Graphical X Splatting (GraphiXS): A Graphical Model for 4D Gaussian Splatting under Uncertainty

DOĞA YILMAZ, University College London, United Kingdom
JIALIN ZHU, Baidu Inc., China
DESHAN GONG, The University of Hong Kong, China
HE WANG*, University College London, United Kingdom

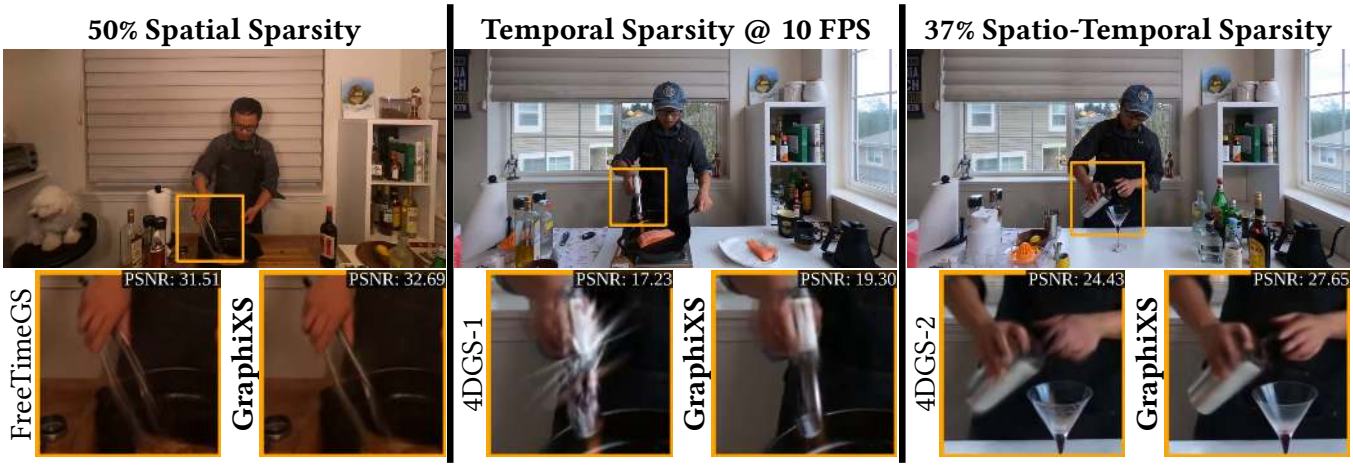

Fig. 1. GraphiXS outperforms existing 4DGS methods under various types of data uncertainty. Left: 50% cameras missing; Middle: 10 FPS low-speed cameras; Right: 37% random frames missing.

We propose a new framework to systematically incorporate data uncertainty in Gaussian Splatting. Being the new paradigm of neural rendering, Gaussian Splatting has been investigated in many applications, with the main effort in extending its representation, improving its optimization process, and accelerating its speed. However, one orthogonal, much needed, but under-explored area is data uncertainty. In standard 4D Gaussian Splatting, data uncertainty can manifest as view sparsity, missing frames, camera asynchronization, *etc*. So far, there has been little research to holistically incorporating various types of data uncertainty under a single framework. To this end, we propose Graphical X Splatting, or *GraphiXS*, a new probabilistic framework that considers multiple types of data uncertainty, aiming for a fundamental augmentation of the current 4D Gaussian Splatting paradigm into a probabilistic setting. GraphiXS is general and can be instantiated with a range of primitives, *e.g*. Gaussians, Student's-t. Furthermore, GraphiXS can be used to 'upgrade' existing methods to accommodate data uncertainty. Through exhaustive evaluation and comparison, we demonstrate that GraphiXS can systematically model various uncertainties in data, outperform existing methods in many settings where data are missing or polluted in space and time, and therefore is a major generalization of the current 4D Gaussian Splatting research.

CCS Concepts: • **Computing methodologies** → **Machine learning approaches**; **Rendering**.

ACM Reference Format:
Doğa Yılmaz, Jialin Zhu, Deshan Gong, and He Wang. 2026. Graphical X Splatting (GraphiXS): A Graphical Model for 4D Gaussian Splatting under Uncertainty. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA*. ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3799902.3811085

1 Introduction

As the latest paradigm of 3D reconstruction and neural rendering, Gaussian Splatting (GS) has served as a fundamental component of many systems [Xiang et al. 2025; Zhou et al. 2024]. At the high level, the current research effort can be broadly categorized as extending the representation [Hamdi et al. 2024; Liu et al. 2025b; Zhu et al. 2025], designing new optimization strategies [Kheradmand et al. 2024; Kim et al. 2025; Zhu et al. 2025], speeding inference processes or/and scaling up the model [Feng et al. 2025; Kerbl et al. 2024; Mallick et al. 2024]. The three lines of research have spawned new applications in novel view synthesis [Xie et al. 2024; Yang et al. 2025], SLAM [Matsuki et al. 2024], geometric reconstruction [Dai et al. 2024; Huang et al. 2024], *etc*.

*Corresponding author.

Authors' Contact Information: Doğa Yılmaz, University College London, London, United Kingdom, doga.yilmaz@ucl.ac.uk; Jialin Zhu, Baidu Inc., Beijing, China, misaliet@outlook.com; Deshan Gong, The University of Hong Kong, Hong Kong, China, deshan@hku.hk; He Wang, University College London, London, United Kingdom, he_wang@ucl.ac.uk.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811085

One under-explored theme that is orthogonal to all the aforementioned research is data uncertainty, which universally exists in real-world applications. Taking multi-view 4DGS as an example, most existing research focuses on learning the dynamics of Gaussian components, with the aim of high reconstruction quality [Wu et al. 2024; Yang et al. 2024b], generalization on unseen motions [Li et al. 2024; Zhu et al. 2024], large complex 3D scenes [Xie et al. 2025], *etc*. However, all of them explicitly or implicitly assume that sufficient data can be obtained with high quality. In practice, this often means enough cameras with views covering all angles, and good camera calibration and synchronization. We argue that such assumptions can be too restrictive as the data collection setup can be constrained by many factors such as security, safety, operational constraints, *e.g*. cameras at traffic junctions might only cover a few angles.

Recently, some methods, including contemporaneous ones, have started to consider certain data uncertainties in GS. This includes probabilistic inference for online learning [Guo et al. 2025; Savant et al. 2024; Van de Maele et al. 2024], dynamically learning moving and static Gaussians [Deng et al. 2025; Gao et al. 2024; Wang et al. 2025a], reconstruction from limited camera views [Jeong et al. 2024; Jiang et al. 2023; Yılmaz and Kıraç 2023]. However, the data uncertainty explicitly or implicitly considered is mostly specific to one application scenario. Therefore, it is desirable to have one unified framework that can incorporate multiple types of data uncertainty.

We propose Graphical X Splatting (GraphiXS), a new framework which can explicitly incorporate data uncertainty in the form of missing or asynchronized observations in 4DGS. The 'X' in GraphiXS is not necessarily Gaussian so we use the word 'component'. The term 'data uncertainty' broadly refers to missing camera views, sparse camera configurations (*i.e*. position and orientations), missing frames from cameras, imperfectly synchronized cameras, *etc*. Probabilistic learning is a natural solution to data uncertainty, but the greatest challenge is to design a flexible probabilistic framework that can model different types of uncertainty in 4DGS. To this end, we formulate GraphiXS as a generative process and propose a new graphical model, by introducing stochasticity into the individual steps of 4DGS. This includes treating all the learnable parameters (*e.g*. component location) as latent variables which are to be inferred via Maximum a Posteriori (MAP). Also, unlike existing methods which treat only the images as observations, we also treat the camera pose and frame time as samples of random variables, which enables us to incorporate the uncertainty in them. Finally, the flexibility of GraphiXS allows us to impose various prior distributions to regulate the behaviors of the components.

We instantiate GraphiXS with different components including Gaussians and Student's-t to show its generality. Through exhaustive evaluation under various combinations of data uncertainty, we demonstrate that GraphiXS can outperform existing methods across different scenes and metrics. More broadly, we demonstrate that GraphiXS is not a specific model but a framework which can be used to upgrade and improve existing methods. To the best of our knowledge, this is the first probabilistic 4DGS framework targeting multiple types of data uncertainty. Our contributions include:

- A new probabilistic framework to holistically incorporate data uncertainty in the form of sparse spatial and temporal sampling in 4DGS.
- A new graphical model which can be instantiated with different primitives and used to 'upgrade' existing 4DGS methods.
- A new way of introducing stochasticity in the steps of 4DGS.
- New priors that can effectively regulate the model behaviors, leading to effective optimization.

An open-source implementation of our method is available at https://github.com/realcrane/Graphical-X-Splatting.

2 Related Work

Traditional Methods. Multi-View Stereo (MVS) [Seitz et al. 2006] is well studied before deep learning. Software such as COLMAP [Schönberger et al. 2016] has been widely utilized in 3D reconstruction. It reconstructs the geometrically consistent points between images and restores camera poses by calculating corresponding features in images from multiple perspectives. Denser and more precise geometries can be obtained using the Structure from Motion (SFM) [Ullman 1979]. Meanwhile, additional data (*e.g*. depth) captured from Time of Flight (ToF) or Light Detection And Ranging (Lidar) sensors can be accessed from some other methods such as SLAM [Bailey and Durrant-Whyte 2006; Durrant-Whyte and Bailey 2006] and KinectFusion [Newcombe et al. 2011] for 3D reconstruction.

Learning-based Reconstruction Method. For a single scene, there are two main cornerstone methods in this sub-area: Neural Radiance Field (NeRF) [Mildenhall et al. 2021] and 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023]. NeRF utilizes a neural network to implicitly learn the 3D radiance field. It can achieve novel view synthesis by querying different points' color and density values from the neural network. NeRF accomplishes good reconstruction results, but its rendering efficiency is low, making it unsuitable for real-time rendering tasks. Comparatively, 3DGS can achieve real-time rendering, but requires more memory compared with NeRF. 3DGS uses 3D Gaussians as components in space. To optimize their attributes (means, covariances, colors, *etc*.), it uses a rasterization method called Splatting [Zwicker et al. 2002] to obtain rendering results from different perspectives. Many subsequent methods are then proposed to improve 3DGS. Among them, some attempt to improve 3DGS in the fundamental paradigm, *e.g*. using different primitives other than Gaussian including SSS [Zhu et al. 2025], DBS [Liu et al. 2025b], and 2DGS [Huang et al. 2024], while others try to improve the training processing and adaptive density control in vanilla 3DGS, such as sampling-based [Kheradmand et al. 2024; Kim et al. 2025; Zhu et al. 2025] methods, elevating rendering quality to a new level.

Dynamic Reconstruction. Most traditional methods either perform a frame-by-frame reconstruction and lack motion continuity, or require calculating optical flow maps to simulate dynamics but cannot reconstruct dense scene flows. In comparison, dynamic reconstruction based on NeRF and 3DGS yields better results. D-NeRF [Pumarola et al. 2021] extend NeRFs to dynamic scenes by introducing an MLP to learn the implicit deformation field in every time interval for the motion. Because of the good learning capability of the implicit representation of NeRF, most NeRF-based

dynamic reconstruction methods adopt concepts that are similar to D-NeRF. On the other hand, there are currently two main threads for dynamic reconstruction using 3DGS-based methods. The first is learning the trajectories of explicit primitives at different times. Deformable 3DGS [Yang et al. 2024a], 3DGStream [Sun et al. 2024], and 4DGS-2 [Wu et al. 2024] follow the idea of D-NeRF, using neural networks/Tri-planes/HexPlanes to learn the motion and deformation of 3D Gaussians. Building on this, E-D3DGS [Bae et al. 2024] uses per-Gaussian and temporal embeddings for coarse-to-fine motion, while Dynamic 3D Gaussians [Luiten et al. 2024] learn per-frame mean and rotation with fixed appearance parameters. MotionGS [Zhu et al. 2024], SplineGS [Park et al. 2025], and FreetimeGS [Wang et al. 2025b], on the other hand, simulate the dynamics of 3D Gaussians in space through optical flow, spline, and linear flow. The second thread involves building higher-dimensional primitives based on the properties of explicit primitives in the 3DGS method. Then, these properties of primitives at different times in 3D space can be calculated by marginalizing the time dimension. Research such as 4DGS-1 [Yang et al. 2024b], 4DRotorGS [Duan et al. 2024], 7DGS [Gao et al. 2025], and UBS [Liu et al. 2025a] are all under this direction.

Uncertainty in Reconstruction. Existing works such as SGS [Savant et al. 2024] and USPLAT4D [Guo et al. 2025] have attempted to improve reconstruction quality by estimating the uncertainty of primitives in Gaussian Splatting. However, only few methods take into account the inherent uncertainty of the training data besides the primitive uncertainty. Yang et al. [2024a] argued that inaccuracies in pose estimation can cause spatial jitter between frames. Research including LongSplat [Lin et al. 2025], DG-SLAM [Xu et al. 2024] and GS-CPR [Liu et al. 2024] adjusts camera poses during the training process. Besides, Bui et al. [2025] pointed out that the temporal uncertainty should not be neglected, especially for the photon integration process within the physical shutter time. Nevertheless, there is no work modeling further data uncertainty such as asynchronous cameras and performing probabilistic modeling of multiple types of uncertainty for 4DGS.

3 Methodology

3.1 Preliminaries

3D/4DGS as a mixture model. GS represents a 3D radiance field using a large set of 3D Gaussians, each parameterized by

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{1}$$

where μ and Σ are the location and shape (with truncation). In addition, each component has additional attributes, including opacity o, and color s which is based on spherical harmonics sh. During rendering, each 3D Gaussian is transformed into a 2D Gaussian on the image plane. The rendering is then computed by:

$$C(d) = \sum_{i=1}^{N} s_i o_i G_i^{2D}(d) \prod_{j=1}^{i-1}(1 - o_j G_j^{2D}(d)). \tag{2}$$

where $C(d)$ is the final color of the dth pixel, N is the number of the Gaussians that intersect with the ray cast from the pixel. All Gaussian parameters are learned from the 2D images.

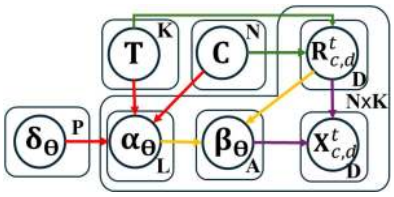

Fig. 2. A graphical model for GraphiXS. The colors correspond to the 4 steps of the generative process (1: red, 2: green, 3: yellow, 4: purple)

Essentially 3D/4DGS can be seen as learning a (unnormalized) mixture model with Gaussians [Zhu et al. 2025]:

$$F(x) = \sum w_i G_i(x) \tag{3}$$

where w_i is determined by o_i, s_i and the rendering process. Our GraphiXS generalizes this concept by a new graphical model and a corresponding generative process based on the mixture model.

Graphical model. Graphical Model is a probabilistic model which uses a graph to express the conditional dependencies of random variables. It allows structured dependencies to be introduced among random variables to describe a data generation process. Then the knowledge of this process can be used as inductive bias for model design. This is particularly suitable for 4DGS since it describes a multi-step process of rendering (Eq.1-3) and each step involves several quantities which can be treated as random variables. Also, Graphical Model naturally enables Bayesian inference. This is achieved by factorizing the joint probability of all variables into a series of conditional probabilities based on the graph, each of which can be independently modeled and evaluated. For a complex process like 4DGS, as shown later, this provides the flexibility for us to design separate priors to regulate overall model behaviors.

