




Temporally Aware Denisfication for Dynamic 3D Gaussian Splatting

Vikram Sandu¹, Mayurdeep Pathak¹, and Rajiv Soundararajan¹

Indian Institute of Science, Bengaluru, India
 {vikramsandu,mayurdepp,rajivs}@iisc.ac.in

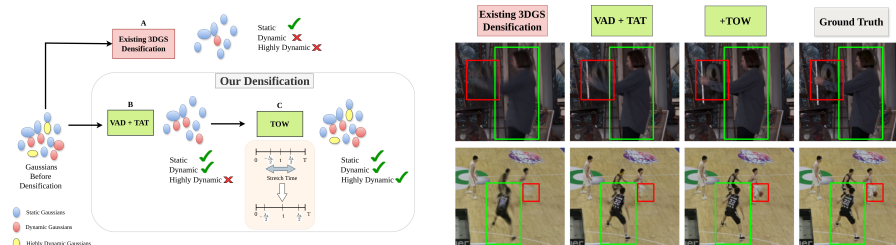


Fig. 1: Left) (A) Existing 3DGS densification fails to densify short-lived dynamic Gaussians. (B) VAD with visibility-weighted gradients and TAT with lifespan-aware temporal thresholds enable densification of short-lived dynamic Gaussians. (C) TOW warps temporal coordinates around center t within window λ_t , boosting deformation capacity and densifying highly dynamic Gaussians. **Right)** Improvement over the densification baseline as VAD, TAT, and TOW are incrementally incorporated, resulting in sharper and more consistent dynamic reconstructions. VAD and TAT enhance moderately dynamic regions shown in green box, while TOW further improves reconstruction in highly dynamic regions such as the hand (top image) and basketball (bottom image) as shown in red box.

Abstract. Despite modeling temporal motion, dynamic 3D Gaussian Splatting (3DGS) methods still inherit a static densification strategy ill-suited for dynamic scenes. This neglect of temporal behavior leads to under-reconstructed and blurry dynamic regions, as short-lived Gaussians receive sparse supervision and fail to densify effectively. We propose a *Visibility-Aware Denisfication* (VAD) framework that integrates temporal visibility into the densification process, ensuring that Gaussians are refined based on their actual temporal presence. A *Temporally-Adaptive Thresholding* (TAT) mechanism further adjusts each Gaussian’s densification threshold according to its temporal lifespan, promoting balanced refinement of both static and dynamic regions. Finally, a *Temporal Offset Warping* (TOW) design enhances deformation capacity around temporal centers, extending the lifespan of highly dynamic Gaussians and facilitating more effective densification. Our approach achieves substantial improvements in the visual quality of dynamic regions, outperforming existing methods across three dynamic multi-view benchmark datasets. Moreover, the proposed VAD module generalizes across diverse dynamic 3DGS methods, consistently improving dynamic reconstruction as a plug-and-play component. The project page is available [here](#).

Keywords: Novel View Synthesis · Gaussian Splatting · Denisfication

1 Introduction

Novel view synthesis (NVS) is a fundamental problem in 3D vision that aims to generate realistic views of a scene from unseen viewpoints by reasoning about its underlying geometry, appearance, and illumination, making it essential to modern 3D vision, graphics, and robotics. Since the advent of Neural Radiance Fields (NeRF) [16], view synthesis has witnessed rapid progress, with numerous follow-up works [1, 3, 5, 17, 24, 25, 27] enhancing reconstruction quality and efficiency, yet still constrained by slow rendering. More recently, 3D Gaussian Splatting (3DGS) [9] has emerged as a compelling alternative, enabling real-time rendering while maintaining photorealistic fidelity. Building upon this foundation, several approaches have extended 3DGS to dynamic scenes - by introducing deformation fields [14, 15, 31], modeling Gaussians directly in 4D space [4, 35], or leveraging temporal keyframe interpolation [11]. While these methods differ in how they represent scene dynamics, they all fundamentally rely on the same densification strategy as the original 3DGS, which was primarily designed for static scenes. Densification enhances scene representation by introducing new Gaussians through splitting or cloning existing ones in regions where reconstruction lacks detail. Gaussians are selected for densification when their accumulated positional gradients, averaged over a predefined interval, exceed a fixed threshold.

However, we observe that the existing densification strategy overlooks the temporal behavior of Gaussians, making it suboptimal for dynamic 3DGS frameworks and often leading to under-reconstructed or blurry dynamic regions in rendered views. Notably, dynamic Gaussians typically exhibit shorter temporal lifespans to effectively represent complex scene dynamics [34, 37]. Consequently, their visibility, parametrized by the opacity of each Gaussian, is limited to only a few frames. This sparse visibility yields equally sparse supervision and gradient updates during training, which prevents these Gaussians from accumulating the positional gradients required for densification, leaving them permanently under-densified. This motivates a densification strategy that explicitly accounts for temporal visibility.

In this work, we propose a *visibility-aware densification* (VAD) criterion that integrates Gaussian visibility into the densification process by accumulating visibility-weighted positional gradients and normalizing them by total visibility. This ensures that the effective densification signal for each Gaussian is primarily influenced by the frames in which it is actually visible. To further promote the densification of short-lived Gaussians, we introduce a *temporally-adaptive thresholding* (TAT) mechanism that dynamically adjusts each Gaussian’s densification threshold in proportion to its temporal lifespan, facilitating densification of such Gaussians. We further observe that certain Gaussians exhibit extremely short temporal lifespans, typically those representing highly dynamic regions—due to the deformation field’s limited ability to model abrupt or complex motion. This results in near-zero sampling probability of such Gaussians during the densification process. To alleviate this issue, we adopt multiple temporal centers, inspired by [11, 14], and propose a *temporal offset warping* (TOW) mechanism that allocates greater deformation capacity near each Gaussian’s temporal center. This

design enhances the modeling of complex motion and effectively extends the temporal lifespan of highly dynamic Gaussians, thereby improving densification in such regions.

