Phys-GS Lite: Material-Aware Dynamic Gaussian Splatting via Semantic Labeling and Lightweight Physics

Tarun Gangadhar Vadaparthi

Abstract

Static scenes can now be rendered efficiently and realistically thanks to recent developments in 3D Gaussian Splatting (3DGS). These methods, however, are still not very good at capturing material-dependent motion or physically realistic dynamics. This project introduces **Phys-GS Lite**, a lightweight prototype that adds *dynamic behavior* and *material awareness* to 3DGS. A sinusoidal wind field and graph Laplacian smoothing are used to produce coherent, physically-inspired deformation after the method introduces semantic separation of rigid and deformable regions using geometry-driven heuristics.Deformable regions show wave-like dynamics while rigid regions stay static, as demonstrated by evaluation on a ship-and-water scene. The findings show that physics-inspired extensions to Gaussian Splatting can be accomplished in a lightweight way that is appropriate for limited environments like Google Colab, even in the absence of extensive neural training. In dynamic Gaussian splatting, Phys-GS Lite presents a useful proof-of-concept that connects geometry, physics, and semantic reasoning.

1 Introduction

A cutting-edge method for effectively rendering intricate scenes that offers real-time performance and high-quality reconstructions is three-dimensional Gaussian Splatting (3DGS). Its effectiveness comes from representing a scene as a set of directly rasterizable anisotropic Gaussian primitives, which eliminates the need for bulky volumetric grids or intricate neural networks. Conventional 3DGS is powerful for static scenes, but it is fundamentally limited when it comes to representing material-dependent behavior or physical dynamics.

Since many real-world scenes involve deformable and environmental interactions, dynamic extensions to 3DGS are becoming more and more popular in computer graphics and vision. Current approaches to dynamics frequently rely on neural simulation pipelines or computationally costly training regimes, which restrict accessibility and lightweight deployment. Between physically grounded dynamic behavior that can be demonstrated without extensive resources and photorealistic rendering, there is a practical gap.

This project, called **Phys-GS Lite**, aims to investigate a lightweight proof-of-concept for material-aware dynamics in the Gaussian Splatting framework. The method distinguishes between deformable and rigid regions by introducing semantic separation through geometric heuristics, as opposed to depending on deep semantic segmentation networks or physics solvers. The deformable regions are then subjected to a sinusoidal wind field, and locally coherent deformations are guaranteed by graph Laplacian smoothing. The end effect is a dynamic visualization in which deformable materials move like waves while rigid structures stay stable.

This contribution is not meant to be a system that is ready for production, but rather a conceptual demonstration. However, the project shows how Gaussian Splatting can be greatly enhanced by even basic semantic cues and lightweight physical models, indicating new possibilities for combining geometry, physics, and semantics in dynamic scene reconstruction.

2 Method

Three main steps make up the design of Phys-GS Lite: applying physics-inspired motion through a wind field, semantically identifying deformable regions, and fine-tuning this motion using graph Laplacian smoothing. Each step is thoroughly explained in this section, with a focus on the method's effectiveness, interpretability, and simplicity.

2.1 Semantic Labeling of Deformable Regions

The ability to distinguish between rigid and deformable materials is a crucial prerequisite for dynamic rendering. For lightweight labeling, this project proposes a geometric heuristic in place of heavy semantic segmentation networks. Using scene statistics like radial distance from the scene center and percentile thresholds on the z-axis, the process starts by defining a bounding region that corresponds to the water basin. This depicts the water surface's roughly horizontal plane.

A hull-exclusion refinement is used to prevent the ship's base hull from being mistakenly classified as deformable. After identifying rigid splats above the waterline, water-labeled splats that are too close to these rigid core regions are filtered out using a nearest-neighbor search. The Gaussian splats are divided into two disjoint sets by the resulting semantic mask:

$$m_i \in \{0,1\}, \quad \begin{cases} m_i = 1 & \text{if splat } i \text{ belongs to the deformable subset } \mathcal{D} \text{ (water)}, \\ m_i = 0 & \text{if splat } i \text{ belongs to the rigid subset } \mathcal{R} \text{ (ship)}. \end{cases}$$

This binary labeling offers a lightweight, scene-adaptable, material-aware distinction.

