GP-4DGS: Probabilistic 4D Gaussian Splatting from Monocular Video via Variational Gaussian Processes

Mijeong Kim[1] Jungtaek Kim[3] Bohyung Han[1,2]
[1]ECE and [2]IPAI, Seoul National University, Korea [3]University of Wisconsin–Madison, USA
{mijeong.kim, bhhan}@snu.ac.kr jungtaek.kim@wisc.edu

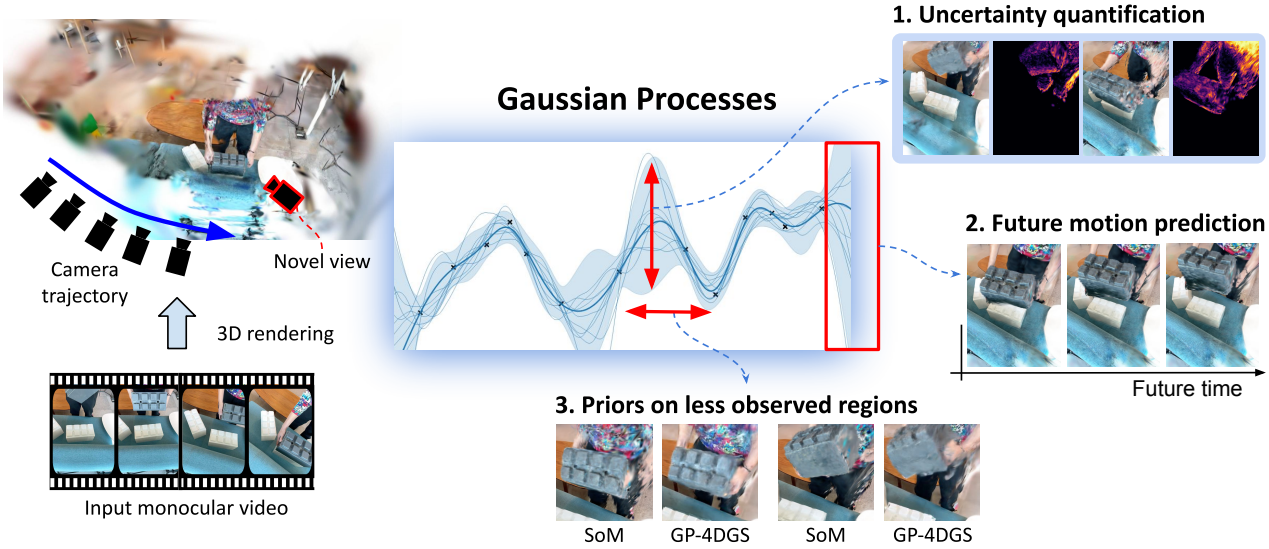


Figure 1. We propose GP-4DGS, a novel integration of Gaussian Processes (GPs) [37] into 4D Gaussian Splatting (4DGS). Unlike existing deterministic approaches, this formulation enables robust uncertainty quantification, future motion prediction, and prior estimation for unobserved regions.

Abstract

We present GP-4DGS, a novel framework that integrates Gaussian Processes (GPs) into 4D Gaussian Splatting (4DGS) for principled probabilistic modeling of dynamic scenes. While existing 4DGS methods focus on deterministic reconstruction, they are inherently limited in capturing motion ambiguity and lack mechanisms to assess prediction reliability. By leveraging the kernel-based probabilistic nature of GPs, our approach introduces three key capabilities: (i) uncertainty quantification for motion predictions, (ii) motion estimation for unobserved or sparsely sampled regions, and (iii) temporal extrapolation beyond observed training frames. To scale GPs to the large number of Gaussian primitives in 4DGS, we design spatio-temporal kernels that capture the correlation structure of deformation fields and adopt variational Gaussian Processes with inducing points for tractable inference. Our experiments show that GP-4DGS enhances reconstruction quality while providing reliable uncertainty estimates that effectively identify regions of high motion ambiguity. By addressing these challenges, our work takes a meaningful step toward bridging probabilistic modeling and neural graphics.

1. Introduction

Dynamic scene reconstruction from monocular videos has recently become an important problem due to its significant practical impact, enabling 3D capture in unconstrained environments for robotics [28, 39, 52], autonomous systems [31, 35, 50, 54], and large reconstruction models [13, 53]. Building on recent advances in static scene reconstruction [3, 9, 16, 18, 30, 32], this field has rapidly evolved by extending static-scene base architectures with additional temporal modeling to handle scene dynamics.

The dominant paradigm in recent years is 4D Gaussian Splatting (4DGS) [19, 23, 38, 46–49], which extends 3D Gaussian Splatting (3DGS) to dynamic scenes. These methods have achieved impressive visual quality in dynamic scene reconstruction task by representing scenes as collections of time-varying Gaussian primitives, optimized end-to-end through differentiable rendering. However, these approaches treat motion as a deterministic optimization problem, imposing hand-crafted motion priors, *e.g.*, polynomial deformations [23], rigidity constraints [2], or other fixed parametrizations [20, 46], that are applied uniformly across all primitives without learning from data. When primitives are poorly observed or occluded, these fixed priors become inappropriate or overly restrictive. Moreover, existing methods lack principled mechanisms for uncertainty estimation and motion extrapolation beyond training frames.

In this work, we propose GP-4DGS, a novel integration of Gaussian Processes (GPs) [37] into 4DGS that provides a principled probabilistic framework for dynamic scene reconstruction, as shown in Figure 1. This simultaneously enables three capabilities absent in existing 4DGS algorithms: (i) uncertainty quantification for motion predictions, (ii) temporal extrapolation beyond observed frames, and (iii) learned motion priors that adapt to observation patterns. Rather than imposing fixed motion constraints, we learn motion priors directly from well-observed primitives via kernel-based probabilistic modeling. The key insight is to represent primitive deformations through GP posteriors, enabling the framework to automatically adjust regularization strength based on observation confidence and to naturally handle sparse or unobserved regions. Notably, the first two capabilities emerge directly from the probabilistic formulation of GPs, without any additional modeling overhead.

