SparseDynGS: Learning Physically Grounded Gaussian Dynamics from Sparse-View Videos

Jiwoong Lee[1] Hyeokjun Kwon[2] Jiwon Jang[1]
Seoul National University
[1]Graduate School of Data Science [2]Department of Electrical and Computer Engineering
{dlwldnd5689, hyeokjun, jjw6424}@snu.ac.kr

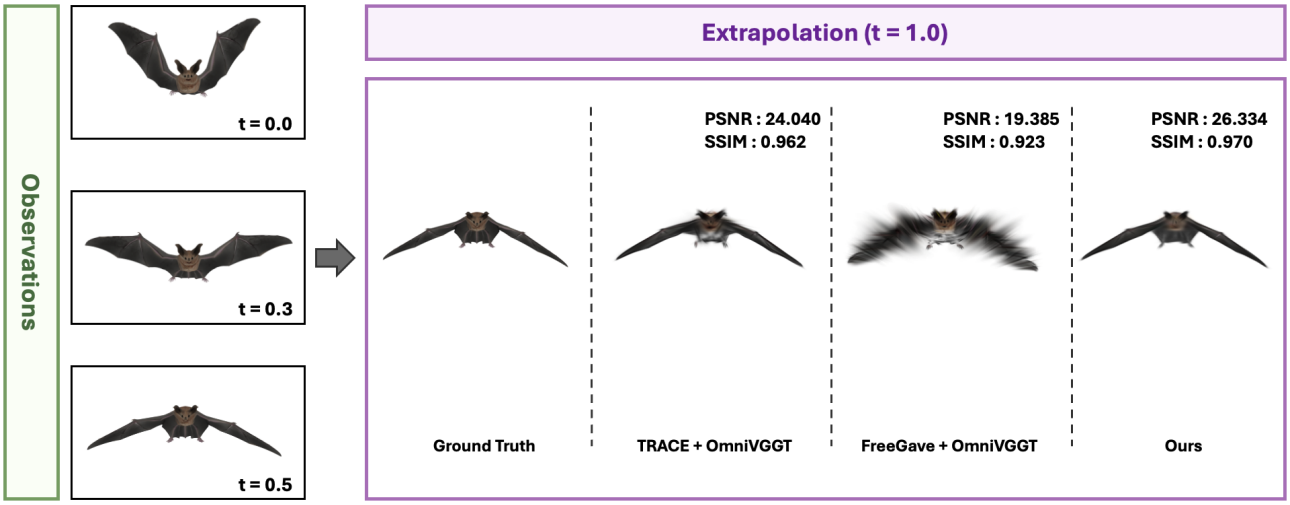

Figure 1. Given sparse-view RGB observations from early timesteps, the goal is to extrapolate the scene to unseen future timesteps. Existing physics-aware 3DGS methods [7, 8] become unstable under sparse-view supervision. In contrast, **SparseDynGS** combines compact, geometry-consistent Gaussian refinement with progressive physics learning from short to long horizons, thereby improving the stability and fidelity of long-horizon extrapolation.

Abstract

*Physics-aware 3D Gaussian Splatting methods learn per-Gaussian dynamics for future-frame extrapolation, but they typically rely on dense multi-view supervision. We present **SparseDynGS**, a framework for learning physically grounded Gaussian dynamics from sparse-view videos. To address unreliable geometry under limited views, We obtain an initial Gaussian representation from a pretrained feed-forward 3D reconstruction model and improve its compactness and geometric consistency through **Regularized Gaussian Refinement**. We further introduce **Progressive Physics Learning**, which stabilizes short-horizon dynamics with a sliding temporal window and improves long-horizon prediction through late-horizon sampling and early-horizon replay regularization. Experiments on Dynamic Objects and Dynamic Indoor Scene datasets demonstrate the effectiveness of SparseDynGS in sparse-view physical dynamics learning*

1. Introduction

Understanding the physical evolution of a dynamic 3D scene is a fundamental requirement for visual systems that must go beyond reproducing observed frames. A robot interacting with deformable or articulated objects, or a digital twin predicting future outcomes, requires a scene representation that can infer how geometry evolves over time, rather than merely reconstructing its appearance at sampled frames.

Recent advances in neural 3D representations have greatly improved the quality and efficiency of scene reconstruction. Neural Radiance Fields (NeRF) [14] enabled high-fidelity novel-view synthesis, while 3D Gaussian Splatting (3DGS) [4] introduced an explicit point-based representation for real-time rendering. Subsequent works extend these representations to dynamic or 4D reconstruction through time-conditioned deformation [15, 16, 18, 20], scene flow [10, 11, 13], or spatio-temporal feature fields [1–3, 9, 12, 19]. These methods achieve strong per-

formance in novel-view synthesis within the observed temporal range. However, their temporal mappings are primarily optimized to reproduce the training frames rather than to satisfy an underlying physical dynamics. Consequently, they often struggle to extrapolate to unseen future timesteps.

To address this limitation, recent studies have attempted to enable 3D representations to learn physical laws. NVFi [5] augments a NeRF-based representation with velocity fields to model dynamic scenes under physical constraints. More recently, 3DGS-based methods such as FreeGave [8] and TRACE [7] treat Gaussian primitives as rigid particles and learn their per-primitive velocities and accelerations. By explicitly modeling such particle-level dynamics, these methods improve the ability to predict scene evolution beyond the observed frames.