2.2 Wind Field Deformation

A sinusoidal wind field is used to introduce dynamics after deformable regions have been identified. For each deformable splat $\mathbf{x}_i \in \mathcal{D}$, displacement at time t is defined as:

$$\Delta \mathbf{x}_i(t) = A \sin(\omega t + \phi + k x_i)\, \alpha_i\, \hat{w}.$$

The amplitude of oscillation is controlled by A, the angular frequency is ω, the global phase is ϕ, a spatial phase offset is introduced across the x-axis by k, and the normalized wind direction vector is $\hat{w}$. The water surface will oscillate periodically in accordance with wind-driven motion thanks to this formulation.

By modulating displacement according to each splat's vertical position, the coefficient α_i reduces motion close to the hull to avoid unrealistic interaction and permits larger oscillations in open water areas. Without the use of fluid solvers, this spatially varying modulation ensures believable material behavior.

2.3 Graph Laplacian Smoothing

Even though the sinusoidal field creates coherent oscillations, each splat's independent displacement may result in motion that is noisy or irregular. This is addressed by using graph Laplacian smoothing to enforce local neighborhood consistency. On $\mathcal{D}$, a k-nearest neighbor graph is created, and each splat is iteratively updated in accordance with:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda \left(\frac{1}{k} \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j - \mathbf{x}_i \right),$$

where λ is the smoothing strength and $\mathcal{N}(i)$ denotes the neighbor set of point i. By encouraging each point to match the average of its neighbors, this operation diffuses local irregularities and creates surfaces that resemble smooth waves. The model strikes a balance between excessively smoothed motion and highly dynamic but noisy displacements by varying λ.

2.4 Rendering and Visualization

Standard Gaussian Splatting rasterization is used to render the final positions of both rigid and deformable splats. While deformable splats are depicted clearly to emphasize dynamics, rigid splats are kept in neutral hues to highlight structural stability. With two complementary visualization modes, animations are produced over a number of frames. A fixed close-up view highlights local water motion, while an orbiting camera offers a global perspective on rigid-versus-deformable separation. Furthermore, a heatmap mode provides an interpretable analysis of motion intensity across the deformable surface by encoding the displacement magnitude per splat.

3 Experiments

3.1 Setup

The Google Colab environment was used to develop the Phys-GS Lite prototype, which was fully implemented in Python. Gaussian Splatting models were imported from `.ply` files produced by reconstruction pipelines, and standard scientific libraries were used for numerical computation and visualization. A ship-and-water scene was used for the experiments because it naturally includes both rigid (ship) and deformable (water) regions. Because the project focused on lightweight feasibility rather than large-scale training, the environment was limited to a single NVIDIA T4 GPU. At 24 frames per second, each animation had between 100 and 150 frames. This configuration guaranteed the method's reproducibility and accessibility within reasonable computational budgets.

3.2 Semantic Labeling Evaluation

Determining whether geometry-driven heuristics could accurately distinguish between deformable water and rigid ship geometry was the first experimental goal. The water basin was isolated using a bounding region and z-thresholding, and then mislabeled splats close to the ship's base were removed using a hull-exclusion step. According to quantitative data, roughly 25–35% of the splats in the tested scene were categorized as deformable, which is in line with the proportion of water to rigid geometry. Semantic visualization qualitatively verified that the rigid mask maintained the ship's structural integrity, while the deformable mask matched the anticipated water surface region. This demonstrated that interpretable material-aware separation was possible even in the absence of pretrained semantic networks.