Together, these components establish a unified framework for visibility-driven Gaussian densification in dynamic 3DGS models as shown in Fig 1. The key contributions of our work are summarized below.

1. *Visibility-aware densification* criterion that incorporates per-frame temporal visibility into the densification process, seamlessly integrating with existing dynamic 3DGS frameworks to improve reconstruction quality in dynamic regions.
2. *Temporally-adaptive thresholding* mechanism that adjusts the densification gradient threshold based on each Gaussian’s temporal lifespan, enabling effective densification of short-lived Gaussians.
3. *Temporal offset warping* that allocates greater deformation capacity near temporal centers, enhancing the modeling of complex motions and effectively improving densification in highly dynamic regions.

2 Related Work

Static 3DGS. Since the introduction of 3DGS [9], numerous extensions have aimed to enhance its rendering quality, generalization, and efficiency across diverse scenarios [7, 18, 38, 40, 41]. These works primarily focus on improving static scene representations through better appearance modeling, adaptive Gaussian management, or geometric regularization.

Dynamic 3DGS. To handle dynamic scenes, several approaches extend 3DGS by incorporating temporal modeling [11, 14, 15, 23, 26, 31, 35, 36]. Existing methods can be broadly categorized as: (1) 4D Gaussian representations, and (2) deformable 3D Gaussians. 4D-based methods, such as 4DGS [35] and 4DRotorGS [4], represent spatiotemporal variations directly in a unified 4D space. 4DGS leverages 4D primitives and spherical harmonics to jointly encode view- and time-dependent appearance, whereas 4DRotorGS models frame-wise dynamics through temporally varying Gaussian rotations.

Deformable Gaussian methods, on the other hand, maintain a canonical 3D representation and learn a deformation field to capture motion over time. Examples include 4DGaussians [31], which employs multi-resolution voxel planes for deformation; STG [14], which parameterizes trajectories with low-order polynomials; SplineGS [19] uses B-spline instead for motion trajectories and GaussianFlow [15], which uses a dual deformation field combining polynomial and Fourier representations. Deformable-3DGS [36] and SWinGS [23] adopt MLP-based temporal deformation for motion modeling. More recently, Ex-4DGS [11] employs keyframe interpolation for dynamic modeling. Other approaches, such as Swift4D [32] and LongVolCap [33], focus on improving computational efficiency and scalability for long-term dynamic reconstruction.

Densification. A substantial amount of work has been done on densification of 3D Gaussians for static scenarios. Kim et al. [10] introduce additional color

gradients into the densification process to achieve more efficient scene modeling. PGD-GS [8] progressively adjusts the densification threshold to improve reconstruction in sparse input settings. Revising Densification [21] proposes error-based densification, which distributes pixel-level errors to individual Gaussians and decides densification accordingly. Pixel-GS [40] weighs positional gradients by pixel coverage area to achieve better reconstruction in regions with poor initialization. Grubert et al. [6] employ exponential scheduling of a predefined gradient threshold for faster convergence and improved densification in complex regions. HDA-GS [29] classifies Gaussians into multiple categories based on their size and applies different densification strategies to each. More recently, AD-GS [20] presents a regulated densification process that alternates between high and low densification phases to improve reconstruction under sparse-view settings.

However, only a limited number of works have addressed densification in dynamic 3DGS. SaRO-GS [34] identifies the issue of short-lived Gaussians and introduces an adaptive optimization scheme that moderates each Gaussian’s learning rate and densification threshold based on a temporally integrated state function to improve reconstruction in dynamic regions. STG [14] employs guided sampling by introducing Gaussians in regions with high reconstruction error to enhance scene reconstruction. More recently, Anchored-4DGS [13] guides densification in dynamic regions by adapting the scaling threshold according to each Gaussian’s temporal coverage to optimize anchors for efficient storage. While these methods offer improvements, they do not fully address the fundamental mismatch between static densification strategies and dynamic scene requirements. The core issue of temporal visibility bias in gradient accumulation remains unexamined, and existing solutions lack a unified framework that jointly optimizes densification criteria, threshold adaptation, and deformation capacity allocation.

In contrast, our work provides a comprehensive analysis of densification challenges in dynamic settings and proposes a detailed solution along with a generalized VAD module that can be seamlessly integrated into any existing dynamic 3DGS framework to improve dynamic reconstruction.

3 Preliminaries

3D Gaussian Splatting (3DGS) represents a scene as a set of Gaussian primitives $\mathcal{G} = \{G_i\}$, where each primitive G_i is parameterized by $\theta_i = (\mu_i, \Sigma_i, \sigma_i, f_i)$. Here, $\mu_i \in \mathbb{R}^3$ denotes the 3D position, $\Sigma_i \in \mathbb{R}^{3 \times 3}$ the covariance, $\sigma_i \in [0, 1]$ the opacity, and f_i the spherical harmonics coefficients (or color feature vector). The covariance Σ_i is further decomposed as

$$\Sigma_i = R_i S_i S_i^\top R_i^\top,$$

where R_i represents the rotation and $S_i \in \mathbb{R}^{3 \times 3}$ is a diagonal scale matrix. This decomposition ensures that Σ_i remains positive semi-definite, allowing for stable and smooth optimization during training.