Nevertheless, existing physics-aware 3DGS methods have been predominantly developed and evaluated in dense multi-view settings, where training can involve as many as 24 calibrated views. While such dense capture setups provide strong geometric and temporal constraints, they are expensive and difficult to deploy in practical scenarios.

Toward practical physics-aware 3DGS, We first establish sparse-view protocols on commonly used dynamic-scene benchmarks, DynIndoorScene [5] and DynObjects [5], by sampling training views to evaluate both viewpoint interpolation and extrapolation, as illustrated in Fig. 2. A natural first attempt is to directly apply existing physics-aware 3DGS methods, such as FreeGave [8] and TRACE [7], to this sparse-view setting. However, this leads to two major challenges. First, unlike dense multi-view settings, where SfM initialization is often reliable, sparse observations provide limited geometric constraints and can result in unstable point-cloud initialization. Second, existing methods [8] [7] mainly learn physical dynamics over short temporal intervals, often relying on deformation-based modeling commonly used in 4D reconstruction. Under sparse-view supervision, such short-horizon optimization can easily converge to poor local optima, resulting in unstable or physically implausible long-term dynamics.

To address these challenges, we propose SparseDynGS, a framework for learning physically grounded Gaussian dynamics from sparse-view videos. SparseDynGS builds upon the staged optimization strategy of recent physics-aware 3DGS methods, such as FreeGave [8] and TRACE [7], which separates training into a warm-up stage and a physics-learning stage. During the warm-up stage, we address the unreliable initialization problem by initializing the Gaussian representation with OmniVGGT [17], a feed-forward 3D reconstruction model that provides robust geometric priors under sparse-view inputs. To further mitigate overfitting and redundant Gaussian primitives caused by sparse supervision and feed-forward reconstruction artifacts, we introduce **Regularized Gaussian Refinement**.

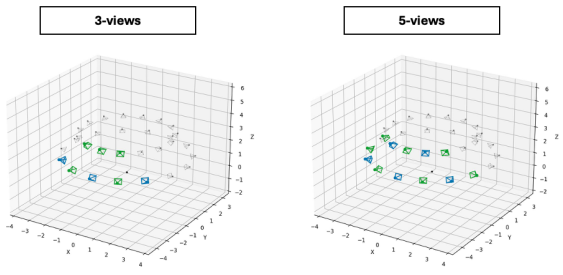

Figure 2. Camera configurations sampled on the Dynamic Indoor Scene dataset [5]. Blue cameras indicate the sampled training views, while green cameras denote the validation views. The validation cameras cover both interpolated and extrapolated viewing directions with respect to the sampled training views.

This refinement combines opacity sparsity regularization with depth supervision from OmniVGGT [17], encouraging a compact and geometrically consistent Gaussian representation for learning physical dynamics.

To facilitate physics learning under sparse-view supervision, we introduce **Progressive Physics Learning**, a temporal curriculum framework based on a Sliding Temporal Window. This curriculum naturally divides training into two phases. In Short-Horizon Physics Learning, the model samples timesteps within the currently active window and focuses on stable short-range dynamics before the full temporal horizon is exposed. Once the window reaches the full sequence, we transition to Long-Horizon Physics Learning. In this phase, we propose late-horizon sampling to prioritize later timesteps, thereby promoting physically consistent dynamics over long temporal horizons. To prevent the model from forgetting dynamics learned from previously exposed timesteps, the long-horizon phase further incorporates early-horizon replay regularization, which reintroduces supervision from earlier timesteps. The main contributions are summarized as follows:

- We introduce **SparseDynGS**, a framework for learning physically grounded Gaussian dynamics from sparse-view videos. To evaluate this practical setting, we establish sparse-view protocols on DynIndoorScene [5] and DynObjects [5], covering both viewpoint interpolation and extrapolation.
- We propose **Opacity-Depth Regularized Gaussian Refinement** for robust warm-up under sparse-view supervision. By initializing Gaussians with OmniVGGT [17] and jointly regularizing opacity and rendered depth, our method mitigates unstable point-cloud initialization, overfitting, and redundant Gaussian primitives.
- We develop **Progressive Physics Learning**, a temporal curriculum for stable long-horizon dynamics learning. Built upon a Sliding Temporal Window, it first learns short-range dynamics through Short-Horizon Physics

Learning and then extends to Long-Horizon Physics Learning with late-horizon sampling and early-horizon replay regularization.

2. Related Work

2.1. 3D Gaussian Splatting in Sparse View

Sparse-view reconstruction is challenging due to insufficient visual constraints, making 3DGS prone to overfitting through floaters, distorted geometry, and view-specific artifacts. Recent sparse-view 3DGS methods mitigate this spatial under-constraint by incorporating additional geometric priors or regularization. For example, FSGS [21] leverages depth-guided augmented views to improve unseen-view coverage, while DNGaussian [6] introduces global-local normalized depth regularization to stabilize Gaussian geometry under sparse-view supervision. Inspired by these regularization-based approaches, we employ Regularized Gaussian Refinement during the warm-up stage, enabling us to obtain compact and geometrically consistent Gaussian representations before learning physical dynamics under sparse-view settings.

