 5: **Visualization of Gaussian motion distributions.** The left column shows KDEs of the raw translation magnitudes $\|\Delta\boldsymbol{\mu}\|_1$, while the right column shows the corresponding densities in the log-transformed domain $\log(\|\Delta\boldsymbol{\mu}\|_1 + \epsilon)$.

# D Datasets and Metrics

## D.1 Datasets

We evaluate FS-GStream on four multi-view dynamic scene datasets with different motion scales, sequence lengths, and scene complexities. Since temporal inconsistency and flickering artifacts become more noticeable in scenes with large and complex motions, we describe the datasets in the order of motion complexity and evaluation relevance.

**Common preprocessing and evaluation protocol.** For all datasets, we follow the standard multi-view streaming evaluation protocol used in prior works Yan et al. (2025); Sun et al. (2024). The video captured by the center reference camera is held out for testing, while the remaining camera views are used for training. For each scene, we run COLMAP Schonberger & Frahm (2016) on the first frame to estimate the camera poses and generate the initial point cloud. The estimated camera poses are then fixed throughout the streaming reconstruction process. Following prior streaming 3DGS settings Sun et al. (2024); Yan et al. (2025), we downsample the videos of N3DV, VRU, and the Corgi scene in SelfCap by a factor of two. For the Bike scene in SelfCap and the Meet Room dataset, we use the original resolution without downsampling. For temporal evaluation, we use 250 frames for VRU, which contains 10-second videos captured at 25 FPS. For Meet Room and N3DV, we use 300 frames from each evaluated sequence. For the long-term SelfCap dataset, we evaluate each selected scene using 300 consecutive frames.

**The Basketball Court dataset (VRU)** Wu et al. (2025b). The Basketball court dataset is captured using a multi-view system with 34 cameras, recording dynamic scenes at a resolution of $1920 \times 1080$ and 25 FPS. It contains a large-scale basketball scene with challenging dynamic events, including

fast player motions, bouncing balls, severe occlusions, and transient objects. Each sequence spans 10 seconds, resulting in 250 frames, and we use all 250 frames for evaluation. These characteristics make the dataset particularly suitable for evaluating temporal consistency in streaming reconstruction, as flickering artifacts and background erosion become more apparent under rapid and complex motions.

**The SelfCap dataset** Xu et al. (2024). The SelfCap dataset provides long multi-view dynamic videos captured at 60 FPS with fixed multi-camera systems. It includes several challenging sequences, such as Corgi, Bike, Hair, Dance, and Yoga, with high-resolution videos and calibrated camera parameters. In our experiments, we use the Corgi and Bike scenes, captured with 24 and 22 cameras, respectively. Corgi contains 3,500 frames at 2160p resolution, while Bike contains 37,377 frames at $1024 \times 1024$ resolution. Since SelfCap provides long-term videos, we evaluate each selected scene using 300 consecutive frames to maintain a consistent evaluation length with Meet Room and N3DV. Although the cameras are fixed, we observed visible camera jitter in some scenes, which may introduce temporal artifacts unrelated to streaming reconstruction. Therefore, we exclude jitter-affected scenes and use only Corgi and Bike, where camera jitter is not apparent, to evaluate temporal consistency under reliable camera conditions.

**The Meet Room dataset** Li et al. (2022a). The Meet Room dataset is captured using a multi-view system with 13 cameras, recording indoor dynamic scenes at a resolution of $1280 \times 720$ and 30 FPS. For evaluation, we use 300 frames from each sequence. Compared to the Basketball court and SelfCap datasets, Meet Room contains more moderate scene motion, but still includes human movement, view-dependent appearance changes, and occlusion, making it useful for evaluating the robustness of online reconstruction in indoor multi-view settings.