This integration, however, is non-trivial and requires addressing fundamental modeling and computational challenges. First, standard GP kernels assume isotropic correlations, fundamentally mismatched to spatio-temporal data where spatial dimensions (x, y, z) and time t have drastically different correlation structures. To address this, we introduce a composite kernel design that explicitly separates spatial Matérn kernels for geometric smoothness from periodic temporal kernels for motion periodicity. Second, exact GP posterior computation has $\mathcal{O}(N^3)$ time complexity where N is the number of Gaussian primitives, which is prohibitive as N typically reaches tens of thousands. We develop a scalable formulation combining variational inference with inducing points, reducing complexity to $\mathcal{O}(NM^2+M^3)$, where M is the number of inducing points with $M \ll N$. Third, integrating probabilistic priors with vision-based optimization requires careful design. We propose a novel GP-GS optimization algorithm that forms synergy between the two by alternating between confidence-weighted GP training on well-observed data and GP-guided regularization during appearance optimization. This creates a self-reinforcing loop where reliable observations progressively refine motion priors, which reciprocally stabilize reconstruction in poorly observed regions. Our main contributions are summarized as follows:

- We introduce a probabilistic framework that integrates GPs with 4DGS, enabling uncertainty quantification for motion predictions, temporal extrapolation for future motion prediction, and observation-adaptive motion priors.
- We develop a spatio-temporal GP kernel, a scalable variational inference scheme, and a GP-GS dual optimization strategy for principled probabilistic dynamic scene reconstruction.
- We demonstrate that our method improves dynamic scene reconstruction, especially in occluded or sparsely observed regions, while providing reliable motion extrapolation and uncertainty estimates.

2. Related Work

2.1. Dynamic Novel View Synthesis

Novel View Synthesis (NVS) is a form of inverse graphics that reconstructs a 3D scene from observed 2D images and renders novel images at arbitrary viewpoints. Dynamic Novel View Synthesis (DyNVS) extends this paradigm to 4D dynamic scenes, where the input is a video containing motion rather than static 2D images. Early progress was driven by variants of Neural Radiance Fields (NeRF) [4, 6, 7, 30, 33, 41], which implicitly represent 4D scenes via neural networks optimized through differentiable rendering. More recently, 4D Gaussian Splatting (4DGS) methods [5, 11, 15, 21, 23, 25–27, 29, 42, 45, 48, 51] have emerged as the state-of-the-art by explicitly deforming Gaussian primitives over time with a deformation field. For instance, D-3DGS [51] employs an MLP, 4DGS [48] uses a HexPlane representation, and STG [23] utilizes polynomial functions to model such deformations. Some works [21, 42] further incorporate priors from pretrained models for depth estimation and 2D tracking, yet these priors are mostly deterministic and confined to observed regions. In contrast, our method introduces a probabilistic framework that learns data-adaptive priors, enabling robust generalization even to unobserved regions.

2.2. Probabilistic Modeling in Gaussian Splattings

Gaussian Splatting [16] represents scenes as a collection of Gaussian primitives, which is inherently a deterministic representation. However, recent works [12, 17, 40, 44] have begun to reformulate Gaussian Splatting as a probabilistic framework to quantify uncertainty or improve optimization. For instance, Stochastic GS [40] models Gaussian attributes as random variables with learnable variances, enabling un-

certainty quantification. Kheradmand et al. [17] reformulate Gaussian primitives as samples drawn from a spatial distribution and adopt Stochastic Gradient Langevin Dynamics (SGLD) to refine densification and reduce sensitivity to initialization. VBGS [44] adopts a Bayesian-based Gaussian mixture model to solve the challenge of catastrophic forgetting in continual learning settings. GP-GS [12] integrates GP priors to improve Structure-from-Motion initialization by inferring missing regions via GP posteriors. However, these probabilistic approaches are confined to static scene representations. In contrast, our work is the first to integrate GP priors into 4D Gaussian Splatting, enabling probabilistic motion modeling for robust dynamic scene reconstruction.

3. Preliminaries

We briefly review 4D Gaussian Splatting and Gaussian Processes, which form the foundation of our method.

3.1. 4D Gaussian Splatting

Gaussian primitives 3DGS achieves high-quality real-time rendering of static scenes by employing an explicit 3D scene representation. This representation consists of a set of N 3D Gaussian primitives, denoted by $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$. Each Gaussian primitive γ_k is represented by an unnormalized 3D Gaussian kernel $\mathcal{G}_k(\mathbf{x}_s)$ as

$$\mathcal{G}_k(\mathbf{x}_s;\boldsymbol{p}_k,\boldsymbol{\Sigma}_k) = \exp\left(-\frac{1}{2}(\mathbf{x}_s-\boldsymbol{p}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_s-\boldsymbol{p}_k)\right), \quad (1)$$

where $\boldsymbol{p}_k \in \mathbb{R}^3$ is a mean vector, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times3}$ is an anisotropic covariance matrix, and $\mathbf{x}_s \in \mathbb{R}^3$ is an arbitrary location in 3D space. The covariance matrix $\boldsymbol{\Sigma}_k$ has to be positive semi-definite, which is challenging to hold during optimization. Instead, to ensure this condition, we learn $\boldsymbol{\Sigma}_k$ by decomposing it into two learnable components, a rotation matrix $\mathbf{R}_k$ and a diagonal scaling matrix $\mathbf{S}_k$ as follows:

$$\boldsymbol{\Sigma}_k = \mathbf{R}_k\mathbf{S}_k\mathbf{S}_k^\top\mathbf{R}_k^\top. \quad (2)$$

In addition to $\boldsymbol{p}_k, \mathbf{R}_k$, and $\mathbf{S}_k$, each Gaussian primitive requires two additional learnable parameters for its opacity $\alpha_k \in [0,1]$ and feature $\boldsymbol{f}_k$. The feature vector is typically represented by RGB colors or spherical harmonic coefficients for rendering view-dependent lighting and color effects. Consequently, a single Gaussian primitive γ_k is defined with its complete set of learnable parameters, $\{\boldsymbol{p}_k, \mathbf{R}_k, \mathbf{S}_k, \alpha_k, \boldsymbol{f}_k\}$.

Deformation modeling Dynamic scene modeling requires extending the 3D formulation to capture temporal variations. Most algorithms [15, 23, 46, 48, 51] deform the 3D Gaussian primitives from their canonical states to a target state over time. The transformed position $\boldsymbol{p}_{k,t}$ and rotation $\mathbf{R}_{k,t}$ at time t are given by

$$(\boldsymbol{p}_{k,t}, \mathbf{R}_{k,t}) = (\boldsymbol{p}_k + \phi_p(\boldsymbol{p}_k, \mathbf{R}_k, t), \mathbf{R}_k + \phi_r(\mathbf{R}_k, t)), \quad (3)$$

where $\phi_p(\cdot)$ and $\phi_r(\cdot)$ are the deformation operations.