2.2. 4D Gaussian Splatting

Early NeRF-based methods model dynamics by conditioning radiance fields on time or learning deformations from a canonicalspace. Representative works include D-NeRF [18], Nerfies [15], and HyperNeRF [16], which handle non-rigid motion through time-dependent or continuous deformation fields. Following the success of 3DGS [4], recent methods extend Gaussian primitives to dynamic 4D reconstruction. Deformable 3D Gaussians [20] and 4D Gaussian Splatting [19] learn time-dependent transformations of Gaussian kernels for efficient dynamic rendering. While these methods improve visual quality and rendering speed, they primarily target reconstruction within the captured time range, and their deformation fields are not explicitly constrained by physical dynamics. As a result, their predictions may become unreliable when extrapolated beyond the observed timesteps.

2.3. Physics-Informed 3D Gaussian Splatting

FreeGave [8] and TRACE [7] are the closest to our target setting because they learn physical dynamics directly from multi-view dynamic videos without explicit object labels. FreeGave assigns latent physics codes to Gaussians and estimates a divergence-free per-Gaussian velocity field, enabling future-frame extrapolation and motion decomposition. TRACE treats each Gaussian as a rigid particle with size and orientation, then learns translation–rotation dynamics parameters that govern the particle's motion. Both methods demonstrate that 3DGS can serve not only as a rendering primitive but also as a carrier of physical dynamics. However, their experimental settings are still mainly based on dense multi-view observations, which provide strong geometric constraints for both reconstruction and dynamics learning. Our work instead focuses on the sparse-view regime, where the lack of visual constraints makes physically meaningful Gaussian dynamics harder to learn and more important to regularize.

3. Method

3.1. Preliminaries

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [4] represents a scene as a set of anisotropic Gaussian primitives. Each Gaussian primitive is parameterized by its 3D position $\mathbf{x}_i \in \mathbb{R}^3$, opacity o_i, color feature $\mathbf{c}_i$, and covariance matrix $\mathbf{\Sigma}_i$. In practice, the covariance is decomposed into a rotation $\mathbf{R}_i$ and a scaling matrix $\mathbf{S}_i$, such that

$$\mathbf{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^\top \mathbf{R}_i^\top. \tag{1}$$

The color feature is represented by spherical harmonics coefficients, allowing view-dependent appearance to be modeled during differentiable rendering.

Given a camera pose $\mathbf{P}^c$, 3DGS projects the Gaussian primitives onto the image plane and alpha-composites them in depth order to produce a rendered image:

$$\hat{\mathbf{I}}^c = \mathcal{R}(\mathcal{G}; \mathbf{P}^c), \tag{2}$$

where $\mathcal{G} = \{\mathbf{x}_i, \mathbf{R}_i, \mathbf{S}_i, o_i, \mathbf{c}_i\}_{i=1}^N$ denotes the set of Gaussian primitives and $\mathcal{R}$ is the differentiable Gaussian splatting renderer.

Canonical 3D Gaussians. Given a multi-view RGB video sequence, we denote the image captured by camera c at timestamp t as $\mathbf{I}_t^c$, where $c \in \{1, \ldots, N\}$ and $t \in \{0, \ldots, T\}$. Here, N is the number of training cameras and T is the last timestamp used for training. Existing physics-aware 3DGS methods, such as TRACE [7] and FreeGave [8], initialize the scene by reconstructing a point cloud from initial-frame multi-view images $\{\mathbf{I}_0^1, \cdots, \mathbf{I}_0^N\}$ using Structure-from-Motion (SfM), followed by standard 3DGS optimization as a warm-up stage. While this strategy is effective in dense multi-view settings, where the scene geometry is well constrained by sufficient cross-view observations, it becomes unreliable under sparse-view inputs. In such settings, the limited number of views often leads to noisy and incomplete SfM initialization. This is particularly detrimental to physics learning, as physical dynamics are estimated on top of the reconstructed Gaussian primitives. To address this issue, we leverage OmniVGGT [17], a pretrained feed-forward 3D reconstruction model, to obtain robust geometric priors under sparse-view observations. We use the reconstructed point cloud from OmniVGGT [17] to initialize the Gaussian configuration $\mathcal{G}_0$.