For rendering, the Gaussians are first sorted in front-to-back order, and the final color at pixel p in camera plane π is computed as

$$f_p = \sum_{i \in \mathcal{G}_p} \phi(f_i, \mathbf{d}_\pi) \alpha_i^{2D} \prod_{j \in \mathcal{G}_p, j < i} (1 - \alpha_j^{2D}), \quad (1)$$

where \mathcal{G}_p is the set of Gaussians influencing pixel p , and

$$\alpha_i^{2D} = \sigma_i G_i^\pi(p), \quad (2)$$

with $G_i^\pi(p)$ denoting the value of the 2D Gaussian function at pixel p , derived from its 3D mean μ_i and covariance Σ_i under camera projection. The function $\phi(f_i, \mathbf{d}_\pi)$ maps the spherical harmonics coefficients f_i of Gaussian i to an RGB color along the viewing direction \mathbf{d}_π .

Denisfication is a key mechanism in 3DGS that introduces new Gaussians in under-reconstructed regions, typically by splitting or cloning existing ones. Regions requiring denisfication are identified using the screen-space positional gradient, defined as $g_i = \|\nabla_{\mu_i^{2D}} L\|_2$, where L denotes the photometric reconstruction loss and μ_i^{2D} is the 2D projection of μ_i . Denisfication is performed at regular intervals of N iterations based on the accumulated gradient signal. Specifically, a Gaussian G_i is eligible for denisfication if

$$\frac{1}{N} \sum_{n=1}^N g_i^{(n)} > \tau_{\text{pos}}, \quad (3)$$

where N is set to 100 iterations and τ_{pos} to 0.0002 in the 3DGS implementation.

4 Method

We propose a framework that consists of three critical components designed to enhance the denisfication process in dynamic 3DGS models as shown in Fig 2, thus improving reconstruction fidelity in dynamic regions: **(i)** VAD, **(ii)** TAT, and **(iii)** TOW. The first two components, VAD and TAT, reformulate the existing 3DGS denisfication strategy by incorporating temporal variations, while the third component, TOW, indirectly facilitates denisfication by improving deformation modeling, allowing for longer temporal lifespan of Gaussians and supervision from more frames. We first describe the deformation model on which we impose our denisfication ideas.

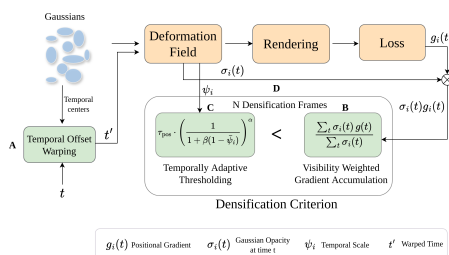


Fig. 2: Overview of our proposed denisfication framework. (A) Temporal Offset Warping adaptively warps the input time around each Gaussian’s temporal center, stretching regions near the center and compressing those farther away. (B) We accumulate the visibility weighted gradient signal for the N denisfication frames. (C) We dynamically adjust the denisfication threshold based on temporal scale ψ averaged over the N denisfication frames. (D) A Gaussian is denisfied when its visibility-weighted gradient accumulation exceeds the temporally adaptive threshold.

4.1 Deformation Model

We model the temporal evolution of each Gaussian through a Fourier-based deformation for motion, a temporal radial basis function (RBF) for opacity, and polynomial functions for rotation and scale, while view- and time-dependent color variations are represented using spherical harmonics.

Temporal RBF for Opacity. We employ a temporal radial basis function, similar to [14], to model time-dependent opacity. The opacity at time t is defined as

$$\sigma_i(t) = \sigma_i^s e^{-\psi_i(t-t_i)^2}, \quad (4)$$

where σ_i^s denotes the time-independent spatial opacity, and t_i and ψ_i represent the temporal center and temporal scale of the Gaussian, corresponding to the time instant of maximum visibility and its effective temporal lifespan, respectively. This formulation naturally provides the temporal lifespan of each Gaussian, which is later utilized in Section 4.3 for temporally adaptive thresholding.

Fourier-based Motion. Inspired by [15], for each Gaussian G_i , the Gaussian mean at time t is defined as

$$\mu_i(t) = \mu_i + \sum_{f=1}^K \left(\mathbf{M}_{i,f}^{(c)} \cos(\Delta t_{i,f}) + \mathbf{M}_{i,f}^{(s)} \sin(\Delta t_{i,f}) \right), \quad (5)$$

where the phase term $\Delta t_{i,f}$ is given by

$$\Delta t_{i,f} = 2\pi f(t - t_i) + \phi_i.$$

Here, μ_i denotes the canonical Gaussian mean. The Fourier coefficients $\mathbf{M}_{i,f}$ model motion as a sum of smooth periodic oscillations, while the phase offsets ϕ_i introduce beneficial asymmetry around each Gaussian’s temporal center t_i , enabling the modeling of complex, non-uniform motion. All parameters μ_i , $\mathbf{M}_{i,f}$, and ϕ_i are learnable. We set $K = 4$ in our experiments.

Polynomial Rotation and Scale. We parameterize rotation using quaternions and employ a polynomial formulation, following prior works [2, 14], to represent time-dependent rotation and scale. The quaternion and scale at time t are defined as

$$q_i(t) = \sum_{k=0}^{N_q} c_{ik}(t - t_i)^k, \quad s_i(t) = \sum_{k=0}^{N_s} d_{ik}(t - t_i)^k, \quad (6)$$

where c_{ik} and d_{ik} are learnable coefficients, and N_q and N_s are set to 1 in our implementation. We use low-degree polynomials for rotation and scale as they typically exhibit smoother temporal variations compared to positional motion.

Each Gaussian G_i is associated with a temporal center t_i , which is fixed at initialization. We initialize Gaussians across multiple temporal centers similar to [14, 34], offering two key benefits: (i) richer temporal coverage during initialization, and (ii) deformation learning that is localized and more stable, as it is

performed relative to each Gaussian’s own temporal center. A similar concept is adopted in [11], where it is referred to as keyframe interpolation. We initialize the temporal centers uniformly, with the spacing empirically set to 20 frames in our implementation.

