Editable dynamic scenes using physics-integrated gaussian splatting

Taekbeom Lee Yeonkyu Lee Hongmo Jung
Seoul National University
{ltb1128, yeonkyu011, jjhm2910}@snu.ac.kr

Abstract

In this paper, we present a novel approach to dynamic scene modeling and editing by integrating physics-based simulations with Gaussian Splatting. Existing methods that directly model deformation often fall short in rendering quality for fast-moving scenes and demand significant computational resources. To address these limitations, we leverage the explicit representation and real-time rendering capabilities of Gaussian Splatting, further enhanced by embedding physical properties to model underlying dynamics. Our method reconstructs dynamic scenes from multi-view image sequences, capturing both geometric and physical properties to enable realistic, complex motion modeling and editing. We evaluate our approach on 2 datasets, showing the possibility of a difficult task of modeling geometry and physical properties from scratch using only multi-view images captured from a single dynamic scene. We expect further development of training process and exploring best hyperparameters can significantly improve current performance.

1. Introduction

The synthesis and manipulation of dynamic environments in virtual reality, gaming, and film industries are vital for creating immersive experiences. Traditional methods relied on multi-view camera settings to model dynamic environments, offering high-quality outputs but often falling short in practicality due to their extensive hardware requirements. The advent of Dynamic Neural Radiance Fields (NeRF) has facilitated the modeling of dynamic scenes with single-camera setups. However, while enabling easier capture, these methods often struggle with rendering quality, especially in fast-moving scenes, and are hindered by significant computational or memory demands.

Recently, Gaussian Splatting has emerged as a promising solution, offering rapid training times and real-time rendering capabilities. This method, with its explicit representation, has shown potential in more accurately depicting dynamic scenes compared to implicit methods like NeRF. Despite its advantages in modeling and synthesizing dynamic scenes, the editing of such environments—particularly in an interactive or complex manner—remains underexplored. Existing studies have generally focused on rudimentary tasks such as object removal or color changes [6, 10], lacking in more sophisticated editing functionalities.

In response to these limitations, recent advancements have introduced text-based dynamic environment editing using Gaussian Splatting [11], showcasing impressive potential. However, these approaches predominantly target high-level, holistic edits [2] rather than precise, localized motion control. To address this, new techniques such as SC-GS [4] have been developed, which learn explicit elements (e.g., control points) from dynamic scenes that can manipulate transformations of Gaussians to model plausible motions. These techniques, however, have been constrained to minor movements and have struggled with complex motions [16].

To address the aforementioned limitations in dynamic environment editing, techniques embedding physical properties, like PhysGaussian, have been introduced. These approaches use well-reconstructed 3D Gaussian models as inputs, which, while effective, limit their practicality due to the specificity of the required input data. Recognizing the need for a more adaptable and widely applicable solution, we propose a new method that integrates the restoration of given dynamic scenes and physics simulations directly from multi-view image inputs of dynamic scenes. This approach not only enhances the realism and complexity of the modeled motions but also broadens the potential applications of dynamic environment modeling and editing. Through this innovative integration, our methodology aims to substantially improve the capacity for realistic and complex motion editing in highly dynamic settings. Our contributions are as follows:

- We try to enhance dynamic scene modeling by embedding physical properties into the Gaussian Splatting framework, allowing for realistic and complex motion generation directly from multi-view image sequences.
- We propose novel multi-stage joint training schemes