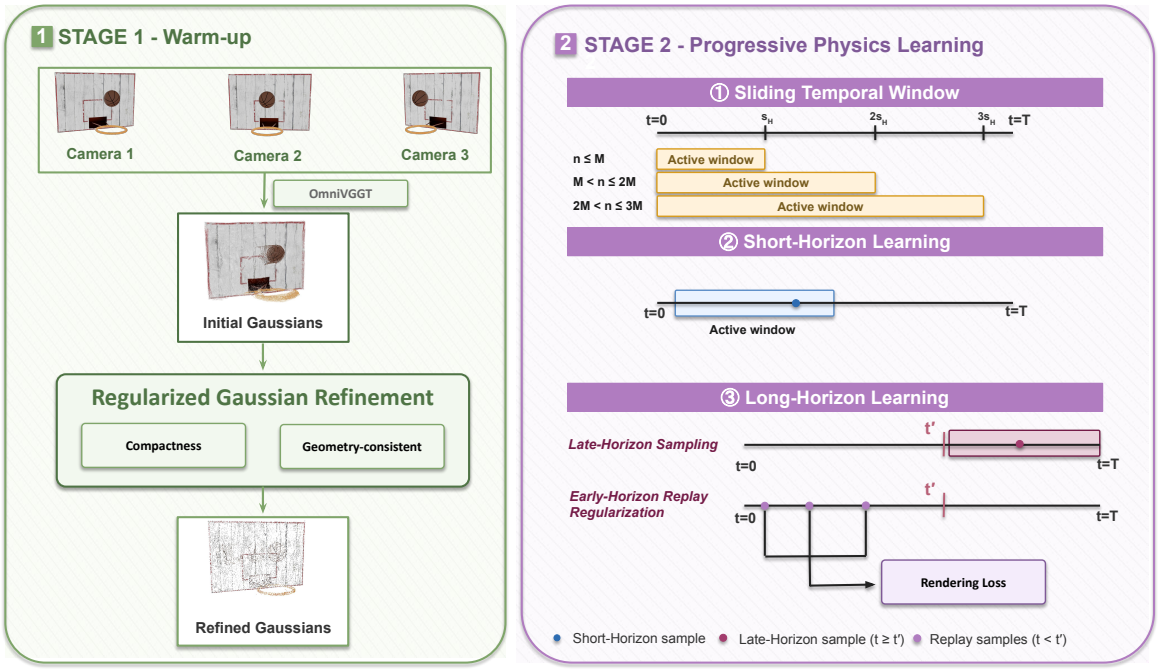

Figure 3. Overview of SparseDynGS. Given sparse multi-view videos, Stage 1 initializes the canonical Gaussian configuration using OmniVGGT [17] and refines it through Regularized Gaussian Refinement, producing a compact and geometry-consistent representation. Stage 2 progressively learns physically grounded Gaussian dynamics with an expanding temporal window. Given the current physics-training iteration n, the active horizon grows by s_H every M iterations. During short-horizon learning, target timesteps are sampled only within the active window until it covers the full sequence. In long-horizon learning, late-horizon sampling focuses supervision on later timesteps, $t \geq t'$, while early-horizon replay regularization reintroduces supervision from earlier timesteps, $t < t'$, to prevent forgetting.

Particle-based Gaussian Dynamics. Following TRACE [7] and FreeGave [8], we treat each Gaussian primitive as a rigid-body particle whose motion is governed by learned physical parameters. For the i-th Gaussian primitive at time t, we predict its translational velocity $\mathbf{v}_{i,t}$, translational acceleration $\mathbf{a}_{i,t}$, angular velocity $\boldsymbol{\omega}_{i,t}$, and angular acceleration $\boldsymbol{\beta}_{i,t}$ using a neural dynamics network:

$$(\mathbf{v}_{i,t}, \mathbf{a}_{i,t}, \boldsymbol{\omega}_{i,t}, \boldsymbol{\beta}_{i,t}) = f_\theta(\mathbf{x}_{i,t}, t), \tag{3}$$

where $f_{i,t}$ denotes the Gaussian center and Φ_θ is the learned dynamics field. Given these physical quantities, we propagate each Gaussian to the next timestep using a second-order Runge–Kutta integration operator:

$$\mathcal{G}_{t+\Delta t} = \Phi_{\mathrm{RK2}}\left(\mathcal{G}_t; f_\theta(\mathbf{x}_t, t), \Delta t\right), \tag{4}$$

where Φ_{RK2} updates the positions and orientations of the Gaussian primitives while keeping their scales and opacities unchanged.

3.2. Regularized Gaussian Refinement.

Although feed-forward initialization provides more robust geometric priors than SfM under sparse-view inputs, the reconstructed point clouds often contain redundant points. These redundant points are directly converted into unnecessary Gaussian primitives, increasing rendering cost and slowing down subsequent optimization. To encourage a compact representation, we impose an opacity sparsity regularization on the Gaussian configuration $\mathcal{G}_0$ during the warm-up stage:

$$\mathcal{L}_{\mathrm{opacity}} = \frac{1}{|\mathcal{G}_0|} \sum_{i \in \mathcal{G}_0} |o_i|. \tag{5}$$

This regularization encourages redundant Gaussians to have low opacity, allowing them to be naturally removed by the pruning step in the standard 3DGS densification strategy.