3.2 GraphiXS as a Graphical Model

Notations. Given videos in a multi-view setting, we aim to reconstruct the 4D radiance field. We define random variables C for the cameras pose, T for frame time, I for video frame with pixels X. The data consists of M camera poses, each recording K frames, giving a total of $M \times K$ frames and each frame with D pixels. We use superscripts for time, superscripts with brackets for derivatives, and subscripts for other indices, *e.g.* $X^t_{c,d}$ indicates the dth pixel of the image I^t_c from camera c at time t.

Our new GraphiXS is a graphical model (Fig. 2) which represents a generative process from a (unnormalized) mixture model with a potentially infinite number of components $F = \sum_{i=1}^{N} \delta_{\theta_i}$, where $N \to +\infty$. Each component is given by a distribution δ_{θ_i} parameterized by θ_i (*e.g.* Gaussian, Student's t-distribution, or other parametric form). The generative process is then described as follows:

(1) Given the mixture model F, a camera $c \sim C$ and a time $t \sim T$, sample a subset of F which is called $\{\alpha_i\}$ with **L** components, for an image I^t_c.
(2) For every $X^t_{c,d} \in I^t_c$ indexed by c, t, and d, sample a ray $R^t_{c,d}$.
(3) Given $\{\alpha_i\}$ and $R^t_{c,d}$, sample a subset of $\{\alpha_i\}$ which is called $\{\beta_j\}$ with **A** components.
(4) Finally, generate the color for $X^t_{c,t}$ based on $\{\beta_j\}$ and $R^t_{c,d}$.

Figure 2 describes most of the existing GS work where some steps are realized as rule-based deterministic processes. For instance, the

original 3DGS is special case of GraphiXS without T and with δ_{θ_i} being Gaussian. Then the last two steps correspond to a per-pixel intersection test between a ray and the Gaussians, and rasterization process. The subset $\{\alpha_i\}$ corresponds to the Gaussians that are projected onto the image plane of camera c at time t. The subset $\{\beta_j\}$ corresponds to the Gaussians hit by the ray $R^t_{c,d}$ cast for a pixel.

The key learnable parameter is θ. It is ideal to make a full Bayesian inference on the posterior distribution of θ, since there is more than one set of θs which can provide good reconstruction. In practice, this is extremely challenging due to the large number of θ. Models such as Hierarchical Dirichlet Processes could describe our generative process by assuming conjugate priors (*e.g.* Dirichlet-Multinomial), so that intermediate latent variables such as α, β can be marginalized. However, some steps of the generative process are dictated by light transport which cannot be easily described by probabilistic distributions. Furthermore, even if we forcefully used a full probabilistic model, the optimization would involve intensive sampling or large-scale variational inference [Van de Maele et al. 2024], both being prohibitively slow in the presence of millions of θs. Therefore, we choose Maximum a Posteriori (MAP):

$$\arg\max_{\theta} P(X, \mid R, \beta, \alpha, C, T, \delta_\theta) P(\delta_\theta) \tag{4}$$

where we assume uninformative priors for C and T. Despite not directly modeling their priors, we do implicitly consider their influence on the distribution of the components as random variables, explained later. We first factorize the likelihood following the graphical model:

$$\begin{aligned} P(X \mid R, \beta, \alpha, C, T, \delta_\theta) &= P(X \mid \bullet) \\ &= P(X \mid \beta, R) P(\beta \mid \alpha, R) P(R \mid C, T) P(\alpha \mid \delta_\theta, C, T), \end{aligned} \tag{5}$$

where we use $\bullet$ to represent variables in the condition for brevity. Also, among the possible options for instantiating δ in GraphiXS, *e.g.* Gaussians, Student's-t, Beta, *etc.*, we assume the component is a distribution parameterized by basic attributes including mean μ, variance Σ, color s, opacity o, and other dynamics related attributes introduced later. Although this assumption excludes free-form parameterizations such as shapes [Held et al. 2025], it is still valid for many GS frameworks. Below, we give the key equations of our instantiations of different distributions in Eq. (5) and provide the details in the supplementary material.

3.3 Probabilistic Image Reconstruction

$P(X \mid \bullet)$ represents the image generative process with 4 distributions. For completeness, we explicitly express the full image formation process in the following factorized form with four components. However, several of these steps are dictated by light transport and are inherently deterministic, so we realize them deterministically following existing practice and introduce stochasticity only in the remaining components:

$$P(X \mid \bullet) = \underbrace{P(X \mid \beta, R)}_{\text{rasterization}} \underbrace{P(\beta \mid \alpha, R)}_{\text{per-pixel comp.}} \underbrace{P(R \mid C, T)}_{\text{per-pixel ray}} \underbrace{P(\alpha \mid \delta_\theta, C, T)}_{\text{per-image comp.}} \tag{6}$$

Following 3DGS [Kerbl et al. 2023], we realize $P(X \mid \beta, R)$ as rasterization:

$$Ras(X; \beta, R) = \sum_{i=1}^{N} \lambda_i \beta_{2D_i}(X), \ \lambda_i = s_i o_i \prod_{j=1}^{i-1} (1 - o_j \beta_{2D_j}(X)) \tag{7}$$

where β_{2D} is a component projected onto the image space. Similarly, $P(\beta \mid \alpha, R) = Intersect(R, \alpha)$ is the per-pixel intersection test to identify the relevant components β. $P(R \mid C, T) = RayCasting(C, T)$ is ray casting which describes a ray R from a camera c into the space at time t. Combining the first three steps gives:

$$X = Ras(X; Intersect(R, \alpha), RayCasting(C, T)) = Ras(X; \bullet), \tag{8}$$

which is not a distribution and therefore cannot be directly used for MAP. So we use an energy-based distribution [Zhu et al. 2025]:

$$\begin{aligned} P(I \mid \beta, R, \alpha, C, T) &\propto exp(-\sum_{c}^{M} \sum_{t}^{K} L_{img}(I^t_c)) \\ L_{img} &= (1 - \epsilon_{D-SSIM}) L_1 + \epsilon_{D-SSIM} L_{D-SSIM} \\ &+ \epsilon_o \sum_i ||o_i||_1 + \epsilon_\Sigma \sum_i \sum_j ||\sqrt{\lambda_{i,j}}||_1 \end{aligned} \tag{9}$$

where L_1 and L_{D-SSIM} are the L_1 norm and the structural similarity loss between the reconstructed image (by Eq. (8)) and the ground-truth. λs are the eigenvalues of Σ. The regularization applied to the opacity ensures that the opacity is big only when a component is absolutely needed. The regularization on λ ensures the model uses components as spiky as possible (i.e. small variances). Together, the regularization terms minimize the needed number of components.

Next, since there are theoretically an infinite number of configurations of θs which can reconstruct the same radiance field, we argue that it is important to impose preferences on well-behaved θs. We impose this preference as stochasticity where well-behaved components have higher probabilities, via:

$$P(\alpha \mid \delta_\theta, C, T) = E\{P(r^t_{c,d} \mid \alpha)\} \approx \frac{\sum_c^M \sum_t^K \sum_d^D P(r^t_{c,d} \mid \alpha)}{\sum_p^N \sum_c^M \sum_t^K \sum_d^D P(r^t_{c,d} \mid \delta_{\theta_p})} \tag{10}$$

where $P(r^t_{c,d} \mid \alpha)$ and $P(r^t_{c,d} \mid \delta_{\theta_p})$ are the likelihoods of a ray $r^t_{c,d}$ from camera c at time t with respect to a component α and δ_{θ_p}. M, K, D, N are the total number of cameras, frames, pixels per image, and components. $P(r^t_{c,d} \mid \delta_{\theta_p})$ is the soft visibility of δ_{θ_p} to a pixel $x^t_{c,d}$. When δ_{θ_p} has low variance and high $P(r^t_{c,d} \mid \delta_{\theta_p})$, δ_{θ_p} is highly visible to $X^t_{c,d}$. Overall, in Eq. (10), the numerator is the soft visibility of component α to all pixels of all images across all times, while the denominator is the sum of all the soft visibility of all components. Therefore, we call $P(\alpha \mid \delta_\theta, C, T)$ the component confidence. A component with high confidence is more visible to all pixels in all frames relative to other components.

Aside from the visibility of components to cameras in time, conversely, $P(\alpha \mid \delta_\theta, C, T)$ also implicitly considers the influence of the distributions of C and T on the components. This is important for accommodating sparse views or missing frames. Maximizing this probabilistic distribution means placing components with high probabilities in the overlapped visible regions of multiple cameras and times. If the data is missing from any camera or time, the correctly placed components will be able to make good 'guesses' of the 4D

radiance field for the missing cameras and frame times, hence robust to these uncertainties.

3.4 Prior for Component Parameters

After the likelihood function, we model the prior $P(\delta_\theta)$ by modeling the distribution of the parameter θ. We assume our components move in space and time, and parameterize $P(\theta)$ to capture their dynamics, assuming $\theta(t)$ is function of t:

$$P(\theta(t)) = P(\theta(1)) \prod_{t=2}^{K} P(\theta(t) \mid \theta(t-1)) \tag{11}$$

The full specification of $\theta = \{\mu, \Sigma, o, sh, g, u, v, a, j, s\}$ includes a set of learnable parameters of the components, where μ, Σ are the mean and covariance. o is the base opacity. sh denotes its color represented through spherical harmonics. In time, a component can appear at any time g and last for a period of time u. In addition, we also introduce other dynamics related variables v, a, j, s where are detailed later. Similar to [Li et al. 2024], all parameters in θ are learned on a per-component basis.

We do not treat all variables in θ as functions of time explicitly. Instead, we explicitly model μ as it is the main variable governing the location of the components. We also make sh dependent on μ. The rest is learned independently. Note that the current θ specification is the minimal set of variables to model component dynamics. They are broadly shared in multiple types of components, *e.g.* Gaussian, Beta, Student's-t. There might be other parameters for certain choices of components, such as the control parameter in Student's-t. In this case, these additional parameters are learned independently.

First, we model $\mu(t)$ as Brownian motions:

$$\frac{d\mu}{dt} = f(\mu, t) + \mathcal{N}(0, \epsilon^2 \mathbf{I}), \tag{12}$$

where t denotes time, $\mathcal{N}$ is a Normal distribution with standard deviation ϵ. $\mathbf{I}$ is identity matrix. The reason is that we observe that the motions of objects are arbitrary and there is even no guarantee of motion smoothness. Therefore, it is crucial to be able to learn arbitrary and even potentially discontinuous motions. Discretizing Eq. (12) in time gives:

$$\mu(t) = \mu(t-1) + f(\mu, t)\triangle t + \mathcal{N}(0, \epsilon^2 \mathbf{I})\triangle t \tag{13}$$

where the dynamics is governed by $f(\mu, t)$. One simple way of modeling $f(\mu, t)$ is to consider the velocity of the components as in existing methods [Wu et al. 2024]. However, to also consider highly discontinuous motions, we learn several orders of motion derivatives, velocity v, acceleration a, jerk j and snap s:

$$f(\mu, t)\triangle t = v\,(t-g) + \tfrac{1}{2}a\,(t-g)^2 + \tfrac{1}{6}j\,(t-g)^3 + \tfrac{1}{24}s\,(t-g)^4 \tag{14}$$

Besides, we also design priors on other parameters. This is because MAP for GraphiXS involves millions of parameters. It can easily overfit without prior knowledge to regulate the variables. First, we impose a prior on the base opacity o following [Wang et al. 2025b]:

$$P(o) \propto \frac{1}{N}\sum_{1}^{N} o^2 \exp\left(-\frac{1}{2}\left(\frac{t-g}{u}\right)^2\right) \psi(\cdot), \tag{15}$$

where N is the total number of components, and $\psi(\cdot)$ denotes the evaluated response of the splatting distribution. The inputs to $\psi(\cdot)$ depend on the chosen component and its associated parameters. For Gaussian, $\psi(\cdot)$ is evaluated using the mean and variance, whereas for Student's-t we additionally include its control parameter ν. Please see the supplementary material for details. Overall this prior penalizes excessively large base opacity values.

Next, we also impose a prior on Σ:

$$P(\Sigma_p) \propto exp(-\lambda_\sigma \frac{1}{N^t}\sum_{p}^{N^t} ||\Sigma_p - \hat{\Sigma}^t||_F^2) \tag{16}$$

where Σ_p is the covariance of the pth component, λ_σ is a weight, $\hat{\Sigma}^t$ is the mean covariance of components at time t, $||\cdot||_F$ is the Frobenius norm, N^t is the total number of components whose temporal support overlaps with time t. This prior prefers small shape disparities between components.

We also place a prior on the dynamics related variables:

$$\begin{aligned} P(v_p, a_p, j_p, s_p) \propto exp(-\lambda_h \frac{1}{N^t}\sum_{p}^{N^t} \sqrt{|\det(\Sigma_p)|} \\ (||v_p||^2 + ||a_p||^2 + ||j_p||^2 + ||s_p||^2) \end{aligned} \tag{17}$$

which prefers slowly and smoothly moving components. More importantly, it is weighted by the volume proxy of the component $|\det(\Sigma_p)|$, so that the larger the component is, the slower and smoother its motion should be. This is based on the observation that larger components cover a large area in space. These tend to be in the static background, so they should not move too often and too quickly.

Finally, combining Eq. (12-17), Eq. (11) becomes:

$$\begin{aligned} P(\theta(t)) &= P(o)P(\mu(1)) \\ &\prod_{t=2}^{K} \mathcal{N}(\mu(t-1) + f(\mu, t), \epsilon^2 \mathbf{I}) \prod_{t=1}^{K} P(v, a, j, s)P(\Sigma) \end{aligned} \tag{18}$$

with the rest parameters learned directly without stochasticity. We learn $P(\theta(1))$ by maximizing Eq. (7) on the first frame.

3.5 Objective Function and Optimization

Finally, with Eq. 7-18, we have all the elements for MAP (Eq. (4)):

$$\begin{aligned} &\arg\max_\theta log\ P(X \mid R, \beta, \alpha, C, T, \theta)P(\theta) \\ \Leftrightarrow &\arg\min_\theta -log\ [P(X \mid R, \beta, \alpha, C, T, \theta)P(\theta)] \end{aligned} \tag{19}$$

where the final loss function is:

$$\mathcal{L}_{\text{full}} = \underbrace{\mathcal{L}_{img}}_{-logP(I|\bullet)} + \underbrace{\mathcal{L}_{\alpha}}_{-logP(\alpha|\bullet)} + \underbrace{\mathcal{L}_{\theta}}_{-logP(\theta)} \tag{20}$$

For optimization, we use Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [Zhu et al. 2025], which injects momentum and controlled stochasticity into gradient updates. In addition, we design component addition/removal sampling and initialization strategies for GraphiXS. Details can be found in the supplementary material.

4 Experiments

Datasets. To evaluate GraphiXS, we need to mimic different types of data uncertainty, which requires the original dataset to have enough cameras, views with good coverage, sufficiently high frequency, *etc*. Based on the criteria, we choose the Neural 3D Video (N3DV) [Li

et al. 2022] and Google Immersive (GI) [Broxton et al. 2020] datasets. Following prior work [Lee et al. 2024; Wang et al. 2025b; Wu et al. 2024; Yang et al. 2024b], we down-sample all N3DV videos to a resolution of 1352×1014 for both training and evaluation. We consistently hold out the first camera as the test view across all experimental settings and evaluate reconstruction quality on this view over 300 consecutive frames at 30 FPS. Details regarding the GI dataset and additional results are provided in the supplementary material.