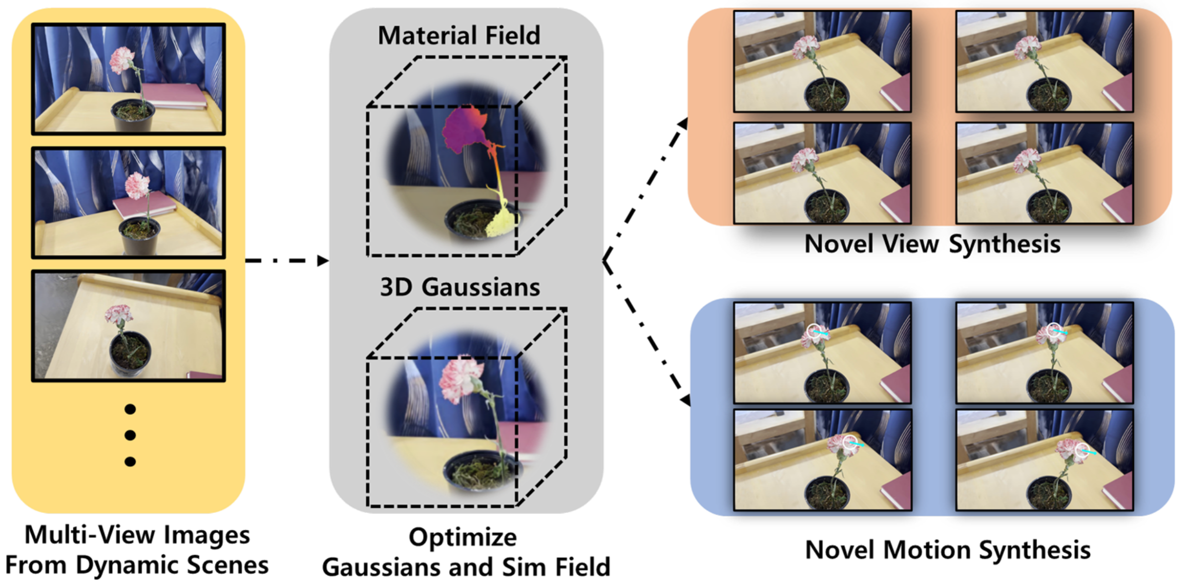

Figure 1. Problem settings of our method. We take multi-view images captured from a single dynamic scene as input, jointly optimize material properties and 3d gaussian, and the learned model is used to synthesize the same dynamic scene in a novel view and generate novel motion.

to stabilize training geometry and physical properties which only take multi-view images of single dynamic episode as input.

2. Related Work

2.1. Dynamic gaussian splatting.

Recent advancements in dynamic scene modeling have seen significant contributions from dynamic Gaussian splatting techniques. These methodologies have fundamentally restructured the way spatial and temporal data are handled, distinguishing between canonical spaces learned through static representations and explicitly modeling temporal deformations over time. Several studies have introduced approaches that separate these spatial and temporal dimensions, learning a canonical space and then explicitly tracking deformations as they evolve [15]. However, the assumption of a canonical space presents limitations, particularly when modeling scenarios involving the emergence or disappearance of objects, and poses challenges for explicit motion modeling necessary for long-term tracking.

Dynmf [8] encoded spatiotemporal information within HexPlane voxels for each Gaussian, enhancing the precision of dynamic interactions captured. Meanwhile, [13] advanced the field by learning 4D primitives, facilitating a more comprehensive modeling of dynamic environments. These methods showcase the continuous evolution towards more sophisticated and precise dynamic scene representations, addressing the complexities inherent in environments with variable and emergent properties. However, the need for improvements in long-term tracking and handling non-permanent scene elements remains, signaling directions for future research in this domain.

2.2. Editing dynamic scenes.

Gaussian splatting-based methods have emerged as potent tools for dynamic scene manipulation due to their explicit nature, allowing for direct control over Gaussian elements. A notable application of this advantage is seen in the work by [17], where control signals are automatically extracted from individual parts of an object, according to user commands. This approach enables the re-animation of only the relevant parts of a dynamic scene, enhancing targeted intervention capabilities.

In another innovative approach, Control4D [11] utilized a text-to-image diffusion model to perform holistic 4D editing across dynamic scenes. This method leverages the power of 4D generative tasks, facilitating broad and integrative changes across time and space with minimal user input required for detailed specifications.