In addition, OmniVGGT [17] provides a depth map for each input view, which we use as a geometric supervision signal during the warm-up stage. Let $\hat{\mathbf{D}}_{0,c}$ denote the depth map rendered from the initial Gaussian configuration $\mathcal{G}_0$ at camera view c, and let $\mathbf{D}_{0,c}^{\mathrm{Omni}}$ denote the corresponding depth prior predicted by OmniVGGT [17]. We define the depth regularization loss as:

$$\mathcal{L}_{\mathrm{depth}} = \sum_c \left\| \mathbf{M}_{0,c} \odot \left(\hat{\mathbf{D}}_{0,c} - \mathbf{D}_{0,c}^{\mathrm{Omni}} \right) \right\|_1, \tag{6}$$

where $\mathbf{M}_{0,c}$ is a valid-pixel mask and $\odot$ denotes element-wise multiplication. The overall optimization objective is then written as:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_{\text{d}}\mathcal{L}_{\text{depth}} + \lambda_{\text{o}}\mathcal{L}_{\text{opacity}}, \tag{7}$$

where $\mathcal{L}_{\text{rgb}}$ is the standard image reconstruction loss, consisting of L1 and d-SSIM loss and λ_{d} and λ_{o} control the strength of the depth and opacity regularization terms, respectively. By applying depth and opacity regularization during the warm-up stage, our optimization reduces redundancy and overfitting, providing a more reliable representation for estimating velocity and acceleration fields, as qualitatively shown in

3.3. Progressive Physics Learning

Under sparse-view supervision, directly optimizing physical dynamics over the entire temporal sequence from the beginning is unstable. To address this issue, we introduce a Sliding Temporal Window that progressively expands the active temporal range used for training. Before this window spans the full sequence, we perform Short-Horizon Physics Learning, where the model samples timesteps within the currently active window and focuses on learning stable short-range dynamics. Once the window covers the full sequence, we transition to Long-Horizon Physics Learning, where the model prioritizes later timesteps while preserving the dynamics learned from earlier timesteps.

Sliding Temporal Window. Let T denote the maximum training timestep and let n be the physics-training iteration after Gaussian warm-up. The active horizon is defined as

$$H(n) = \min\left(T,\ s_H\left(\left\lfloor \frac{n-1}{M} \right\rfloor + 1\right)\right), \tag{8}$$

where s_H is the window expansion size and M is the expansion interval. At iteration n, timesteps are sampled only from the active temporal window $\mathcal{T}_n = \{0, \ldots, H(n)-1\}$.

Short-Horizon Physics. Before the temporal window spans the full sequence, we train the neural dynamics field f_θ within the active temporal window. At each iteration, we sample a target timestep $\tau \in \mathcal{T}_n$ and roll out the canonical Gaussian configuration $\mathcal{G}_0$ to the target timestep by repeatedly applying the second-order Runge–Kutta integration operator Φ_{RK2}. This rollout process is defined as:

$$\mathcal{G}_\tau = \text{Rollout}(\Phi_{\text{RK2}}(\mathcal{G}_0; f_\theta(\mathbf{x}_t, t), \Delta t), \tau) \tag{9}$$

The rollout operator recursively applies the integration operator Φ_{RK2}. Starting from $\mathcal{G}_0$, each intermediate Gaussian configuration $\mathcal{G}_t$ is advanced to $\mathcal{G}_{t+\Delta t}$ by Φ_{RK2}. At each step, the neural dynamics field f_θ predicts the physical quantities of the current Gaussian primitives, and these quantities are used by the integration operator to update their positions and orientations. This short-horizon stage is applied until the progressively expanding temporal window covers the full sequence. Although the window gradually increases over training, earlier timesteps receive more frequent supervision throughout this stage. Consequently, the model is encouraged to first learn stable short-horizon dynamics.

Long-Horizon Physics Learning Once the temporal window covers the full sequence, we switch to long-horizon physics learning. Although the short-horizon stage provides stable supervision for early timesteps, the dynamics of later timesteps remain relatively under-trained. If we continue to sample target timesteps uniformly from the entire sequence, the overall training pipeline becomes biased toward early timesteps. As a result, the model fail to sufficiently learn the physics for late-timestep prediction.

To address this issue, we introduce *late-horizon sampling*. Instead of sampling from the full temporal range, we restrict the target timestep to the late temporal window $\mathcal{T}_{\text{late}} = \{t', t'+1, \ldots, T\}$, where t' denotes the transition timestep between the early and late horizons. At each iteration, we sample $\tau_{\text{late}} \in \mathcal{T}_{\text{late}}$, thereby increasing the amount of supervision assigned to late timesteps and encouraging the model to better learn long-range physical dynamics.

However, training only with late-horizon samples can cause the model to forget the dynamics learned for early timesteps. To mitigate this issue, we introduce *early-horizon replay regularization*. During rollout, we additionally sample K timesteps from the early temporal interval $\mathcal{T}_{\text{early}} = \{0, 1, \ldots, t'-1\}$. we sample K early timesteps $\{\tau_{\text{early},k}\}_{k=1}^{K}$ from $\mathcal{T}_{\text{early}}$ and apply the rendering losses at these timesteps. The long-horizon training objective is defined as:

$$\mathcal{L}_{\text{long}} = \mathcal{L}(\mathcal{G}_{\tau_{\text{late}}}, c_{\text{late}}) + \lambda_{\text{early}} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\mathcal{G}_{\tau_{\text{early},k}}, c_{\text{early},k}) \tag{10}$$

where λ_{early} controls the strength of the early-horizon replay regularization. This objective increases supervision on late timesteps while preserving the short-horizon dynamics learned during the earlier stage.