Metrics. We use PSNR (Peak Signal-to-Noise Ratio), DSSIM (Dissimilarity Structural Similarity Index Measure), and LPIPS (Learned Perceptual Image Patch Similarity) [Zhang et al. 2018]. For LPIPS, we report results computed using AlexNet. DSSIM is derived from the multi-scale structural similarity (MS-SSIM) index by converting similarity to a dissimilarity measure and scaled by a factor of 0.5. In all tables, green and yellow denote the best and second-best scores, respectively.

Baseline Methods. We choose a representative set of state-of-the-art methods as baselines including 4DGS-1 [Yang et al. 2024b], 4DGS-2 [Wu et al. 2024], Ex4DGS [Lee et al. 2024], and FreeTimeGS [Wang et al. 2025b]. We use their official open-source implementations when available, otherwise our own implementation (for FreeTimeGS [Wang et al. 2025b]). For fair comparison, we train all models from scratch and run them 3 times and report the average. We color the best and second best results in green and yellow respectively. We report averaged metrics across scenes in the main paper and provide detailed results and analysis in the supplementary material.

Uncertainty Settings. We design several settings for multiple types of commonly seen data uncertainty. Standard Setting uses exactly the same setting as the baseline methods, regarded as without any data uncertainty. Sparse Views represents missing cameras. Sparse Frames represents only low frequency cameras are used. Unsynchronized Cameras represents a group of cameras which are not synchronized. Faulty Cameras represents a system with random camera malfunctions. Together these settings cover a wide range of possible scenarios where data uncertainty is induced into the data.

Instantiation and Generalization. GraphiXS does not assume specific components, so we instantiate it with two components to show its generality. The first is Student's-t (GraphiTS) and the second is approximate Gaussian (GraphiGS) by fixing the control parameter of Student's-t to a large value. In addition, we show that GraphiXS can be used to 'upgrade' existing 4DGS methods.

4.1 Comparison under Standard Setting

We show the numerical results for both N3DV and GI datasets in Tab. 1. Overall, the DSSIM for most methods are similar showing all methods capture the structure of images well in reconstruction. Both GraphiTS and GraphiGS outperform existing methods, mainly on PSNR and LPIPS. This is somewhat surprising, as the camera angles in N3DV are dense *e.g.* little data uncertainty, so one would assume further modeling of data uncertainty is redundant. However, the experiments demonstrate that even with the full observations, explicitly considering data uncertainty can further enhance the reconstruction quality. This becomes more pronounced on the GI dataset, where the standard camera setup is sparser compared to N3DV. We show one visual result in Fig. 3. This is a difficult scene. One example is that one hand of the person holding a tong moves quickly from time to time, causing visual blur. This motion involves constantly stirring the spinach in random manner both spatially and in terms of its dynamics, causing motion blur. So capturing the high-order dynamics is crucial. Therefore, comparatively GraphiXS reconstructs clearer structures of the hand and tong with details, demonstrating the dynamics of the components are learned well.

Table 1. Comparison under standard setting on N3DV and GI datasets.

Method	N3DV			GI		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	31.18	0.015	0.051	26.76	0.030	0.153
4DGS-2	30.52	0.019	0.060	23.99	0.079	0.290
Ex4DGS	31.71	0.015	0.050	20.39	0.107	0.190
FTGS	31.55	0.016	0.047	28.01	0.022	0.073
Ours (GraphiGS)	31.78	0.015	0.044	28.84	0.017	0.064
Ours (GraphiTS)	32.02	0.015	0.043	29.08	0.016	0.056

Table 2. Comparison at different levels of spatial sparsity.

Method	10%			30%			50%		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	31.37	0.015	0.049	29.11	0.024	0.058	28.54	0.027	0.068
4DGS-2	31.00	0.016	0.057	28.68	0.024	0.069	27.99	0.029	0.074
Ex4DGS	30.94	0.016	0.051	28.86	0.026	0.064	28.33	0.028	0.067
FTGS	31.38	0.016	0.047	29.04	0.026	0.063	28.06	0.031	0.072
Ours (GraphiGS)	31.73	0.016	0.044	29.35	0.025	0.058	28.60	0.026	0.066
Ours (GraphiTS)	31.62	0.016	0.046	29.41	0.024	0.061	28.56	0.027	0.067

4.2 Comparison under Sparse Views

In Sparse Views, we randomly remove 10%, 30%, and 50% of the training cameras, causing view gaps. Table 2 shows the numerical comparison on N3DV dataset. Overall, GraphiXS outperforms or is in par with other methods across all metrics. Due to the dense nature of the cameras in N3DV, a 10% reduction of the cameras does not challenge the methods, sometimes even improve the results (*e.g.* PSNR in 4DGS-1, 4DGS-2 and Ex4DGS). Further analysis suggests that the difference might be due to the randomness in the training for different methods, which might cause bigger variances in different runs in some methods but overall give similar results to the Standard Setting. When dropping 30% and 50% of the cameras, all metrics start to deteriorate, showing Sparse Views induces major data uncertainty. Across all three settings, GraphiXS achieves the best in 8 of 9 experiments and the second best in 1 experiment. We show one visual result in Fig. 1 Left and more results in Fig. 4. When more cameras are missing, it becomes more challenging for all methods where GraphiXS is affected the least.

4.3 Comparison under Sparse Frames

In Sparse Frames, we reduce the frame rate of the training cameras from 30 FPS to 20 and 10 FPS while still evaluating the scene at 30 FPS causing time sparsity. Table 3 shows the numerical results for both settings on N3DV dataset. Overall GraphiXS outperforms other

Table 3. Comparison at different levels of temporal sparsity.

Method	20 FPS			10 FPS		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	31.26	0.015	0.049	30.72	0.016	0.051
4DGS-2	30.46	0.016	0.058	31.01	0.016	0.059
Ex4DGS	31.39	0.015	0.050	31.15	0.016	0.053
FTGS	31.39	0.017	0.046	31.36	0.017	0.046
Ours (GraphiGS)	31.88	0.015	0.043	31.87	0.015	0.044
Ours (GraphiTS)	31.62	0.015	0.043	31.63	0.015	0.044

Table 4. Comparison under different levels of Unsynchronization.

Method	10% @ 20 FPS			50% @ 20 FPS		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	31.57	0.015	0.049	31.54	0.015	0.050
4DGS-2	30.68	0.016	0.059	30.92	0.016	0.059
Ex4DGS	31.34	0.016	0.050	31.01	0.016	0.051
FTGS	31.44	0.017	0.047	31.34	0.017	0.047
Ours (GraphiGS)	31.78	0.015	0.044	31.76	0.015	0.044
Ours (GraphiTS)	31.87	0.014	0.043	31.85	0.015	0.043

methods. Different from Sparse Views, Sparse Frames mimics low frequency cameras which are not suitable for recording fast motions. Since motions in N3DV are generally not fast, and all methods have dedicated parts to learn the dynamics of the components, Sparse Frames is generally less challenging than Sparse Views.

Furthermore, the key difference between different methods stem from how dense in time they require samples to be. However individual model behaviors do not share the same pattern. For the baseline methods, when FPS is lower, the results become worse as expected. The only exception is 4DGS-2 which is slightly improved. Our speculation is that since 4DGS-2 exhaustively learns pair-wise relationships between the x, y, z coordinates of the Gaussian mean and time t, there is a chance their network overfits when the sample density in time is too high. So down-sampling actually improves the results. Last, GraphiGS does not deteriorate from Standard Setting, GraphiTS deteriorates at 20 FPS but not further in 10 FPS. We show one visual result in Fig. 1 Middle and more results in Fig. 5.

4.4 Comparison under Unsynchronized Cameras

In data capture, synchronizing cameras require extra hardware, software and calibration effort. So we test if methods could work with unsynchronized cameras. We simulate this scenario by setting cameras at different FPS. We randomly select 10% and 50% of cameras and reduce their FPS from 30 to 20, yielding 2 settings with two levels of partial temporal sparsity. Table 4 shows the results on N3DV dataset. Overall, we find Unsynchronized Cameras is an easier setting for all methods compared with Sparse Views and Sparse Frames. This is expected as all methods learn the dynamics of the components which does not require observations to be available for arbitrary time. Unsynchronized Cameras could be a bigger issue if cameras are not dense in space where every frame from every camera is crucial in capturing the motions. But in N3DV, all the cameras are in front of the person. Overall, GraphiXS gives the best results.

Table 5. Comparison with Faulty Camera settings.

Method	Faulty Cam 1			Faulty Cam 2			Faulty Cam 3		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	30.77	0.017	0.051	30.31	0.020	0.055	19.29	0.136	0.314
4DGS-2	30.56	0.018	0.061	29.78	0.021	0.065	14.93	0.202	0.491
Ex4DGS	30.52	0.019	0.053	30.03	0.020	0.056	16.38	0.203	0.359
FTGS	30.46	0.020	0.051	29.90	0.026	0.051	20.49	0.115	0.228
Ours (GraphiGS)	30.90	0.018	0.048	30.39	0.020	0.052	20.65	0.118	0.250
Ours (GraphiTS)	30.79	0.019	0.050	30.18	0.021	0.053	20.82	0.119	0.253

Table 6. Comparison in before and after upgrade for FTGS across Standard and Faulty Camera settings.

Method	Standard			Faulty Cam 1			Faulty Cam 2		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
FTGS	31.55	0.016	0.047	30.46	0.020	0.051	29.90	0.026	0.051
FTGS UG	31.61	0.016	0.044	30.80	0.019	0.050	30.20	0.021	0.054

4.5 Comparison under Faulty Cameras

During data recording, any camera can malfunction which will cause a combination of all the types of the data uncertainty before this section. This data uncertainty spans across space and time. We design three settings to simulate Faulty Camera. These settings aggregate the previous parameters, resulting in total space-time sparsities of approximately 13% for Faulty Cam 1, 37% for Faulty Cam 2 and 84% for Faulty Cam 3. Table 5 shows the numerical results for these settings on N3DV dataset. One visual example is shown in Fig. 1 Right and more are in Fig. 6.

4.6 GraphiXS as an Upgrade

GraphiXS is not only a specific method, but a framework that can be used to 'upgrade' other compatible methods. This involves turning their deterministic models into probabilistic ones, using formulations similar to Eq. (7), then modeling stochasticity. The former step varies depending on the specific method, and the latter step is to add our stochastic components such as $P(\alpha \mid \theta, C, T)$ and $P(\Sigma_p)$. We directly show results here and give the details in the supplementary material. We use the Standard Setting and the Faulty Camera 1 and 2 settings as it includes all types of data uncertainty. Among the latest methods, we choose FTGS [Wang et al. 2025b] and Table 6 shows the numerical results. One visual example is shown in Fig. 7.

4.7 Ablation Study

Since we decompose GraphiXS into distributions (Eq. (5)) and propose various parameterization for each distribution, we show their respective effectiveness. The key distributions include the Higher Order Dynamics (Eq. (12)) and the component confidence $P(\alpha \mid \delta_\theta, C, T)$(Eq. (10)). We also include a deterministic variant of GraphiXS to assess the benefit of our probabilistic formulation (see the supplementary material for details). Therefore, we conduct an ablation study on GraphiGS under the three settings on Faulty Cam 1 and 2. We show the quantitative results in Table 7 and qualitative results in Fig. 8. In general, removing any component will deteriorate the results. This is more obvious in Faulty Cams 2 which has a higher level of uncertainty. 'W/O Higher Order Dynamics' only considers the position and velocity of the component, which is a strategy for

Table 7. Quantitative results of our ablation study on GraphiGS.

Method	Faulty Cam 1			Faulty Cam 2		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
W/O Higher Order Dynamics	30.84	0.019	0.049	30.03	0.021	0.054
W/O $P(\alpha \mid \theta, C, T)$	30.74	0.019	0.051	30.04	0.022	0.054
W/O Stochasticity	28.64	0.030	0.100	27.66	0.036	0.113
Ours (GraphiGS)	30.90	0.018	0.048	30.39	0.020	0.052

many existing 4DGS methods [Lee et al. 2024; Li et al. 2024; Wang et al. 2025b]. But the results shows that considering higher order dynamics will improve the results. Furthermore, 'W/O $P(\alpha \mid \theta, C, T)$' shows that the component confidence part also improves the results. This part is designed to regulate component locations and shapes to make them visible to cameras in time. Lastly, the deterministic variant shows a significant performance drop, demonstrating the effectiveness of our framework.

5 Conclusion and Discussion

We have proposed the first probabilistic 4DGS framework, GraphiXS, that holistically considers multiple types of data uncertainty. GraphiXS is general in that it can be instantiated with different components or used to upgrade existing methods. Through exhaustive evaluation, we have demonstrated the effectiveness of GraphiXS. A major limitation is GraphiXS assumes the components are probabilistic distributions parameterized by a set of common parameters. This makes it unsuitable for upgrading methods with other types of components such as geometric primitives. Also, GraphiXS does not do full Bayesian inference, *i.e.* no posterior distribution learned for θ, which would be ideal given the stochastic nature of 4DGS models. Lastly, GraphiXS assumes uninformative priors for C and T, and could be extended to incorporate camera pose priors $P(C)$; we leave this to future work.

Acknowledgments

This work was supported in part by the Dr. Rabin Ezra Scholarship (Charity No. 1116049), awarded to Doğa Yılmaz, and the UK Research and Innovation AIRR Innovator Award (0261-5654-9320-1).

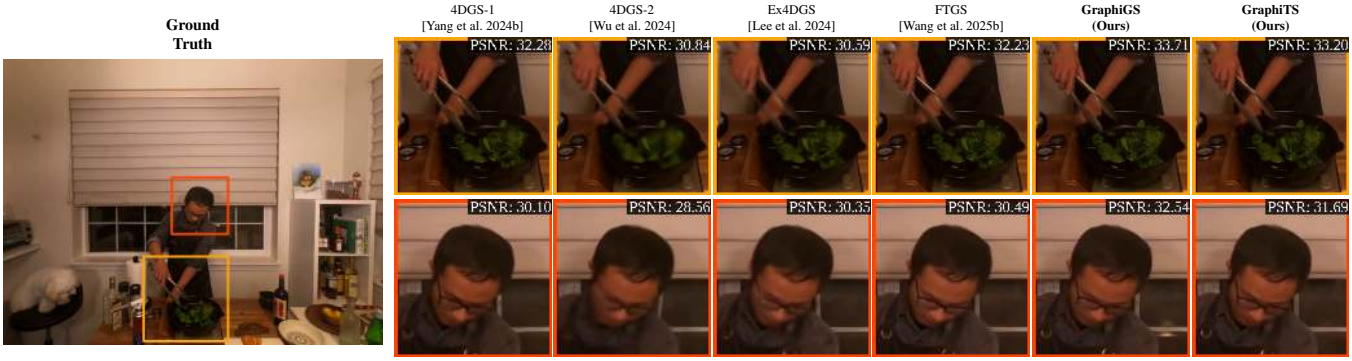

Fig. 3. Comparison using N3DV dataset under standard setting. Per-region PSNR scores are given at the top right of each image. Enlarged regions contain complex and fast motions *e.g.* the tongs motion. GraphiXS reconstruction is more clear and richer in details than other methods.

Fig. 4. Comparison using GI dataset under 10%, 30%, and 50% spatial sparsity. Per-region PSNR scores are given at the top right of each crop. GraphiXS is affected the least when the percentage of missing cameras increases.