Differentiable rasterization Before rendering with a set of deformed Gaussian primitives Γ on an image space, each deformed Gaussian kernel $\mathcal{G}_{k,t}(\mathbf{x}_s;\boldsymbol{p}_{k,t},\boldsymbol{\Sigma}_{k,t})$ is projected onto a 2D image space and forms a 2D Gaussian kernel $\mathcal{G}^\pi_{k,t}(\mathbf{r};\boldsymbol{p}^\pi_{k,t},\boldsymbol{\Sigma}^\pi_{k,t})$, where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ denotes a projection from a world coordinate to an image space and t is the target time. In the projected Gaussian representation, $\mathbf{r} \in \mathbb{R}^2$ indicates a pixel location in an image, and the 2D mean $\boldsymbol{p}^\pi_{k,t} \in \mathbb{R}^2$ and covariance $\boldsymbol{\Sigma}^\pi_{k,t} \in \mathbb{R}^{2\times2}$ are given by

$$\boldsymbol{p}^\pi_{k,t} = \pi(\boldsymbol{p}_{k,t}) \quad \text{and} \quad \boldsymbol{\Sigma}^\pi_{k,t} = \mathbf{JW}\boldsymbol{\Sigma}_{k,t}\mathbf{W}^\top\mathbf{J}^\top, \quad (4)$$

where $\mathbf{J}$ denotes the Jacobian of the affine approximation of the projective transformation, and $\mathbf{W}$ is the world-to-camera transform matrix. When rendering the primitives in Γ to a target camera, they are sorted by their depths with respect to the camera center. The color of a pixel $\mathbf{r}$ is then obtained by α-blending, which is given by

$$\hat{\mathbf{I}}(\mathbf{r}) = \sum_{k=1}^{N} c_k\, \omega^\pi_{k,t}(\mathbf{r}), \quad (5)$$

where $\omega^\pi_{k,t}(\mathbf{r})$ represents a relative contribution of each Gaussian primitive to pixel $\mathbf{r}$ at time t and c_k is the color of the corresponding primitive.

3.2. 1D Gaussian Processes

A Gaussian Process (GP) defines a probability distribution over functions, such that any finite collection of function values follows a joint Gaussian distribution as follows:

$$f(x) \sim \mathcal{GP}(m(x),\, k(x,x')), \quad (6)$$

where $x \in \mathbb{R}$ and $f(x) \in \mathbb{R}$ are both scalar-valued in the simplest case, $m(x)$ and $k(x,x')$ are the mean function and kernel function, respectively. Given observations $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ with $\mathbf{X} = \{x_n\}_{n=1}^N$ and $\mathbf{y} = \{y_n\}_{n=1}^N$, the kernel hyperparameters are optimized by maximizing the marginal likelihood to capture the correlation structure of the observed data.

For inference, the posterior distribution at a new query point x^* is Gaussian, $f(x^*) \sim \mathcal{N}(\bar{\mu}(x^*), \bar{\sigma}^2(x^*))$, conditioned on the observed data $\mathcal{D}$. The predictive mean and variance are respectively derived as

$$\bar{\mu}(x^*) = m(x^*) + \mathbf{k}_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}(\mathbf{y} - m(\mathbf{X})), \quad (7)$$

$$\bar{\sigma}^2(x^*) = k(x^*,x^*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_*, \quad (8)$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the covariance matrix with entries $[\mathbf{K}]_{ij} = k(x_i, x_j)$, and $\mathbf{k}_* \in \mathbb{R}^N$ denotes the vector of covariances between x^* and the training inputs. Intuitively, the predictive mean $\bar{\mu}(x^*)$ is a weighted combination of the observed residuals $(\mathbf{y} - m(\mathbf{X}))$ in output space, where the weights are determined by the kernel-defined similarity between the query point and the training data.

4. Method

We present GP-4DGS, a principled integration of GPs into 4DGS for monocular video reconstruction. Specifically, we design spatial and temporal kernels to capture geometric and motion correlations in primitive deformations in Section 4.1, adopt variational inference for scalable computation in Section 4.2, and introduce a GP-GS optimization strategy for joint training of GP and 4DGS in Section 4.3. This naturally provides uncertainty quantification and temporal extrapolation, capabilities absent in existing 4DGS frameworks, as described in Sections 4.4 and 4.5.

4.1. Motion Modeling with Gaussian Processes

We model the temporal deformation of Gaussian primitives with a multi-input multi-output GP, using a composite kernel with spatial and temporal components for canonical geometry and motion periodicity, respectively.

4.1.1. Probabilistic Deformation

Given a 4D input $\mathbf{x} = (\boldsymbol{p}, t) \in \mathbb{R}^4$, where $\boldsymbol{p} = (p_x, p_y, p_z)$ denotes the canonical 3D position of an arbitrary primitive and t represents the target time, our GP outputs a d-dimensional deformation vector $\mathbf{y} = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x})) \in \mathbb{R}^d$. We set $d = 9$ with three dimensions for translational deformation and six for the 6D continuous rotation representation [55] of Gaussian primitive orientations. Each output $f_i(\mathbf{x})$ is modeled as an independent GP with mean function $m_i(\mathbf{x})$ and kernel function $k_i(\mathbf{x}, \mathbf{x}')$ as follows:

$$f_i(\mathbf{x}) \sim \mathcal{GP}(m_i(\mathbf{x}), k_i(\mathbf{x}, \mathbf{x}')). \tag{9}$$

This defines a probabilistic distribution over functions, where a prediction at any input $\mathbf{x}$ is Gaussian-distributed, $f_i(\mathbf{x}) \sim \mathcal{N}(\bar{\mu}_i(\mathbf{x}), \bar{\sigma}_i^2(\mathbf{x}))$, with both mean and variance derived from the GP posterior.

4.1.2. Kernel Modeling

A kernel defines the correlation structure between GP inputs, determining how the model generalizes from observed to unobserved regions. While standard GP kernels assume isotropic correlations across all input dimensions, our input space exhibits fundamentally different correlation characteristics between the spatial dimensions $\boldsymbol{p}$ and the temporal dimension t. To capture this heterogeneity, we propose a composite kernel that separates spatial and temporal components as follows:

$$k_i(\mathbf{x}, \mathbf{x}') = k_i^{\text{spatial}}(\boldsymbol{p}, \boldsymbol{p}') + k_i^{\text{temporal}}(\mathbf{x}, \mathbf{x}'), \tag{10}$$

where $\boldsymbol{p} = (p_x, p_y, p_z)$ denotes the spatial canonical component in 3D, and $i \in \{1, \ldots, d\}$ is the index of the function.