4. Experiments

4.1. Experimental Setup

Datasets and Metrics. Following TRACE [7] and FreeGave [8], we consider dynamic multi-view datasets that are designed to evaluate not only reconstruction quality but also future frame extrapolation. We evaluates physical dynamics model on two existing datasets from NVFi [5]: Dynamic Object dataset, Dynamic Indoor Scene dataset. The Dynamic Object dataset contains 6 dynamic object sequences captured by 15 calibrated cameras, while the Dynamic Indoor Scene dataset consists of 4 indoor scenes with multiple moving objects captured by 25 calibrated cameras. For

Table 1. Quantitative comparison on the Dynamic Objects [5] and Dynamic Indoor Scene [5] under sparse-view settings. We report PSNR, SSIM, and LPIPS for 3-view and 5-view training settings.

Method	Dynamic Objects						Dynamic Indoor Scene					
	3-view			5-view			3-view			5-view		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
TRACE+OmniVGGT	24.725	0.956	0.048	25.229	0.963	0.042	**22.758**	0.733	0.227	**23.709**	0.799	0.177
FreeGave+OmniVGGT	23.700	0.942	0.062	24.127	0.943	0.069	22.357	0.699	0.260	23.383	0.797	0.187
SparseDynGS	**26.019**	**0.965**	**0.036**	**26.280**	**0.969**	**0.033**	22.459	**0.740**	**0.216**	23.516	**0.822**	**0.157**

both datasets, we construct sparse-view settings by selecting either 3 or 5 training views. We choose validation cameras such that the evaluation includes both interpolation and extrapolation viewpoints with respect to the selected training views, as illustrated in Fig. 2. For the Dynamic Object dataset, we use 5 validation views in the 3-view setting and 7 validation views in the 5-view setting. For the Dynamic Indoor Scene dataset, we use 4 validation views for both the 3-view and 5-view settings. We evaluate all methods using standard image quality metrics, including peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS).

Implementation Details. Following the protocol of NVFi [5], we normalize each video sequence to the temporal range $[0, 1]$. Frames with $t \leq 0.75$ are used for training, while future frames in the interval $(0.75, 1.0]$ are used for extrapolation evaluation. During the warm-up stage, we set the weights for depth and opacity regularization to $\lambda_{\mathrm{d}} = 5 \times 10^{-3}$ and $\lambda_{\mathrm{o}} = 10^{-2}$ for all experiments. For Progressive Physics Learning, we set the window expansion size to $s_H = 4$ and the expansion interval to $M = 500$ in the Sliding Temporal Window. In the Short-Horizon Physics Learning, we use the same neural dynamics network architecture f_θ as TRACE [7] and set the integration step to $\Delta t = 2$ for rollout. In the Long-Horizon Physics stage, we set the transition timestep for *late-horizon sampling* to $t' = 35$, use $K = 4$ early timesteps for *early-horizon replay regularization*, and set $\lambda_{\mathrm{early}} = 0.5$.

Baselines. We use TRACE [7] and FreeGave [8], two 3DGS-based methods that learn physical dynamics, as our main baselines. Since these methods were originally evaluated under dense multi-view settings, they rely on SfM-based initialization, which becomes unreliable in sparse-view scenarios. For a fair comparison under sparse-view inputs, we additionally construct strengthened baselines by replacing their SfM initialization with the OmniVGGT-based initialization used in our method. We denote these variants as TRACE+OmniVGGT and FreeGave+OmniVGGT, respectively.

4.2. Results & Analysis

Table 1 reports the quantitative comparison under sparse-view settings on the Dynamic Objects [5] and Dynamic Indoor Scene [5] datasets. On Dynamic Objects [5], SparseDynGS consistently outperforms both TRACE+OmniVGGT and FreeGave+OmniVGGT across all metrics in both 3-view and 5-view settings. In the 3-view setting, SparseDynGS improves PSNR from 24.725 to 26.019 and reduces LPIPS from 0.048 to 0.036 compared to TRACE+OmniVGGT. A similar trend is observed in the 5-view setting, where SparseDynGS achieves the best PSNR, SSIM, and LPIPS. These results indicate that our method improves sparse-view physical dynamics learning beyond the gain obtained from OmniVGGT initialization alone.

On Dynamic Indoor Scene [5], SparseDynGS shows stronger structural and perceptual quality, achieving the best SSIM and LPIPS in both 3-view and 5-view settings. Although TRACE+OmniVGGT obtains slightly higher PSNR, SparseDynGS achieves lower LPIPS and higher SSIM, suggesting that our method produces more coherent geometry and visually faithful dynamics under challenging indoor scenes. This is particularly meaningful because Dynamic Indoor Scene contains more complex multi-object motion, making sparse-view dynamics learning more difficult.

The qualitative results in Fig. 4 further illustrate the advantage of SparseDynGS in future-time extrapolation. In the Bat and Whale examples, the baselines produce visible motion blur and distorted dynamic regions, especially around wings and tails. SparseDynGS better preserves the object silhouette and maintains sharper local structures. In the Darkroom scene, FreeGave+OmniVGGT introduces strong ghosting artifacts and TRACE+OmniVGGT still shows noticeable distortion in the highlighted foreground region, whereas SparseDynGS produces a cleaner dynamic region with fewer artifacts. These visual results are consistent with the quantitative improvements in SSIM and LPIPS, indicating that SparseDynGS provides more coherent and perceptually faithful extrapolations.