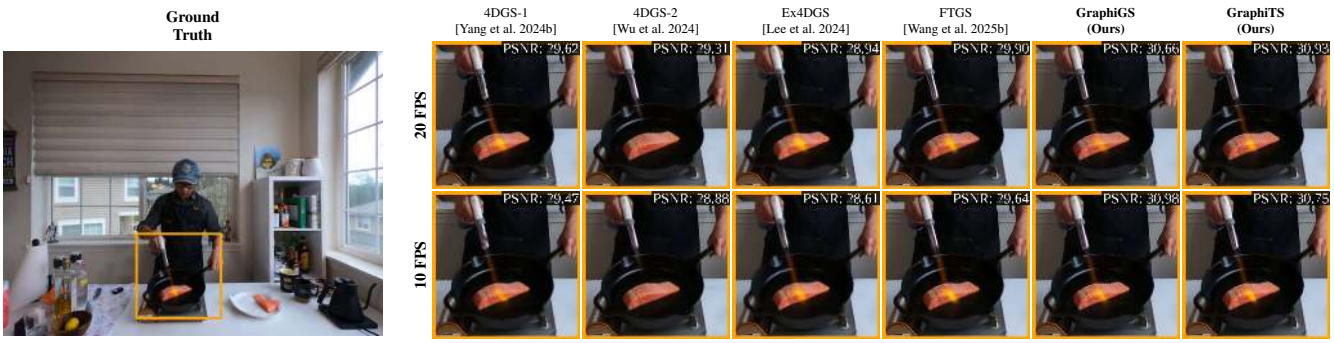

Fig. 5. Comparison using N3DV dataset under 20 FPS and 10 FPS temporal sparsity. Per-region PSNR scores are given at the top right of each crop. GraphiXS is affected the least when the training camera FPS drops.

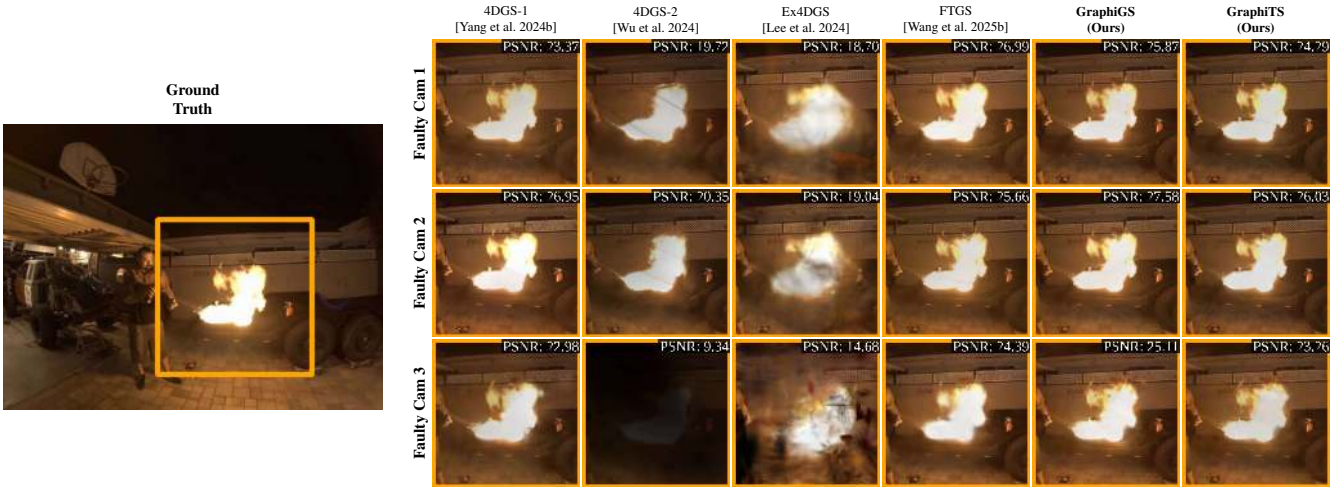

Fig. 6. Comparison using GI dataset under faulty camera settings. Per-region PSNR scores are given at the top right of each crop. GraphiXS is affected the least when spatio-temporal sparsity is increased.

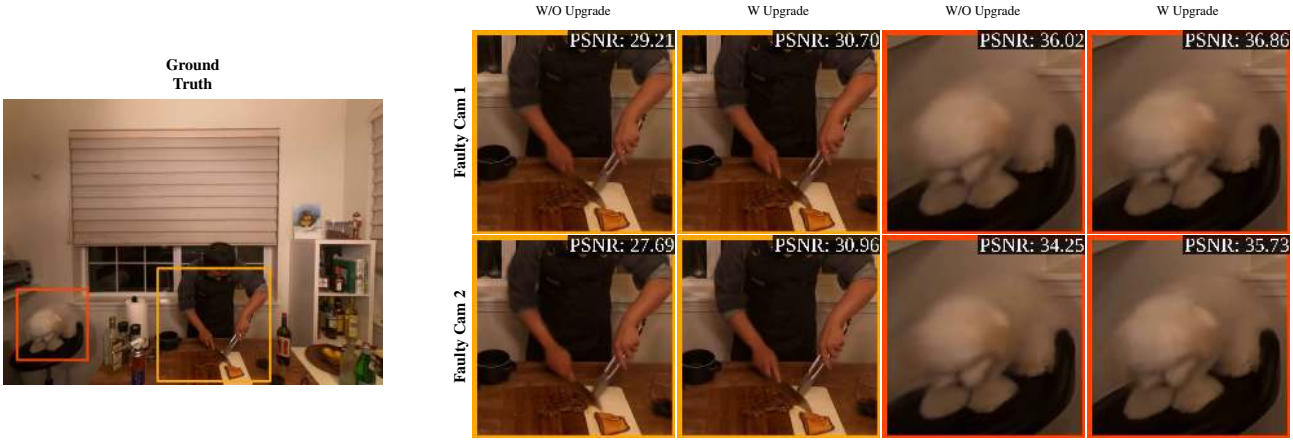

Fig. 7. Comparison of FTGS [Wang et al. 2025b] with and without upgrading under faulty camera 1 and 2 settings. Per-region PSNR scores are given at the top right of each crop. Upgrading FTGS using GraphiXS improves visual quality under various levels of spatio-temporal uncertainty.

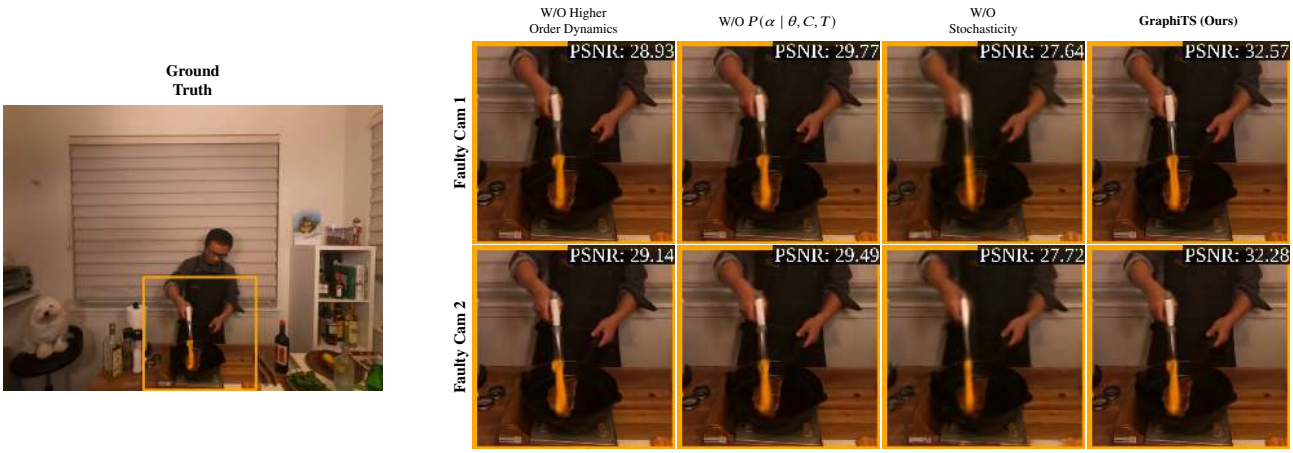

Fig. 8. Visual results of our ablation study. Per-region PSNR scores are given at the top right of each crop.

References

Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. 2024. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In *European Conference on Computer Vision*. Springer, 321–335.

Tim Bailey and Hugh Durrant-Whyte. 2006. Simultaneous localization and mapping: part II. *IEEE robotics & automation magazine* 13, 3 (2006), 108–117.

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. 39, 4 (2020), 86:1–86:15.

Minh-Quan Viet Bui, Jongmin Park, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. 2025. MoBGS: Motion Deblurring Dynamic 3D Gaussian Splatting for Blurry Monocular Video. *arXiv preprint arXiv:2504.15122* (2025).

Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. 2024. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 conference papers*. 1–11.

Junli Deng, Ping Shi, Qipei Li, and Jinyang Guo. 2025. DynaSplat: Dynamic-Static Gaussian Splatting with Hierarchical Motion Decomposition for Scene Reconstruction. *arXiv preprint arXiv:2506.09836* (2025).

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

Hugh Durrant-Whyte and Tim Bailey. 2006. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine* 13, 2 (2006), 99–110.

Guofeng Feng, Siyan Chen, Rong Fu, Zimu Liao, Yi Wang, Tao Liu, Boni Hu, Linning Xu, Zhilin Pei, Hengjie Li, et al. 2025. Flashgs: Efficient 3d gaussian splatting for large-scale and high-resolution rendering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 26652–26662.

Qiankun Gao, Yanmin Wu, Chengxiang Wen, Jiarui Meng, Luyang Tang, Jie Chen, Ronggang Wang, and Jian Zhang. 2024. Relaygs: Reconstructing dynamic scenes with large-scale and complex motions via relay gaussians. *arXiv preprint arXiv:2412.02493* (2024).

Zhongpai Gao, Benjamin Planche, Meng Zheng, Anwesa Choudhuri, Terrence Chen, and Ziyan Wu. 2025. 7DGS: Unified Spatial-Temporal-Angular Gaussian Splatting. *arXiv preprint arXiv:2503.07946* (2025).

Fengzhi Guo, Chih-Chuan Hsu, Sihao Ding, and Cheng Zhang. 2025. Uncertainty Matters in Dynamic Gaussian Splatting for Monocular 4D Reconstruction. *arXiv preprint arXiv:2510.12768* (2025).

Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. 2024. Ges: Generalized exponential splatting for efficient radiance field rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19812–19822.

Jan Held, Renaud Vandeghen, Abdullah Hamdi, Adrien Deliege, Anthony Cioppa, Silvio Giancola, Andrea Vedaldi, Bernard Ghanem, and Marc Van Droogenbroeck. 2025. 3D convex splatting: Radiance field rendering with 3D smooth convexes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21360–21369.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*. 1–11.

Yoonwoo Jeong, Junmyeong Lee, Hoseung Choi, and Minsu Cho. 2024. RoDyGS: Robust Dynamic Gaussian Splatting for Casual Videos. *arXiv preprint arXiv:2412.03077* (2024).

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. 2023. Consistent4d: Consistent 360 {\deg} dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848* (2023).

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–15.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 2024. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems* 37 (2024), 80965–80986.

Hyunjin Kim, Haebeom Jung, and Jaesik Park. 2025. Metropolis-Hastings Sampling for 3D Gaussian Reconstruction. *arXiv preprint arXiv:2506.12945* (2025).

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

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

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. 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.

Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. 2025. Longsplat: Robust unposed 3d gaussian splatting for casual long videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 27412–27422.

Changkun Liu, Shuai Chen, Yash Bhalgat, Siyan Hu, Ming Cheng, Zirui Wang, Victor Adrian Prisacariu, and Tristan Braud. 2024. GS-CPR: Efficient camera pose refinement via 3d gaussian splatting. *arXiv preprint arXiv:2408.11085* (2024).

Rong Liu, Zhongpai Gao, Benjamin Planche, Meida Chen, Van Nguyen Nguyen, Meng Zheng, Anwesa Choudhuri, Terrence Chen, Yue Wang, Andrew Feng, et al. 2025a. Universal Beta Splatting. *arXiv preprint arXiv:2510.03312* (2025).

Rong Liu, Dylan Sun, Meida Chen, Yue Wang, and Andrew Feng. 2025b. Deformable beta splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11.

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

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. 2024. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. 2024. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18039–18048.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 127–136.

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

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

Luca Savant, Diego Valsesia, and Enrico Magli. 2024. Modeling uncertainty for gaussian splatting. *arXiv preprint arXiv:2403.18476* (2024).

Johannes L. Schönberger et al. 2016. COLMAP: Structure-from-Motion and Multi-View Stereo. https://colmap.github.io/. Accessed: 2026-01-11.

Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, Vol. 1. IEEE, 519–528.

Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 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.

Shimon Ullman. 1979. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203, 1153 (1979), 405–426.

Toon Van de Maele, Ozan Çatal, Alexander Tschantz, Christopher L Buckley, and Tim Verbelen. 2024. Variational Bayes Gaussian Splatting. *arXiv preprint arXiv:2410.03592* (2024).

Rui Wang, Quentin Lohmeyer, Mirko Meboldt, and Siyu Tang. 2025a. Degauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6294–6303.

Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. 2025b. FreeTimeGS: Free Gaussian Primitives at Anytime Anywhere for Dynamic Scene Reconstruction. In *CVPR*. https://zju3dv.github.io/freetimegs

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 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.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2025. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21469–21480.

Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 2025. Compositional generative model of unbounded 4D cities. *arXiv preprint arXiv:2501.08983* (2025).

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4389–4398.

Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. 2024. Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization. *Advances in Neural Information Processing Systems* 37 (2024), 51577–51596.

Qitong Yang, Mingtao Feng, Zijie Wu, Weisheng Dong, Fangfang Wu, Yaonan Wang, and Ajmal Mian. 2025. Hierarchical Gaussian Mixture Model Splatting for Efficient and Part Controllable 3D Generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 11104–11114.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. 2024a. 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.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. 2024b. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In *International Conference on Learning Representations (ICLR)*.

Doğa Yılmaz and Furkan Kıraç. 2023. Illumination-guided inverse rendering benchmark: Learning real objects with few cameras. *Computers & Graphics* 115 (2023), 107–121. doi:10.1016/j.cag.2023.07.002

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. 2024. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 21634–21643.

Jialin Zhu, Jiangbei Yue, Feixiang He, and He Wang. 2025. 3D Student Splatting and Scooping. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21045–21054.

Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. 2024. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems* 37 (2024), 101790–101817.