4.2 Visibility-Aware Denisfication

We first identify a key limitation in the existing denisfication strategy (Eq. 3), and subsequently introduce a temporal visibility-aware formulation that promotes the denisfication of short-lived dynamic Gaussians.

Consider a Gaussian G_i that is visible in a subset $\mathcal{M} \subset \mathcal{T}$, where \mathcal{T} is the entire set of frames. The cardinality of \mathcal{M} , $|\mathcal{M}| = M$. Also $|\mathcal{T}| = T$. Typically, $M \ll T$, corresponding to a short-lived dynamic Gaussian within the total video duration of T frames. Let $\mathcal{N} \subset \mathcal{T}$ correspond to set of N frames considered for denisfication. As a result, G_i contributes sparsely in \mathcal{N} . Let $g_i(t)$ be the gradient signal of frame t . The consolidated denisfication gradient signal \bar{g}_i under the standard formulation can be decomposed as

$$\bar{g}_i = \frac{1}{N} \sum_{t \in \mathcal{M} \cap \mathcal{N}} g_i(t) + \frac{1}{N} \sum_{t \in \mathcal{M}^c \cap \mathcal{N}} g_i(t). \quad (7)$$

Here, the first term contributes weakly, as the limited number of visible frames is penalized by the large denominator, while the second term vanishes since $g_i(t)$ is zero for frames where G_i is not visible. Consequently, dynamic Gaussians receive disproportionately low gradient signals that rarely exceed the denisfication threshold τ_{pos} , thereby suppressing their denisfication. This imbalance becomes more pronounced as the total video duration T increases.

To address this issue, we propose a simple yet effective denisfication strategy that reweights the gradient signal by the temporal visibility of each Gaussian. The consolidated gradient signal for denisfication is reformulated as

$$\bar{g}_i = \frac{\sum_{t \in \mathcal{N}} \sigma_i(t) g_i(t)}{\sum_{t \in \mathcal{N}} \sigma_i(t)}, \quad (8)$$

where $\sigma_i(t)$ denotes the opacity of Gaussian G_i in frame t . Note that if a Gaussian remains completely visible or opaque across all frames (i.e., $\sigma_i(t) = 1$ for all t , corresponding to the static case), our denisfication strategy naturally reduces to the original 3DGS formulation defined in Eq. 3.

4.3 Temporally Adaptive Thresholding

Even after applying visibility-aware denisfication, we observe that certain short-lived dynamic Gaussians struggle to exceed the fixed denisfication threshold τ_{pos} , originally tuned for static scenes in 3DGS.

To address this, we introduce a temporally adaptive thresholding strategy that modulates the denisfication threshold according to each Gaussian’s normalized temporal lifespan $\psi \in [0, 1]$. Specifically, we redefine the threshold for a

Gaussian G_i as

$$\tau_{\text{pos}}^i = \tau_{\text{pos}} \cdot \left(\frac{1}{1 + \beta(1 - \bar{\psi}_i)} \right)^\alpha, \quad (9)$$

where τ_{pos} denotes the base threshold, and $\bar{\psi}_i = \frac{1}{N} \sum_{n \in \mathcal{N}} \psi_i^{(n)}$ represents the average temporal scale of Gaussian G_i over the set of frames considered for densification. Note that $\psi_i^{(n)}$ corresponds to the temporal scale of the Gaussian when the n^{th} frame is sampled. This adaptive relaxation of the threshold for short-lived Gaussians (small ψ_i), promotes more frequent densification, while maintaining a stricter criterion for long-lived (large ψ_i) ones. The hyperparameters α and β control the curvature and sensitivity of this temporal modulation. We choose this function with sub-linear decay to achieve more aggressive decay close to life-span 1 and slower decay close to life-span 0. Thus, other polynomial or exponential functions with appropriate parameters may also be used instead.

Since ψ_i is a learnable parameter updated at every iteration, we use its averaged form $\bar{\psi}_i$ to determine the effective temporal lifespan. In practice, we set $\beta = 0.3$ to provide moderate threshold relaxation and $\alpha = 1.0$ for linear scaling, empirically balancing densification quality and stability. Notably, for Gaussians with full temporal lifespan ($\psi_i = 1$), the adaptive threshold simplifies to $\tau_{\text{pos}}^i = \tau_{\text{pos}}$, recovering the standard static 3DGS behavior.

Incorporating both the visibility-aware densification and temporally adaptive thresholding, our final densification criterion for a Gaussian G_i is expressed as

$$\frac{\sum_{t \in \mathcal{N}} \sigma_i(t) g_i(t)}{\sum_{t \in \mathcal{N}} \sigma_i(t)} > \tau_{\text{pos}} \cdot \left(\frac{1}{1 + \beta(1 - \bar{\psi}_i)} \right)^\alpha \quad (10)$$

Here, the left-hand side aggregates visibility-weighted gradients, while the right-hand side relaxes the threshold for short-lived Gaussians, together facilitating their effective densification.

4.4 Temporal Offset Warping

Despite our visibility-aware and temporally adaptive densification strategy, we observe that certain Gaussians, particularly those corresponding to highly dynamic regions, remain under-densified. This issue arises from the limited capacity of the deformation field to model complex or rapid motion. Consequently, these Gaussians exhibit extremely low temporal scales (ψ_i), which substantially reduces their sampling probability among the frames considered for densification.

The core limitation is that a standard Fourier basis, with its globally-uniform frequencies, allocates modeling capacity uniformly across time. This is inefficient for dynamic scenes where motion complexity is non-uniform, often requiring high-frequency modeling near a Gaussian’s peak activity and only low-frequency modeling elsewhere.

