CPOStream: Collaborating Prediction and Observation for Flicker-Free Streamable Free-Viewpoint Video with 3DGS

Zhenyu Bao[1,2], Qing Li[2], Jinhan Xie[1,2], Kanglin Liu[2*]

[1]School of Electronic and Computer Engineering, Peking University
[2]Pengcheng Laboratory
{zybao, xiejinhan0428}@pku.edu.cn, {lqing900205, max.liu.426}@gmail.com

Abstract

3D Gaussian Splatting (3DGS) has recently demonstrated significant potential for streaming dynamic scenes, enabling the synthesis of photo-realistic and real-time free-viewpoint videos (FVVs). Conventional streaming pipelines optimize each frame independently, *i.e.*, the attribute of the 3D Gaussians (3DGs) responsible for the static regions are supposed to be identical across all frames but are changed in the optimization process, thus causing temporal color inconsistency and visual flickering artifacts in the static regions. To tackle this, we propose CPOStream, which utilizes a prediction and observation module to determine the state of 3DG. Specifically, the prediction module records those 3DGs that are inactive in the past K frames and those would be ignored in the optimization process of the current frame reconstruction. Thus, the attributes of those 3DGs would be kept consistent across the past K frames, guaranteeing the temporal consistence. Additionally, the observation module conducts motion detection, and recognizes those new 3DGs which are not recorded in the prediction module and are first detected by the observation module in the past K frames. The attributes of those 3DGs are optimized during the current frame reconstruction. Experiments on multiple real-world FVV benchmarks show that CPOStream substantially reduces temporal flickering and improves reconstruction fidelity, achieving state-of-the-art performance.

Introduction

Free-viewpoint Video (FVV) synthesis from multi-view video sequences has emerged as a critical research focus due to its promising applications in live broadcasting and streamings sports broadcasting. Early approaches rely on dynamic geometric primitives (Collet et al. 2015; Innmann et al. 2016) or view-interpolation techniques (Zitnick et al. 2004; Chen and Williams 2023) to achieve it, which are struggled to faithfully reconstruct complex real-world illumination and textures. Motivated by the photorealistic view synthesis capacity of Nerual Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023), researchers have begun to utilize these techniques to achieve FVV.

One line of research follows an offline paradigm, which requires full accessibility of videos (Fridovich-Keil et al. 2023; Cao and Johnson 2023; Yang et al. 2023; Duan et al. 2024; Li et al. 2024; Wu et al. 2024). These methods build spatio-temporal models to support video playback from arbitrary viewpoints and times. Due to full-sequence access and shared spatio-temporal modeling, offline methods are of high temporal consistency by performing global optimization across all frames. However, such approaches have several limitations. Firstly, they usually require a fixed length video as input, which makes them incapable of handling arbitrary-length video sequences. Secondly, they are inherently incompatible with real-time streaming applications. These limitations make offline methods impractical for many real-world live streaming scenarios.

Another line of work adopts an online paradigm, reconstructing scenes moment-by-moment by optimizing per frame in a sequential manner (Li et al. 2022a; Wang et al. 2023; Sun et al. 2024; Girish et al. 2024; Gao et al. 2024; Yun et al. 2025; Hu et al. 2025). This per-frame training strategy supports real-time streaming and arbitrary-length sequences. Despite these advantages, existing online methods often overlook flickering artifacts, which severely degrades the perceptual quality of generated videos. This flickering arises because Gaussians primitives corresponding to static regions are changed during current frame reconstruction, introducing frame-by-frame inconsistencies. Such discrepancies manifest as noticeable flickering artifacts within static regions of synthesized videos, as shown in Fig. 2 (c).

To address this issue, we introduce CPOStream, a novel framework that combines a prediction module and an observation module to selectively freeze the 3D Gaussians (3DGs) corresponding to static contents during training, enabling flicker-free FVV streaming in dynamic real-world scenes. The framework is designed based on the fact that regions or objects that have remained static over the past K frames (e.g., a wall) are highly likely to be static in the subsequent frame. Consequently, the attributes of the associated 3DGs should remain unchanged. This assumption may only fail when previously static objects exhibit sudden movements, such as a bottle being picked up. Inspired by this assumption, CPOStream integrates a prediction module and an observation module to jointly separate all 3DGs into *to-be-optimized* and *to-be-frozen* categories at the beginning

*Corresponding author.