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

Graphical X Splatting (GraphiXS): A Graphical Model for 4D Gaussian Splatting under Uncertainty Supplementary Material

DOĞA YILMAZ, University College London, United Kingdom
JIALIN ZHU, Baidu Inc., China
DESHAN GONG, The University of Hong Kong, China
HE WANG*, University College London, United Kingdom

CCS Concepts: • **Computing methodologies** → **Machine learning approaches**; **Rendering**;

ACM Reference Format:
Doğa Yılmaz, Jialin Zhu, Deshan Gong, and He Wang. 2026. Graphical X Splatting (GraphiXS): A Graphical Model for 4D Gaussian Splatting under Uncertainty Supplementary Material. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3799902.3811085

1 Additional Experimental Results

1.1 Quantitative Results

We report comprehensive per-scene quantitative metrics on the Neural 3D Video (N3DV) [Li et al. 2022] dataset in Tabs. 1 to 5. We further present analogous results on the Google Immersive (GI) [Broxton et al. 2020] dataset in Tabs. 6 to 10. In addition, we include results on a synthetic scene where we have full control over camera configurations in Tab. 11. In all tables, green and yellow denote the best and second-best scores, respectively. Overall, these results are consistent with the average metrics reported in the main manuscript. However, the Coffee Martini scene exhibits a trend that differs from the other sequences. This discrepancy is caused by a known temporal synchronization artifact in Camera 13 of the Coffee Martini scene. Camera 13 is temporally misaligned with respect to the other cameras, leading to consistent frame-level offsets between its observations and the true scene dynamics. As a result, images from Camera 13 correspond to different time instants than those assumed by methods that rely on strict multi-view temporal alignment. This issue primarily affects approaches that explicitly model dynamic geometry and appearance from synchronized multi-view inputs. In particular, GraphiXS and other explicit dynamic reconstruction methods assume consistent temporal correspondence across cameras and therefore interpret the desynchronized observations as conflicting geometric or appearance changes, which degrades reconstruction quality and quantitative performance. In contrast, 4DGS-2 outperforms GraphiXS and other baselines in this setting. We attribute this behavior to the implicit nature of the 4DGS-2 representation, which makes it more tolerant to moderate temporal misalignment in the input camera streams. As a result, 4DGS-2 is less affected by the Camera 13 synchronization error, leading to improved quantitative performance on the Coffee Martini scene despite the underlying dataset error. This effect is also reflected in the Coffee Martini columns of Tabs. 1 to 5, where 4DGS-2 achieves the highest PSNR in seven out of ten evaluated settings.

For the GI dataset, we evaluate on a representative subset of scenes: Welder, Flames, Horse, Goats, and Alexa10. These scenes provide diverse challenges, including fast-moving particles (Welder, Flames), outdoor environments (Horse, Goats), and complex color patterns (Alexa10). For all scenes, we designate the first camera as the test view and evaluate over 30 frames at quarter resolution. The GI capture setup employs outward-facing cameras on a spherical rig, in contrast to N3DV where cameras are inward-facing toward the subject. This results in reduced view overlap and a more spatially sparse capture regime. Under these conditions, the performance gap between GraphiXS and competing baselines becomes more pronounced, highlighting the advantages of our probabilistic modeling under uncertainty.

In the synthetic CityCrowd scene, which consists of multiple characters walking in a city environment, we place cameras in a random, CCTV-like configuration to simulate a real-world unconstrained multi-view setup. We report results for both the standard setting and a 50% spatial sparsity configuration. Consistent with the other datasets, GraphiXS outperforms baselines under these CCTV-style camera distributions.

Overall, we see that different methods are affected differently across datasets, uncertainty types, and uncertainty levels. This is expected as all methods have generality. This is why GraphiXS does not outperform all existing methods across all scenes under all uncertainty settings. However, when analyzing the high-level statistics, GraphiXS provides the best or second-best result in 172 out of a total of 198 individual metrics reported on N3DV dataset in Tabs. 1 to 5 and 159 in 165 on GI dataset in Tabs. 6 to 10.

1.2 Qualitative Results

We present further visual comparisons on the N3DV dataset in Figs. 2 and 3, on the GI dataset in Figs. 4 and 5, and on the CityCrowd dataset in Fig. 6. Consistent with the findings in the main manuscript, in general, GraphiXS recovers finer high-frequency details across all

*Corresponding author.

Authors' Contact Information: Doğa Yılmaz, University College London, London, United Kingdom, doga.yilmaz@ucl.ac.uk; Jialin Zhu, Baidu Inc., Beijing, China, misaliet@outlook.com; Deshan Gong, The University of Hong Kong, Hong Kong, China, deshan@hku.hk; He Wang, University College London, London, United Kingdom, he_wang@ucl.ac.uk.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811085

Table 1. Quantitative comparison on the Neural 3D Video dataset with standard setting.

Method	PSNR ↑ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	DSSIM ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	LPIPS ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.
4DGS-1	28.29	32.76	33.37	28.86	31.48	32.29	31.18	0.023	0.012	0.010	0.021	0.012	0.010	0.015	0.075	0.042	0.041	0.070	0.036	0.035	0.051
4DGS-2	27.68	30.40	32.17	28.84	31.82	32.21	30.52	0.025	0.017	0.027	0.022	0.012	0.012	0.019	0.084	0.055	0.040	0.081	0.043	0.042	0.060
Ex4DGS	28.54	32.86	33.26	28.95	33.17	33.49	31.71	0.023	0.013	0.012	0.022	0.011	0.011	0.015	0.073	0.042	0.044	0.068	0.036	0.037	0.050
FTGS	27.36	33.28	33.76	28.06	33.49	33.32	31.55	0.030	0.012	0.011	0.024	0.010	0.010	0.016	0.076	0.039	0.037	0.064	0.035	0.031	0.047
Ours (GraphiGS)	28.19	33.43	33.70	28.50	33.49	33.40	31.78	0.024	0.011	0.011	0.024	0.010	0.009	0.015	0.067	0.035	0.034	0.063	0.033	0.030	0.044
Ours (GraphiTS)	28.34	33.49	34.01	28.95	33.85	33.47	32.02	0.024	0.011	0.010	0.022	0.010	0.010	0.015	0.066	0.038	0.036	0.059	0.029	0.028	0.043

Table 2. Quantitative comparison on the Neural 3D Video dataset with 10%, 30%, and 50% spatial sparsity.

Method	PSNR ↑ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	DSSIM ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	LPIPS ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.
10% spatial Sparsity																					
4DGS-1	28.12	33.25	33.53	28.12	32.06	33.13	31.37	0.023	0.011	0.010	0.023	0.011	0.010	0.015	0.074	0.040	0.039	0.068	0.036	0.034	0.049
4DGS-2	28.73	32.14	31.90	28.91	31.51	32.84	31.00	0.022	0.014	0.014	0.021	0.012	0.011	0.016	0.080	0.051	0.053	0.077	0.044	0.040	0.057
Ex4DGS	27.47	32.76	31.36	28.27	32.88	32.89	30.94	0.027	0.013	0.013	0.024	0.010	0.011	0.016	0.074	0.043	0.043	0.072	0.036	0.038	0.051
FTGS	27.18	33.09	33.50	28.37	33.06	33.09	31.38	0.030	0.012	0.011	0.025	0.011	0.010	0.016	0.073	0.041	0.038	0.066	0.035	0.030	0.047
Ours (GraphiGS)	27.77	33.37	33.67	28.68	33.52	33.35	31.73	0.028	0.012	0.011	0.023	0.010	0.009	0.016	0.071	0.037	0.036	0.062	0.030	0.030	0.044
Ours (GraphiTS)	27.86	32.87	33.47	28.91	33.30	33.28	31.62	0.028	0.012	0.011	0.023	0.010	0.009	0.016	0.073	0.042	0.037	0.061	0.034	0.029	0.046
30% spatial Sparsity																					
4DGS-1	25.20	30.90	29.35	25.11	32.22	31.91	29.11	0.038	0.016	0.016	0.038	0.022	0.013	0.024	0.094	0.048	0.049	0.089	0.031	0.039	0.058
4DGS-2	25.77	30.01	29.92	25.81	30.96	29.60	28.68	0.037	0.021	0.019	0.035	0.016	0.016	0.024	0.098	0.058	0.062	0.096	0.048	0.049	0.069
Ex4DGS	24.15	30.70	30.92	25.47	31.30	30.64	28.86	0.048	0.018	0.017	0.035	0.016	0.018	0.026	0.101	0.050	0.052	0.088	0.044	0.048	0.064
FTGS	24.55	31.10	30.64	24.91	31.61	31.44	29.04	0.051	0.016	0.018	0.044	0.014	0.014	0.026	0.103	0.046	0.057	0.092	0.042	0.038	0.063
Ours (GraphiGS)	24.29	31.16	30.88	25.61	31.93	32.20	29.35	0.051	0.016	0.015	0.040	0.014	0.013	0.025	0.102	0.043	0.044	0.087	0.037	0.037	0.058
Ours (GraphiTS)	24.56	31.23	31.07	25.61	31.65	32.34	29.41	0.049	0.016	0.015	0.038	0.015	0.013	0.024	0.100	0.049	0.047	0.086	0.046	0.037	0.061
50% spatial Sparsity																					
4DGS-1	24.12	30.13	29.85	24.52	31.40	31.24	28.54	0.049	0.019	0.017	0.044	0.015	0.014	0.027	0.110	0.052	0.057	0.099	0.044	0.043	0.068
4DGS-2	24.11	29.26	29.00	24.71	30.62	30.25	27.99	0.048	0.025	0.020	0.040	0.020	0.019	0.029	0.109	0.063	0.064	0.098	0.055	0.054	0.074
Ex4DGS	23.88	29.99	30.41	24.55	31.03	30.14	28.33	0.052	0.021	0.018	0.042	0.018	0.019	0.028	0.110	0.053	0.052	0.092	0.046	0.048	0.067
FTGS	22.81	30.42	29.79	23.78	31.20	30.33	28.06	0.065	0.019	0.017	0.050	0.016	0.018	0.031	0.118	0.051	0.052	0.125	0.045	0.043	0.072
Ours (GraphiGS)	24.11	30.65	30.10	24.59	31.11	31.05	28.60	0.048	0.018	0.016	0.045	0.016	0.015	0.026	0.110	0.053	0.048	0.095	0.044	0.045	0.066
Ours (GraphiTS)	23.92	30.71	30.09	24.41	31.13	31.07	28.56	0.048	0.018	0.017	0.048	0.016	0.015	0.027	0.111	0.052	0.051	0.098	0.046	0.043	0.067

Table 3. Quantitative comparison on the Neural 3D Video dataset with temporal sparsity where the 30 FPS videos are trained with 20 FPS and 10 FPS.

Method	PSNR ↑ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	DSSIM ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	LPIPS ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.
20 FPS																					
4DGS-1	28.19	32.29	33.35	28.48	32.18	33.05	31.26	0.023	0.012	0.010	0.023	0.010	0.010	0.015	0.074	0.042	0.041	0.069	0.035	0.034	0.049
4DGS-2	28.85	31.12	31.46	28.61	31.71	31.02	30.46	0.021	0.015	0.014	0.022	0.012	0.012	0.016	0.082	0.053	0.053	0.078	0.042	0.042	0.058
Ex4DGS	28.27	32.46	32.84	28.66	32.98	33.12	31.39	0.023	0.014	0.012	0.023	0.010	0.011	0.015	0.072	0.044	0.044	0.069	0.035	0.037	0.050
FTGS	27.11	33.11	33.68	27.93	32.95	33.57	31.39	0.031	0.012	0.011	0.026	0.011	0.010	0.017	0.076	0.038	0.037	0.065	0.033	0.030	0.046
Ours (Gaussian)	27.82	33.49	33.96	28.41	33.64	33.98	31.88	0.026	0.011	0.010	0.022	0.010	0.009	0.015	0.069	0.036	0.035	0.062	0.030	0.029	0.043
Ours (Student's T)	28.00	33.05	33.80	28.63	33.05	33.17	31.62	0.026	0.012	0.010	0.022	0.010	0.010	0.015	0.069	0.036	0.035	0.060	0.030	0.029	0.043
10 FPS																					
4DGS-1	28.28	32.81	32.96	28.36	31.95	30.00	30.72	0.023	0.012	0.011	0.023	0.011	0.014	0.016	0.074	0.041	0.042	0.070	0.036	0.042	0.051
4DGS-2	28.60	30.24	32.62	29.10	33.13	32.39	31.01	0.023	0.016	0.014	0.021	0.011	0.011	0.016	0.083	0.053	0.053	0.080	0.041	0.041	0.059
Ex4DGS	28.20	32.03	33.14	28.81	31.83	32.90	31.15	0.025	0.014	0.012	0.022	0.011	0.012	0.016	0.074	0.047	0.044	0.072	0.039	0.041	0.053
FTGS	27.36	33.14	33.44	27.97	33.07	33.18	31.36	0.029	0.012	0.011	0.027	0.011	0.010	0.017	0.074	0.037	0.037	0.067	0.033	0.031	0.046
Ours (GraphiGS)	27.94	33.40	33.90	28.59	33.64	33.75	31.87	0.027	0.012	0.010	0.022	0.010	0.009	0.015	0.070	0.036	0.036	0.062	0.032	0.030	0.044
Ours (GraphiTS)	27.63	33.15	33.44	28.32	33.57	33.66	31.63	0.027	0.012	0.011	0.022	0.010	0.009	0.015	0.069	0.035	0.035	0.060	0.031	0.030	0.044

uncertainty settings. Notably, dynamic scene components are reconstructed with significantly higher fidelity compared to baselines. This is particularly evident as temporal sparsity increases. The crops highlighting the subject's hands and motion in the rows corresponding to 10 FPS, 10% @ 20 FPS, 50% @ 20 FPS, and Random Camera Failures 1 in Figs. 2 and 3. We attribute this performance gap to the expressiveness of our higher-order motion model.

A trade-off of our approach is that when most cameras are directed away from the primary region of interest, the model assigns lower confidence to under-observed areas, which can lead to reduced quality in those regions. This effect is visible in the Flame Salmon and Coffee Martini scenes in Figs. 2 and 3, where outdoor regions have lower visibility, while the representation budget is concentrated on the well-observed areas, resulting in higher-quality reconstruction and motion fidelity in those regions.

1.3 Higher Order Motion

Figure 1 visualizes the magnitudes of the motion derivatives, providing empirical evidence for our choice of modeling the component dynamics up to the fourth order. As expected, dynamic regions exhibit strong responses in the lower-order terms, corresponding to

Table 4. Quantitative comparison on the Neural 3D Video dataset with non-synchronous temporal sparsity where 10% and 50% of the views are observed at 20 FPS.

Method	PSNR ↑ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	DSSIM ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	LPIPS ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.
10% of the views observed at 20 FPS																					
4DGS-1	28.11	33.14	33.18	28.26	33.44	33.30	31.57	0.023	0.011	0.011	0.023	0.010	0.010	0.015	0.074	0.040	0.040	0.070	0.034	0.034	0.049
4DGS-2	28.68	31.85	30.58	28.80	32.49	31.70	30.68	0.022	0.013	0.014	0.022	0.012	0.012	0.016	0.082	0.051	0.054	0.082	0.043	0.043	0.059
Ex4DGS	27.76	32.59	32.75	28.66	33.67	32.63	31.34	0.026	0.013	0.012	0.022	0.010	0.011	0.016	0.077	0.043	0.041	0.069	0.034	0.038	0.050
FTGS	27.06	33.22	33.82	28.24	33.17	33.15	31.44	0.031	0.012	0.011	0.025	0.010	0.010	0.017	0.075	0.039	0.037	0.066	0.032	0.031	0.047
Ours (GraphiGS)	28.08	33.34	33.83	28.29	33.54	33.62	31.78	0.026	0.012	0.011	0.023	0.010	0.009	0.015	0.069	0.037	0.036	0.062	0.031	0.029	0.044
Ours (GraphiTS)	28.08	33.45	33.69	29.38	33.27	33.36	31.87	0.024	0.011	0.010	0.021	0.010	0.009	0.014	0.066	0.036	0.035	0.060	0.030	0.029	0.043
50% of the views observed at 20 FPS																					
4DGS-1	28.08	33.11	33.50	28.24	33.29	33.01	31.54	0.023	0.011	0.011	0.023	0.010	0.010	0.015	0.076	0.041	0.040	0.071	0.035	0.035	0.050
4DGS-2	28.75	31.45	32.32	28.60	32.42	31.97	30.92	0.022	0.014	0.013	0.022	0.011	0.011	0.016	0.082	0.052	0.053	0.082	0.043	0.042	0.059
Ex4DGS	28.23	32.60	31.76	28.18	32.93	32.35	31.01	0.023	0.013	0.013	0.024	0.011	0.012	0.016	0.071	0.046	0.044	0.069	0.037	0.041	0.051
FTGS	27.14	33.18	33.43	27.75	33.10	33.45	31.34	0.031	0.012	0.011	0.026	0.010	0.010	0.017	0.076	0.038	0.038	0.067	0.032	0.030	0.047
Ours (GraphiGS)	27.81	33.51	33.75	28.59	33.41	33.50	31.76	0.026	0.011	0.011	0.023	0.010	0.009	0.015	0.068	0.036	0.038	0.064	0.030	0.029	0.044
Ours (GraphiTS)	27.87	33.40	33.63	28.80	33.64	33.74	31.85	0.025	0.011	0.010	0.022	0.010	0.009	0.015	0.068	0.036	0.035	0.061	0.030	0.029	0.043

Table 5. Quantitative comparison on the Neural 3D Video dataset with random camera failures.