**The Neural 3D Video dataset (N3DV)** Li et al. (2022b). The Neural 3D Video dataset is captured using a multi-view system with 18–21 cameras, recording dynamic scenes at a resolution of $2704 \times 2028$ and 30 FPS. It includes diverse dynamic scenarios, such as fire, reflections, and newly appearing objects. For evaluation, we use 300 frames from each selected sequence. Although N3DV generally contains less abrupt object motion than the Basketball court dataset, its complex appearance variations and dynamic visual effects provide a complementary benchmark for evaluating rendering quality and temporal stability.

### D.2 METRICS

**Streaming Payload.** We report the average streaming payload size, which measures the serialized binary data transmitted per frame to synchronize the receiver-side Gaussian state. The payload consists of updated Gaussian attributes, newly added Gaussians, deformation parameters, and pruning indices. This metric reflects the practical communication cost of real-time streaming reconstruction.

**The pairwise warping error.** The pairwise warping error is defined as:

$$E_{\text{pair}}(I_i, I_j) = \|\mathcal{M}_{i,j} \odot (I_i - \mathcal{W}_{j \to i}(I_j))\|_1 , \tag{13}$$

and the overall temporal warping error is computed as:

$$E_{\text{warp}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \left[ E_{\text{pair}}(I_t, I_{t-1}) + E_{\text{pair}}(I_t, I_0) \right] , \tag{14}$$

where $I_i$ and $I_j$ denote rendered frames, $\mathcal{W}_{j \to i}$ warps $I_j$ to $I_i$ using optical flow, $\mathcal{M}_{i,j}$ is the valid flow mask, and $T$ is the total number of evaluated frames. A lower $E_{\text{warp}}$ indicates fewer flickering artifacts and better temporal consistency. We also report the average per-frame training time and streaming payload over the evaluated frames to measure computational efficiency.

## E MORE QUALITATIVE RESULTS

In this section, we provide additional qualitative results to further demonstrate the visual improvements achieved by our method. As discussed in Sec. 4.3, existing streaming baselines often struggle with rendering moving objects, resulting in blurry artifacts and loss of detail. Fig. 7 and Fig. 8 show that integrating FS-GStream into the baselines enables more precise reconstruction of highly dynamic regions. Across diverse datasets, including VRU Wu et al. (2025b) and SelfCap Xu et al. (2024), our FS-GStream consistently preserves sharper structures and finer details around moving objects compared to the original baselines. Please see the videos in supplementary material for more results.

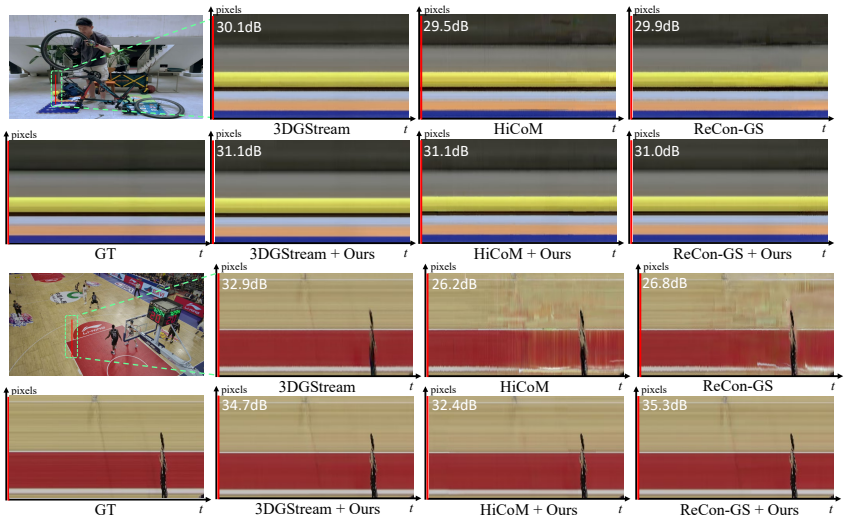


Figure 6: **Qualitative comparison of spatiotemporal images.** We compare 3DGStream, HiCoM and ReCon-GS with and without FS-GStream on the VRU and SelfCap datasets.