3.3 Dynamic Behavior under Wind Field

The deformable splats were subjected to sinusoidal wind fields in order to assess the physics-inspired deformation. The displacement amplitude A was varied in the experiments from 0.01 to 0.03 times the water bounding box's diagonal. Larger amplitudes created noticeable oscillations throughout the basin, whereas smaller amplitudes created faint ripples. To simulate faster or slower environmental conditions, the temporal frequency ω was changed. The findings demonstrated that the water surface reacted uniformly to changes in parameters: raising ω reduced the oscillation period, while adjusting the spatial phase term kx regulated the wave's wavelength and direction of propagation. These findings show that interpretable control of dynamic behavior is possible with the suggested formulation.

3.4 Effect of Graph Laplacian Smoothing

A graph To ensure coherence among deformable splats, Laplacian smoothing was implemented. Results from experiments were compared using $\lambda = 0$ (no smoothing), $\lambda = 0.2$ (mild smoothing), and $\lambda = 0.4$ (strong smoothing). Unrealistic flickering resulted from the surface's noisy and haphazard displacements in the absence of smoothing. Strong smoothing oversmoothed the surface, lowering visible dynamics, whereas mild smoothing produced organic ripple-like motion with local coherence. A value of $\lambda = 0.3$ produced visually convincing wave motion and offered the best balance between expressiveness and stability. The significance of local neighborhood coupling in generating coherent deformable dynamics is emphasized by this analysis.

3.5 Visualization Modes

Three modes of visualization were assessed. Global context was given by the orbiting camera view (Fig. 1), which made it evident that rigid and deformable separation occurred during full rotation. The fixed close-up view made it possible to examine the wave motion at the waterline in great detail, highlighting how the ship's hull stayed stable despite the oscillations of the surrounding water. Lastly, to make subtle motion easier to understand, the heatmap view (Fig. 2) encoded displacement magnitude. With the heatmap allowing for spatial analysis of wave intensity, the fixed close-up stressing motion fidelity, and the semantic view emphasizing material separation, this combination of views offered complementary insights.

3.6 Performance and Feasibility

Runtime for frame generation was tracked in order to gauge performance. On a Colab T4 GPU, processing of about 100k splats, including semantic labeling, deformation, smoothing, and rendering, was finished in a matter of minutes per animation sequence. The pipeline's lightweight design was confirmed by the fact that no external neural models or specialized acceleration were needed. Dynamic extensions to Gaussian Splatting are possible without requiring a lot of infrastructure, as demonstrated by the ability to run the entire dynamic visualization pipeline in a cloud notebook. This bolsters the project's assertion that physics and semantics can be incorporated into 3DGS in a way that is understandable and accessible.

3.7 Figures

Figure 1 and Figure 2 display representative frames from the generated visualizations. With the ship depicted in neutral gray tones and the water's surface highlighted in turquoise, the semantic visualization draws attention to areas that are rigid and deformable. This perspective explicitly distinguishes between rigid and deformable.

A clearer analysis of motion amplitude across the surface is made possible by the heatmap visualization, which encodes the displacement magnitude of each deformable splat. Whereas calmer areas stay in cooler tones, areas with greater displacement intensity show up in warmer hues. By showing how wave strength varies spatially and confirming that areas close to the hull stay stable while open water areas show more motion, this complementary rendering offers interpretability.

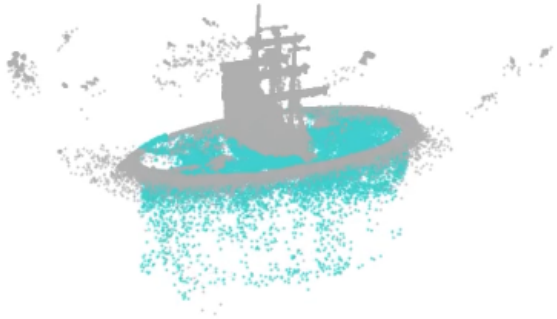

Figure 1: Semantic visualization: rigid ship shown in gray, deformable water shown in turquoise. This view emphasizes material-aware separation.