Method	PSNR ↑ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	DSSIM ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.	LPIPS ↓ Cof. Mart.	Cook Spin.	Cut Beef	Flame Salm.	Flame Steak	Sear Steak	Avg.
Faulty Cam 1																					
4DGS-1	26.58	32.31	33.16	27.43	32.67	32.50	30.77	0.028	0.013	0.012	0.026	0.011	0.010	0.017	0.080	0.042	0.040	0.077	0.034	0.034	0.051
4DGS-2	27.28	31.77	31.84	27.32	32.80	32.34	30.56	0.027	0.015	0.015	0.027	0.013	0.012	0.018	0.084	0.053	0.055	0.083	0.045	0.043	0.061
Ex4DGS	27.28	31.96	31.87	27.32	32.01	32.68	30.52	0.030	0.015	0.015	0.028	0.012	0.012	0.019	0.077	0.047	0.046	0.074	0.038	0.039	0.053
FTGS	25.30	32.51	33.04	26.65	32.85	32.39	30.46	0.039	0.014	0.013	0.031	0.012	0.011	0.020	0.083	0.040	0.044	0.073	0.033	0.032	0.051
Ours (GraphiGS)	26.10	32.67	33.34	27.03	33.05	33.21	30.90	0.035	0.013	0.012	0.028	0.011	0.011	0.018	0.078	0.038	0.037	0.068	0.034	0.033	0.048
Ours (GraphiTS)	26.29	32.36	33.46	26.94	32.86	32.85	30.79	0.033	0.014	0.013	0.029	0.011	0.011	0.019	0.077	0.046	0.044	0.071	0.032	0.031	0.050
Faulty Cam 2																					
4DGS-1	25.83	32.06	32.16	26.79	32.47	32.53	30.31	0.034	0.015	0.014	0.028	0.014	0.013	0.020	0.088	0.044	0.044	0.078	0.036	0.037	0.055
4DGS-2	25.70	31.04	31.35	26.74	31.95	31.88	29.78	0.036	0.018	0.016	0.031	0.015	0.013	0.021	0.096	0.058	0.058	0.086	0.048	0.045	0.065
Ex4DGS	26.38	31.28	32.03	25.70	32.61	32.18	30.03	0.033	0.016	0.015	0.032	0.013	0.014	0.020	0.085	0.048	0.047	0.079	0.039	0.040	0.056
FTGS	24.83	32.48	31.94	25.62	32.59	31.94	29.90	0.042	0.014	0.014	0.037	0.013	0.038	0.026	0.089	0.043	0.042	0.079	0.038	0.013	0.051
Ours (GraphiGS)	25.57	32.34	32.61	26.17	32.64	33.01	30.39	0.040	0.014	0.014	0.031	0.012	0.011	0.020	0.088	0.044	0.040	0.070	0.038	0.033	0.052
Ours (GraphiTS)	25.35	32.53	32.49	26.08	32.56	32.08	30.18	0.042	0.014	0.013	0.033	0.012	0.013	0.021	0.088	0.041	0.040	0.075	0.037	0.036	0.053
Faulty Cam 3																					
4DGS-1	22.33	17.26	18.14	20.01	19.38	18.94	19.34	0.065	0.135	0.135	0.137	0.153	0.147	0.128	0.141	0.302	0.300	0.279	0.300	0.316	0.273
4DGS-2	12.72	9.29	8.33	11.30	8.92	12.41	10.49	0.285	0.245	0.247	0.300	0.226	0.230	0.255	0.578	0.564	0.566	0.611	0.567	0.538	0.571
Ex4DGS	18.40	26.98	26.87	20.12	27.77	26.21	24.40	0.105	0.028	0.026	0.075	0.022	0.025	0.047	0.191	0.069	0.067	0.153	0.059	0.061	0.100
FTGS	22.46	29.07	30.07	23.54	29.33	29.44	27.32	0.077	0.025	0.022	0.054	0.021	0.019	0.036	0.143	0.087	0.079	0.116	0.076	0.051	0.092
Ours (GraphiGS)	21.86	29.32	29.82	24.06	29.35	29.81	27.37	0.080	0.023	0.021	0.053	0.018	0.018	0.036	0.190	0.082	0.068	0.108	0.058	0.061	0.094
Ours (GraphiTS)	21.79	29.37	29.94	24.10	29.78	30.00	27.50	0.079	0.023	0.020	0.052	0.019	0.017	0.035	0.146	0.081	0.060	0.105	0.070	0.057	0.086

Table 6. Quantitative comparison on the Google Immersive Dataset with standard setting.

Method	PSNR ↑ Welder	Flames	Horse	Goats	Alexa	Avg.	DSSIM ↓ Welder	Flames	Horse	Goats	Alexa	Avg.	LPIPS ↓ Welder	Flames	Horse	Goats	Alexa	Avg.
4DGS-1	24.24	28.58	26.68	25.01	29.31	26.76	0.042	0.015	0.020	0.057	0.019	0.031	0.193	0.056	0.131	0.270	0.117	0.153
4DGS-2	22.47	27.74	23.32	20.52	25.90	23.99	0.060	0.029	0.054	0.188	0.066	0.079	0.252	0.145	0.288	0.503	0.266	0.291
Ex4DGS	19.80	20.59	23.24	13.52	24.81	20.39	0.060	0.089	0.038	0.313	0.036	0.107	0.145	0.199	0.088	0.441	0.082	0.191
FTGS	27.61	30.18	25.22	24.63	32.40	28.01	0.015	0.010	0.013	0.066	0.008	0.022	0.061	0.032	0.059	0.195	0.021	0.074
Ours (GraphiGS)	25.03	31.29	29.74	25.31	32.85	28.84	0.018	0.009	0.009	0.045	0.008	0.018	0.059	0.026	0.039	0.183	0.017	0.065
Ours (GraphiTS)	27.52	31.81	29.57	25.67	30.85	29.08	0.016	0.008	0.008	0.043	0.008	0.017	0.060	0.025	0.038	0.140	0.018	0.056

velocity and acceleration. As the derivative order increases, the magnitudes progressively decay. By the fourth order (snap), the response in moving regions is substantially attenuated, appearing as darker areas in the visualization. This systematic decay suggests that the dominant motion dynamics in the data are well captured by the first four derivatives. Consequently, extending the model beyond the fourth order is unlikely to yield meaningful improvements in reconstruction quality, while incurring additional computational and storage costs.

1.4 Training and Storage Requirements

We report the training time, storage requirements, training memory usage, and inference FPS of GraphiXS, and compare them against baseline methods in Tab. 12. All training times are measured on an NVIDIA RTX 4090 GPU. GraphiXS has slightly higher training cost primarily due to the component confidence computation, which requires multi-view rendering of the scene at each iteration. Overall, this is expected as the introduction of stochasticity into various steps of 4DGS requires more computation for the optimization. We argue the extra time cost is meaningful to enable the model to accommodate different types of data uncertainty. For inference, GraphiXS achieves a mid-range FPS among the baselines, primarily due to the increased complexity of its motion modeling.

Table 7. Quantitative comparison on the Google Immersive Dataset with 10%, 30%, and 50% spatial sparsity.

Method	PSNR ↑						DSSIM ↓						LPIPS ↓					
	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.
10% spatial Sparsity																		
4DGS-1	24.08	29.57	26.60	23.78	29.42	26.69	0.040	0.013	0.020	0.064	0.018	0.031	0.183	0.049	0.127	0.276	0.105	0.148
4DGS-2	23.69	27.04	20.68	18.25	29.14	23.76	0.069	0.031	0.081	0.251	0.024	0.091	0.324	0.160	0.321	0.559	0.127	0.298
Ex4DGS	23.20	21.10	25.12	13.35	28.24	22.20	0.038	0.070	0.022	0.317	0.020	0.093	0.104	0.168	0.062	0.466	0.038	0.168
FTGS	27.45	30.08	26.73	22.54	32.58	27.88	0.016	0.011	0.017	0.104	0.008	0.031	0.061	0.030	0.102	0.257	0.018	0.094
Ours (GraphiGS)	25.81	29.85	29.69	27.52	31.54	28.88	0.020	0.011	0.009	0.031	0.008	0.016	0.077	0.028	0.036	0.128	0.018	0.057
Ours (GraphiTS)	26.53	30.09	29.45	28.21	32.67	29.39	0.017	0.010	0.009	0.028	0.008	0.014	0.061	0.028	0.046	0.128	0.017	0.056
30% spatial Sparsity																		
4DGS-1	20.44	28.87	26.26	22.31	29.86	25.55	0.067	0.017	0.020	0.076	0.018	0.040	0.206	0.054	0.128	0.281	0.105	0.155
4DGS-2	22.11	25.09	15.84	18.01	25.78	21.37	0.073	0.059	0.171	0.200	0.061	0.113	0.339	0.266	0.417	0.420	0.292	0.347
Ex4DGS	19.64	25.08	17.60	13.56	23.55	19.49	0.076	0.040	0.088	0.293	0.057	0.111	0.166	0.103	0.161	0.437	0.116	0.197
FTGS	21.82	30.91	28.53	22.59	30.85	26.94	0.075	0.010	0.011	0.101	0.010	0.041	0.299	0.029	0.042	0.211	0.023	0.121
Ours (GraphiGS)	22.77	30.58	28.10	23.34	31.19	27.20	0.027	0.010	0.011	0.056	0.009	0.023	0.081	0.027	0.039	0.145	0.018	0.062
Ours (GraphiTS)	23.41	31.26	28.27	24.97	31.99	27.98	0.025	0.009	0.010	0.044	0.010	0.020	0.069	0.028	0.040	0.142	0.017	0.059
50% spatial Sparsity																		
4DGS-1	16.91	21.86	25.01	22.36	27.76	22.78	0.137	0.048	0.028	0.103	0.024	0.068	0.267	0.105	0.140	0.292	0.109	0.183
4DGS-2	20.12	20.09	18.03	16.10	27.35	20.34	0.127	0.062	0.128	0.259	0.029	0.121	0.362	0.186	0.376	0.493	0.099	0.303
Ex4DGS	18.45	16.96	16.98	12.69	18.83	16.78	0.110	0.162	0.114	0.317	0.140	0.169	0.192	0.282	0.204	0.467	0.225	0.274
FTGS	20.85	27.81	22.75	19.34	30.43	24.24	0.049	0.024	0.032	0.175	0.014	0.059	0.099	0.113	0.149	0.239	0.045	0.129
Ours (GraphiGS)	23.07	28.89	24.99	19.32	27.70	24.79	0.034	0.019	0.018	0.149	0.015	0.047	0.081	0.090	0.056	0.229	0.031	0.097
Ours (GraphiTS)	21.91	29.12	25.60	18.75	29.67	25.01	0.047	0.017	0.016	0.145	0.015	0.048	0.171	0.074	0.073	0.216	0.047	0.116

Table 8. Quantitative comparison on the Google Immersive Dataset with temporal sparsity where the 30 FPS videos are trained with 20 FPS and 10 FPS.

Method	PSNR ↑						DSSIM ↓						LPIPS ↓					
	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.
20 FPS																		
4DGS-1	22.90	29.13	25.85	23.15	29.22	26.05	0.053	0.018	0.027	0.074	0.025	0.039	0.206	0.064	0.144	0.299	0.126	0.167
4DGS-2	23.60	25.50	19.18	20.29	27.29	23.17	0.070	0.043	0.076	0.168	0.036	0.078	0.317	0.215	0.314	0.399	0.177	0.284
Ex4DGS	21.50	24.12	25.83	13.15	20.60	21.04	0.056	0.048	0.020	0.295	0.053	0.095	0.132	0.124	0.056	0.467	0.101	0.176
FTGS	25.80	30.35	27.22	24.63	30.94	27.79	0.024	0.010	0.014	0.065	0.009	0.025	0.100	0.030	0.078	0.204	0.022	0.087
Ours (Gaussian)	27.30	30.22	29.10	24.55	32.21	28.68	0.018	0.011	0.010	0.072	0.008	0.024	0.063	0.035	0.044	0.178	0.020	0.068
Ours (Student's T)	26.52	30.12	28.75	24.31	32.01	28.34	0.020	0.010	0.011	0.064	0.009	0.023	0.067	0.028	0.057	0.168	0.020	0.068
10 FPS																		
4DGS-1	22.61	27.88	23.63	23.93	27.27	25.06	0.079	0.030	0.066	0.085	0.048	0.062	0.231	0.100	0.196	0.299	0.167	0.199
4DGS-2	21.27	25.33	18.33	19.21	31.02	23.03	0.114	0.047	0.115	0.190	0.016	0.096	0.443	0.208	0.374	0.381	0.051	0.291
Ex4DGS	21.36	25.09	24.98	13.71	18.63	20.75	0.060	0.039	0.025	0.321	0.090	0.107	0.137	0.111	0.068	0.466	0.166	0.190
FTGS	24.33	28.25	25.51	22.82	29.95	26.17	0.043	0.017	0.026	0.103	0.017	0.041	0.134	0.045	0.112	0.248	0.054	0.119
Ours (GraphiGS)	24.87	28.32	27.33	24.29	29.71	26.90	0.044	0.017	0.023	0.072	0.019	0.035	0.120	0.046	0.079	0.213	0.068	0.105
Ours (GraphiTS)	24.71	27.93	26.64	24.96	29.54	26.76	0.044	0.018	0.026	0.062	0.020	0.034	0.120	0.047	0.081	0.203	0.056	0.101

Table 9. Quantitative comparison on the Google Immersive Dataset with non-synchronous temporal sparsity where 10% and 50% of the views are observed at 20 FPS.

