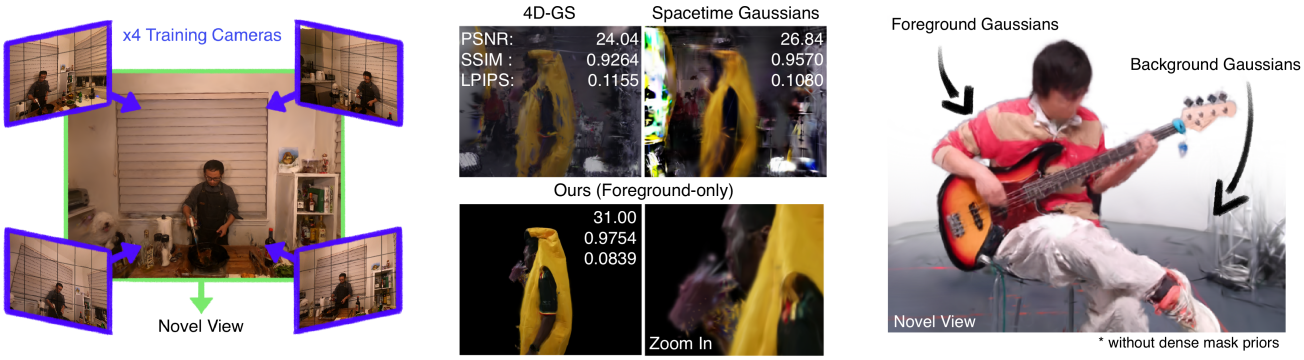
Splatography: Sparse multi-view dynamic Gaussian Splatting for filmmaking challenges

Adrian Azzarelli
Bristol Visual Institute
University of Bristol, UK
a.azzarelli@bristol.ac.uk

Nantheera Anantrasirichai
Bristol Visual Institute
University of Bristol, UK
n.anantrasirichai@bristol.ac.uk

David R Bull
Bristol Visual Institute
University of Bristol, UK
dave.bull@bristol.ac.uk



(a) Sparse-view dynamic 3-D reconstruction (b) High quality results (c) Segmentation capabilities

Figure 1. **Sparse view 3-D reconstruction:** Our dynamic representation offers foreground-background separability and high quality 3-D reconstruction without the need for dense mask priors. This paper focuses on filmmaking challenges, including but not limited to sparse view and reflective, transparent and dynamic textures

Abstract

Deformable Gaussian Splatting (GS) accomplishes photorealistic dynamic 3-D reconstruction from dense multi-view video (MVV) by learning to deform a canonical GS representation. However, in filmmaking, tight budgets can result in sparse camera configurations, which limits state-of-the-art (SotA) methods when capturing complex dynamic features. To address this issue, we introduce an approach that splits the canonical Gaussians and deformation field into foreground and background components using a sparse set of masks for frames at $t = 0$*. Each representation is separately trained on different loss functions during canonical pre-training. Then, during dynamic training, different parameters are modeled for each deformation field following common filmmaking practices. The foreground stage contains diverse dynamic features so changes in color, position and rotation are learned. While, the background containing film-crew and equipment, is typically dimmer and less dynamic so only changes in point position are learned. Experiments on 3-D and 2.5-D entertainment datasets show that our method produces SotA qualitative and quantitative results; up to* 3 *PSNR higher with half the model size on 3-D scenes. Unlike the SotA and without the need for dense mask supervision, our method also produces segmented dynamic reconstructions including transparent and dynamic textures. Code and video comparisons are available online:* http://bit.ly/4oqzZrO

1. Introduction

Gaussian Splatting (GS) is increasingly used for photorealistic dynamic 3-D reconstruction from multi-view video (MVV) data. While GS methods can unlock creative potential in filmmaking applications [3, 7, 12, 14, 18, 37], current approaches [10, 18, 25, 27, 36] rely on a large number of cameras for robust dynamic reconstruction. This can be impractical for some filmmaking productions, where the number of cameras is limited by filming budgets, or issues with acting challenges arise due to isolating filming spaces designed to minimize background noise [9, 19, 28, 32]. Hence, improving sparse view 3-D reconstruction (SV3D) for dynamic scenes is a key challenge.

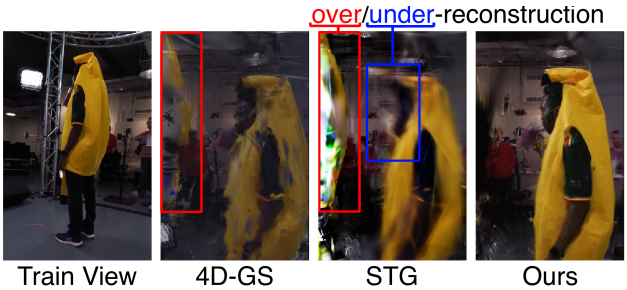


Figure 2. Novel views (right) reveal over-reconstructed backgrounds and under-reconstructed foregrounds in SotA

In sparse MVV configurations, fewer cameras typically result in densely viewed foregrounds but sparsely viewed backgrounds. This conflicts with the workings of state-of-the-art (SotA) approaches that attribute Gaussian point importance based on full-image reconstruction quality [16, 27, 36]. Thus, backgrounds receive more attention during training due to their larger area, leading to over-reconstruction in backgrounds and under-reconstruction in foregrounds; see Fig. 2. While this could be resolved by masking out backgrounds for each frame using priors generated with SAM2/MiVOS/etc. [8, 33], the issue of densely masking reflective, transparent and dynamic (RTD) textures (like fire and smoke) has not been resolved. Transparency is especially problematic as disentangling dimming and lighting material effects from a background cannot be achieved with simple binary masks. RTD textures are common in filmmaking, so dense mask priors would limit the use of costumes and props. For SV3D we also find that, even before dynamic training, SotA methods cannot reconstruct robust canonical representations. These methods work by learning to deform a canonical (time-independent) 3-D GS. Despite its importance, robust canonical pre-training has yet to be investigated for enhancing dynamic reconstruction.