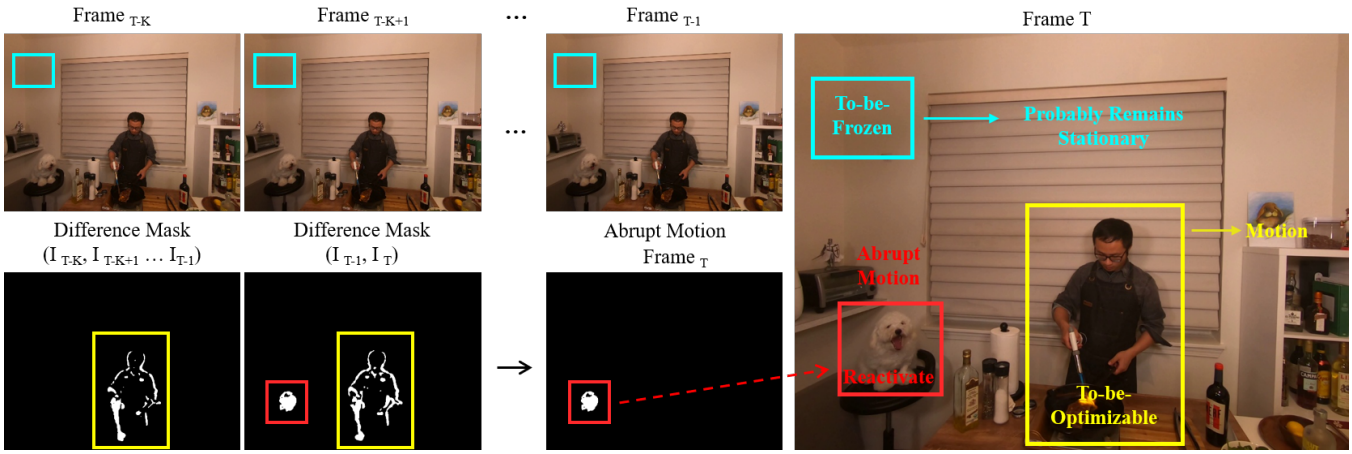


Figure 1: CPOStream is motivated by a key insight: regions that remain static over the past K frames are highly likely to remain static in the subsequent frame (*e.g.* the wall framed by a cyan box). The only exception arises when a previously stationary object suddenly starts moving in the current frame, for example, the red-boxed dog.

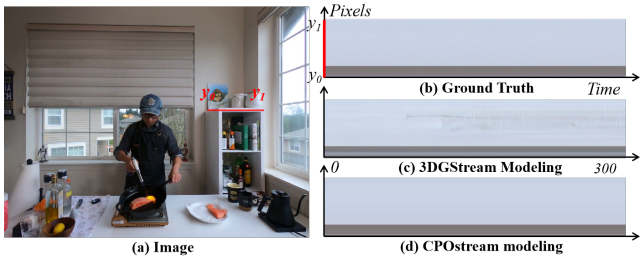


Figure 2: Flickering issue. We present a time-color plot of red line over 300 frames, with the horizontal axis representing time and the vertical axis representing pixel indexes.

of each frame's training stage. In the prediction module, a 3DG is marked as frozen for the current frame if it has remained inactive over the preceding K frames. In the observation module, the current frame's abrupt motion mask is analyzed to detect unexpected movements, reactivating previously frozen 3DGs associated with suddenly moving regions. In summary, our contributions are as follows:

- We present a novel 3DGS-based framework for streaming FVVs that achieves superior temporal consistency and state-of-the-art reconstruction quality.
- We design a prediction module and an observation module to effectively identify the state of 3DGs.
- We introduce a Gaussian Activity Classifier that accurately separates 3DGs into active or inactive based on their 2D variation of angles during optimization.
- We propose a new approach for identifying sudden motion regions in the current frame by leveraging both historical 2D frame-difference masks and the current frame's difference mask to detect abrupt motions.
- Extensive experiments on multiple datasets demonstrate that our method outperforms prior approaches in terms of both reconstruction fidelity and temporal consistency.

Related Work

Building upon high-fidelity reconstruction of static scenes, recent works have extended differentiable volumetric rendering to construct photo-realistic FVV of dynamic scenes, yielding two predominant lines of research distinguished by the availability of the full video frame sequence.

Offline methods. These methods assume the entire video is available before training, and reconstruct 4D spatio-temporal dynamic models and allow time-consuming optimization that delivers smooth, high-quality playback. K-Planes (Fridovich-Keil et al. 2023) and HexPlane (Cao and Johnson 2023) decouple the spatio-temporal field into several 2D planes, markedly improving storage efficiency while preserving fidelity. 4DGS (Yang et al. 2023) and 4D RotorGS (Duan et al. 2024) raise 3D Gaussian to 4D space by adding a temporal dimension, enhancing interpretability and dynamic modeling capacity. 4D deformable Gaussians (Yang et al. 2024) and 4DGaussians (Wu et al. 2024) employ MLPs to regress per-frame Gaussian residuals, whereas SpacetimeGS (Li et al. 2024) parameterizes translation, rotation and occupancy jointly over time. Although these systems achieve compelling results, their reliance on full-length videos conflicts with real-world streaming and live-capture use cases.

Online methods. Online approaches take sequential frames as input and update the model on the fly, sequentially reconstructing the scene at each moment, which aligns with modern applications such as live short-video broadcasting. StreamRF (Li et al. 2022a) is the first to introduce a differentiable volumetric rendering-based online reconstruction pipeline, building upon Plenoxels-NeRF (Fridovich-Keil et al. 2022) to reduce temporal storage requirements. 3DGStream (Sun et al. 2024) combines 3DGS with a hash grid to achieve on-the-fly construction of photo-realistic, real-time renderable FVV. Methods like (Girish et al. 2024; Hu et al. 2025) focus on compression by quantifying Gaussian parameters, dramatically shrinking the bandwidth. Yan

et al. (Yan et al. 2025) leverage a pre-trained multi-view stereo (MVS) model and key-points interpolation to reduce per-frame optimization time. Yet, these methods overlook temporal flicker induced by independent per-frame optimization. To enhance the temporal consistency of streaming modeling, MGStream (Bao et al. 2025) introduces optical flow masks and convex-hull morphology to selectively update moving 3DGs, thereby mitigating flicker and enhancing temporal consistency. Nonetheless, the robustness of optical flow estimation remains a challenge in real-world scenarios, as inevitable noise can adversely affect both temporal coherence and rendering quality. Yun *et al.* (Yun et al. 2025) uses the single-frame residual masks to compensate for sensor noise without the need for pre-trained models, alleviating flicker during streaming reconstruction. Yet, this approach does not explicitly address temporal artifacts in static regions caused by inconsistent optimization across frames, and the additional residual optimization complicates the convergence of per-frame training frameworks.