Method	PSNR ↑						DSSIM ↓						LPIPS ↓					
	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.
10% of the views observed at 20 FPS																		
4DGS-1	23.60	29.36	26.37	24.60	29.35	26.66	0.043	0.015	0.022	0.065	0.019	0.033	0.193	0.054	0.130	0.289	0.108	0.155
4DGS-2	24.01	26.23	19.28	19.88	24.04	22.69	0.057	0.049	0.075	0.165	0.098	0.089	0.271	0.230	0.297	0.419	0.207	0.285
Ex4DGS	24.10	23.94	21.77	13.09	27.39	22.06	0.031	0.056	0.025	0.333	0.025	0.094	0.099	0.128	0.070	0.502	0.062	0.172
FTGS	25.38	31.36	26.51	22.16	32.04	27.49	0.020	0.009	0.013	0.094	0.009	0.029	0.064	0.026	0.059	0.223	0.023	0.079
Ours (GraphiGS)	27.30	30.89	29.48	26.01	31.60	29.06	0.015	0.009	0.008	0.033	0.008	0.015	0.056	0.029	0.037	0.137	0.019	0.055
Ours (GraphiTS)	24.66	31.05	29.63	26.81	32.12	28.85	0.026	0.009	0.009	0.034	0.008	0.017	0.101	0.026	0.042	0.154	0.022	0.069
50% of the views observed at 20 FPS																		
4DGS-1	24.80	25.25	26.75	24.58	29.69	26.21	0.039	0.021	0.020	0.062	0.019	0.032	0.189	0.065	0.130	0.275	0.110	0.154
4DGS-2	20.30	24.75	21.17	19.60	29.18	23.00	0.150	0.048	0.060	0.172	0.024	0.091	0.510	0.228	0.265	0.477	0.104	0.317
Ex4DGS	24.34	24.66	24.26	13.48	26.70	22.69	0.031	0.042	0.027	0.316	0.029	0.089	0.102	0.107	0.070	0.462	0.054	0.159
FTGS	25.20	30.69	29.34	22.92	32.16	28.06	0.022	0.010	0.011	0.085	0.008	0.027	0.079	0.030	0.055	0.233	0.017	0.083
Ours (GraphiGS)	25.97	30.98	29.08	27.45	30.97	28.89	0.019	0.009	0.009	0.031	0.009	0.015	0.077	0.026	0.043	0.143	0.022	0.062
Ours (GraphiTS)	26.21	30.83	28.79	25.13	31.03	28.40	0.018	0.010	0.009	0.060	0.010	0.021	0.066	0.030	0.041	0.159	0.032	0.066

In terms of storage, GraphiXS exhibits a moderate footprint. Compared to FTGS, the increased storage consumption is mainly caused by the additional parameters introduced by the higher-order motion model and the per-component confidence scores. Among the baselines, 4DGS-2 is the most storage-efficient, as it maintains only a canonical set of components and employs an MLP-based deformation model to generate time-dependent components. However, this storage efficiency comes at the cost of reduced visual quality compared to the other methods. At the opposite end of the spectrum, 4DGS-1 represents scene motion using a large number of very short-lived Gaussians. These components are limited to zeroth-order dynamics, resulting in minimal temporal sharing compared to Ex4DGS, FTGS,

Table 10. Quantitative comparison on the Google Immersive Dataset with random camera failures.

Method	PSNR ↑						DSSIM ↓						LPIPS ↓					
	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.	Welder	Flames	Horse	Goats	Alexa	Avg.
Faulty Cam 1																		
4DGS-1	23.26	27.91	25.87	23.37	30.05	26.09	0.046	0.020	0.023	0.070	0.018	0.035	0.198	0.060	0.137	0.286	0.110	0.158
4DGS-2	19.54	23.60	18.93	17.50	29.17	21.75	0.142	0.062	0.099	0.206	0.023	0.106	0.455	0.245	0.327	0.452	0.103	0.316
Ex4DGS	21.31	22.12	20.64	13.56	26.44	20.81	0.051	0.075	0.053	0.302	0.032	0.103	0.125	0.178	0.113	0.446	0.068	0.186
FTGS	26.33	30.75	28.74	23.90	31.55	28.25	0.019	0.010	0.011	0.075	0.009	0.025	0.069	0.040	0.047	0.203	0.021	0.076
Ours (GraphiGS)	25.64	31.10	28.79	26.22	30.68	28.49	0.020	0.009	0.010	0.034	0.009	0.016	0.063	0.026	0.038	0.133	0.021	0.056
Ours (GraphiTS)	24.99	30.29	27.57	22.83	31.19	27.37	0.024	0.009	0.011	0.097	0.009	0.030	0.092	0.029	0.045	0.222	0.024	0.072
Faulty Cam 2																		
4DGS-1	23.67	29.90	25.71	24.07	27.74	26.22	0.049	0.015	0.025	0.074	0.026	0.038	0.208	0.057	0.147	0.290	0.129	0.166
4DGS-2	20.91	25.71	21.12	17.14	28.36	22.65	0.111	0.038	0.071	0.208	0.024	0.090	0.389	0.162	0.304	0.466	0.071	0.278
Ex4DGS	22.56	23.83	23.61	12.52	27.26	21.96	0.046	0.061	0.043	0.332	0.038	0.104	0.125	0.134	0.091	0.491	0.070	0.182
FTGS	24.90	30.34	26.27	22.18	28.71	26.48	0.031	0.011	0.015	0.104	0.011	0.034	0.132	0.031	0.045	0.234	0.025	0.093
Ours (GraphiGS)	24.05	30.86	28.32	23.57	30.53	27.47	0.030	0.010	0.012	0.072	0.010	0.027	0.111	0.028	0.042	0.178	0.023	0.076
Ours (GraphiTS)	24.83	30.77	28.07	22.77	31.46	27.58	0.025	0.009	0.011	0.093	0.010	0.030	0.072	0.028	0.042	0.193	0.023	0.072
Faulty Cam 3																		
4DGS-1	16.54	21.59	18.48	17.83	22.03	19.29	0.183	0.080	0.126	0.191	0.100	0.136	0.374	0.185	0.311	0.413	0.289	0.314
4DGS-2	13.83	14.61	15.46	13.63	17.12	14.93	0.217	0.163	0.168	0.301	0.160	0.202	0.506	0.442	0.447	0.601	0.456	0.490
Ex4DGS	16.83	17.26	18.18	12.26	17.37	16.38	0.157	0.180	0.103	0.345	0.230	0.203	0.292	0.374	0.193	0.551	0.387	0.359
FTGS	18.57	26.40	19.43	15.56	22.50	20.49	0.120	0.031	0.094	0.256	0.075	0.115	0.252	0.072	0.199	0.449	0.166	0.228
Ours (GraphiGS)	18.68	26.57	20.13	14.43	23.44	20.65	0.133	0.033	0.078	0.281	0.063	0.118	0.382	0.080	0.211	0.420	0.157	0.250
Ours (GraphiTS)	18.73	26.31	20.12	14.69	24.26	20.82	0.128	0.033	0.083	0.286	0.063	0.119	0.374	0.078	0.185	0.423	0.204	0.253

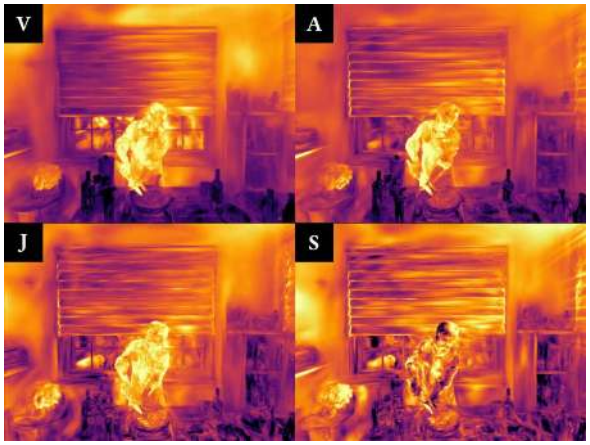

Fig. 1. Magnitude of motion derivatives. Each image is independently normalized. V, A, J, and S denote the magnitudes of velocity, acceleration, jerk, and snap, respectively.

Table 11. Quantitative comparison on the CityCrowd dataset (scene: city_01) under the Standard and 50% Spatial Sparsity settings.

Method	Standard			50% Spatial Sparsity		
	PSNR↑	DSSIM↓	LPIPS↓	PSNR↑	DSSIM↓	LPIPS↓
4DGS-1	27.36	0.028	0.100	24.33	0.040	0.112
4DGS-2	15.71	0.276	0.512	15.17	0.275	0.504
Ex4DGS	19.34	0.149	0.301	17.36	0.243	0.374
FTGS	28.09	0.019	0.068	25.96	0.029	0.083
Ours (GraphiGS)	28.02	0.017	0.061	25.97	0.029	0.076
Ours (GraphiTS)	28.17	0.016	0.057	26.42	0.028	0.076

Table 12. Average performance comparison on N3DV dataset [Li et al. 2022] using standard setting.

Method	Storage Size↓	Training Time↓	Training Memory↓	Inference FPS↑
4DGS-1	3128 MB	1.5 Hours	25814 MB	120 FPS
4DGS-2	90 MB	40 Minutes	15428 MB	34 FPS
Ex4DGS	115 MB	1 Hour	3759 MB	46 FPS
FTGS	366 MB	2.5 Hours	3751 MB	126 FPS
Ours (GraphiGS)	423 MB	4.5 Hours	4701 MB	113 FPS
Ours (GraphiTS)	429 MB	4.5 Hours	6471 MB	89 FPS

and GraphiXS. GraphiXS occupies the middle ground in this trade-off space, leveraging short-lived yet temporally shared components to balance storage efficiency and reconstruction quality.

2 Additional Details of Methodology

This section provides additional details about GraphiXS and hyperparameters used in the trainings.

2.1 Variants of GraphiXS

As established in our generative formulation, GraphiXS is a generalized framework in which the component δ_θ is modeled as a probabilistic distribution parameterized by θ, rather than being restricted to a fixed geometric primitive. This design enables flexible instantiations of the framework using different distribution families.

GraphiGS. As the first variant of GraphiXS we initialize δ_θ using 3D Gaussian primitives and refer this version as GraphiGS. In this variant, the component δ is defined as a Gaussian distribution parameterized by a mean $\boldsymbol{\mu} \in \mathbb{R}^3$ and a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$. This formulation is consistent with the standard 3D Gaussian Splatting paradigm, where density decays exponentially with distance from the mean. Formally, the distribution for a single component in GraphiGS is given by

$$\delta_{\mathrm{GS}}(\mathbf{x};\boldsymbol{\mu},\Sigma) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{1}$$

GraphiTS. While Gaussian primitives provide compact spatial support, they may be limited in capturing broader uncertainty regions. To address this, we introduce a second variant, GraphiTS, which instantiates δ using Student's t distribution. Following the formulation in 3DSSS [Zhu et al. 2025], we augment the parameterization with a learnable degrees-of-freedom parameter $\nu \in [1, +\infty)$, in addition to $\boldsymbol{\mu}$ and Σ. The parameter ν explicitly controls the tail heaviness of the distribution, allowing GraphiTS to smoothly interpolate between

heavy-tailed Cauchy-like behavior and Gaussian-like behavior. The corresponding distribution is defined as

$$\delta_{\mathrm{TS}}(\mathbf{x};\boldsymbol{\mu},\Sigma,\nu)=\left[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]^{-\frac{\nu+3}{2}}. \tag{2}$$

More Variants. Beyond the two variants described above, the probabilistic formulation of GraphiXS naturally supports alternative distribution families for the component δ, as long as they can be parameterized by the shared latent variables θ. This flexibility enables the incorporation of distributions with different support characteristics and uncertainty profiles. Beta distribution based primitives, as introduced in [Liu et al. 2025a,b], may serve as a promising candidate for a future GraphiBS variant.

2.2 Prior on Color

To capture the view-dependent appearance of the moving components, we condition the color s on $\mu(t)$ by [Wang et al. 2025]:

$$s=\sum_{l=0}^{L}\sum_{m=-l}^{l} sh_{lm}\,Y_{lm}\big(D(\mu(t))\big) \tag{3}$$

where s is the time and view dependent color, L is the degree of the spherical harmonics, D is the viewing direction at position $\mu(t)$, and Y is the spherical harmonics basis function evaluated at the viewing direction $D(\mu(t))$. By explicitly conditioning the color evaluation on the dynamic position $\mu(t)$, we ensure that the view-dependent effects remain consistent with the component's learned trajectory and higher-order dynamics.

2.3 Sampling via Stochastic Gradient Hamiltonian Monte Carlo

To optimize the posterior distribution defined in the main manuscript, we adopt Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [Zhu et al. 2025]. SGHMC is applied jointly to all learnable component parameters θ, enabling efficient exploration of the posterior. This choice is motivated by the highly non-convex solution space induced by our objective, where parameters governing shape, motion, and appearance are tightly coupled.

2.4 Adding and Recycling Components in 4D

During optimization, components with near-zero opacity contribute negligibly to the rendered scene while unnecessarily consuming the component budget. We identify such components using an opacity threshold and mark them as unused. Unused components are recycled regardless of budget, while new components are added only when the active set falls below the budget. We regulate both component addition and recycling using a scalar importance score

$$score_{\delta}^{t}=\lambda_{sg},\bar{g}_{\delta}^{t}+\lambda so,o_{\delta}^{t}, \tag{4}$$

where $\bar{g}_{\delta}^{t}$ denotes the accumulated positional gradient of components δ at time t, o_{δ}^{t} its opacity, and λsg and λ_{so} are weighting coefficients. Active components are sampled according to the resulting importance distribution

$$\tau_{\delta}^{t}=\frac{score_{\delta}^{t}}{\sum_{\delta}^{\mathbf{pt}} score_{\delta}^{t}+\epsilon}. \tag{5}$$

The sampled components guide both the placement of newly added components and the reassignment of recycled ones. As a result, component capacity is dynamically focused on regions that most strongly influence the 4D reconstruction.

2.5 Initialization

Following [Wang et al. 2025], we initialize our model from temporally coherent 4D point clouds reconstructed from multi-view image sequences. For each video frame, dense 3D point clouds are reconstructed independently by matching keypoints across views using RoMa [Edstedt et al. 2024] and triangulating the resulting correspondences. These per-frame point clouds are subsequently merged into a unified 4D point set by establishing temporal correspondences via k-nearest-neighbor matching between points in successive frames. The temporal coordinate of each point is initialized to the corresponding video timestamp, while point velocities are derived from inter-frame displacements along the temporal correspondences. Higher-order motion derivatives are initialized using a temporal binning scheme. For each derivative order, we compute deviations from the preceding order within fixed temporal bins and assign the resulting statistics to the corresponding points. This procedure yields bounded and physically plausible motion initialization.

2.6 Hyperparameters

We implement our framework using PyTorch and CUDA. Optimization of the component parameters θ is performed using SGHMC with a friction coefficient of $C = 120$ and a noise scale of 1.0. To stabilize early training, we employ a burn-in phase of 7,000 iterations, during which the friction coefficient is increased to $C_{\text{burn-in}} = 5 \times 10^5$ in order to suppress excessive oscillations. For higher-order dynamics, the standard deviation of the Brownian motion noise is set to $\epsilon = 1 \times 10^{-4}$.

The learning rates for dynamics-related parameters—velocity v, acceleration a, jerk j, snap s, and component duration u—are exponentially annealed from an initial value of 1×10^{-2} to a final value of 1×10^{-4} over the course of training.

We assign fixed weights to the loss terms and regularization priors. The component confidence regularization weight λ_{α} for $\mathcal{L}_{\alpha}$ is set to 1×10^{-3}. The temporal opacity regularization weight is set to 0.01, and the scale regularization weight associated with ϵ_{Σ} is also set to 0.01. For the smoothness prior on the dynamics derivatives, the weights for velocity, acceleration, jerk, and snap are set uniformly to $\lambda_h = 1 \times 10^{-4}$. For importance sampling in component addition and recycling, we assign equal weights to the gradient and opacity terms, with $\lambda_{sg} = 0.5$ and $\lambda_{so} = 0.5$, respectively. Across all GraphiXS experiments, we cap the total number of components at 1.5 million.