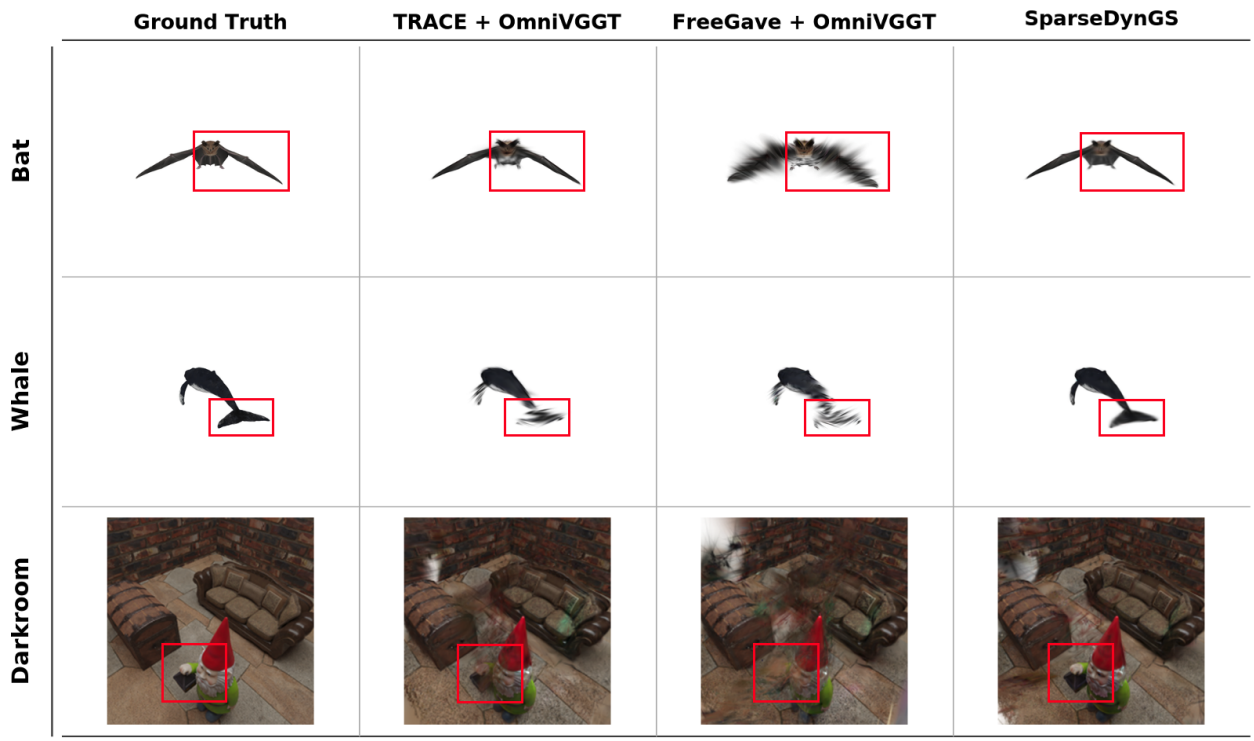

Figure 4. Qualitative results of future-time extrapolation on the Dynamic Objects Dataset (Bat and Whale) and the Dynamic Indoor Scene Dataset (Darkroom). Red boxes highlight dynamic regions where extrapolation errors are most visible. SparseDynGS preserves sharper object geometry and more coherent future motion compared with TRACE+OmniVGGT and FreeGave+OmniVGGT, which often produce motion blur, ghosting, or distorted dynamic structures.

4.3. Ablation Study

In this section, we conduct ablation studies to analyze the contribution of each component in SparseDynGS. All ablations are performed on the Dynamic Objects [5] under the 3-view training setting.

Regularized Gaussian Refinement. We analyze the effect of the proposed regularization terms during the warm-up stage. As shown in Table 2, opacity sparsity regularization substantially reduces redundant Gaussian primitives, decreasing the number of Gaussians from 487,369 to 9,255, corresponding to a 98.10% reduction. Despite using significantly fewer Gaussians, opacity regularization improves PSNR by 5.737 dB, SSIM by 0.039, and reduces LPIPS by 0.049 compared to the vanilla warm-up. Adding depth regularization further improves PSNR to 23.982 while maintaining a compact representation with only 8,957 Gaussians. These results indicate that our regularized refinement produces a compact and reliable Gaussian representation for subsequent physics learning.

Progressive Physics Learning. We ablate the key components of Progressive Physics Learning in Table 3. First, we evaluate the effect of the window expansion size s_H. The best performance is achieved with $s_H = 4$, suggesting that the expansion size plays an important role in progressive physics learning. Smaller expansion sizes keep the model concentrated on short-horizon dynamics for a prolonged period, thereby delaying supervision for later timesteps. In contrast, larger expansion sizes introduce long-horizon rollout supervision before short-horizon dynamics are sufficiently stabilized, which can make physics learning less stable. Second, we compare full-horizon sampling with the proposed late-horizon sampling. Full-horizon sampling denotes the strategy of sampling a target timestep from the entire temporal range at each iteration in Long-Horizon Physics Learning. the temporal window covers the full sequence. Late-horizon sampling improves PSNR from 25.855 to 26.019 and reduces LPIPS from 0.038 to 0.036, showing that focusing supervision on later timesteps is important for learning long-range physical dynamics. Finally, we analyze the number of early-horizon replay samples K. Using $K = 4$ achieves the best performance across all metrics. A smaller replay size provides limited coverage of early timesteps, making it less effective at preserving short-horizon dynamics. Increasing the replay size beyond $K = 4$ does not further improve performance, suggesting that additional early-horizon samples provide diminishing

Table 2. Ablation study on the warm-up regularization strategy under the 3-view training setting on DynObjects [5]. We report average results over all the objects.