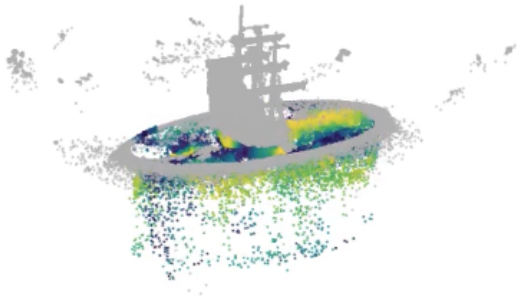

Figure 2: Heatmap visualization: displacement magnitudes encoded with a colormap, showing stronger motion in open water regions and stability near the ship hull.

4 Results and Discussion

The outcomes of Phys-GS Lite show that visually coherent dynamics within Gaussian Splatting can be produced with just physics-inspired deformation and lightweight semantic labeling. The semantic visualization

verified that rigid geometry, like the ship, was unaffected while deformable splats were limited to the water's surface. Fig. 1 illustrates how this separation allowed for distinct material-dependent behavior.

Wave-like displacements were created by applying the sinusoidal wind field, and these displacements spread throughout the deformable surface. In order to prevent isolated noise or unrealistic jittering, graph Laplacian smoothing was added to guarantee that motion remained spatially coherent. The motion could be adjusted using parameters governing amplitude, frequency, and smoothing strength, and it showed periodicity consistent with environmental wind-driven dynamics.

Beyond qualitative examination, the heatmap visualization offered an interpretability layer. Points near the ship hull stayed relatively stable, but areas in open water showed larger displacement magnitudes, as shown in Fig. 2. This demonstrated that the exclusion strategy allowed realistic fluid surface oscillation while preventing undesired deformation of rigid geometry.

All things considered, the project shows that expensive simulations or massive neural networks are not necessary for physically inspired dynamics. Lightweight motion models and straightforward semantic partitioning greatly improve Gaussian Splatting's expressive power, even in a limited environment. The approach shows the potential of incorporating physics and semantics into point-based rendering, but it is not a substitute for complete fluid simulation.

5 Conclusion and Future Work

This project presented **Phys-GS Lite**, a lightweight version of 3D Gaussian Splatting that incorporates dynamics inspired by physics and material awareness. Deformable regions were animated using a sinusoidal wind field with graph Laplacian smoothing, and rigid and deformable splats were distinguished using geometry-based labeling. The resulting visualizations demonstrated that basic heuristics can enhance Gaussian Splatting with material-dependent dynamics by displaying coherent wave motion in water while maintaining the ship's static structure.

The existing approach does not capture full fluid behavior or offer quantitative validation, and it is scene-specific and restricted to sinusoidal wind-driven motion. Future developments could expand the use of physics-aware and semantic Gaussian Splatting by incorporating learning-based semantic segmentation, lightweight fluid solvers, and interactive editing. All things considered, Phys-GS Lite shows that it is possible to integrate physics, geometry, and semantics within 3DGS, even with limited resources, and it provides a promising path for dynamic scene representation.

References

[1] T. Kerbl, G. Riegler, T. Schwarzler, V. Koltun, and T. Müller, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," in *ACM SIGGRAPH*, 2023.

[2] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *ECCV*, 2020.

[3] K. Park, U. Srinivasan, R. Barron, S. Bouaziz, D. Goldman, and S. Seitz, "Nerfies: Deformable Neural Radiance Fields," in *ICCV*, 2021.

[4] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *CVPR*, 2021.

[5] H. Gao, J. Li, C. Zhang, and Z. Shen, "Dynamic View Synthesis via Deformable 3D Gaussians," arXiv preprint arXiv:2212.00262, 2022.

[6] Y. Yao, J. Zhang, Y. Liu, T. Funkhouser, and S. Rusinkiewicz, "Semantic Neural Radiance Fields for Editable Scene Rendering," in *CVPR*, 2023.

[7] R. Bridson, *Fluid Simulation for Computer Graphics*. A. K. Peters, 2008.