Distinct from prior methods, our framework fuses the past inactive information cached during reconstruction and sudden-motion mask from the current frames to identify and freeze the 3DGs of the static content, facilitates a temporal-consistent training process, requires no pre-trained models, and does not increase training complexity.

Preliminaries

3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents 3D scenes using explicit Gaussian primitives. Given multi-view images I^M, the method first recovers an initial point cloud $\mathcal{P}$ and camera poses $\mathcal{T}$ via Structure-from-Motion (SfM). Each point p_i is then parameterized as a 3D Gaussian primitive G_i, equipped with position (u_i), opacity (o_i), scale (s_i), rotation (r_i), and spherical harmonic color coefficients (shs_i). The color of a pixel (C_u) could be rendered by alpha-blending algorithm:

$$C_u = \sum_{i=1} \mathbf{c}_i(shs_i)\alpha_i \prod_{j=1}^{j-1}(1-\alpha_j), \tag{1}$$

where α_i is the rasterized opacity of $i-$th 3DG, computed by $\alpha_i = o_i\mathcal{G}_i(s_i, r_i, u_i)$, and $\mathcal{G}_i$ denotes projected 2D Gaussian value. The parameters $\Theta = \{u_i, \alpha_i, s_i, r_i, shs_i\}$ are optimized by minimizing the L_1 loss and $D-SSIM$ loss between rendered and ground-truth images.

$$\mathcal{L}_{color} = (1-\lambda)\mathcal{L}_1(\hat{I}, I) + \lambda\mathcal{L}_{D-SSIM}(\hat{I}, I). \tag{2}$$

The λ is set to 0.2 in vanilla 3DGS.

Method

CPOStream constructs photo-realistic, flicker-free FVVs through a per-frame training paradigm. Starting from the T_0 frame, it initializes and trains the 3DGs derived from a sparse COLMAP reconstruction. Each subsequent frame is then optimized using the 3DGs from the previous frame as initialization. To suppress flickering in static regions, CPOStream analyzes the past activity status of 3DGs and selectively freezes those associated with static content before optimizing each frame T ($T \geq 1$). Fig. 3 illustrates our architecture, and we detail it in the following three subsections. Section **Prediction Module** determines whether a 3DG should be optimized or frozen based on its activity over past frames. Section **Observation Module** reactivates 3DGs associated with objects that exhibit unexpected motion, using a customized 2D abrupt mask. Finally, Section **Online Training** presents the online optimization procedure.

Prediction Module

The prediction module estimates each 3DG's state from its activity status over the past K frames. It is built on the assumption that regions remaining static over the past K frames are likely to remain static in the subsequent frame as well (see Fig. 1). It comprises two components: a *Gaussian Activity Classifier*, which labels each 3DG as active or inactive per frame based on its post-optimization 2D angular displacement, and a *Freeze Candidate Predictor*, which classifies each 3DG as either to-be-frozen or to-be-optimized for the current frame utilizing its activity labels sequence.

Gaussian Activity Classifier. This classifier labels the activity status of each 3DG in previous frames as either active or inactive, storing the result in a status list for the subsequent *Freeze Candidate Predictor*. For a 3DG G_i^{T-1} at frame $T-1$, its activity status is determined by the average angular displacement across multiple camera views, yielding an activity score $P_{T-1} \in (0, 1)$:

$$P_{T-1} = \mathbb{I}[(\frac{1}{R}\sum_{i=1}^{R}\frac{\theta_i^j(u_{t-2}, u_{t-1})}{p_i^j}) > \epsilon], \tag{3}$$

where R is the number of camera views in single frame, $\theta_i^j(u_{t-2}, u_{t-1})$ is the angular displacement of the j-th 3DG in i-th 2D camera planes, p_i^j denotes the number of pixels covered by the projected 3DG^j in view i, and ϵ is a predefined threshold. This score represents a binary activity status, where $P = 0$ indicates *inactive* and $P = 1$ indicates *active*. We choose the average angular displacement instead of the Gaussian's 3D spatial movement to distinguish between actual motion and minor jitter introduced during optimization, which ensures robustness against small view-dependent noise.

Freeze Candidate Predictor. Before optimizing the current frame T, the predictor classifies each 3DG as either *to-be-frozen* or *to-be-optimized*, based on its cached activity status over the past K frames, denoted by $\{P_{T-K}, \ldots, P_{T-1}\}$. To bridge the Prediction Module with the assumption that regions remaining static over consecutive frames are likely to stay static in the next frame, we designate a 3DG as a freeze candidate if it has remained *inactive* throughout all of the past K frames. This decision is made by:

$$G_p = \mathbb{I}\left(\sum_{k=1}^{K} P_{T-k} = 0\right), \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function. A value of $G_p = 1$ indicates that the 3DG will be frozen and excluded from gradient updates during the optimization of the current frame.