Further advancing the capabilities of GS, SC-GS [4] introduced the use of sparse control points to learn motions in a canonical space. This methodology allows for the manipulation of these control points, enabling motion editing that was previously challenging with conventional dynamic GS techniques. By providing a framework where users can directly influence motion parameters, this approach significantly extends the flexibility and applicability of GS for

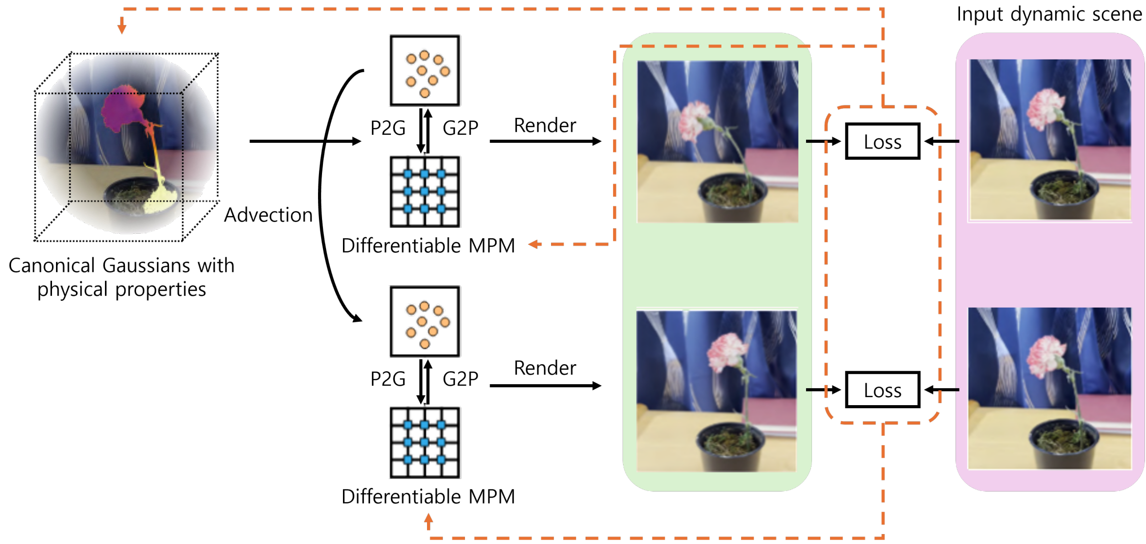

Figure 2. Overview of proposed method.

dynamic scene editing. These developments collectively enhance the utility and precision of GS in dynamic environment applications, setting a new standard for interactive and user-driven scene manipulation.

2.3. Physics-based deformation in 3dGS

Recent innovations in Gaussian Splatting have incorporated physics-based models to enhance the simulation and management of complex dynamic scenes. VR-GS [7] enhances virtual reality (VR) interactivity by dynamically adjusting Gaussian splats based on physical interactions and user inputs. This approach provides immersive experiences with physically plausible responses, aligning closely with real-world physics, thereby improving the user's sense of presence and engagement in virtual environments. The most closely related work to ours is PhysGaussian [14], which embeds physics-driven transformations directly into 3D Gaussians, allowing for the simulation of fluid dynamics and other complex deformations. It also supports the generation of highly dynamic scenes where physical laws dictate the movement and interactions of particles. While PhysGaussian leverages the advantages mentioned earlier and serves as a foundation for our method, it depends on a well-trained 3D Gaussian Splatting model. Additionally, it has not been tested for its capability to accurately generate complex motions based on the underlying physics from multi-view image sequences of dynamic scenes. In contrast, our proposed method effectively adjusts motions to adhere to physical laws even within dynamic environments.

3. Method

In our research, we focus on extracting both physical properties and geometric representation from dynamic scenes using posed multi-view image sequences and the corresponding time data. Our objective is to leverage this data to create complex and realistic user-interactive motions. Inspired by previous studies on modeling dynamic scenes by predicting canonical Gaussians and their deformations, as well as PhysGaussian [14] that treats Gaussian kernels as discrete particle clouds representing simulated continua, we propose a pipeline as illustrated in Fig. 2.