Consequently, this paper's primary focus is to tackle SV3D reconstruction challenges by: (1) proposing a new dynamic scene representation that disentangles the foreground and background to deal with point-importance, and (2) developing a new strategy for training canonical representations to deal with poor initialization. Linking this work to filmmaking, we leverage foreground-background separation to apply background-specific constraints based on filmmaking practices; discussed in the following paragraphs. This aims to suppress the reconstruction artifacts associated with background and foreground reconstruction, which we identify in Fig. 2. We also place emphasis on reconstructing RTD textures as they are more common in filmmaking than most other practices.

The proposed scene representation in Fig. 3 uses only a single mask per view at $t = 0$ to segment the initial point cloud into canonical foreground and background representations. During canonical training, the representations are separately trained using specialized loss functions that rely on the aforementioned masks. The foreground loss discourages floaters and smooth-edge artifacts, and the background loss suppresses view-dependent occlusions and foreground artifacts present along the mask-edge. To model dynamics, we propose representing the foreground and background as separate hex-plane deformation fields, jointly trained without mask supervision. The assumption made is that, during filming, backgrounds are weakly dynamic as the only temporal changes are due to film crew moving around. Color changes are not expected as background lighting is usually static and dim. So, dynamic background features are captured using only point displacement. By doing this, the background is less likely to over-compensate for under-reconstructed foregrounds. Regarding foreground dynamics, we model changes in position, rotation and color via a hex-plane model as we expect the presence of RTD textures. We also modify the temporal opacity function in [27] to better represent deformable textures and introduce regularizers for each relative parameter.

Dealing with point densification, prior works rely on cloning and splitting strategies that assess volumetric importance (i.e. where to instance points) based on full image reconstruction quality [22, 27, 36]. For SV3D, sparsely viewed backgrounds dominate these metrics leading to over-reconstruction in unimportant regions of space. This also leads to temporal bias, as instances in time with greater reconstruction error may produce clones that negatively affect temporal consistency in other views. We consequently leverage the separation of foreground and background representations to limit cloning only for foreground points. To avoid view dependent and temporal bias during densification, we propose a reference-free strategy that samples point motion over time and uses quantile selection to clone the most dynamic points. As this approach is reference-free, it does not require tracking additional point parameters for densification, as in prior works.

In summary, this paper aims to resolve current issues when reconstructing dynamic multi-view SV3D content, including RTD textures, via the following contributions:

1. A novel approach to disentangling and separately training canonical foreground and background representations using a sparse set of mask; including the proposal of representation-specific loss functions
2. A new formulation for dynamic representations that treat foregrounds and backgrounds differently by limiting the expressiveness of background features. By also limiting the number of background Gaussians being modeled, this encourages the model to focus more on the foreground representation
3. A reference-free and minimally invasive approach to foreground-only point densification. This mitigates is-