	PSNR ↑	SSIM ↑	LPIPS ↓	# Gaussians
No Regularization	18.179	0.895	0.116	487,369
Opacity	23.916	**0.934**	**0.067**	9,255
Opacity + Depth	**23.982**	**0.934**	0.068	**8,957**

Table 3. Ablation study of Progressive Physics Learning on-DynObjects [5] under the 3-view traning setting. We analyze the window expansion size s_H, late-horizon sampling, and the number of early-horizon replay samples K.

Setting	PSNR ↑	SSIM ↑	LPIPS ↓
Window expansion size			
$s_H = 2$	25.344	0.962	0.038
$s_H = 3$	25.778	0.964	0.038
$s_H = 4$	**26.019**	**0.965**	**0.036**
$s_H = 5$	25.837	0.963	0.039
Late-horizon sampling			
Full-horizon	25.855	0.964	0.038
Late-horizon	**26.019**	**0.965**	**0.036**
Early-horizon replay size			
No Replay	25.788	0.962	0.039
$K = 2$	25.978	0.964	0.037
$K = 4$	**26.019**	**0.965**	**0.036**
$K = 6$	25.955	0.964	0.038

returns and may introduce constraints that are less aligned with late-horizon optimization.

5. Conclusion

We presented SparseDynGS, a framework for learning physically grounded Gaussian dynamics from sparse-view videos—a practical setting that has been largely overlooked by existing physics-aware 3DGS methods, which are typically developed and evaluated under dense multi-view supervision. To overcome the unreliable geometry that arises when only a few calibrated views are available, we replaced SfM-based initialization with a feed-forward reconstruction prior from OmniVGGT and introduced Regularized Gaussian Refinement, which jointly applies opacity sparsity and depth supervision to obtain a compact and geometrically consistent representation for subsequent dynamics learning. Building on this representation, we proposed Progressive Physics Learning, a temporal curriculum based on a Sliding Temporal Window that first stabilizes short-horizon dynamics and then extends to longer horizons through late-horizon sampling and early-horizon replay regularization. Experiments on the Dynamic Objects and Dynamic Indoor Scene datasets under both 3-view and 5-view settings demonstrate the effectiveness of our approach. SparseDynGS consistently outperforms the strengthened TRACE+OmniVGGT and FreeGave+OmniVGGT baselines on object-centric scenes and remains competitive on the more challenging indoor scenes, where it achieves the best SSIM and LPIPS. Our ablation studies further confirm that each component contributes meaningfully: opacity regularization removes over 98% of redundant Gaussians while substantially improving rendering quality, and both late-horizon sampling and early-horizon replay are important for learning stable long-range physical dynamics. Despite these results, several limitations remain. The quality of our dynamics learning depends on the geometric prior provided by the feed-forward initialization, which can degrade for highly cluttered or textureless indoor scenes, and our particle-based formulation primarily targets rigid-body motion rather than strongly non-rigid or deformable dynamics. Extending SparseDynGS toward even sparser or monocular inputs, richer deformation models, and longer prediction horizons in real-world captures is a promising direction for future work.

References

[1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[2] Jiemin Fang, Xinggang Wang, and Matthias Nießner. Fast dynamic radiance fields with time-aware neural voxels. *ACM Transactions on Graphics*, 2022.

[3] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1, 3

[5] Jinxi Li, Ziyang Song, and Bo Yang. Nvfi: Neural velocity fields for 3d physics learning from dynamic videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 5, 6, 7, 8

[6] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20775–20785, 2024. 3

[7] Jinxi Li, Ziyang Song, and Bo Yang. Trace: Learning 3d gaussian physical dynamics from multi-view videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2508.09811. 1, 2, 3, 4, 5, 6

[8] Jinxi Li, Ziyang Song, Siyuan Zhou, and Bo Yang. Freegave: 3d physics learning from dynamic videos by gaussian velocity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 3, 4, 5, 6

[9] Tianye Li, Miroslava Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 1

[10] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[11] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[12] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[13] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[14] 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 *European Conference on Computer Vision (ECCV)*, 2020. 1

[15] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 3

[16] 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 Transactions on Graphics*, 2021. 1, 3

[17] Haosong Peng, Hao Li, Yalun Dai, Yushi Lan, Yihang Luo, Tianyu Qi, Zhengshen Zhang, Yufeng Zhan, Junfei Zhang, Wenchao Xu, and Ziwei Liu. OmniVGGT: Omni-modality driven visual geometry grounded transformer. *arXiv preprint arXiv:2511.10560*, 2025. 2, 3, 4

[18] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[19] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3

[20] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3

[21] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, pages 145–163. Springer, 2024. 3