In the canonical field, we extend the properties defined by traditional Gaussian splatting—position, rotation, scale, opacity, and spherical harmonics coefficients—to include physics simulation properties such as Young's modulus, Poisson's ratio, and mass, following PhysDreamer. These parameters are optimized to minimize rendering loss with the given dynamic video, detailed further in Section 3.2

3.1. Preliminaries

3.1.1 3D Gaussian Splatting

3D Gaussian splatting explicitly represents a 3D scene using a set of unstructured Gaussian kernels $G = \{x_p, \sigma_p, A_p, C_p\}_{p\in\mathcal{P}}$, where x_p, α_p, Σ_p, sh_p represent the position, opacity, covariance, and spherical harmonics coefficients, respectively. The covariance matrix is decomposed as $\Sigma = RSS^TR^T$ for optimization, where R is a rotation matrix, and S is a diagonal matrix representing 3D scale. The presence of a Gaussian at a location x is expressed as:

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

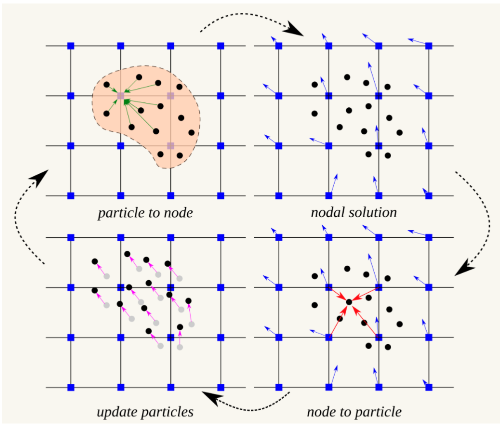

Figure 3. Material Point Method.

During rendering, 3D Gaussian splatting projects these Gaussians onto the image plane as 2D Gaussians, which are efficiently aggregated using α-blending to compute the color $C(u)$ of pixel u in real-time:

$$C(u) = \sum_{i \in N} \alpha_i SH(d_i, sh_i) \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{2}$$

Here, i denotes the index according to the z-depth order within the frame, and d_i represents the view direction from the camera origin to x_i. The parameters of the Gaussians are optimized using L1 and SSIM Loss between the rendered and ground truth images to enable realistic rendering. The explicit nature of the Gaussians also facilitates direct manipulation of the scene, including physics-integrated manipulation, which will be discussed in subsequent sections.

3.1.2 Material Point Method

The Material Point Method (MPM) is a numerical technique used for simulating the behavior of materials under various conditions, such as deformation, fracture, and other physical processes. It combines the advantages of both particle methods and grid-based methods to provide a comprehensive and flexible approach to computational material modeling [3, 5, 12].

In MPM, the material to be simulated is represented by a collection of material points, which carry information about the material's properties like mass, velocity, stress, and strain. These material points move through a background computational grid, which is used to solve the equations of motion. A schematic diagram is presented in Fig. 3. The process works as follows. First, Material points are initialized with material properties including volume V_p, mass m_p, position x_p^t, velocity v_p^t, deformation gradient F_p^t, and local velocity field gradient C_p^t at time step t. Then, particle-to-grid (P2G) transfers properties of the material points including mass and momentum to the background computational grid as follows.

$$m_i^t v_i^t = \textstyle\sum_p N(x_i - x_p^t)[m_p v_p^t + (m_p C_p^t - \frac{4}{(\Delta x)^2} \Delta t V_p \frac{\partial \psi}{\partial F} {F_p^t}^T)(x_i - x_p^t)] + \boldsymbol{f}_i^t \tag{3}$$

$m_i^t = \sum_p N(x_i - x_p^t) m_p$, where $N(x_i - x_p^t)$ is B-spline kernel. Then the transferred properties are used to solve equations of motion, typically using finite difference methods. Then grid-to-particle (G2P) process is performed to transfer the computed grid velocities, local velocity gradient, and deformation gradient.