## F LIMITATIONS

FS-GStream is designed for two-stage streaming 3DGS pipelines that provide per-Gaussian motion estimates followed by continual Gaussian refinement. Therefore, directly applying it to streaming frameworks without explicit per-Gaussian motion updates may require additional adaptation. Moreover, since DAMM relies on the learned per-Gaussian translation magnitudes for motion separation, FS-GStream inherits the modeling limitations of the underlying motion representation. Inaccurate or insufficiently expressive motion estimates may therefore degrade the reliability of the dynamic-static separation.

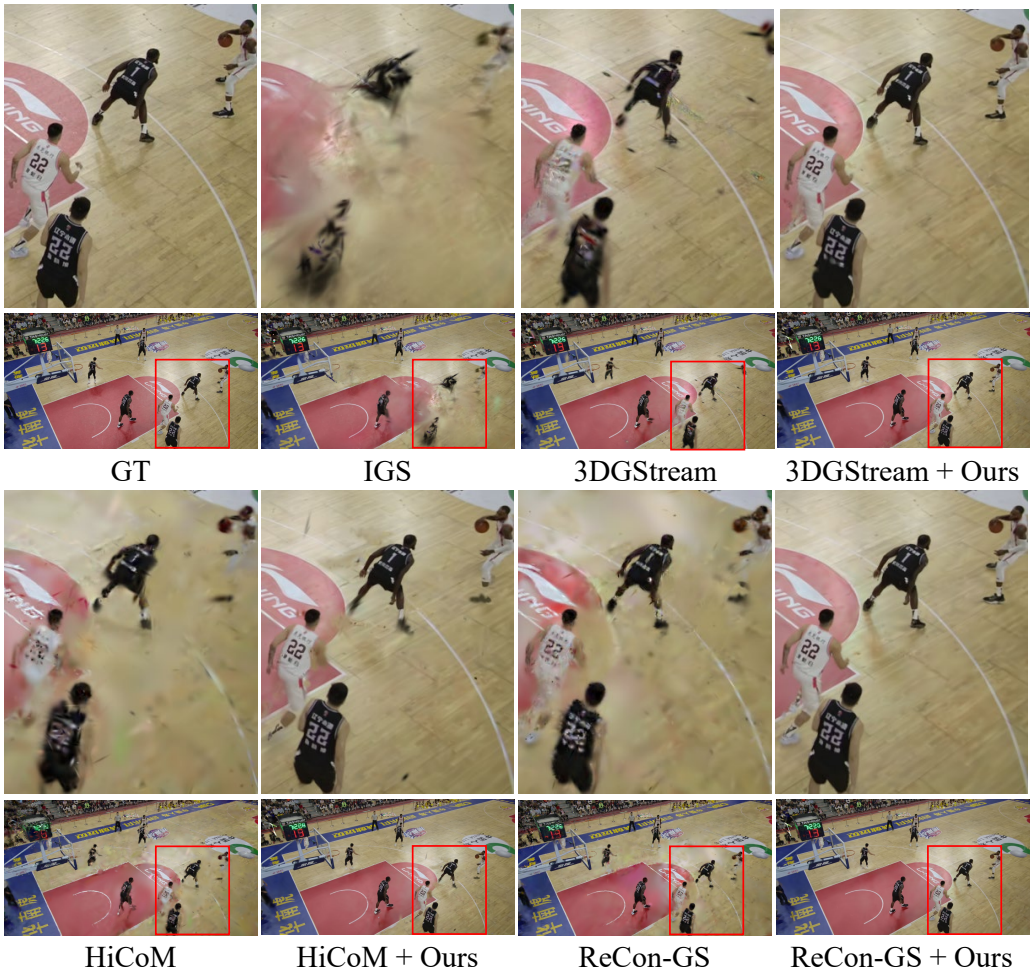


Figure 7: **Qualitative results on VRU Wu et al. (2025b).**

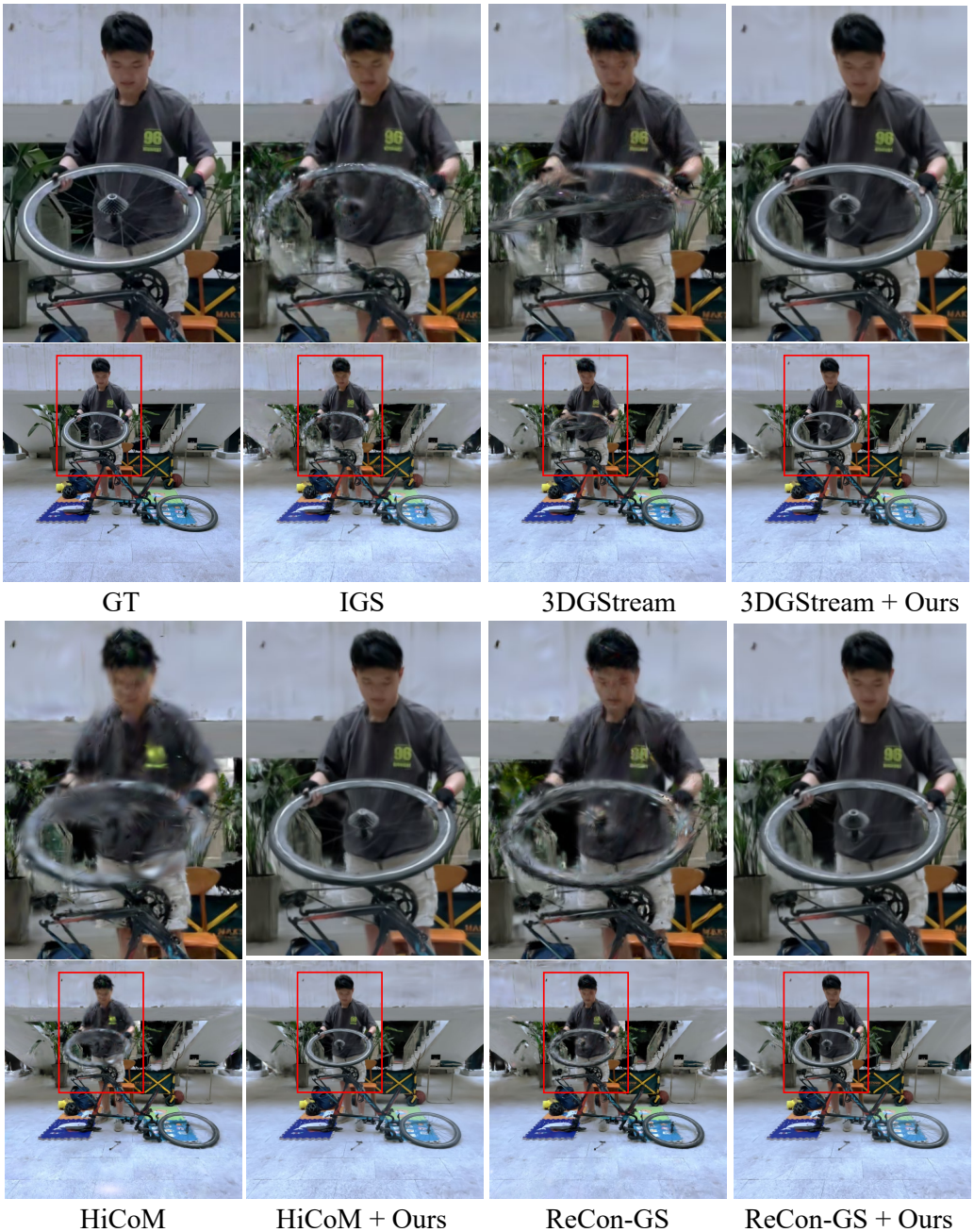


Figure 8: **Qualitative results on SelfCap Xu et al. (2024).**