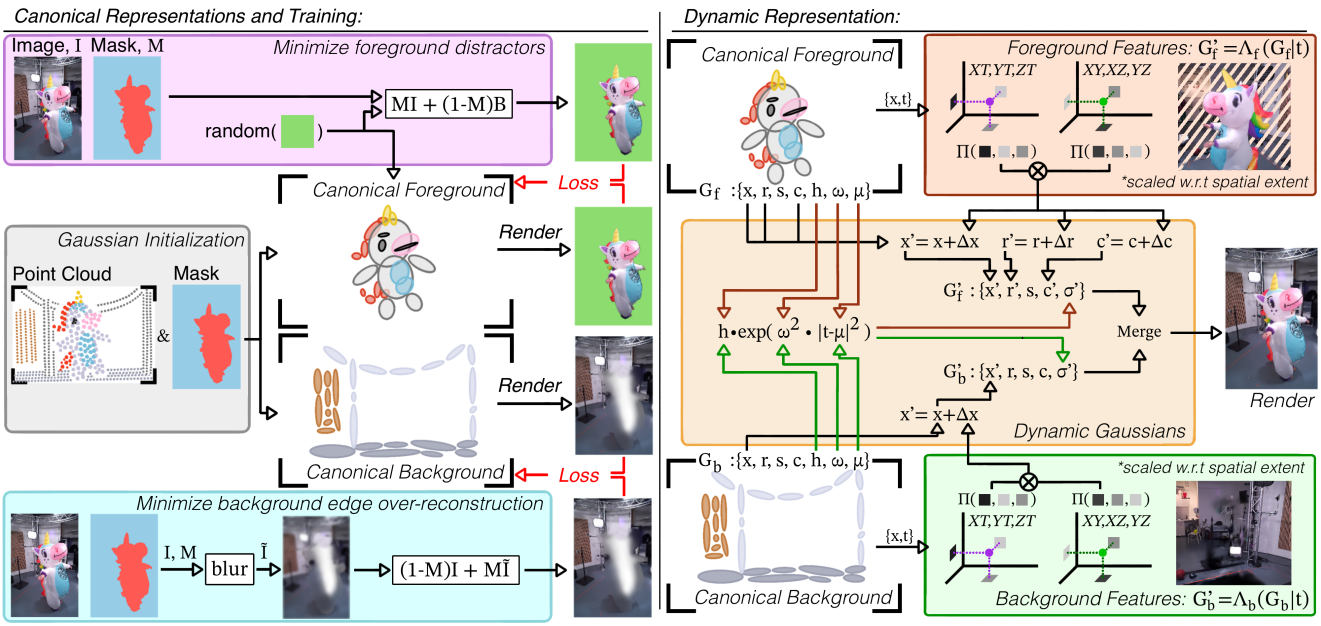


Figure 3. **Left:** The canonical representation is constructed by masking the initial point cloud and training the foreground and background representations G_f and G_b on specialized loss functions that minimize over-reconstruction. **Right:** Dynamic features for G_f and G_b are jointly trained using the proposed plane-based design. For G_b, we only learn motion. For G_f, we learn motion, rotation and color change using a novel combination of plane features, and also temporal opacity using an exponential peaking function.

sues with current approaches, that utilize full-reference reconstruction quality to mediate point densification

4. As a by-product of the foreground-background separation, our approach is also capable of foreground segmentation, including RTD textures, with minimal supervision

To our knowledge, this approach is the first to separate and condition a scene based on foreground and background observations. This paper is also the first to investigate dynamic SV3D for scene-based reconstructions and the first to investigate improvements for robust canonical initialization.

2. Related Work

In this section we discuss related work on dynamic SV3D. A review of static SV3D research is available in the appendix, however as static methods do not account for RTD textures most solutions are not applicable to the dynamic SV3D paradigm.

General Discussion: Research on dynamic 3-D reconstruction began with D-NeRF [31], an implicit deformable neural radiance field (NeRF). However, it was soon found that key-frame base solutions produced better results [1, 26, 30, 34, 35]. DyNeRF [26] proposed a training strategy for key-frame NeRFs that relied on attributing higher loss (via an importance weight) to pixels with larger motion vectors. HyperReel [1] proposed a hierarchical interpolation strategy for a set of tri-plane key-frame representations [6]. Similarly, HexPlanes [5] and K-Planes [11] proposed modeling key-frames as 2-D spatiotemporal feature grids, mapping $(x,t),(y,t),(z,t) \rightarrow f_{XT}, f_{YT}, f_{ZT}$ feature space. Combining this with a tri-plane representation, $(x,y),(x,z),(y,z) \rightarrow f_{XY}, f_{XZ}, f_{YZ}$, produces a hex-plane feature representation.

More recently, research on dynamic GS has become prevalent [15, 16, 18, 27, 36, 39]. Since, we have noticed a divide between fully explicit and implicit dynamic representations. Explicit models [13, 18, 27, 39] learn to deform Gaussian points by, for example, using polynomials to model motion [27]. Polynomial functions offer the opportunity to apply parameter-specific constraints, but require more computation than implicit solutions and are degenerate for longer videos. Instead, implicit representations like hex-planes [2, 17, 36, 38] learn features that have a one-to-many relationship w.r.t GS point parameters. They are compact and do not degenerate w.r.t video duration but are harder to precisely constrain dynamic components.

Filmmaking Applications: Research on 3-D human-centric/avatar reconstruction has tended to focus on cases where dense mask priors are attainable [19, 21, 28, 29, 32]. However, more recently there has been increased interest in applying these methods to sparse view configurations [20]. Since these methods rely on dense mask priors in order

to achieve high quality foreground reconstructions, significant problems remain, particularly when applying them to scenes that capture dynamic and transparent textures.

For scene-based reconstruction, [4] tested 4D-GS [36], STG [27] and SCGS [18] on several entertainment scenes using a multi-view SV3D set-up. 4D-GS achieves reconstruction using the 3D-GS [22] model with spatiotemporal changes driven by a K-Planes feature representations. It uses the Gaussian cloning and splitting strategy, proposed in [22]. STG introduces a representation and rendering method driven by a temporal event-based parameter. The parameter is involved in polynomial point displacement, temporal rotation functions, and in an exponential change in opacity function. The proposed densification strategy optimizes the number of Gaussians in sparsely viewed spatial regions. SC-GS is a hybrid approach for mesh-like Gaussians, where motion is driven by a small set of local control points and the linked Gaussians are trained as surface-like residuals using a rigidity loss to maximize uniform local motions.

These methods place equal importance across the whole scene. Hence, when the spatial extent of the background is much greater than the foreground, the background receives more attention during training and densification. As backgrounds in SV3D are non-salient across view, this leads to under-reconstructed foreground and over-reconstructed background content. This analysis is supported by Fig. 2 and highlights the need to investigate the dynamic SV3D paradigms.

3. Preliminaries

In Sec. 3.1 we present background on 3-D GS rendering and in Sec. 3.2 we present background hex-plane deformation fields, which we use to model the various dynamic foreground and background parameters.