$$v_p^{t+1} = \textstyle\sum_i N(x_i - x_p^t) v_i^t, \quad x_p^{t+1} = x_p^t + \Delta t v_p^{t+1} \tag{4}$$

$$C_p^{t+1} = \frac{4}{(\Delta x)^2} \textstyle\sum_i N(x_i - x_p^t) v_i^t (x_i - x_p^t)^T, \quad F_p^{t+1} = (\boldsymbol{I} + \Delta t C_p^{t+1}) F_p^t \tag{5}$$

After G2P process, material points are moved according to their velocities and computational grid is reset for next timestep.

3.2. Joint Training

In this section, we introduce our system design, which is depicted in Figure 2. Similar to various deformable Gaussian splitting approaches [8, 13, 15], we decouple static geometry from dynamic scenes and represent it in a canonical space. The parameters for this static geometry, referred to as canonical Gaussian parameters, are initialized following the method described in 4dGS [15]. These initialized Gaussians are coupled with physical parameters—Young's modulus $\boldsymbol{E} = [E_1, \cdots, E_P]$, Poisson's ratio $\boldsymbol{\nu} = [\nu_1, \cdots, \nu_P]$, and volume $\boldsymbol{V} = [V_1, \cdots, V_P]$ (hereafter collectively referred to as θ)—and include initial velocity field as an additional parameter to enable physical simulations. The initialization of these physical parameters and the Gaussians' initial velocity follow the protocols outlined in [18].

During each training iteration, Gaussians are sampled from the canonical Gaussians to perform physical simulations based on the Material Point Method (MPM). After each time step, rasterization is conducted using the resulting Gaussians to compare against images from the input dynamic sequence. The simulation between adjacent video frames can be expressed by the following equation:

$$x^{t+1}, v^{t+1}, F^{t+1}, C^{t+1} = \mathcal{S}(x^t, v^t, F^t, C^t, m\theta, \Delta t, N), \tag{6}$$

Following the simulation, the resulting Gaussians at each time step are rendered according to Eq. 2. The rendered image $\hat{I}_t$ is then subjected to a per-frame loss function as follows, which is used to update the parameters of the canonical Gaussians and the differentiable parameters of the MPM.

$$L^t = \lambda L_1(\hat{I}^t, I^t) + (1 - \lambda) L_{\mathrm{D-SSIM}}(\hat{I}^t, I^t), \tag{7}$$

where we set $\lambda = 0.1$ in our experiments.

Figure 4. Comparison of novel view dynamic video synthesis on 3 selected PAC-NeRF scenes

We have structured our framework for fast rendering and training using two triplanes and three-layer MLP for both material and initial velocity fields. Triplanes store features and these features are decoded through MLP to predict its field property. To ensure the spatial smoothness of the material field and initial velocity field, we incorporated a total variation regularization term for each 2D spatial plane.

$$L_{\text{tv}} = \sum_{i,j} \|u_{i+1,j} - u_{i,j}\|_2^2 + \|u_{i,j+1} - u_{i,j}\|_2^2, \tag{8}$$

where $u_{i,j}$ is a feature vector on the 2D plane. This regularization helps maintain a smooth gradient in spatial properties, enhancing the stability and accuracy of our simulations.

We employ a two-stage optimization process to ensure stable training instead of jointly learning the canonical Gaussian, initial velocity field, and physical properties from the start. In the first stage, we focus on training the canonical Gaussian and initial velocity field. Optimization uses only the first 4 frames to establish a stable base. Similar to the warm-up process in 4dGS [15], we avoid performing simulations during the initial optimization steps to prevent instability in learning. Instead, we optimize the canonical Gaussians first and then proceed to train them with the initial velocity field jointly. In the second stage, we fix the canonical Gaussian and initial velocity field and use the entire image sequence to optimize the physical properties. To prevent gradient explosion or vanishing, similar to the approach in PhysDreamer [18], we restrict the gradient flow to only the previous frame.