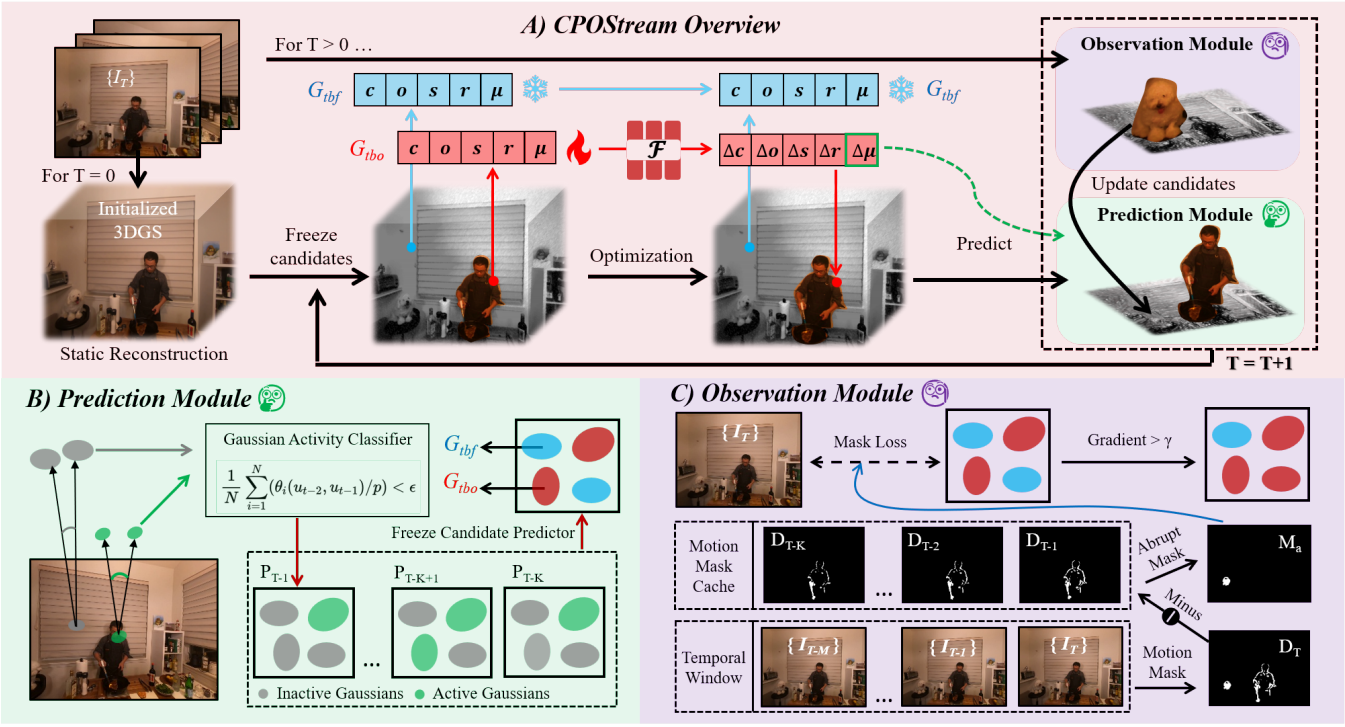


Figure 3: The architecture of **CPOStream**. **A)** presents the CPOStream overview. At frame T_0, we initialize the 3DGs from sparse point clouds estimated by SFM. For subsequent frames, 3DGs optimized in the previous frame serve as initializations. To mitigate flickering artifacts in static regions, prior to optimization at each frame, we classify 3DGs into *to-be-frozen* and *to-be-optimized* subsets using our prediction and observation modules. **B)** The prediction module labels each 3DG as "active" or "inactive" by the Gaussian Activity Classifier. 3DGs labeled as "inactive" for K consecutive frames are marked as *to-be-frozen* G_{tbf}, thus skipping attribute optimization in the following frame. Otherwise, they remain *to-be-optimized* G_{tbo}. **C)** The observation module detects sudden motion regions absent in the previous K frames by generating an abrupt mask M_a, reactivating corresponding 3DGs as *to-be-optimized* through the Freeze Candidate Updater.

Observation Module

The prediction module predicts the state of the current 3DG based on their historical activity status, but fails when previously static objects exhibit sudden motion in the current frame. To address this issue, we introduce an observation module that aims to reactivate 3DGs erroneously marked as to-be-frozen by the prediction module, if they fall within sudden motion regions unique to the current frame. The observation module consists of two components: a *Sudden Motion Detector* and a *Freeze Candidate Updater*.

Sudden Motion Detector. This detector identifies motion regions that are unique to the current frame that exhibit motion patterns not present in any of the previous K frames. To achieve this, we first extract a set of 2D motion masks $\{D_{T-k} \mid k = 1, 2..., K\}$ from the preceding K frames and a current motion mask D_T, where D_t is generated by calculating the standard deviation across a short temporal window of M frames, followed by Gaussian smoothing:

$$\begin{aligned} D_t &= \mathbb{I}(\mathcal{G}\left[S(I_t^M)\right] \geq \eta), \\ I_t^M &= \{I_{t-m} \,|m \in (0, 1, ..., M)\}, \end{aligned} \tag{5}$$

where $S(\cdot)$ denotes per-pixel standard deviation, $\mathcal{G}[\cdot]$ denotes Gaussian blur, and η is an empirically chosen threshold (set to 0.01). The indicator function $\mathbb{I}(\cdot)$ binarizes the result into a motion mask. We then compute the abrupt motion mask M_a as:

$$M_a = \left(\bigcap_{k=1}^{K} \neg D_{T-k}\right) \cap D_T, \tag{6}$$

where $\neg$ denotes logical negation. This mask highlights regions exhibiting motion exclusively in the current frame. To associate 2D sudden motion regions with 3DGs, we derive 2D gradients from the masked photometric loss. A 3DG is considered abruptly moving if it produces a strong gradient signal within M_a:

$$G_o = \nabla_{(u,v)} \left[M_a(u,v) \cdot \mathcal{L}_{\text{color}}\left(\tilde{I}(\mathcal{G}; u, v), I(u, v)\right)\right] > \gamma, \tag{7}$$

where $\nabla(u, v)$ denotes the 2D gradient for 3DG $\mathcal{G}$ in image coordinates (u, v), and γ is a threshold value.