3.1. Canonical 3-D Gaussians

3-D Gaussians [22] are characterized as $G = \{x, s, r, c, \sigma\}$, representing point position $x \in \mathcal{R}^3$, covariance constructed from $\Sigma = RSS^TR^T$, where R is the 3×3 rotation transform represented by $r \in \mathcal{R}^4$ in quaternion form and S is the scaling transform represented as $s \in \mathcal{R}^3$, color $c \in \mathcal{R}^3$ and opacity $\sigma \in \mathcal{R}$. The set G is differentiably rendered as a 2-D image via splatting [41]. This involves: (1) sorting G w.r.t point distance from the camera, (2) projecting Σ into image space using $\Sigma' = JW\Sigma W^TJ^T$, where W is the viewing transform, J is a Jacobian of the affine approximation of the projective transform and Σ' is the 2-D covariance of Gaussian on the image plane, and (3) alpha blending 2-D splats using the formula for pixel color:

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i), \tag{1}$$

where i is the index of each splat's color and alpha w.r.t the aforementioned sorting and α is the influence of a splat on a pixel with displacement $\mathcal{X}$ from the 2-D splat's center, given by the Mahalanobis distance function:

$$\alpha = \sigma \cdot \exp(-\frac{1}{2}\mathcal{X}^T\Sigma^{-1}\mathcal{X}). \tag{2}$$

Ultimately, we choose MipSplatting [40] as our canonical representation as it resolves 3D-GS anti-aliasing due to non-salient object resolutions across views. Knowledge of MipSplatting is not required to understand the main paper so details are provided in the appendix.

3.2. Hex-Plane Deformation Fields

Hex-Planes [2, 5, 11, 18, 36] model the change in each Gaussian parameter over time, t. The point position, x, is concatenated with the time t, forming $q = \{x, y, z, t\}$. The normalized 2-D coordinates q_ψ are projected onto a grid-plane $P_\psi \in \mathcal{R}^{J \times K \times L}$ for $\psi \in [XY, XZ, YZ, XT, YT, ZT]$ and bilinearly sample a feature $f_\psi = \theta(q_\psi)$, where $J \times K$ is the pixel resolution of the grid-planes, L is the feature size, and θ represents the projection and sampling operations. Per point, all features are multiplied forming $f(q) = \prod_{i \in \psi} f_i(q)$. The final feature is decoded via a shallow MLP per Gaussian component. This entire process is defined as $\Lambda(G|t)$ leading to $G' = G + \Lambda(G|t)$, where G' is the deformed representation at time-step t.

4. Methodology

Our approach to tackling dynamic SV3D for MVV content is summarized in Fig. 3. Sec. 4.1 presents our canonical representation and optimization strategy, responsible for decoupling and separately training background and foreground Gaussians. Sec. 4.2 presents our formulation of the foreground and background deformation fields, and optimization. In Sec. 4.3 we present the temporal opacity function and regularization approach, and in Sec. 4.4 we propose our reference-free densification strategy, which focuses on reducing under-reconstruction artifacts in the foreground. Further details are available in the appendix.

4.1. Canonical Foreground and Background Gaussians

Representation. Accurate initialization is critical for dynamic reconstruction and works to optimize the quality of the canonical representation before learning dynamics. Prior work [2, 11, 27, 36] has tended to over-reconstruct backgrounds and under-reconstruct foregrounds during this stage of SV3D reconstruction. To address this imbalance, we first separate the initial point cloud into foreground and background GS representations, G_f and G_b respectively,

using a sparse set of masks, $M^* \in \mathcal{R}^{H\times W}$ for MVV frames at $t = 0$. This can be accomplished by projecting the initial point cloud onto the 2-D plane of each mask and assigning points that appear in all masks to G_f.

Considering that our approach to densification is fully explicit (see in Sec. 4.4), we can estimate the final number of points with reasonable accuracy. This facilitates mitigation of over and under-reconstruction errors that arise from G_f and G_b being too sparsely or densely populated at the beginning of training. This is done by applying voxel up/down sampling to the initial G_f and G_b representations. In all of our experiments (on 1080p and 2K resolution images) we found $\sim 50,000$ points to be sufficient.

Optimization. Prior works that rely on hex-planes to model deformations [2, 18, 36] train G during the canonical training stage using all the ground truth images, $I^* \in \mathcal{R}^{H\times W\times 3}$ for $t \geq 0$. This means that G in not interpretable by the renderer without using $G' = \Lambda(G|t)$ to update G. This makes developing loss functions for canonical training challenging. Instead, during the this stage we anchor G_f and G_b to $t = 0$ by only using ground truth data at $t = 0$. This means G_f and G_b can be rendered as standalone components. This allows us to develop the following mask-based loss functions to deal with the challenges discussed in Sec. 1.

To minimize foreground floaters and smooth-edge artifacts, we propose Eq. (3) for training G_f, where the first component alpha blends a random background color, $B \in \mathcal{R}^3$, with the masked ground truth image and the second component alpha blends B with the image rendered from G_f. Here, $\alpha_f \in \mathcal{R}^{H\times W}$ is the alpha of G_f. B is randomly sampled every iteration.

$$\mathcal{L}_{\text{f}} = ||(M^* I^* + (1 - M^*)B) - (\alpha_f I_f + (1 - \alpha_f)B)|| \quad (3)$$

To avoid over-fitting background colors that fill in the gaps left by under-reconstructed foreground volumes, we optimize the similarity between the edge Gaussians and surrounding Gaussians using Eq. (4). As in Fig. 3, this bleeds the edges of the background into the masked region using $\tilde{I}^b = \text{blur}((1 - M^*)I^*)$ and alpha blends the blurred and original image to preserve the background detail.