Spatial correlations To capture smooth geometric correlations in canonical space, we adopt the Matérn kernel:

$$k_i^{\text{spatial}}(\boldsymbol{p}, \boldsymbol{p}') = \sigma_{s,i}^2 \frac{2^{1-\nu_i}}{\Gamma(\nu_i)} r_{s,i}^{\nu_i} K_{\nu_i}(r_{s,i}), \tag{11}$$

where $r_{s,i} = \sqrt{2\nu_i \sum_{j\in\{x,y,z\}} (p_j - p_j')^2 / \ell_{s,i,j}^2}$ is the anisotropic scaled distance, K_{ν_i} is the modified Bessel function, $\sigma_{s,i}^2$ is the signal variance, and ν_i controls smoothness. This encodes the prior that nearby primitives in canonical space exhibit similar deformations. Note that we choose the Matérn over the RBF kernel for its ability to handle discontinuities, which is essential for modeling spatially disconnected objects in dynamic scenes.

Temporal correlations To capture motion patterns along time, we model temporal correlations as a sum of per-axis Matérn kernels weighted by a periodic kernel as follows:

$$k_i^{\text{temporal}}(\mathbf{x}, \mathbf{x}') = \sum_{j\in\{x,y,z\}} k_{i,j}(p_j, p_j')\, k_{i,j}^{\text{periodic}}(t, t'), \tag{12}$$

where $k_{i,j}$ is a one-dimensional Matérn kernel over the corresponding axis, and $k_{i,j}^{\text{periodic}}$ is a periodic kernel over time t, defined as follows:

$$k_{i,j}^{\text{periodic}}(t, t') = \sigma_{i,j}^2 \exp\left(-\frac{2\sin^2(\pi|t - t'|/\tau_{i,j})}{\ell_{i,j}^2}\right), \tag{13}$$

with period $\tau_{i,j}$, length scale $\ell_{i,j}$, and signal variance $\sigma_{i,j}^2$. The periodic kernel provides strong inductive bias for temporal extrapolation by capturing patterns in motion, enabling robust predictions beyond the observed time range.

4.2. Variational Gaussian Processes for 4DGS

Standard GP posterior computation requires constructing an $N \times N$ kernel matrix and $\mathcal{O}(N^3)$ matrix inversion, where N is the number of data points. In our setting, N corresponds to the number of Gaussian primitives, which typically reaches tens of thousands, making exact inference computationally intractable. To address this, we adopt variational GPs [43].

Inducing points We approximate the full GP with M inducing points $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$, $\mathbf{z}_m \in \mathbb{R}^4$, $M \ll N$, which serve as a learned summary of the deformation field in the 4D input space $\mathbf{x} = (\mathbf{x}_s, t)$. With this approximation, we only compute two kernel matrices: $\mathbf{K}_{ZZ}^{(i)} = \mathbf{K}_i(\mathbf{Z}, \mathbf{Z}) \in \mathbb{R}^{M\times M}$ among inducing points and $\mathbf{K}_{XZ}^{(i)} = \mathbf{K}_i(\mathbf{X}, \mathbf{Z}) \in \mathbb{R}^{N\times M}$ between primitives and inducing points. This reduces complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2 + M^3)$, making inference tractable even with tens of thousands of Gaussian primitives.

For initialization of $\mathbf{Z}$, spatial locations $\mathbf{x}_s$ are selected by extracting time-series features from primitive trajectories using Chronos [1] and clustering them via k-means to select M representative canonical positions. Temporal locations t are sampled uniformly over the observed time range.

Training In addition to the inducing point locations $\mathbf{Z}$, we parameterize a variational posterior $q(\mathbf{u}_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$ over the function values at all inducing points, where $\mathbf{u}_i \in \mathbb{R}^M$ collects the values for output dimension $i \in \{1, \dots, d\}$. The kernel hyperparameters (*e.g.*, length scales, signal variances, and periods), inducing point locations $\mathbf{Z}$, and variational parameters $\{(\mathbf{m}_i, \mathbf{S}_i)\}_{i=1}^{d}$ are jointly optimized via the ELBO as follows:

$$\mathcal{L}_{\text{ELBO}} = \sum_{i=1}^{d} \left[\mathbb{E}_q[\log p(\mathbf{y}_i|\mathbf{u}_i)] - \text{KL}[q(\mathbf{u}_i)\|p(\mathbf{u}_i)]\right], \quad (14)$$

where $p(\mathbf{u}_i) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{ZZ}^{(i)})$ is the GP prior, the expectation term encourages fitting the observed deformations, and the KL term regularizes the posterior toward the prior.

Inference After training, given an arbitrary query point $\mathbf{x}^*$, we compute the mean and variance of the deformation from the variational posterior as follows:

$$\bar{\mu}_i^* = \mathbf{k}_*^{(i)\top}(\mathbf{K}_{ZZ}^{(i)})^{-1}\mathbf{m}_i \quad (15)$$

$$\bar{\sigma}_i^{*2} = k_i^* - \mathbf{k}_*^{(i)\top}\mathbf{\Sigma}_i\mathbf{k}_*^{(i)}, \quad (16)$$

where $\mathbf{k}_*^{(i)} = \mathbf{k}_i(\mathbf{Z}, \mathbf{x}^*) \in \mathbb{R}^M$ is the cross-covariance vector between the inducing points and the query, $k_i^* = k_i(\mathbf{x}^*, \mathbf{x}^*)$ is the kernel variance at $\mathbf{x}^*$, and $\mathbf{\Sigma}_i = (\mathbf{K}_{ZZ}^{(i)})^{-1} - (\mathbf{K}_{ZZ}^{(i)})^{-1}\mathbf{S}_i(\mathbf{K}_{ZZ}^{(i)})^{-1}$. This formulation scales as $\mathcal{O}(M)$ per query $\mathbf{x}^*$, enabling efficient inference without accessing all training primitives. The predicted mean and variance vectors over all output dimensions are $\bar{\boldsymbol{\mu}} = [\bar{\mu}_1, \dots, \bar{\mu}_d]^\top$ and $\bar{\boldsymbol{\sigma}}^2 = [\bar{\sigma}_1^2, \dots, \bar{\sigma}_d^2]^\top$.

4.3. GP-GS Optimization

This section describes our GP-GS optimization strategy, as outlined in Algorithm 1, which alternates between GP training on confident observations and 4DGS optimization guided by GP predictions, creating a feedback loop that progressively refines both representations.