Freeze Candidate Updater. The detector identifies 3DGs affected by sudden motion that were mistakenly classified as *to-be-frozen* by the Prediction Module. The Updater then reactivates them as *to-be-optimized*, $G_{re} \in (G_p \cap G_o)$, effectively correcting the prediction. The final G_{tbo} that requires

optimization in frame T is defined as follows:

$$G_{tbo} = \Gamma(\mathcal{G}, r), \quad \text{where} \quad \mathcal{G} \in (\neg G_p \cup G_{re}). \tag{8}$$

Here, Γ represents spatial clustering that ensures the coherence of moving objects, and r is a hyperparameter that defines the clustering radius.

Online Training

Given multi-camera images at frame T, the *to-be-optimized* 3DGs G_{tbo}, and the *to-be-frozen* 3DGs G_{tbf}, we draw on insights from prior works (Sun et al. 2024; Bao et al. 2025) and employ a lightweight hash MLP to estimate intermediate optimization residuals. In contrast to 3DGStream (Sun et al. 2024), which restricts updates to rigid transformations, we optimize all attributes of the moving 3DGs. This more flexible strategy has proven effective for modeling newly emerging objects, as demonstrated in recent works such as MGStream (Bao et al. 2025) and Dynamic3DGS (Luiten et al. 2024). Although both G_{tbo} and G_{tbf} participate in rendering, the attributes of G_{tbf} remain fixed, and only the following aspects of G_{tbo} are considered for optimization:

$$\begin{aligned} \Delta u_t, \Delta o_t, \Delta s_t, \Delta r_t, \Delta c_t &= \mathcal{F}(u_{t-1}(G_{tbo})), \\ u_t, s_t, q_t &= u_{t-1} + \Delta u_t, s_{t-1} + \Delta s_t, q_{t-1} \times \Delta q_t, \\ o_t, c_t &= o_{t-1} + \Delta o_t, c_{t-1} + \Delta c_t, \end{aligned} \tag{9}$$

where subscripts t and $t-1$ indicate attributes at respective frames, and $\mathcal{F}$ is the hash MLP used for smoothly modeling local attribute changes. Training loss follows the Eq. 2.

Initial Frame Training. In constructing streamable FVVs of dynamic scenes, the quality of the first frame plays a crucial role, as the optimization of each subsequent frame builds upon the previous one. To ensure robustness and stability during initialization, we perform per-image exposure compensation in the first-frame training phase. Specifically, we learn a camera-specific color correction matrix $\eta \in \mathbb{R}^{3\times3}$ and a bias vector $\psi \in \mathbb{R}^{3\times1}$ to address photometric inconsistencies such as exposure differences. The final rendered image is formulated as:

$$\tilde{I} = I_{splat} \cdot \eta + \psi, \tag{10}$$

where I_{splat} denotes the raw image rendered by the rasterizer. This correction step ensures that the initial reconstruction serves an accurate and photometrically consistent baseline for the rest of the video sequence.

First K Frames Training. For the first K frames, the *Prediction Module* cannot be applied due to the absence of temporal history. Instead, we adopt a coarse motion range selection for learning these frames. Specifically, we utilize only the motion mask D_t between each frame and its previous frame ($M = 1$). If the 2D projection of a 3DG falls within the range of this motion mask, we regard it as G_{tbo}.

Experiment

Setup

Datasets. We evaluate our method on two real-world datasets: N3DV (Li et al. 2022b) and MeetRoom (Li et al. 2022a). The N3DV dataset consists of 21 forward-facing camera views capturing a static indoor scene. Each camera records video at a resolution of 2704×2028 and a frame rate of 30 FPS. Following the protocol established in StreamRF (Li et al. 2022a) and 3DGSstream (Sun et al. 2024), we downsample all videos by a factor of 2 and use the undistorted pose from the first frame of each video as the camera pose. In all experiments, the central camera is designated as the test view, while the remaining cameras are used for training. The MeetRoom dataset captures a dynamic scene with a moving human subject using 13 synchronized cameras, each recording at 1280×720 resolution and 30 FPS. As in prior streaming methods (Li et al. 2022a; Sun et al. 2024), we initialize the scene using the undistorted pose from the first frame and a sparse point cloud. Camera 00 is used as the test view, and the other 12 cameras are used for training.

Metrics. To quantitatively evaluate temporal consistency, we adopt the E_{warp} metric introduced in video de-flickering works (Qiu et al. 2024; Lei et al. 2023; Lei, Xing, and Chen 2020), using the same optical flow model (Xu et al. 2023) as MGStream for fair comparison. Specifically, E_{warp} is computed as follows:

$$\begin{aligned} E_{\text{pair}}\ (I_i, I_j) &= \left\| M_{i,j} \odot (I_i - \mathcal{W}(I_j)) \right\|_1 \\ E^T_{\text{warp}} &= \frac{1}{T-1} \sum_{t=1}^{T} \{ E_{\text{pair}}\ (I_t, I_{t-1}) + E_{\text{pair}}\ (I_t, I_0) \}. \end{aligned} \tag{11}$$

Here, I_i and I_j are rendered images at different frames, $\mathcal{W}$ warps I_j to I_i via estimated optical flow, $M_{i,j}$ denotes the valid optical flow mask, and T is the total number of modeled frames. Additionally, we report PSNR, training and inference time, and storage cost to comprehensively benchmark against state-of-the-art streaming methods.