$$\mathcal{L}_{\text{b}} = ||((1 - M^*)I^* + M^* \tilde{I}^b) - I^b|| \quad (4)$$

4.2. Foreground and Background Deformation Fields

Representation. 4D-GS [36] introduces the hex-plane deformation model $G' = \Lambda(G|t)$ to approximate the linear deformations $(\Delta x, \Delta r, \Delta s, \Delta c, \Delta\sigma)$ for all components in G w.r.t time. This is also applicable to G_f and G_b. Though, as G_f and G_b contribute differently to the final render, we choose to formulate G'_f and G'_b differently. The foreground model $G'_f = \Lambda_f(G_f|t)$ is hex-plane representation that approximates $(\Delta x, \Delta r, \Delta c)\forall \in G_f$. The background model $G'_b = \Lambda_b(G_b|t)$, is a different hex-plane representation that approximates $(\Delta x)\forall \in G_b$. $\Lambda_f(\cdot)$ and $\Lambda_b(\cdot)$ are scaled w.r.t the spatial extents of the initial G_f and G_b point clouds, respectively.

The expressivity of Λ_b is limited to Δx following assumptions made regarding common filmmaking practices. The assumptions are that the area around the performance stage (the background) is dimly lit, and the film crew limit movement to avoid distracting the performer. As dim background lighting reduces the likelihood of dynamic texture effects, learning $(\Delta r, \Delta c)$ for G'_b imposes an unnecessary challenge for reconstruction. Conversely, foregrounds are assumed to contain a broad range of texture effects, so they learn the full set of dynamic parameters, except Δs following [27].

Optimization. We train dynamic content by merging (G'_b, G'_f) and rendering the image I_t for timestep $0 \leq t$ and applying the panoptic loss,

$$\mathcal{L} = ||I^*_t - I_t|| \quad \text{for} \quad 0 \leq t \leq 1 \quad (5)$$

4.3. Dynamic Textures via Opacity Constraints

Representation. Event-based opacity [27] is effective at reconstructing RTD textures. This models $\sigma(t)$ as a normal distribution, where the half-life, mean temporal position and amplitude are learnable components of G. Thus, dynamic textures like fire have a short life and solid materials have long life. To better represent these opposing features, we modify the original function by replacing ω from [27] with ω^2 to bias learning either instantaneous or stationary $0 \leq \sigma(t) \leq 1$.

$$\sigma(t) = h \cdot \exp(-\omega^2 |t - \mu|^2), \quad (6)$$

where h is the peak opacity, ω is the bandwidth and μ is the temporal center. This evolves G_f and G_b to include $\{x, r, s, h, \omega, \mu, c\}$.

Optimization. For training we propose regularizing h and ω using

$$\mathcal{L}_{h,\omega} = \lambda_h |1 - h| + \lambda_\omega |\omega|, \quad (7)$$

where $\lambda_h = 0.1$ and $\lambda_\omega = 1.$. This encourages points to be dense and temporally consistent.

4.4. Densification

Prior works on adaptive density for dynamic GS [22, 36] track parameters that indicate point importance based on full-reference reconstruction quality. In SV3D, sparsely

Method	PSNR		SSIM		LPIPS-Alex		LPIPS-VGG		Size
	Full	Mask	Full	Mask	Full	Mask	Full	Mask	(MB)
3-D ViVo dataset [4]: Bassist, Pianist, Weights, Fruit, Pony									
4D-GS [36]	14.22	21.22	0.6727	0.9328	0.3832	0.1160	0.4281	0.1449	134
STG [27]	13.83	21.72	0.6759	0.9398	0.4005	0.1189	0.4276	0.1431	280
SC-GS [18]	13.81	20.57	0.4244	0.8536	0.5047	0.1727	0.5320	0.1906	320
Ours	16.05	24.80	0.7503	0.9537	0.3245	0.0792	0.3839	0.1270	60
2.5-D DyNeRF dataset [26]: Flame steak, Flame salmon, Cook spinach									
STG	20.81	21.71	0.8456	0.9851	0.1523	0.0127	0.2499	0.0157	34
ITGS [16]	21.95	24.93	0.8312	0.9867	0.1908	0.0114	0.2824	0.0154	119
4D-GS	24.51	26.45	0.8548	0.9871	0.1308	0.0118	0.2265	0.0155	37
W4D-GS [2]	24.32	26.56	0.8552	0.9873	0.1319	0.0117	0.2294	0.0157	46
Ours	24.41	26.28	0.8606	0.9872	0.1444	0.0110	0.2493	0.0149	47

Table 1. Average results from the 2.5-D and 3-D entertainment datasets. The selected scenes are named alongside each dataset. *Full/Mask* evaluates full and foreground-masked test videos.

Figure 4. **Full and Zoom Temporal Comparison:** The zoom results show that our method is the only one capable of capturing the visual dynamics of the semi-transparent key-chain. Using the ViVo-Bassist scene [4]

viewed background dominate the full-image metrics leading to higher point instancing in less important background regions, leaving the foreground under-reconstructed. To handle over-reconstructed backgrounds, we initialize G_b with sufficient points to avoid requiring densification during training; see Sec. 4.1. In contrast, under-reconstruction in foregrounds predominantly occurs in regions exhibiting larger point displacement with few initial points. To address this, we densify points with higher temporal displacement statistics. To identify these points, we linear sample $x' \in G'_f(x,t)$ across 10 time-steps, from $t = 0$ to $t = 1$. We then find the displacement of each point position x'_t, w.r.t to the mean position over time, $\tilde{x}'$, using the Euclidean norm $X'_t = ||x'_t - \tilde{x}'||_2$. The average change in displacement is then computed with $\tilde{X}' = \frac{1}{10}\sum_{t=0}^{10} X'_t$, and thresholded via quantile selection, where the top 10% largest $\tilde{X}'$ value are selected for point duplication.