To address this, we introduce Temporal Offset Warping (TOW), a simple input warping scheme that enables a fixed set of Fourier basis functions to achieve **adaptive frequency allocation**. By non-linearly warping the input offset, we

concentrate the model’s capacity to represent high-frequency details near each Gaussian’s temporal center while representing motion at lower frequencies farther away. The warped time t' is defined as:

$$t' = t_i + \mathcal{W}(t - t_i; \lambda_t, \rho_t), \quad (11)$$

where $\lambda_t \in (0, 1)$ denotes the normalized focus window size (as a fraction of the total temporal range T), and $\rho_t \in (0, 1)$ represents the fraction of temporal capacity allocated within the focus window. The piecewise-linear warping function \mathcal{W} is formulated as:

$$\mathcal{W}(\Delta t) = \begin{cases} s_{\text{near}} \cdot \Delta t, & |\Delta t| \leq \frac{\lambda_t}{2}, \\ s_{\text{far}} \cdot \Delta t, & \text{otherwise,} \end{cases} \quad (12)$$

where $\Delta t = t - t_i$, and the near- and far-region scaling factors are defined as:

$$s_{\text{near}} = \frac{\rho_t}{\lambda_t}, \quad s_{\text{far}} = \frac{1 - \rho_t}{1 - \lambda_t}. \quad (13)$$

This formulation guarantees that exactly a fraction ρ_t of the total temporal deformation capacity is allocated within the focus window.

Adaptive Frequency Interpretation.

The key insight is that warping the input time effectively retunes the frequencies of the Fourier basis relative to physical time. The derivative of the warp function, $d\mathcal{W}/d(\Delta t)$, defines the local frequency scaling factor. Within the focus window ($|\Delta t| \leq \lambda_t/2$), the scaling factor $s_{\text{near}} > 1$ stretches the input, causing the fixed Fourier frequencies to represent higher effective frequencies in physical time, enabling the modeling of complex, rapid motion. Outside the window, $s_{\text{far}} < 1$ compresses the input, causing the same basis functions to represent lower effective frequencies, suitable for smoother motion as shown in Fig. 3. Unlike adding Fourier coefficients, which increases the parameter count and amplifies high-frequency noise, TOW achieves adaptive resolution without new parameters.

This warped time t' replaces t in our deformation model (Equations 4, 5, and 6). The enhanced deformation capability, particularly the ability to model high-frequency motion near t_i , prevents the collapse of temporal scales to extremely

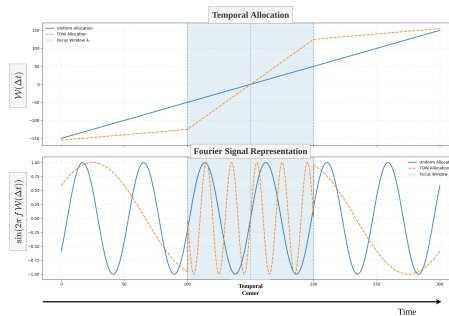


Fig. 3: Temporal Offset Warping (TOW). **Top:** Warping function $\mathcal{W}(\Delta t)$ vs. time. Uniform mapping has constant unit slope ($s = 1$), whereas TOW applies a piecewise-linear reparameterization with slope $s_{\text{near}} > 1$ inside the focus window and $s_{\text{far}} < 1$ outside, expanding near-center offsets and compressing distant ones while preserving the total temporal span. **Bottom:** Signals generated under uniform and warped temporal offsets. TOW enables finer temporal variation within the focus window while enforcing smoother evolution in distant regions.

low values. This enables Gaussians in highly dynamic regions to maintain a longer effective temporal lifespan, thereby increasing their sampling probability during densification and leading to superior reconstruction. In practice, for a sequence of T frames, we set $\lambda_t = \frac{50}{T}$ and $\rho_t = 0.75$, thereby concentrating 75% of the temporal modeling capacity within the 50-frame neighborhood around t_i .

5 Experiments

We utilize three benchmark multiview datasets for our experiments: **Neural 3D Video** (N3DV) [12] dataset comprises six indoor scenes captured by 17–21 time synchronized cameras recorded at 2704×2028 resolution, each featuring intricate motion confined to a small region of the frame. We use the central camera for evaluation. **Interdigital** [22] consists of five indoor scenes captured by a 4×4 camera rig at 2048×1088 resolution. In contrast to Neural 3D Video, these sequences exhibit long-range motion and abrupt object appearances and disappearances. We use the camera in the second row and the second column for evaluation. **VRU Basketball** [32] features two real-world basketball scenes captured with 34 cameras at 1920×1080 resolution. The scenes include fast human and ball movements, making them highly challenging for accurately modeling complex non-rigid motion and interactions. We evaluate on the camera at the center. For all experiments, we use videos downsampled to half of their original resolution.

Setup. Two experimental setups are commonly used across dynamic 3D Gaussian Splatting methods. Approaches such as [11, 14] train on shorter temporal segments (e.g., 50 frames) and report results aggregated over the full 300-frame sequence. In contrast, most methods like [31, 32, 34, 35] train a single model over all 300 frames. We adopt the latter setting, as densification challenges become more pronounced when training on longer continuous sequences due to sparser temporal sampling, whereas short-segment training increases total training time and can introduce flicker at segment boundaries. For completeness, we further benchmark our method against the short-segment training configuration in the supplementary.

5.1 Implementation Details & Evaluation Metrics

Please refer to the supplementary for detailed discussions on benchmarking protocols and hyperparameter settings. All our method specific hyperparameters are kept fixed across all datasets and scenes. These datasets span a wide range of motion regimes, including localized rapid motion (N3DV), mixed slow and fast motion (Interdigital), and high-speed real-world sports motion (VRU).