Implementation. All experiments were conducted on a Windows 11 system equipped with a single NVIDIA GeForce RTX 3090 GPU. The initial 3D Gaussian Splatting (3DGS) model is trained for 15000 iterations, after which the Gaussian components of each subsequent frame are optimized for only 200 iterations. During the sequential modeling process, we incorporate the information from the preceding five frames ($K = 5$) into the prediction and observation module. The Standard Deviation mask for each frame is computed utilizing information from its two adjacent frames. Hyperparameters ϵ and γ are empirically set to 0.0006 and 0.001, respectively.

Comparisons

Quantitative Evaluation. We compare our method against several state-of-the-art per-frame reconstruction baselines, including StreamRF (Li et al. 2022a), Dynamic3DG (Luiten et al. 2024), 3DGStream (Sun et al. 2024), MGStream (Bao et al. 2025), Hicom (Gao et al. 2024), Queen (Girish et al. 2024), and 4DGC (Hu et al. 2025). As shown in Tab. 1, our approach achieves superior temporal consistency across both the N3DV and MeetRoom datasets. Specifically, on N3DV, our method reduces the Ewarp metric by 0.0022 compared to the next best approach, MGStream. Moreover, we deliver the highest rendering quality, evidenced by our PSNR surpasses Queen by 1.32 dB. In terms of

Dataset	Method	$E_{warp}\downarrow$	PSNR (dB)↑	Storage (MB)↓	Train (s)↓	Render (FPS)↑
N3DV	StreamRF (Li et al. 2022a)	0.0103	30.66	17.7	15.0	8
	Dynamic-3DGS (Luiten et al. 2024)	0.0278	29.75	11.1	138.6	195
	3DGStream (Sun et al. 2024)	0.0126	31.84	7.6	13.8	145
	MGStream (Bao et al. 2025)	0.0100	32.02	2.1	13.2	176
	Hicom (Gao et al. 2024)	0.0158	30.85	0.7	**6.7**	**274**
	Queen (Girish et al. 2024)	0.0163	32.19	0.8	7.9	248
	4DGC (Hu et al. 2025)	0.0088	31.58	**0.5**	49.8	168
	Ours	**0.0078**	**33.51**	1.5	10.2	182
MeetRoom	StreamRF (Li et al. 2022a)	0.0100	26.72	5.7	10.2	10
	Dynamic-3DGS (Luiten et al. 2024)	0.0250	27.93	3.9	72.1	280
	3DGStream (Sun et al. 2024)	0.0110	30.31	4.0	7.9	219
	MGStream (Bao et al. 2025)	0.0090	31.02	0.7	7.6	255
	Hicom (Gao et al. 2024)	0.0139	28.68	0.4	3.9	284
	Queen (Girish et al. 2024)	0.0151	26.23	**0.3**	**2.2**	**473**
	4DGC (Hu et al. 2025)	0.0102	28.08	0.4	43.9	213
	Ours	**0.0079**	**33.18**	0.6	5.2	307

Table 1: Quantitative comparison with online methods on the N3DV and MeetRoom datasets. We report the E_{warp} for temporal consistency, PSNR for per-frame quality, storage, training and rendering time for efficiency. The E_{warp}, PSNR, storage memory, training and rendering time are averaged over 300 frames for each scene.

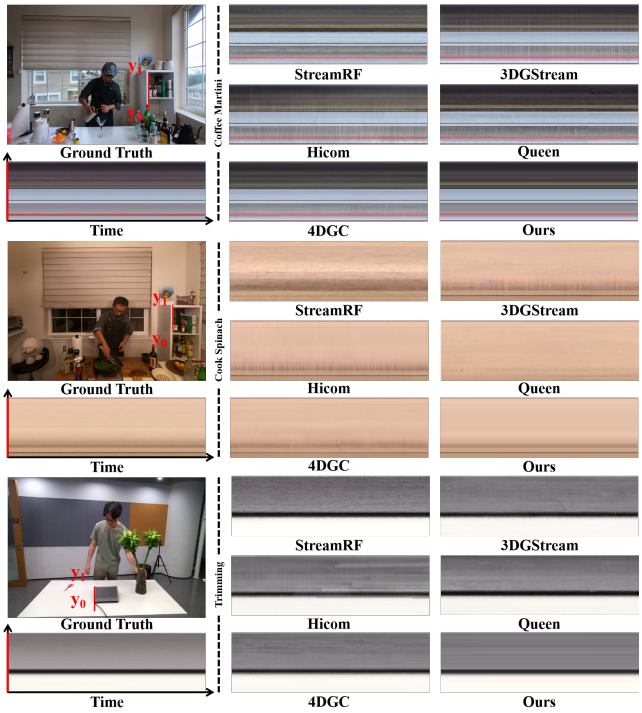


Figure 4: Qualitative comparison of temporal consistency, with frame index (0–300) on the x-axis and red line region (y_0–y_1) on the y-axis.

computational performance, our method maintains competitive storage efficiency, and demonstrates efficient training and inference time. On MeetRoom, our approach not only outperforms prior flicker-reduction-focused methods like MGStream in training efficiency, but also achieves higher single-frame reconstruction quality.