4. Experiment

We conducted experiments on two distinct tasks. The first task involves using a multi-view image sequence of a dynamic scene as input to render that scene from novel viewpoints (novel view dynamic video synthesis). The second task focuses on rendering new motions by applying external forces to the learned physical properties (novel motion generation).

4.1. Setup

4.1.1 Dataset

PAC-NeRF dataset. The PAC-NeRF dataset utilizes a photo-realistic simulation engine to simulate and render images of various objects under different lighting conditions. This dataset provides multi-view image sequences that capture significant shape transformations of different objects. The PAC-NeRF dataset supports a variety of material types, including elasticity (torus, bird), plasticine (Play-Doh, cat), Newtonian fluid (cream, toothpaste), and non-Newtonian fluid (droplet, letter). Each scene is captured from 11 uniformly distributed viewpoints on the upper hemisphere, with 14 time-synchronized frames captured from each viewpoint.

PhysDreamer dataset. In the PhysDreamer dataset, 8 real-world static scenes are provided, each accompanied by a multi-view image sequence and corresponding static 3D Gaussian sets. Only selected parts of these sets are manually specified for simulation. For training, dynamic scene videos are created by selecting 1-2 images from the image sequence and generating a 14-frame video for each image using the Stable Video Diffusion model [1]. Our method requires predicting both physical properties and geometry from video; therefore, the desired videos are captured from various viewpoints. Since the PhysDreamer dataset does not provide such videos, we rendered multi-view dynamic videos using the originally trained PhysDreamer model to construct the training data for both our method and the baseline.

Table 1. Novel view dynamic video synthesis result on 4 PAC-NeRF scenes

Methods	**Droplet**		**Cream**		**Torus**		**Cat**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
PAC-NeRF	32.93	0.987	35.87	0.992	32.86	0.985	32.60	0.991
Ours	21.84	0.976	24.88	0.981	19.90	0.971	28.91	0.991

4.1.2 Baselines

We compared our method against two baselines: PAC-NeRF [9] and PhysDreamer [18]. Similar to our approach, PAC-NeRF predicts the geometry and physical parameters of highly dynamic objects through multi-view videos. Like us, it uses the Material Point Method (MPM) for physics simulation to model the dynamic scene. However, PAC-NeRF differs in that it employs a hybrid representation combining Eulerian (static voxel NeRF) and Lagrangian (particles), whereas our method utilizes the same representation for both physics simulation and rendering. We conducted both qualitative and quantitative comparisons of novel view dynamic video synthesis performance between our method and PAC-NeRF using the PAC-NeRF dataset.

PhysDreamer uses a multi-view image dataset captured from static scenes to reconstruct a well-formed set of Gaussians, which it uses as known geometry to learn physical parameters through videos generated by diffusion. For a fair comparison with our method, which uses videos captured from dynamic scenes as training data, we extracted canonical Gaussians from the 4dGS [15] model trained on such dynamic videos to serve as the geometry for training the PhysDreamer model. We qualitatively compared the performance of novel motion generation using the PhysDreamer dataset.

4.1.3 Metrics

For the novel view dynamic video synthesis task, we utilized widely used metrics in novel view synthesis such as PSNR and SSIM to report performance. For the novel motion synthesis task, while PhysDreamer employs a user study with a two-alternative forced choice (2AFC) protocol to evaluate the realism of the motions perceived by humans. However, such metric is hard to conduct in this project. Therefore, we have relied solely on qualitative results to assess the performance of our generated motions.

4.2. Implementation details

During the first 100 epochs, we focused solely on learning the canonical Gaussian to ensure stable acquisition of position and shape. Subsequently, we jointly trained the canonical Gaussian and initial velocity field for 200 epochs. With the canonical Gaussian and initial velocity field fixed, we then trained the material field over another 200 epochs. The initial velocity field and material field are each modeled using triplanes of resolutions 8^3 and 24^3, respectively, followed by a 3-layer MLP.