Algorithm 1 GP-GS Optimization Strategy

Input: Training images $\mathcal{T}$, Gaussian primitives $\{\gamma_k\}_{k=1}^{N}$
Output: Optimized GS model with GP regularization

while not converged **do**
  Stage 1: GP Training // For every N_{GP} iterations
  Compute confidence: $C_k = \sum_{\mathbf{I}\in\mathcal{T}}\sum_{\mathbf{r}\in\mathbf{I}}\omega_k^{\pi}(\mathbf{r})$
  Select confident data: $\mathcal{D}_C = \{(\mathbf{x}_i, \mathbf{y}_i) : C_k > \tau_C\}$
  Train variational GPs on $\mathcal{D}_C$ with Eq. (14)
  Stage 2: GS Optimization
  Infer variational GP posteriors and cache $\bar{\boldsymbol{\mu}}_{(k,t)}(\mathbf{x})$
  Compute loss: $\mathcal{L}_{\text{GP}} = \mathbb{E}[\delta_k \cdot \|\mathbf{y}_{(k,t)} - \bar{\boldsymbol{\mu}}_{(k,t)}\|^2]$
  Update GS: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{GP}}\mathcal{L}_{\text{GP}}$
end while
return optimized GS and GP models

4.3.1. Stage 1: GP Optimization

Response-aware data sampling We measure the informativeness of each Gaussian primitive via its cumulative contribution to pixel rendering as follows:

$$C_k = \sum_{\mathbf{I}\in\mathcal{T}}\sum_{\mathbf{r}\in\mathbf{I}}\omega_{k,t}^{\pi}(\mathbf{r}), \quad (17)$$

where $\mathbf{r}$ is a pixel in a training image $\mathbf{I} \in \mathcal{T}$, $\omega_{k,t}^{\pi}(\mathbf{r})$ is the α-blending weight of primitive γ_k at pixel $\mathbf{r}$, and t is captured time of image $\mathbf{I}$. We select primitives with $C_k > \tau_C$ to form a confident subset $\mathcal{D}_C \subset \mathcal{D}$. The GP is then trained on $\mathcal{D}_C$ by maximizing the ELBO in Eq. 14, optimizing the kernel hyperparameters, inducing point locations $\mathbf{Z}$, and variational parameters $\{(\mathbf{m}_i, \mathbf{S}_i)\}_{i=1}^{d}$.

Handling noisy inputs Since the spatial coordinates $\boldsymbol{p} = (p_x, p_y, p_z)$ are also optimized parameters rather than exact positions, we inject Gaussian noise into each spatial dimension during GP training as follows:

$$\tilde{\mathbf{x}} = (p_x + \epsilon_x,\ p_y + \epsilon_y,\ p_z + \epsilon_z,\ t), \quad (18)$$

where $\epsilon_x, \epsilon_y, \epsilon_z \sim \mathcal{N}(0, 0.02)$. This acts as regularization, enabling the GP to handle uncertainty in canonical positions and produce more robust predictions.

4.3.2. Stage 2: GS Optimization

GP inference After training on $\mathcal{D}_C$, we run GP inference on all primitives. For each primitive k at time t with input $\mathbf{x}_{(k,t)}^* = (\boldsymbol{p}_k, t)$, we compute the posterior mean $\bar{\boldsymbol{\mu}}_{(k,t)}^*$, which serves as a pseudo-guidance signal. To balance computational efficiency and guidance freshness, we cache GP predictions every $N_{GP} = 2000$ iterations during GS optimization. Between updates, cached predictions are reused across optimization steps.

GP guidance loss We regularize GS optimization with a GP guidance loss as follows:

$$\mathcal{L}_{\text{GP}} = \frac{1}{NT} \sum_{k=1}^{N} \sum_{t=1}^{T} \delta_{(k,t)} \cdot \left\| \mathbf{y}_{(k,t)} - \bar{\boldsymbol{\mu}}^{*}_{(k,t)} \right\|^2, \quad (19)$$

where $\delta_{(k,t)} = \mathbf{1}(\|\mathbf{y}_{(k,t)} - \bar{\boldsymbol{\mu}}^{*}_{(k,t)}\| > \tau_{\delta})$ selects primitives that deviate from GP motion predictions $\bar{\boldsymbol{\mu}}^{*}_{(k,t)}$. The threshold τ_{δ} is annealed during optimization from $\tau_{\delta,\text{start}}$ to $\tau_{\delta,\text{end}}$, progressively tightening constraints as the two representations converge. The total training loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{GP}}\mathcal{L}_{\text{GP}}$, where $\mathcal{L}_{\text{recon}}$ follows SoM [46], consisting of photometric, D-SSIM, flow, and smoothness losses. The balancing weight λ_{GP} is set to 0.1.

4.4. Interpretability via Uncertainty Quantification

As GPs define a distribution over functions, they naturally provide variance estimates alongside mean predictions, which we use to quantify the uncertainty of primitive motions. For translation, uncertainty is directly obtained from the first three dimensions of $\bar{\boldsymbol{\sigma}}^{*}$. For rotation, the 6D-to-matrix conversion via Gram-Schmidt orthogonalization is non-linear, precluding direct variance propagation. Instead, we use Monte Carlo sampling to approximate uncertainty in the final positions as follows:

$$U_{k,t} = \text{Var}(\{\boldsymbol{p}^{(s)}_{k,t}\}^{S}_{s=1}), \quad (20)$$

where $\boldsymbol{p}^{(s)}_{k,t}$ is the s^{th} deformed position of primitive k across S sampled deformations at time t. We then render a motion uncertainty map by projecting $U_{k,t}$ to a target view as follows:

$$\hat{\mathbf{U}}(\mathbf{r}) = \sum_{k=1}^{N} U_{k,t}\, \omega^{\pi}_{k,t}(\mathbf{r}), \quad (21)$$

which is analogous to color rendering in Eq. (5), and examples are shown in Figure 2.

4.5. Extrapolation for Future Motion Prediction

A key advantage of GP-based motion reasoning is the ability to predict future motion beyond the training sequence. Given a canonical position and a future timestamp t_f, we directly query the trained GP in Eq. (15) with $\mathbf{x}^{*}_{f} = (\boldsymbol{p}, t_f)$ to obtain the deformation at that time. This enables motion forecasting without additional training or architectural modifications.

5. Experiments

This section evaluates the efficacy of GP-4DGS, demonstrating that the GP integration provides superior structural priors for sparse-view dynamic reconstruction while achieving reliable uncertainty estimation and motion prediction.