An estimate of the final foreground point count can be made with $N_{final} = ((2^{n_c} \times N_{start}) \times 1.1)^{n_d}$, where n_c and n_d are the number of times that we densify during canonical and dynamic training stages, respectively. This provides guidance when initializing G_f and G_b, as in Sec. 4.1 though it could be helpful for work that focuses on scene compression [23, 24].

5. Experiments

This section evaluates performance on two entertainment datasets, [4, 26]. All tests used an Nvidia RTX 3090 (24GB of VRAM). In the appendix and online, we provide more frame-by-frame and video comparisons.

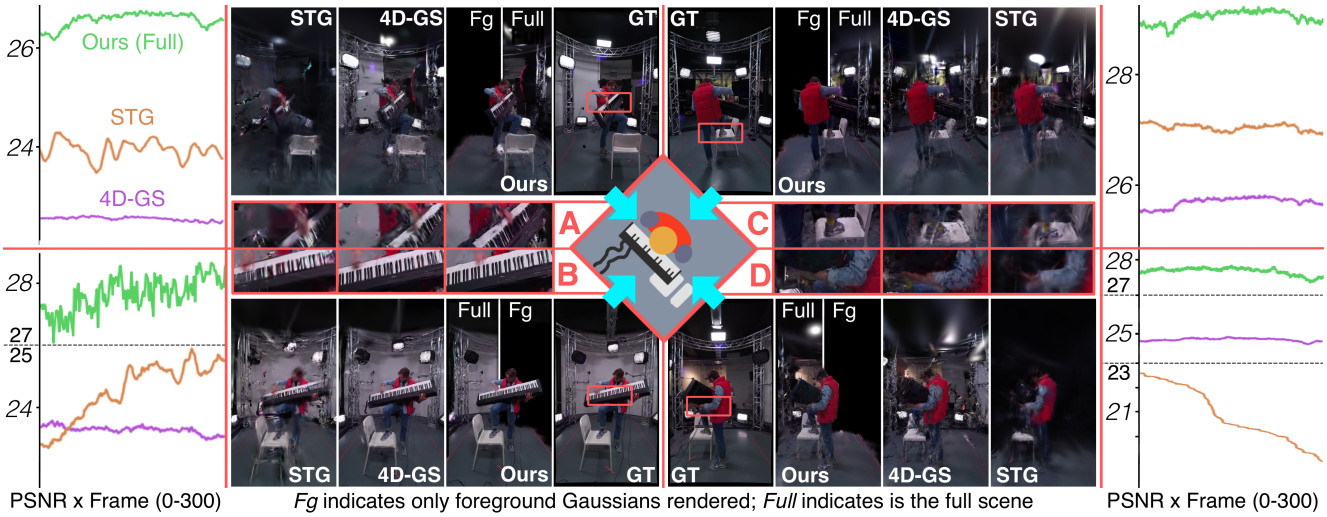


Figure 5. **Per-Frame and Per-View PSNR Plot:** The surrounding plots show the PSNR result and objectively demonstrate our approach is consistently performant. **Full and Zoom Frame Comparison:** Our-Foreground (labeled *Ours*) reconstructs keyboard, arms and feet with more visual appeal. Using the ViVo-Pianist scene [4]

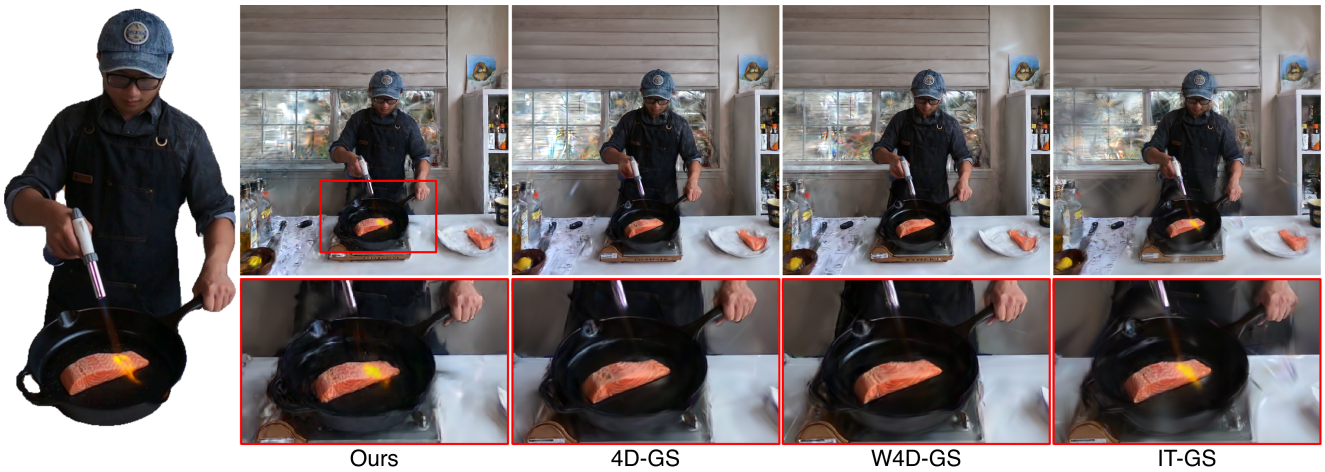


Figure 6. **Frame-by-frame test view** on the sparse DyNeRF dataset [26]; 4 training cameras at the extremities were used instead of 20. The masked ground truth is on the left