For simulations using the Material Point Method (MPM), the resolution of the computational grid was set to 32^3 during the first stage and 64^3 in the second stage. To achieve precise motion, we conducted simulations with 64 sub-steps between consecutive video frames in the first stage and 768 sub-steps in the second stage. To address the memory consumption issues arising from the large number of sub-steps, we implemented simulation state checkpointing along with re-computation during gradient backpropagation.

4.3. Results

4.3.1 Novel view dynamic video synthesis on PAC-NeRF dataset

Firstly, we perform novel view dynamic video synthesis to test if our joint training method effectively learns the geometry and physical properties of the given dynamic scene. This task involves re-simulating the input dynamic scene using the trained canonical Gaussians, initial velocity field, and material property field, and rendering the entire image sequence from new viewpoints not used in the training data. Currently, PAC-NeRF [9] is the only other study that simultaneously learns geometry representation and material properties using only a multi-view image sequence of the dynamic scene as input. Therefore, we compare our performance with PAC-NeRF using the same dataset.

Figure 4 compares the rendering results for objects with three different types of materials. Our method appears to render well in the initial frames but seems to progressively disappear as the object moves away from its original position. While we have not fully analyzed the cause of this issue yet, one possible reason might be that we mistakenly set the simulation bounds as the boundary of the initial sfm point cloud, without considering the volume traversed by the object throughout the frames. This may cause renderings to fail once the Gaussians exceed these bounds. As shown in Table 1, the quantitative results also indicate that our performance is not as good as PAC-NeRF's. However, we anticipate that addressing the aforementioned issue and further tuning the training and simulation-related hyperparameters could significantly improve our results.

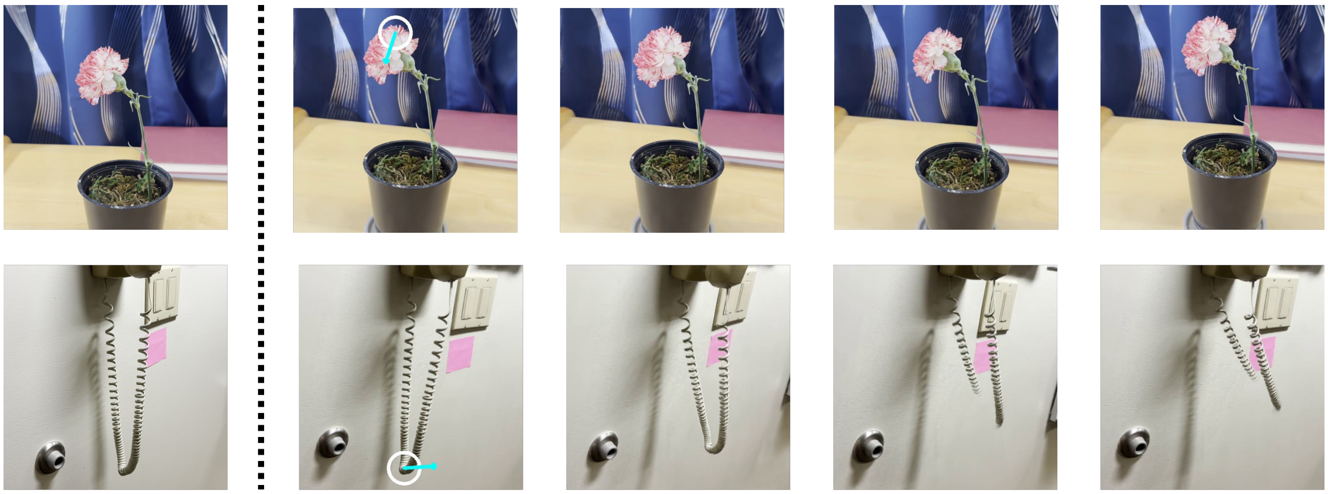

Figure 5. Initial state and novel dynamic scene generated by user interaction on 2 selected PhysDreamer scenes