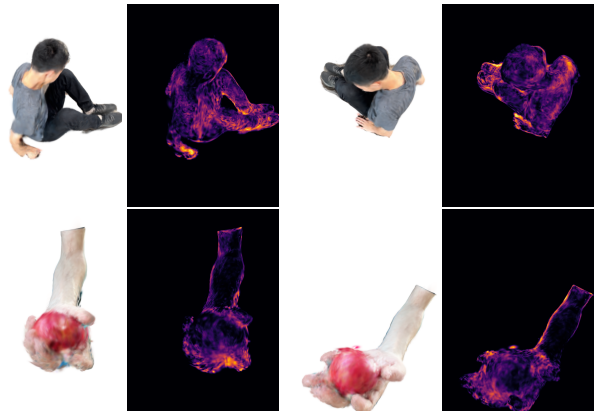


Figure 2. Uncertainty quantification. GP-4DGS provides principled uncertainty estimates for motion, a capability inherently lacking in existing 4DGS methods.

Table 1. Quantitative results on the DyCheck dataset. We evaluate performance on all seven scenes (All), the five scenes used in SoM [46] (SoM 5), and a challenging subset with reduced viewpoint overlap. GP-4DGS consistently achieves superior results, particularly under sparse observations.

Data	Method	mPSNR ↑	mSSIM ↑	mLPIPS ↓
All	Gaussian Marbles [42]	15.84	0.54	0.57
	SoM [46]	17.09	0.65	0.39
	GP-4DGS (ours)	**17.38**	**0.65**	**0.37**
SoM 5	SC-GS [23]	14.13	0.48	0.49
	D-3DGS [51]	11.92	0.49	0.66
	4DGS [48]	13.42	0.49	0.56
	T-NeRF [46]	15.60	0.55	0.55
	HyperNeRF [34]	15.99	0.59	0.51
	DynIBaR [22]	13.41	0.48	0.55
	Gaussian Marbles [42]	16.03	0.54	0.58
	SoM [46]	16.73	0.64	0.43
	GP-4DGS (ours)	**16.92**	**0.66**	**0.41**
Challenging subset [14]	Gaussian Marbles [42]	14.05	0.40	0.61
	SoM [46]	14.56	**0.46**	0.53
	GP-4DGS (ours)	**15.02**	**0.46**	**0.51**

5.1. Experimental Settings

We utilize DyCheck [8], a benchmark comprising handheld monocular videos with rapid motion that presents challenging realistic scenarios for dynamic scene reconstruction. To evaluate robustness under extreme viewpoint shifts, we qualitatively assess our method on DAVIS [36], where camera poses are estimated using Mega-SAM [24]. Our variational GP framework is implemented using GPyTorch [10].

5.2. Performance of Dynamic Novel View Synthesis

Results on DyCheck Following the evaluation protocol of DyCheck [8], we assess novel view synthesis quality using masked versions of standard metrics—mPSNR, mSSIM, and mLPIPS—to focus specifically on co-visible

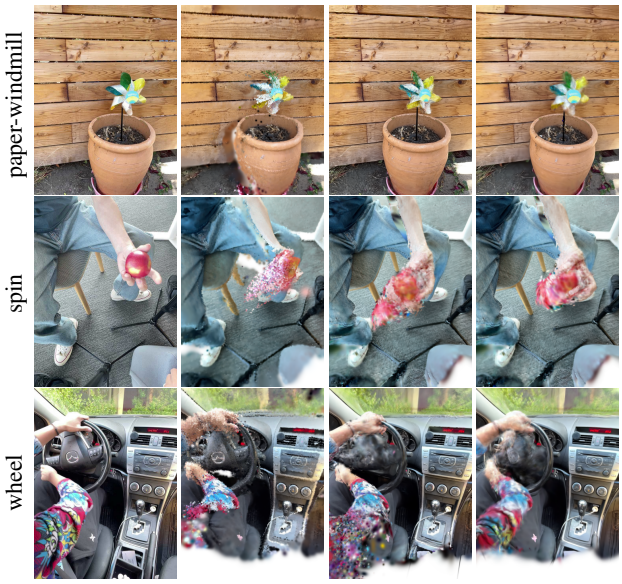


Figure 3. Qualitative comparison of novel view synthesis on the DyCheck dataset. GP-4DGS shows more accurate geometry compared to baselines, particularly in regions with less observation.

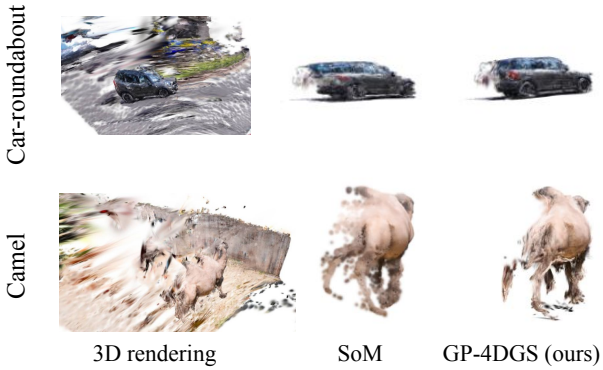


Figure 4. Qualitative comparison on the DAVIS dataset under extreme viewpoint shifts from training view. Unlike the baseline, our spatiotemporal GP prior effectively regularizes the scene by faithfully propagating motion constraints into poorly observed regions.

regions. Table 1 presents that GP-4DGS consistently surpasses all baselines, with the performance gap in mPSNR and mLPIPS widening on the *Challenging subset*[1]. This demonstrates that our GP-based optimization effectively mitigates artifacts from sparse observations by propagating spatiotemporal correlations from confident regions to unobserved ones. Qualitative results in Figure 3 confirm these gains; our method recovers sharper textures and more accurate geometry compared to Gaussian Marbles [42] and SoM [46], which often suffer from floater artifacts or geometric blurring under rapid motion.

[1]The challenging subset follows the evaluation protocol of [14], where training and test viewpoint overlap is significantly reduced.

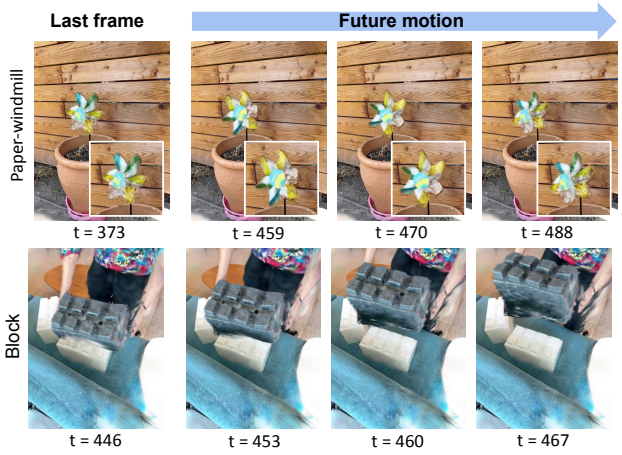


Figure 5. Motion extrapolation results from GP-4DGS. Our GP-based approach naturally predicts future motion by querying the model at timesteps beyond the training range.