5.1. Real 3-D Cinematographic Scenes

We select five scenes from the ViVo dataset [4]. The sequences are trained with 10 cameras for 300 frames with 2K resolution. There are four test cameras placed in-front, behind, left and right of the performance stage. This serves to assess the quality on sparse views that *do not* share background features.

Visual and metric results are provided in Figs. 4 and 5 and Tab. 1. Comparatively, the temporal quality in our method is higher and smoother, as shown by the consistency of the per-frame and per-camera PSNR results in Fig. 5. Our method is also capable of producing clean segmentations with only a sparse set of masks, and more accurately captures fine details such as the semi-transparent key-chain in Fig. 4, which none of the other benchmarks accomplishes.

5.2. Real 2.5-D Cooking Scenes

We select the three cooking scenes from the DyNeRF dataset that focus on RTD textures. This is an SV2.5D data set and is representative of talk-show productions where all cameras point in the same direction. For training, we select the four most distant cameras from the test camera, for 50 frames at 1080p resolution. This experiment serves to assess the quality on sparse views that *do* share background features.

Tab. 1 and Fig. 6 show the metric and visual results. While 4D-GS and W4D-GS produce competitive results,

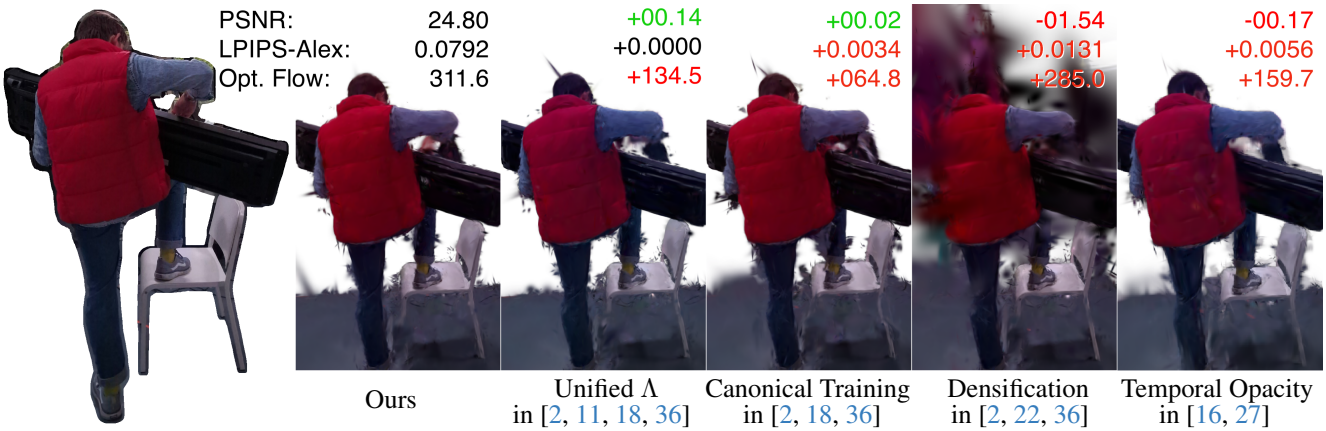


Figure 7. **Visual and average metric results of Ablation.** PSNR and LPIPS-Alex are masked evaluations of the ground truth foreground (left). *Opt. flow* is a MSE optical flow metric (see appendix)

they fail to reconstruct the fire texture in Fig. 6. Comparatively, our approach performs well on both visual and metric tests and is capable of reconstructing RTD textures for all scenes. Several scenes also produce high-quality segmentations, demonstrating that our method is capable of producing segmented representations on a variety of camera configurations.

5.3. Ablations

Results of our ablation study are shown in Fig. 7. By unifying Λ_f and Λ_b such that $G' \rightarrow \Lambda(\{G_f, G_b\}|t)$ as in [2, 11, 18, 36], the quantitative results indicate little difference but the qualitative and optical flow results reveal weaknesses reconstructing high-frequency temporal details such as hands. Our canonical training strategy is compared to the strategy used in [2, 18, 36], where G_f and G_b are jointly trained with $\mathcal{L} = ||I_t^* - I_t||$ for $0 \leq t \leq 0$. Our method produces better qualitative results, including cleaner masks. Then, using the original densification strategy as in [2, 22, 36] is shown to be un-suitable for SV3D problems. Finally, we compare the original temporal opacity function in [16, 27], where $\sigma = h \cdot \exp(\omega|t - \mu|^2)$. The results show our method produces better qualitative and quantitative results, including better high-frequency temporal details, like the hands.

6. Limitations

As with dense-view dynamic GS research [2, 18, 27, 36], issues exist with disentangling view-dependent color changes and point motions in SV3D settings. This occurs due to fast and large motions, as seen by the hand and hand-shadow in Fig. 7 (Ours), where the shadow blends in with the background; over-reconstructing color and under-reconstructing motion. Following work on human-centric reconstruction [20], future work could investigate depth-based learning using real/pseudo depth to improve geometric observations. Depth sensors/data are not uncommon in film-making.