2.7 GraphiXS as an Upgrade

GraphiXS is a generalized probabilistic framework that can be applied to a wide range of 4D Gaussian Splatting (4DGS) representations. In contrast, most existing 4DGS methods formulate reconstruction as a deterministic optimization of photometric error:

$$\min\ \mathcal{L}_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}. \tag{6}$$

To upgrade a baseline 4DGS method using GraphiXS, we reformulate its training objective as a Maximum a Posteriori (MAP) estimation

problem, as defined in the main paper. This reformulation involves two main adaptations: reinterpreting the likelihood and incorporating stochastic priors.

General Adaptation Strategy. First, the baseline rendering loss is converted into an energy-based likelihood distribution. Our energy-based likelihood model in the main paper can be directly used, as it includes Equation (6):

$$P(I \mid \beta, R, \alpha, C, T) \propto exp(-\sum_{c}^{M}\sum_{t}^{K} L_{img}(I_c^t))$$
$$L_{img} = (1-\epsilon_{D-SSIM})L_1 + \epsilon_{D-SSIM}L_{D-SSIM} + \epsilon_o \sum_i ||o_i||_1 + \epsilon_\Sigma \sum_i \sum_j ||\sqrt{\lambda_{i,j}}||_1 \quad (7)$$

where L_1 and L_{D-SSIM} are the L_1 norm and the structural similarity loss between the reconstructed image and the ground-truth. λs are the eigenvalues of Σ. The regularization applied to the opacity ensures that the opacity is big only when a component is absolutely needed. The regularization on λ ensures the model uses components as spiky as possible (i.e. small variances). Together, the regularization terms minimize the needed number of components.

Second, we introduce the *Component Confidence* term,

$$P(\alpha \mid \delta_\theta, C, T), \quad (8)$$

which is incorporated as a regularization term $\mathcal{L}_\alpha$. This term encourages the model to account for data uncertainty by maximizing the joint probability of component visibility across all available views and timestamps per batch. Regardless the formulation and the implementation of the base methods, this component confidence term can be computed.

Finally, the shape prior $P(\Sigma_t^p)$ can also be applied to ensure consistency of primitive covariances over time. This term is also general as long as the component is a probabilistic distribution, as the covariance matrix is a common parameter.

Methods such as FTGS [Wang et al. 2025], Ex4DGS [Lee et al. 2024], and 4DGS-1 [Yang et al. 2024] use explicit 3D/4D Gaussians or similar primitives that move dynamically, allowing the above adaptation strategy to be applied directly. As an example in the paper, we upgraded FTGS. By contrast, methods like 4DGS-2 [Wu et al. 2024], which rely on a canonical Gaussian set deformed via a neural field (*e.g.*, HexPlane or MLP), cannot be upgraded directly. In these cases, GraphiXS priors can be applied to the displaced components at each sampled timestamp, with gradients backpropagated through the deformation network.

2.8 Deterministic Variant of GraphiXS

In our ablation study, we introduce a deterministic variant of GraphiXS to validate the effectiveness of our framework. In this variant, we replace the SGHMC optimizer with standard Adam, thereby removing Hamiltonian Monte Carlo noise from gradient updates. We also disable the component confidence loss, eliminating the incentive to maintain visibility across multiple viewpoints, and remove Brownian motion noise during rasterization. Furthermore, stochastic multinomial sampling for component addition and recycling during densification is replaced with a deterministic strategy that selects the top-N components based on the same scoring criteria.

References

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. 39, 4 (2020), 86:1–86:15.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. 2024. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19790–19800.

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

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

Rong Liu, Zhongpai Gao, Benjamin Planche, Meida Chen, Van Nguyen Nguyen, Meng Zheng, Anwesa Choudhuri, Terrence Chen, Yue Wang, Andrew Feng, et al. 2025a. Universal Beta Splatting. *arXiv preprint arXiv:2510.03312* (2025).

Rong Liu, Dylan Sun, Meida Chen, Yue Wang, and Andrew Feng. 2025b. Deformable beta splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11.

Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. 2025. FreeTimeGS: Free Gaussian Primitives at Anytime Anywhere for Dynamic Scene Reconstruction. In *CVPR*. https://zju3dv.github.io/freetimegs

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 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.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. 2024. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In *International Conference on Learning Representations (ICLR)*.

Jialin Zhu, Jiangbei Yue, Feixiang He, and He Wang. 2025. 3D Student Splatting and Scooping. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21045–21054.

Fig. 2. Qualitative comparison on N3DV dataset. Each row corresponds to a different experimental setting. Columns show six competing methods and the ground truth (GT). Per-region PSNR values are reported in the top-right corner of each cropped region.

Fig. 3. Qualitative comparison on N3DV dataset. Each row corresponds to a different experimental setting. Columns show six competing methods and the ground truth (GT). Per-region PSNR values are reported in the top-right corner of each cropped region.

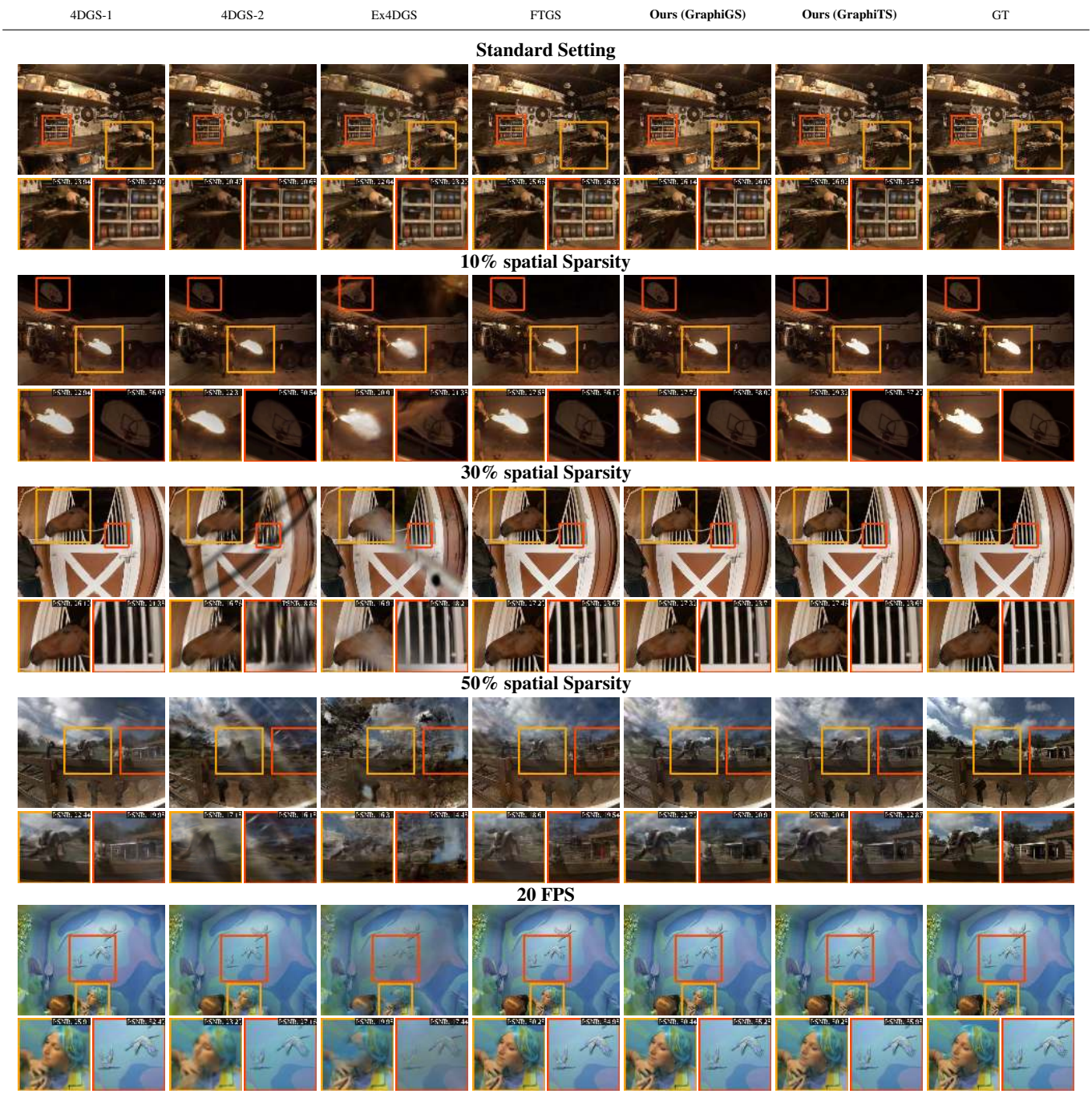

Fig. 4. Qualitative comparison on GI dataset. Each row corresponds to a different experimental setting. Columns show six competing methods and the ground truth (GT). Per-region PSNR values are reported in the top-right corner of each cropped region.

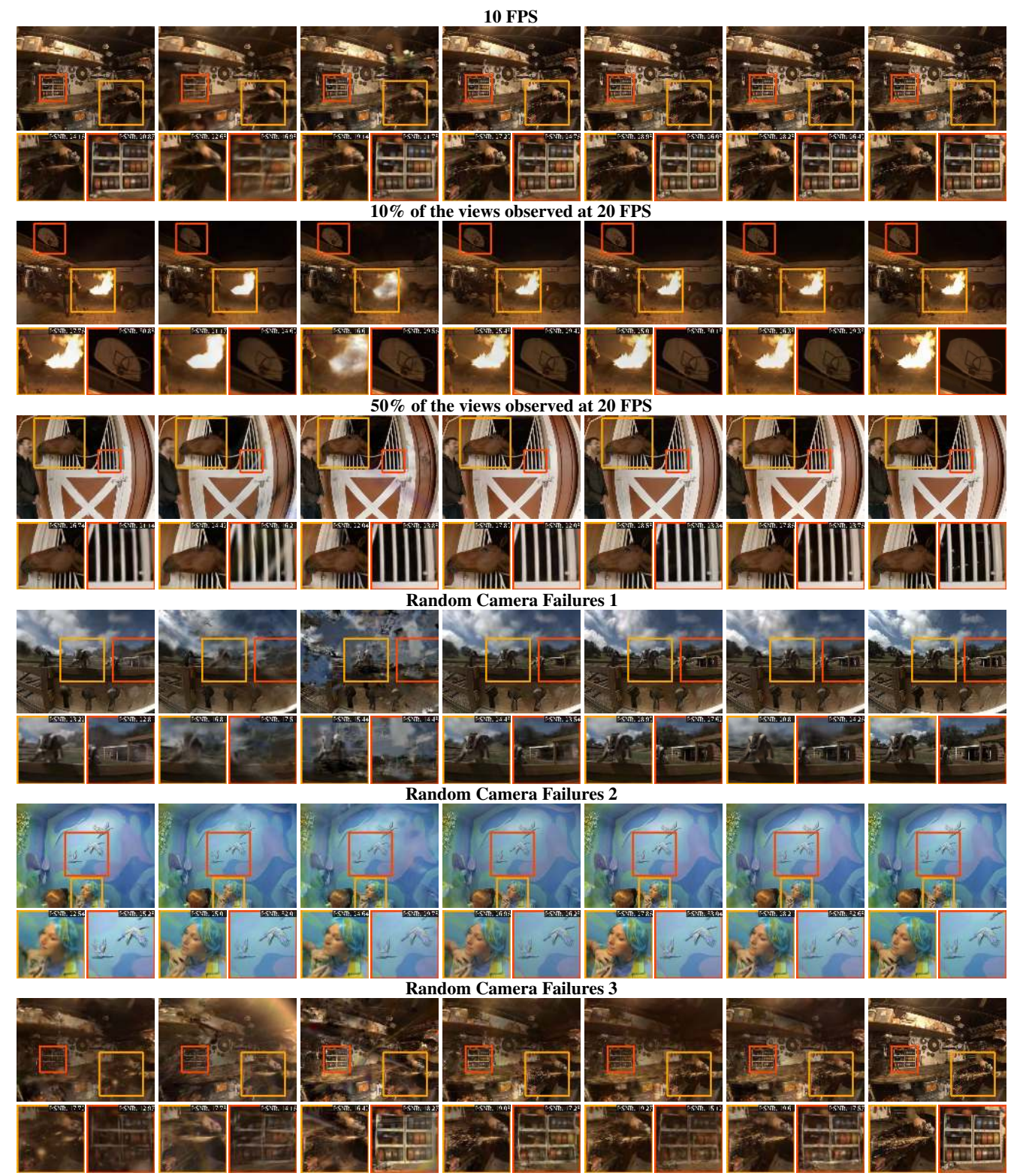

Fig. 5. Qualitative comparison on GI dataset. Each row corresponds to a different experimental setting. Columns show six competing methods and the ground truth (GT). Per-region PSNR values are reported in the top-right corner of each cropped region.

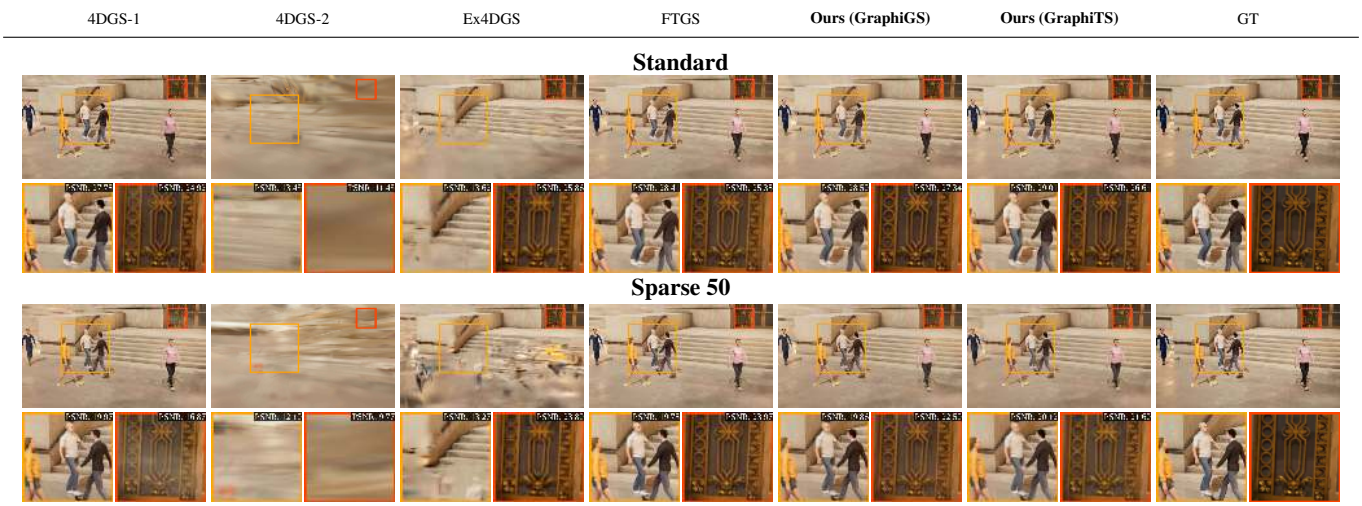

Fig. 6. Qualitative comparison on our CityCrowd dataset. Each row corresponds to a different experimental setting. Columns show six competing methods and the ground truth (GT). Per-region PSNR values are reported in the top-right corner of each cropped region.