Table 2. Future motion extrapolation performance in terms of PSNR (↑). We evaluate performance on the last 5 and 15 frames excluded for training. GP-4DGS outperforms naïve linear extrapolation, especially in periodic scenes where the temporal kernel effectively captures cyclic structures.

Method	Periodic motion		Non-periodic motion	
	5 frames	15 frames	5 frames	15 frames
Linear extrapolation	11.55	8.11	15.02	11.92
GP-4DGS (ours)	**17.62**	**16.65**	**15.27**	**13.22**

Extreme novel view synthesis on DAVIS To assess robustness under extreme viewpoint shifts, we evaluate GP-4DGS on DAVIS by rendering views significantly outside the training distribution. As illustrated in Figure 4, our method preserves geometry and sharp edges more faithfully than the baseline [46], validating the reliability of GP-GS optimization in maintaining structural integrity, even in regions with sparse observations.

5.3. Probabilistic Interpretability of 4DGS

5.3.1. Future Motion Extrapolation

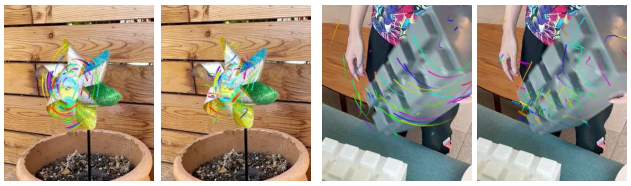
GP-4DGS enables motion extrapolation by querying the model at future timesteps. To demonstrate its effectiveness, we withhold the last 5 or 15 frames of each sequence during training for evaluation. As shown in Table 2, our method significantly outperforms naive linear extrapolation, particularly in periodic scenes, where the temporal kernel captures cyclic dynamics. Figure 5 further confirms that GP-4DGS predicts physically plausible and temporally coherent motion for these unobserved timesteps, demonstrating its ability to learn underlying dynamics rather than merely interpolating training frames.

5.3.2. Uncertainty Quantification

GP-4DGS enables principled uncertainty quantification through its probabilistic formulation. To evaluate its reli-

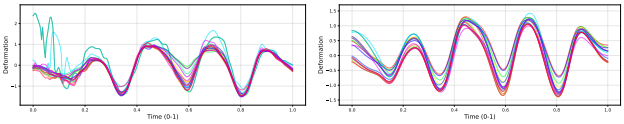
Table 3. Uncertainty quantification results in terms of AUSE-MSE (↓), measured as the area gap between the reconstruction error- and predicted uncertainty-based sparsification curves. Top 20 and 40 denote the frames with the lowest MSEs. GP-4DGS achieves the lowest AUSE across all settings.

Method	AUSE-MSE [40] ($\times 10^{-2}$) ↓		
	Top 20 frames	Top 40 frames	All frames
Random	9.76	9.30	10.98
UA-4DGS [19]	7.60	8.11	8.62
GP-4DGS (ours)	**7.22** (-0.38)	**8.00** (-0.11)	**8.49** (-0.13)



w/ GP w/o GP w/ GP w/o GP

Figure 6. Trajectory comparison on the (left) *paper-windmill* and (right) *block* scene. GP guidance effectively regularizes motion trajectories, reducing noise and producing physically plausible motion patterns, compared to the baseline approach.



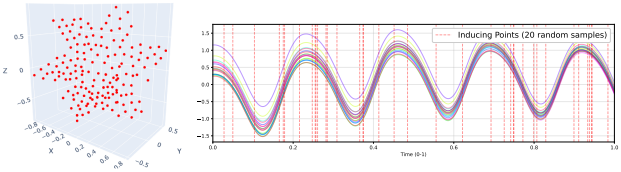
(a) Trajectory w/o GP (b) Trajectory from GP

Figure 7. Trajectory comparison on the *spin* scene between the initial GS reconstruction and the GP guidance in the GP-GS optimization. These graphs correspond to the first dimension in 6D rotation. The GP provides accurate and stable motion priors.

ability, we adopt the Area Under the Sparsification Error (AUSE), which measures the alignment between estimated uncertainty and actual reconstruction error. As shown in Table 3, GP-4DGS consistently outperforms both Random and UA-4DGS [19] baselines. Notably, the performance gap becomes larger when evaluating AUSE on high-quality frames (e.g., top 20 and 40 frames). This indicates that by accounting for both spatial and temporal correlations, our model effectively identifies subtle residual errors in well-reconstructed regions, where baselines fail to produce well-calibrated uncertainty.

5.4. Additional Analysis

GP guidance as a motion prior We present the regularizing effect of GP guidance on Gaussian trajectories in Figures 6 and 7. Without GP guidance, reconstructions suffer from noise and fluctuations, especially in sparsely observed regions. In contrast, GP-GS optimization leverages learned correlation structures to propagate motion priors from confident observations to uncertain regions. This process effectively stabilizes the trajectories, ensuring temporal smoothness and structural consistency throughout the sequence.



(a) Canonical space (b) time axis in $[0.1]$

Figure 8. Visualization of inducing points (*paper-windmill*). Inducing points (red) are well-distributed across the canonical space and temporal axis to ensure comprehensive coverage of the scene.

Table 4. Comparison of inducing point initialization methods in terms of ELBO (↑). Our time-series-based selection achieves superior convergence with higher ELBO than baselines.

Scene	Random Init.	Velocity KNN	Time-series (ours)
paper-windmill	0.85	1.12	**1.28**
apple	1.12	**1.21**	1.19
spin	0.92	1.15	**1.35**
teddy	1.45	1.52	**1.68**
block	1.08	1.31	**1.48**
space-out	0.78	1.18	**1.52**
wheel	1.35	1.68	**1.80**
Average	1.10	1.37	**1.53**

Inducing point coverage To show that the GP faithfully captures scene deformation, we visualize the inducing point distribution in Figure 8. The inducing points are well-distributed in both the canonical space and temporal axis, ensuring comprehensive coverage of motion dynamics over the full sequence. Such an arrangement prevents the GP from biasing toward specific clusters, enabling consistent and stable global deformation.

Inducing point initialization We compare our time-series feature-based selection with random and velocity-based counterpart for inducing point initialization. To isolate the impact of initialization, we evaluate ELBO under a single offline GP optimization without the full GP-GS loop. As shown in Table 4, our method consistently achieves higher ELBO, demonstrating superior convergence and representation of motion dynamics. See Section A of the supplementary document for detailed initialization strategies.