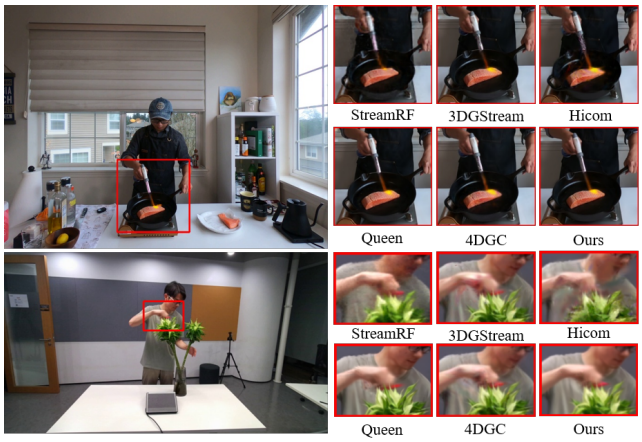
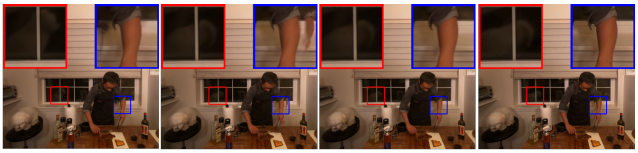


Figure 5: Qualitative comparison of rendering quality with representative online methods. We highlighted local details for better visual comparison.

Qualitative Evaluation. Fig. 4 and Fig. 5 show qualitative comparisons in terms of temporal consistency and rendering quality, respectively. Specifically, Fig. 4 visualizes the rendering color map of a static region over 300 consecutive frames, where the vertical axis corresponds to the red line in the image, and the horizontal axis represents all modeled frames. Our method exhibits notably improved stability in static regions, effectively suppressing flickering artifacts that remain evident in all other methods. In Fig. 5, we zoom in on dynamic local details reconstructed by various streaming approaches. Compared to motion distortions (*e.g.* Methods 3DGStream and Hicom in the first scene) and ghosting artifacts (*e.g.* Methods Queen and 4DGC in the second scene), our method consistently produces reconstructions that are perceptually closer to the ground truth.

Pre	Obv	Clu	Ewarp↓	PSNR↑	Dy-PSNR↑
✓			N/A	N/A	N/A
	✓		0.0081	32.47	29.21
✓		✓	**0.0071**	32.98	30.95
	✓	✓	0.0076	33.06	30.72
✓	✓	✓	0.0078	**33.51**	**31.82**

Table 2: Ablation Study on Architecture, where Pre, Obv and Clu represent the Prediction module, the Observation module and the clustering algorithm, respectively. We further report PNSR in dynamic regions (Dy-PSNR) for effective performance evaluation.



(a) Pre. + Clu. (b) Obv. + Clu. (c) Full Model (d) GT

Figure 6: Ablation Study on Architecture. The red box highlights the shadow of the puppy, while the blue box highlights the continuously moving man's arm.

Ablation Study

Effect of the CPOStream Architecture. To evaluate the individual contributions of CPOStream components, we perform ablation studies isolating the Prediction and Observation Modules (Tab. 2 and Fig. 6). Using only the Prediction Module causes 3DGs to become irreversibly static after approximately 200 frames, halting further optimization. The Observation Module alone reduces rendering quality in dynamic regions by 2 dB. The Prediction Module with clustering fails to detect sudden motion (e.g., the dog's shadow highlighted by the red box in Fig. 6), while the Observation Module with clustering cannot handle sustained motion (e.g., a moving arm). Misclassifying dynamic areas as static leads to slightly lower ewarp scores but significantly degrades dynamic PSNR. Combining both modules yields the best performance and temporal consistency.

Effect of the Number of Prediction Frames (K). We evaluate the Prediction Module under varying values of K to identify the optimal number of historical frames (Tab. 3). A small K (e.g., $K = 1$) leads to frequent misclassification of static 3DGs as dynamic, increasing flicker and storage cost. In contrast, a large K overly restricts dynamics, impairing reconstruction quality. We adopt $K = 5$ as the default to balance accuracy and performance.

Effect of the Number of Observation Frames. To determine the optimal number of frames for computing the STD Mask in the Observation Module, we evaluate various settings (Tab. 4). Results show that additional frames yield only marginal PSNR gains. Thus, we use two historical frames to ensure robustness while avoiding unnecessary computation.

Effect of the Prediction Module Threshold (ϵ). We investigate the impact of the threshold parameter ϵ in the Prediction Module (Tab. 5). Higher ϵ improves storage efficiency but degrades dynamic region reconstruction, while lower values have the opposite effect. We set $\epsilon = 0.0006$ to balance storage and quality.

Pre Frames (K)	Ewarp↓	Storage↓	PSNR↑	dy-PSNR↑
1	0.0087	2.85	28.88	**27.06**
3	0.0066	**2.07**	28.88	26.98
5	0.0061	2.18	**28.90**	27.05
8	**0.0055**	2.19	28.83	26.97
10	0.0060	1.15	28.49	25.59

Table 3: Ablation Study on Prediction Frames.

Obv Frames (M)	Ewarp↓	Storage↓	PSNR↑	dy-PSNR↑
1	**0.0058**	**2.33**	**28.88**	**27.04**
2	0.0060	2.44	28.87	27.03
3	0.0063	2.50	28.88	27.06
5	0.0065	2.57	28.90	27.08

Table 4: Ablation Study on Observation Frames.