For evaluation, we report PSNR and SSIM [30] to measure overall rendering quality. However, since dynamic regions typically occupy small portions of the frame, global metrics may obscure improvements in motion areas. To specifically evaluate dynamic reconstruction quality, we introduce Masked PSNR (M-PSNR)

Table 1: Quantitative comparisons on the Neural 3D Video (N3DV) dataset. We report PSNR, M-PSNR, M-SSIM, LPIPS, rendering speed (FPS), Train Time (TT) in minutes, and Model Size in MB, measured on NVIDIA A4000 16GB GPU. ^(†) indicates results obtained from official pretrained models released by the respective authors. Our model significantly improves reconstruction quality in dynamic regions, as reflected by the Masked metrics (M-PSNR and M-SSIM). Notably, our method achieves the highest FPS since it avoids heavy components like MLP decoders and multi-planar voxels used by other methods.

Method	PSNR \uparrow	M-PSNR \uparrow	M-SSIM \uparrow	LPIPS \downarrow	FPS \uparrow	TT \downarrow	Size \downarrow
4DGaussian [31]	31.21	22.67	0.784	0.071	61	50	42 MB
STG [14]	31.40	22.61	0.792	0.069	93	120	62 MB
Ex4DGS [†] [11]	31.45	23.40	0.814	0.078	29	144	213 MB
SaroGS [†] [34]	32.08	23.62	0.821	0.064	39	189	310 MB
Swift4D [†] [32]	32.12	23.74	0.835	0.061	53	41	147 MB
Ours	32.42	24.68	0.863	0.059	146	62	204 MB

and Masked SSIM (M-SSIM), computed exclusively within dynamic regions identified using RAFT optical flow [28] between frames separated by 5 timesteps. Additionally, we report LPIPS [39] to assess perceptual image quality.

5.2 Results

We compare against 4DGaussian [31], Ex4DGS [11], STG [14], Swift4D [32], and SaroGS [34]. Tab. 1 and Tab. 2 report averaged results across all scenes for the N3DV and Interdigital datasets. SaroGS does not report results on Interdigital in their paper. We encountered CUDA runtime errors while attempting to run this model on Interdigital. Our method consistently outperforms all other methods across all datasets, particularly in dynamic regions. We observe that Swift4D and 4DGS perform relatively well when motion is localized to small regions, as in N3DV, whereas Ex4DGS and STG achieve better results in scenarios with slower, large-scale motion, such as the Interdigital dataset. As shown in Table 1, our training time (TT) is comparable to 4DGaussian and Swift4D, while the resulting model size remains competitive with the top three performing methods. Please see the supplementary for VRU Basketball benchmark and more details on scene-wise comparisons on all datasets.

We present qualitative results in Figures 4, 5, and 6. Notably, 4DGaussian and Swift4D exhibit noticeable loss of detail in regions with rapid or complex

Table 2: Quantitative comparison on the Interdigital dataset. We report PSNR, Masked PSNR (M-PSNR), Masked SSIM (M-SSIM), and LPIPS \downarrow . Red and Orange highlight the best and second-best results, respectively.

Method	PSNR \uparrow	M-PSNR \uparrow	M-SSIM \uparrow	LPIPS \downarrow
4DGaussian [31]	26.74	18.09	0.520	0.173
STG [14]	33.45	27.14	0.860	0.060
Ex4DGS [11]	32.44	26.15	0.844	0.070
Swift4D [32]	31.62	23.01	0.741	0.072
Ours	34.14	28.87	0.901	0.044



Fig. 4: Qualitative comparison on Neural 3D Video [12]. Scene: i) *cook spinach* ii) *sear steak* iii) *cut roasted beef*

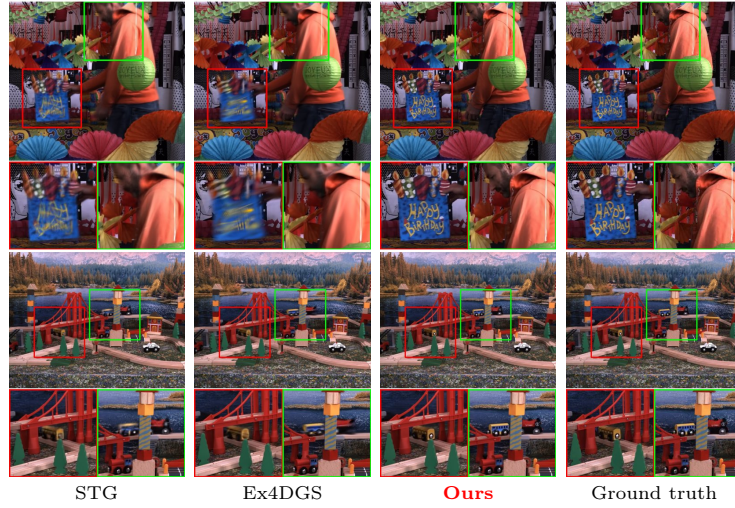


Fig. 5: Qualitative results on the Interdigital Dataset [22]. Scenes shown top to bottom: i) *Birthday*, ii) *Train*: Note that our method preserves finer details in the moving train region.



Fig. 6: Qualitative comparison on VRU Basketball [32]. Scenes top to bottom: i) *DG*, ii) *GZ*

motion. Ex4DGS and STG exhibit blurring artifacts in dynamic regions when trained on longer continuous sequence. Our videos in the supplementary indicate the superior reconstruction of dynamic regions without any flicker artifacts.

5.3 Ablations

Ablation study of the proposed components: Table 3 analyzes the contribution of our three core components: VAD, TAT, and TOW. Each component provides consistent gains across both Interdigital and N3DV datasets, with the full model achieving the highest PSNR, M-PSNR, and M-SSIM.