7. Conclusion

This paper present a new dynamic gaussian splatting pipeline for sparse view 3-D reconstruction from multi-view video, specifically targeting cinematographic applications. This starts by separating the densely viewed foreground and sparsely viewed backgrounds using a sparse set of masks for each frame at $t = 0$. Each representation is then separately trained on different loss functions that utilize the sparse masks to minimize over-reconstruction in foreground and backgrounds. The deformation field is also split into foreground and background components and reformulated to reflect common filmmaking practices; backgrounds are assumed to be less dynamic and dimmer so only point displacements are modeled for these Gaussians. We further propose a new reference-free Gaussian densification approach, and also modify the temporal opacity function to improve opaque or transparent texture reconstruction. To evaluate the efficacy of our approach for cinematographic applications, we benchmark our method on sparse view 3-D and 2.5-D datasets that simulate cinematographic content. Not only does our method produce SotA quantitative and qualitative results, especially outperforming the next best method by > 3 PSNR with half the model size on 3-D scenes, the separation of foreground and background also provides the ability to segment important foreground features - ready for post-production.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil

Kim. Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16610–16620, 2023. 3

[2] Adrian Azzarelli, Nantheera Anantrasirichai, and David R Bull. Waveplanes: A compact wavelet representation for dynamic neural radiance fields. *arXiv preprint arXiv:2312.02218*, 2023. 3, 4, 5, 6, 8

[3] Adrian Azzarelli, Nantheera Anantrasirichai, and David R Bull. Intelligent cinematography: a review of ai research for cinematographic production. *Artificial Intelligence Review*, 58(4):108, 2025. 1

[4] Adrian Azzarelli, Ge Gao, Ho Man Kwan, Fan Zhang, Nantheera Anantrasirichai, Ollie Moolan-Feroze, and David Bull. Vivo: A dataset for volumetric videoreconstruction and compression. *arXiv preprint arXiv:2506.00558*, 2025. 4, 6, 7

[5] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 3, 4

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022. 3

[7] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21476–21485, 2024. 1

[8] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021. 2

[9] Wei Cheng, Ruixiang Chen, Siming Fan, Wanqi Yin, Keyu Chen, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, et al. Dna-rendering: A diverse neural actor repository for high-fidelity human-centric rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19982–19993, 2023. 1

[10] 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*, pages 1–11, 2024. 1

[11] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 3, 4, 8

[12] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024. 1

[13] 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*, 2024. 3

[14] Yijia Guo, Yuanxi Bai, Liwen Hu, Ziyi Guo, Mianzhi Liu, Yu Cai, Tiejun Huang, and Lei Ma. Prtgs: Precomputed radiance transfer of gaussian splats for real-time high-quality relighting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5112–5120, 2024. 1

[15] Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 3

[16] Bingbing Hu, Yanyan Li, Rui Xie, Bo Xu, Haoye Dong, Junfeng Yao, and Gim Hee Lee. Learnable infinite taylor gaussian for dynamic view rendering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26844–26854, 2025. 2, 3, 6, 8

[17] He Huang, Qi Yang, Mufan Liu, Yiling Xu, and Zhu Li. Adcgs: Anchor-driven deformable and compressed gaussian splatting for dynamic scene reconstruction. *arXiv preprint arXiv:2505.08196*, 2025. 3

[18] 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*, pages 4220–4230, 2024. 1, 3, 4, 5, 6, 8

[19] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 1, 3

[20] Yudong Jin, Sida Peng, Xuan Wang, Tao Xie, Zhen Xu, Yifan Yang, Yujun Shen, Hujun Bao, and Xiaowei Zhou. Diffuman4d: 4d consistent human view synthesis from sparse-view videos with spatio-temporal diffusion models. *arXiv preprint arXiv:2507.13344*, 2025. 3, 8

[21] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars. *arXiv preprint arXiv:2312.15059*, 2023. 3

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2, 4, 5, 8

[23] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. Hinerv: Video compression with hierarchical encoding-based neural representation. *Advances in Neural Information Processing Systems*, 36:72692–72704, 2023. 6

[24] Ho Man Kwan, Fan Zhang, Andrew Gower, and David Bull. Immersive video compression using implicit neural representations. *arXiv preprint arXiv:2402.01596*, 2024. 6

[25] Deqi Li, Shi-Sheng Huang, Zhiyuan Lu, Xinran Duan, and Hua Huang. St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1

[26] 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*, pages 5521–5531, 2022. 3, 6, 7

[27] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. 1, 2, 3, 4, 5, 6, 8

[28] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19711–19722, 2024. 1, 3

[29] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1165–1175, 2024. 3

[30] Sungheon Park, Minjung Son, Seokhwan Jang, Young Chun Ahn, Ji-Yeon Kim, and Nahyup Kang. Temporal interpolation is all you need for dynamic neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4212–4221, 2023. 3

[31] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 3

[32] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 1, 3

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[34] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 3

[35] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 76–87, 2023. 3

[36] 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*, pages 20310–20320, 2024. 1, 2, 3, 4, 5, 6, 8

[37] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision*, pages 55–71. Springer, 2024. 1

[38] Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. ! In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7871–7880, 2024. 3

[39] 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. 3

[40] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024. 4

[41] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 4