6. Conclusion

We proposed a novel probabilistic framework that integrates variational Gaussian Processes into the 4D Gaussian Splatting pipeline. By leveraging GP-based motion priors, our method effectively capture complex spatiotemporal correlations and propagates learned dynamics to uncertain regions, resulting in physically plausible motion trajectories. Furthermore, our approach provides principled uncertainty quantification and enables reliable future motion prediction. We believe our framework represents a significant step toward integrating principled probabilistic modeling with high-fidelity neural graphics.

Acknowledgements This work was partly supported by the National Research Foundation of Korea (NRF) grant [RS-2022-NR070855, Trustworthy Artificial Intelligence] and the Institute of Information & communications Technology Planning & Evaluation (IITP) grants [RS-2025-25442338, AI star Fellowship Support Program (Seoul National University); RS-2022-II220959 (No.2022-0-00959), (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making; No.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)] funded by the Korea government (MSIT).

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. 5

[2] Weiwei Cai, Weicai Ye, Peng Ye, Tong He, and Tao Chen. DynaSurfGS: Dynamic surface reconstruction with planar-based Gaussian splatting. *arXiv preprint arXiv:2408.13972*, 2024. 2

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*, 2022. 1

[4] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4D view synthesis and video processing. In *ICCV*, 2021. 2

[5] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *SIGGRAPH*, 2024. 2

[6] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2

[7] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2

[8] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022. 6

[9] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200FPS. In *ICCV*, 2021. 1

[10] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *NIPS*, 2018. 6

[11] Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. In *TCSVT*, 2024. 2

[12] Zhihao Guo, Jingxuan Su, Shenglin Wang, Jinlong Fan, Jing Zhang, Liangxiu Han, and Peng Wang. GP-GS: Gaussian processes for enhanced Gaussian splatting. *arXiv preprint arXiv:2502.02283*, 2025. 2, 3

[13] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *ICLR*, 2024. 1

[14] Jiaxin Huang, Sheng Miao, Bangbang Yang, Yuewen Ma, and Yiyi Liao. Vivid4d: Improving 4d reconstruction from monocular video by video inpainting. In *ICCV*, 2025. 6, 7

[15] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-controlled Gaussian splatting for editable dynamic scenes. In *CVPR*, 2024. 2, 3

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. In *ACM TOG*, 2023. 1, 2

[17] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian splatting as Markov Chain Monte Carlo. In *NeurIPS*, 2024. 2, 3

[18] Mijeong Kim, Seonguk Seo, and Bohyung Han. InfoNeRF: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. 1

[19] Mijeong Kim, Jongwoo Lim, and Bohyung Han. 4D Gaussian splatting in the wild with uncertainty-aware regularization. In *NeurIPS*, 2024. 2, 8

[20] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. DynMF: Neural motion factorization for real-time dynamic view synthesis with 3D Gaussian splatting. In *ECCV*, 2024. 2

[21] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds, 2024. arXiv preprint arXiv:2405.17421. 2

[22] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DynIBaR: Neural dynamic image-based rendering. In *CVPR*, 2023. 6

[23] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime Gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*, 2024. 2, 3, 6

[24] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025. 6

[25] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In *WACV*, 2025. 2

[26] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *CVPR*, 2024.

[27] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lyv, Peng Wang, Wenping Wang, and Junhui Hou. Modgs: Dy-

namic gaussian splatting from casually-captured monocular videos. In *ICLR*, 2025. 2

[28] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. VL-Grasp: a 6-Dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *IROS*, 2023. 1

[29] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *CVPR*, 2024. 2

[30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[31] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, Liping Jing, Yiming Nie, and Bin Dai. DriveWorld: 4D pre-trained scene understanding via world models for autonomous driving. In *CVPR*, 2024. 1

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 1

[33] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 2

[34] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM TOG*, 40(6):1–12, 2021. 6

[35] Yuxin Peng et al. DeSiRe-GS: 4D street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In *CVPR*, 2025. 1

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[37] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 1, 2

[38] Alfredo Rivero, ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Rig3DGS: Creating controllable portraits from casual monocular videos. *arXiv:2402.03723*, 2024. 2

[39] Antoni Rosinol, John J Leonard, and Luca Carlone. NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. In *IROS*, 2023. 1

[40] Luca Savant, Diego Valsesia, and Enrico Magli. Modeling uncertainty for Gaussian splatting. *arXiv preprint arXiv:2403.18476*, 2024. 2, 8

[41] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4D: Efficient neural 4D decomposition for high-fidelity dynamic reconstruction and rendering. In *CVPR*, 2023. 2

[42] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH*, 2024. 2, 6, 7

[43] Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*, 2009. 4

[44] Toon Van de Maele, Ozan Catal, Alexander Tschantz, Christopher L Buckley, and Tim Verbelen. Variational bayes gaussian splatting. In *NeurIPS*, 2025. 2, 3

[45] Joanna Waczynska, Piotr Borycki, Joanna Kaleta, Slawomir Tadeja, and Przemysław Spurek. D-miso: Editing dynamic 3d scenes using multi-gaussians soup. In *NeurIPS*, 2024. 2

[46] Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of Motion: 4D reconstruction from a single video. In *ICCV*, 2025. 2, 3, 6, 7

[47] Shizun Wang, Xingyi Yang, Qiuhong Shen, Zhenxiang Jiang, and Xinchao Wang. Gflow: Recovering 4D world from monocular video. In *AAAI*, 2025.

[48] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 2, 3, 6

[49] Renlong Wu, Zhilu Zhang, Mingyang Chen, Zifei Yan, and Wangmeng Zuo. Deblur4DGS: 4D Gaussian splatting from blurry monocular video. *arXiv preprint arXiv:2412.06424*, 2024. 2

[50] Tianyi Yan, Dongming Wu, Wencheng Han, Junpeng Jiang, Xia Zhou, Kun Zhan, Cheng-zhong Xu, and Jianbing Shen. DrivingSphere: Building a high-fidelity 4D world for closed-loop simulation. In *CVPR*, 2025. 1

[51] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 2, 3, 6

[52] Yijun Yuan and Andreas Nüchter. Online learning of neural surface light fields alongside real-time incremental 3D reconstruction. *IEEE Robotics and Automation Letters*, 8(6): 3844–3851, 2023. 1

[53] Chubin Zhang, Hongliang Song, Yi Wei, Chen Yu, Jiwen Lu, and Yansong Tang. GeoLRM: Geometry-aware large reconstruction model for high-quality 3D Gaussian generation. In *NeuIPS*, 2024. 1

[54] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. DrivingGaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, 2024. 1

[55] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 4