Generalization of VAD: Table 4 and Figure 7 demonstrate that integrating the proposed VAD consistently improves reconstruction especially in dynamic regions across multiple dynamic 3DGS baselines on both Interdigital and N3DV datasets showing its plug-and-play capability. We observe larger improvements in dynamic regions for methods such as [14] and ours that employ multiple anchors, which are more likely to have short-lived Gaussians and thus VAD becomes important. Additionally, TAT can be naturally extended to methods that model temporal opacity using Gaussian RBFs, such as [14]. TOW, however, is intuitively designed for approaches that rely on a Fourier basis for motion modeling.

Table 3: Ablation study showing the impact of VAD, TAT, and TOW on model performance across the **Interdigital** and **N3DV** datasets. “baseline” refers to training with our deformation model only.

VAD	TAT	TOW	PSNR	M-PSNR	M-SSIM	LPIPS
Interdigital						
		baseline	32.80	24.75	0.793	0.065
✓			33.59	27.32	0.869	0.049
✓	✓		33.65	27.51	0.873	0.048
✓	✓	✓	34.14	28.87	0.901	0.044
Neural 3D Video						
		baseline	31.98	21.96	0.781	0.066
✓			32.14	23.40	0.832	0.062
✓	✓		32.17	23.62	0.837	0.061
✓	✓	✓	32.42	24.68	0.863	0.059

Table 4: Ablation of the VAD module. The dynamic metrics M-PSNR and M-SSIM show substantial gains over existing dynamic 3DGS baselines.

Dataset	Method	PSNR↑	M-PSNR↑	M-SSIM↑	LPIPS↓
Interdigital	STG [14]	33.45	27.14	0.860	0.060
	+VAD	33.67	27.91	0.884	0.057
	Ex4DGS [11]	32.44	26.15	0.844	0.070
	+VAD	32.51	26.42	0.853	0.066
	Swift4D [32]	31.62	23.01	0.741	0.072
	+VAD	31.79	23.31	0.762	0.070
Neural 3D Video	STG [14]	31.40	22.61	0.792	0.069
	+VAD	31.71	24.01	0.842	0.065
	Ex4DGS [11]	31.45	23.40	0.814	0.078
	+VAD	31.62	23.79	0.830	0.076
	Swift4D [32]	32.12	23.74	0.835	0.061
	+VAD	32.21	24.04	0.846	0.061

Temporally Aware Denisfication vs. Threshold Reduction: To validate the effectiveness of our method, we compare it with a baseline that simply reduces the denisfication threshold τ_{pos} . As illustrated in Fig. 8, lower thresholds lead to a rapid increase in model size but small performance gain. Our method, on the other hand, achieves higher Masked PSNR at a better model size.

Handling Occlusion: Our method is robust to occlusions due to both its temporal initialization and temporally aware denisfication strategy. By initializing

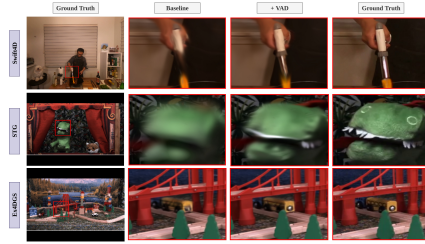


Fig. 7: Generalization of the proposed VAD module across diverse baselines.

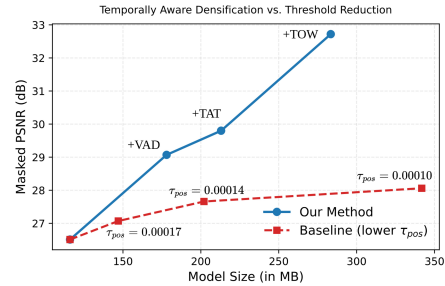


Fig. 8: Ablation on densification efficiency. (Scene: Painter)

Gaussians across multiple temporal centers, we ensure broader temporal coverage, increasing the likelihood that occluded objects are captured during initialization. Furthermore, VAD, TAT, and TOW jointly help densify Gaussians that are visible for only a few frames due to occlusion or late entry, which would otherwise receive insufficient gradients because of limited visibility. Fig 9 illustrates a scenario in which an object re-emerges after occlusion. We also present such cases in the supplementary videos, notably in the Painter (sudden object appearance) and Train scenes.

Additional ablations on the VRU dataset, choice of hyperparameters, and impact of TOW on life spans of Gaussians are provided in the supplementary material.

6 Conclusion

We revisit the densification strategy in dynamic 3DGS and identify its limitations when modeling fast complex motion. To address this, we propose a deformation formulation complemented by our three key contributions. Together, these modules significantly improve reconstruction in dynamic regions, and VAD module can generalize to other models.

Limitations and Future Work. Although effective, our method has not yet been tested on longer video sequences spanning several minutes; evaluating how well our method scales to such durations is an important next step. Additionally, our current focus is multiview capture, and exploring our approach to monocular or sparse-view settings remains a practical next step. We also plan to explore applying TAT and TOW as plug-and-play modules with minimal modifications to other baseline architectures.

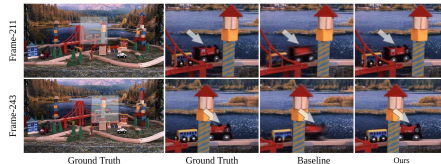


Fig. 9: Occlusion and reappearance. Train scene. Baseline uses only deformation model, while ours better preserves occluded objects upon reappearance.