ϵ	Ewarp↓	Storage↓	PSNR↑	dy-PSNR↑
0.0004	0.0062	3.37	**28.92**	**27.08**
0.0006	0.0061	2.18	28.90	**27.08**
0.0008	**0.0060**	2.15	28.88	27.06
0.0010	**0.0060**	**2.03**	28.87	27.03

Table 5: Ablation Study on the Pre-Module Threshold ϵ.

γ	Ewarp↓	Storage↓	PSNR↑	dy-PSNR↑
0.0001	0.0064	2.50	**28.91**	27.05
0.0005	0.0063	2.48	**28.91**	**27.07**
0.0010	0.0061	**2.18**	28.90	27.05
0.0020	0.0060	2.42	28.89	27.07
0.0050	**0.0059**	2.41	28.87	27.06

Table 6: Ablation Study on the Obv-Module Threshold γ.

Effect of the Observation Module Threshold (γ). We evaluate the threshold parameter γ in the Observation Module using various settings (Tab. 6). Both overly small and large γ values improve storage efficiency but degrade reconstruction quality. We choose $\gamma = 0.001$ to achieve a balanced trade-off between storage and visual fidelity.

Conclusion

In this paper, we present CPOStream, a novel 3DGS-based framework for flicker-free free-viewpoint video (FVV) streaming of real-world dynamic scenes. By leveraging past inactive information of 3DGs to fix the attributes of frozen 3DGs before per-frame modeling, CPOStream significantly improves temporal consistency and reconstruction quality. Extensive experiments show that CPOStream achieves SOTA performance in both fidelity and stability, demonstrating its effectiveness for flicker-free, high-quality FVV applications. Nevertheless, our method still relies on the quality of the initial static reconstruction. Thus the performance may degrade in scenarios with extremely sparse inputs (e.g., only 1–3 cameras).

References

Bao, Z.; Li, Q.; Liao, G.; Zhao, Z.; and Liu, K. 2025. MGStream: Motion-aware 3D Gaussian for Streamable Dynamic Scene Reconstruction. *arXiv preprint arXiv:2505.13839*.

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

Chen, S. E.; and Williams, L. 2023. View interpolation for image synthesis. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 423–432.

Collet, A.; Chuang, M.; Sweeney, P.; Gillett, D.; Evseev, D.; Calabrese, D.; Hoppe, H.; Kirk, A.; and Sullivan, S. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4): 1–13.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5501–5510.

Gao, Q.; Meng, J.; Wen, C.; Chen, J.; and Zhang, J. 2024. Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37: 80609–80633.

Girish, S.; Li, T.; Mazumdar, A.; Shrivastava, A.; De Mello, S.; et al. 2024. Queen: Quantized efficient encoding of dynamic gaussians for streaming free-viewpoint videos. *Advances in Neural Information Processing Systems*, 37: 43435–43467.

Hu, Q.; Zheng, Z.; Zhong, H.; Fu, S.; Song, L.; Zhang, X.; Zhai, G.; and Wang, Y. 2025. 4DGC: Rate-Aware 4D Gaussian Compression for Efficient Streamable Free-Viewpoint Video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 875–885.

Innmann, M.; Zollhöfer, M.; Nießner, M.; Theobalt, C.; and Stamminger, M. 2016. Volumedeform: Real-time volumetric non-rigid reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, 362–379. Springer.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Lei, C.; Ren, X.; Zhang, Z.; and Chen, Q. 2023. Blind video deflickering by neural filtering with a flawed atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10439–10448.

Lei, C.; Xing, Y.; and Chen, Q. 2020. Blind video temporal consistency via deep video prior. *Advances in Neural Information Processing Systems*, 33: 1083–1093.

Li, L.; Shen, Z.; Wang, Z.; Shen, L.; and Tan, P. 2022a. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35: 13485–13498.

Li, T.; Slavcheva, M.; Zollhoefer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; et al. 2022b. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5521–5531.

Li, Z.; Chen, Z.; Li, Z.; and Xu, Y. 2024. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8508–8520.

Luiten, J.; Kopanas, G.; Leibe, B.; and Ramanan, D. 2024. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, 800–809. IEEE.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Qiu, X.; Han, C.; Zhang, Z.; Li, B.; Guo, T.; Wang, P.; and Nie, X. 2024. BlazeBVD: Make Scale-Time Equalization Great Again for Blind Video Deflickering. *arXiv preprint arXiv:2403.06243*.

Sun, J.; Jiao, H.; Li, G.; Zhang, Z.; Zhao, L.; and Xing, W. 2024. 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*, 20675–20685.

Wang, L.; Hu, Q.; He, Q.; Wang, Z.; Yu, J.; Tuytelaars, T.; Xu, L.; and Wu, M. 2023. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 76–87.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20310–20320.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; Yu, F.; Tao, D.; and Geiger, A. 2023. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yan, J.; Peng, R.; Wang, Z.; Tang, L.; Yang, J.; Liang, J.; Wu, J.; and Wang, R. 2025. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 16520–16531.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20331–20341.

Yang, Z.; Yang, H.; Pan, Z.; and Zhang, L. 2023. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*.

Yun, Y.; Bae, J.; Son, H.; Kim, S.; Lee, H.; Bang, G.; and Uh, Y. 2025. Compensating Spatiotemporally Inconsistent Observations for Online Dynamic 3D Gaussian Splatting. *arXiv preprint arXiv:2505.01235*.

Zitnick, C. L.; Kang, S. B.; Uyttendaele, M.; Winder, S.; and Szeliski, R. 2004. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3): 600–608.