4.3.2 Novel motion synthesis on PhysDreamer dataset

We perform novel motion synthesis to verify that our jointly learned geometry and material properties are not merely overfitted to the input motion but accurately reflect the actual properties of the objects. This task also highlights the benefits of physics-integrated dynamic scene modeling, emphasizing the generation of physically and visually plausible new motions. In this task, we do not use the learned initial velocity field specific to the input video. Instead, we apply forces at user-specified locations to generate motions, while the learned canonical Gaussian and material property fields remain unchanged. Since there is no ground truth data for evaluating the newly deformed scenes - except for the lattice deformation benchmark rendered through desired interactions in the simulation engine by PhysGaussian [14], which has not been released - we only report qualitative results. Additionally, as there are no existing works that successfully perform novel motion synthesis under a fair setting, we do not compare our results with other methods.

Figure 5 illustrates the resulting motions generated by applying forces to two different types of objects, depicted across several frames. In the frames where forces are applied, the points of force application, as well as the magnitude and direction of the forces, are displayed. Although both objects move with a slightly larger amplitude than expected considering their properties and the applied forces, the motions are otherwise physically and visually plausible to some extent. Detailed results can be found in the supplementary video. However, for the second object, the telephone, parts of its cord appear to disappear. One possible reason could be that, as mentioned in Section 4.3.1, some parts exceed the pre-set simulation bounds due to larger than anticipated amplitudes, resulting in their disappearance.

5. Conclusion

In this study, we proposed a novel approach to enhancing dynamic scene modeling and editing by integrating physics-based simulations with Gaussian Splatting. Our method effectively addresses the limitations of existing techniques by enabling complex, realistic motion generation and precise editing capabilities directly from multi-view image sequences. Through the combination of canonical Gaussian parameters and physical simulation properties, our approach partially shows potential to improve the representation and manipulation of dynamic environments.

Our experiments demonstrate the capability of our method to reconstruct and synthesize dynamic scenes. In the task of novel view dynamic video synthesis using the PAC-NeRF dataset, our results show promising initial frame renderings, though further refinement is needed to address issues related to disappearing objects. The qualitative analysis of novel motion synthesis on the PhysDreamer dataset highlights the effectiveness of our approach in generating physically plausible motions, although challenges remain in controlling a plausible amount of movement and solving disappearing parts.

Overall, our integration of physics-based properties with Gaussian Splatting represents a potential in dynamic scene modeling. Future work will focus on optimizing the training process, refining simulation parameters, and expanding the applicability of our method to a broader range of dynamic scene datasets.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 5

[2] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 1

[3] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 4

[4] 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. *arXiv preprint arXiv:2312.14937*, 2023. 1, 2

[5] Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*, pages 1–52. 2016. 4

[6] Dadong Jiang, Zhihui Ke, Xiaobo Zhou, and Xidong Shi. 4d-editor: Interactive object-level editing in dynamic neural radiance fields via 4d semantic segmentation. *arXiv preprint arXiv:2310.16858*, 2023. 1

[7] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. *arXiv preprint arXiv:2401.16663*, 2024. 3

[8] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. *arXiv preprint arXiv:2312.00112*, 2023. 2, 4

[9] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. *arXiv preprint arXiv:2303.05512*, 2023. 6

[10] Yi-Ling Qiao, Alexander Gao, and Ming Lin. Neuphysics: Editable neural geometry and physics from monocular videos. *Advances in Neural Information Processing Systems*, 35:12841–12854, 2022. 1

[11] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. *arXiv preprint arXiv:2305.20082*, 2023. 1, 2

[12] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 4

[13] 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. *arXiv preprint arXiv:2310.08528*, 2023. 2, 4

[14] Tianyi Xie, Zeshun Zong, Yuxin Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023. 3, 7

[15] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2, 4, 5, 6

[16] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 1

[17] Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. Cogs: Controllable gaussian splatting. *arXiv preprint arXiv:2312.05664*, 2023. 2

[18] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. *arXiv preprint arXiv:2404.13026*, 2024. 4, 5, 6