Acknowledgements

This work is supported in part by a grant from Anusandhan National Research Foundation, India under grant number ANRF/ARG/2025/002993/ENS and the Kotak IISc AI ML Center.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5855–5864 (2021) [2](#)
2. Bond, A., Wang, J.H., Mai, L., Erdem, E., Erdem, A.: Gaussianvideo: Efficient video representation via hierarchical gaussian splatting. arXiv preprint arXiv:2501.04782 (2025) [6](#)
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) [2](#)
4. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) (July 2024) [2, 3](#)
5. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [2](#)
6. Grubert, G., Barthel, F., Hilsmann, A., Eisert, P.: Improving adaptive density control for 3d gaussian splatting. In: Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. p. 610–621. SCITEPRESS - Science and Technology Publications (2025). <https://doi.org/10.5220/0013308500003912>, <http://dx.doi.org/10.5220/0013308500003912> [4](#)
7. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH). Association for Computing Machinery (2024). <https://doi.org/10.1145/3641519.3657428> [3](#)
8. Huang, H., Zhang, Z., Wu, G., Wang, R.: Pgdgs: Improving few-shot 3d gaussian splatting with progressive gaussian densification. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10889028> [4](#)
9. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics (TOG) **42**(4) (July 2023) [2, 3](#)
10. Kim, S., Lee, K., Lee, Y.: Color-cued efficient densification method for 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 775–783 (June 2024) [3](#)
11. Lee, J., Won, C., Jung, H., Bae, I., Jeon, H.G.: Fully explicit dynamic gaussian splatting. In: Proceedings of the Neural Information Processing Systems (2024) [2, 3, 7, 10, 11, 13](#)

12. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3D video synthesis from multi-view video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5521–5531 (2022) [10](#), [12](#)
13. Li, Y., Pang, B., Chen, Y., Wang, G.: Anchored 4d gaussian splatting for dynamic novel view synthesis. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques - Asia (SIGGRAPH-Asia). Association for Computing Machinery, New York, NY, USA (2025). <https://doi.org/10.1145/3757377.3763898>, <https://doi.org/10.1145/3757377.3763898> [4](#)
14. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) [2](#), [3](#), [4](#), [6](#), [10](#), [11](#), [13](#)
15. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21136–21145 (2024) [2](#), [3](#), [6](#)
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) [2](#)
17. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (TOG) **41**(4), 102:1–102:15 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127> [2](#)
18. Park, H., Ryu, G., Kim, W.: Dropgaussian: Structural regularization for sparse-view gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025) [3](#)
19. Park, J., Bui, M.Q.V., Bello, J.L.G., Moon, J., Oh, J., Kim, M.: Splinesg: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 26866–26875 (June 2025) [3](#)
20. Patle, G., Girgaonkar, N., Somraj, N., Soundararajan, R.: AD-GS: Alternating Denisfication for Sparse-Input 3D Gaussian Splatting. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques - Asia (SIGGRAPH-Asia) (2025). <https://doi.org/10.1145/3757377.3763993> [4](#)
21. Rota Bulò, S., Porzi, L., Kotschieder, P.: Revising denisfication in gaussian splatting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) [4](#)
22. Sabater, N., Boisson, G., Vandame, B., Kerbiriou, P., Babon, F., Hog, M., Langlois, T., Gendrot, R., Bureller, O., Schubert, A., Allie, V.: Dataset and pipeline for multi-view light-field video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017) [10](#), [12](#)
23. Shaw, R., Nazarczuk, M., Song, J., Moreau, A., Catley-Chandar, S., Dharmo, H., Pérez-Pellitero, E.: Swings: sliding windows for dynamic 3d gaussian splatting. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 37–54. Springer (2024) [3](#)
24. Somraj, N., Karanayil, A., Soundararajan, R.: SimpleNeRF: Regularizing sparse input neural radiance fields with simpler solutions. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques - Asia (SIGGRAPH-Asia) (December 2023). <https://doi.org/10.1145/3610548.3618188> [2](#)

25. Somraj, N., Soundararajan, R.: ViP-NeRF: Visibility prior for sparse input neural radiance fields. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) (August 2023). <https://doi.org/10.1145/3588432.3591539> 2
26. Stearns, C., Harley, A.W., Uy, M., Dubost, F., Tombari, F., Wetzstein, G., Guibas, L.: Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques - Asia (SIGGRAPH-Asia). pp. 1–11 (2024) 3
27. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2
28. Teed, Z., Deng, J.Y.: Raft: Recurrent all-pairs field transforms for optical flow. Proceedings of the European Conference on Computer Vision (ECCV) (2020) 11
29. Wang, Z., Wu, G., Wang, Z., Xiao, L., Liu, R., Wu, J., Wang, R.: Hda-gs: Hierarchical density-controlled for anisotropic 3d gaussian splatting. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10887967> 4
30. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861> 10
31. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20310–20320 (June 2024) 2, 3, 10, 11
32. Wu, J., Peng, R., Wang, Z., Xiao, L., Tang, L., Yan, J., Xiong, K., Wang, R.: Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025) 3, 10, 11, 12, 13
33. Xu, Z., Xu, Y., Yu, Z., Peng, S., Sun, J., Bao, H., Zhou, X.: Representing long volumetric video with temporal gaussian hierarchy. ACM Transactions on Graphics (TOG) 43(6) (November 2024) 3
34. Yan, J., Peng, R., Tang, L., Wang, R.: 4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes. In: ACM Multimedia 2024 (2024) 2, 4, 6, 10, 11
35. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024) 2, 3, 10
36. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
37. Yuan, Y., Shen, Q., Yang, X., Wang, X.: 1000+ fps 4d gaussian splatting for dynamic scene rendering (2025), <https://arxiv.org/abs/2503.16422> 2
38. Zhang, J., Li, J., Yu, X., Huang, L., Gu, L., Zheng, J., Bai, X.: Cor-gs: Sparse-view 3d gaussian splatting via co-regularization. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) 3
39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 11

40. Zhang, Z., Hu, W., Lao, Y., He, T., Zhao, H.: Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) [3](#), [4](#)
41. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) [3](#)