

Unleashing the Potential of Multi-modal Foundation Models and Video Diffusion for 4D Dynamic Physical Scene Simulation

Zhuoman Liu¹ Weicai Ye^{2,✉,*} Yan Luximon^{1,✉} Pengfei Wan² Di Zhang²

¹The Hong Kong Polytechnic University ²Kuaishou Technology

maikeyeweicai@gmail.com yan.luximon@polyu.edu.hk zhuo-man.liu@connect.polyu.hk

Abstract

*Realistic simulation of dynamic scenes requires accurately capturing diverse material properties and modeling complex object interactions grounded in physical principles. However, existing methods are constrained to basic material types with limited predictable parameters, making them insufficient to represent the complexity of real-world materials. We introduce **PhysFlow**, a novel approach that leverages multi-modal foundation models and video diffusion to achieve enhanced 4D dynamic scene simulation. Our method utilizes multi-modal models to identify material types and initialize material parameters through image queries, while simultaneously inferring 3D Gaussian splats for detailed scene representation. We further refine these material parameters using video diffusion with a differentiable Material Point Method (MPM) and optical flow guidance rather than render loss or Score Distillation Sampling (SDS) loss. This integrated framework enables accurate prediction and realistic simulation of dynamic interactions in real-world scenarios, advancing both accuracy and flexibility in physics-based simulations. Our code and data are available at <https://zhuomanliu.github.io/PhysFlow>*

1. Introduction

The realistic simulation of dynamic scenes in computer vision and graphics is critical for applications such as robotics manipulation [35, 45] and video generation [3, 8, 46]. However, existing video generation methods often produce visually plausible results that lack physical realism, leading to inaccuracies in object motion and deformation. To enhance simulation realism, it is essential to automatically capture the diverse material properties of objects and model physical interactions under varying forces and conditions.

Among existing methods, PAC-NeRF [22] and GIC [4] aim to learn material parameters by analyzing the deformation of objects in multi-view dynamic videos. PAC-NeRF integrates neural radiance fields (NeRF) [30] with a differentiable Material Point Method (MPM) [17] to facilitate the simulation of various material types and identify ma-

terial properties. However, it is constrained to simpler objects and textures, leading to lower rendering fidelity that limits its effectiveness in high-quality dynamic scene simulations. GIC uses 3D Gaussian Splatting [19] to capture explicit shapes and relies on continuum mechanics to infer implicit shapes, assisting in estimating physical properties. Nonetheless, PAC-NeRF and GIC require multi-view images of deforming objects with known camera poses and prior knowledge of material types, which restricts its applicability in arbitrary dynamic scene videos.

PhysGaussian [44] utilizes 3D Gaussian Splatting with differentiable MPM to enhance rendering fidelity, but requires manual initialization of material properties, limiting its adaptability to complex dynamic scenarios. Recent methods [16, 23, 49] use video diffusion for motion guidance to estimate material properties. However, these methods are restricted to elastic material and minor object movements. Besides, these methods utilize render loss [49] or Score Distillation Sampling (SDS) [34] to estimate material parameters, which presents challenges: render loss is constrained to small motions due to instability and noise in generated videos, while SDS loss demands high computation memory. In contrast, we investigate that optical flow guidance is both memory-efficient and better suited for capturing large and complex motions, enabling more accurate material parameter optimization (see Tab. 4 and visual comparisons in the Appendix).

PhysGen [24] introduces physics reasoning using large pre-trained visual foundation models, eliminating the need for manual parameter initialization. However, it is restricted to single input images and simulates objects at a fixed depth and cannot perform 4D reconstruction or dynamic simulations, limiting its flexibility in simulating complex scenes.

To address these challenges, we propose **PhysFlow**, a novel approach that unleashes the potential of multi-modal foundation models and video diffusion through two key innovations: **1)** physics-conditioned flexible material behavior modeling, and **2)** optical flow-guided iterative refinement for high-fidelity 4D dynamic scene simulation (see Fig. 1). A detailed comparison of the feasibility of our approach versus existing methods across various configurations is provided in Tab. 1.

*: Project Lead, ✉: Corresponding Author

Method	Various Material Types	w/o Manual Init.	Input Type			4D Reconstruction
			Multi-view	Dynamic Video	Single Image	
PAC-NeRF [22]	✓(5 types)			✓		✓
GIC [4]	✓(5 types)			✓		✓
PhysGaussian [44]	✓(6 types)		✓			✓
PhysDreamer [49]			✓			✓
Physics3D [23]			✓			✓
PhysGen [24]	✓	✓			✓	
Ours	✓(7 types)	✓	✓	✓	✓	✓

Table 1. Comparison of methods by material types, manual parameter initialization, input flexibility, and 4D reconstruction capabilities.

Our method leverages large pre-trained visual foundation models (*e.g.*, GPT-4 [1]) to identify material types and initialize material parameters based on visual and text-based queries, ensuring that initial simulations are grounded in accurate material properties. Depending on the input type (*i.e.*, multi-view images, dynamic scene videos, or a single image), we employ different scene reconstruction methods [19, 36, 43] to generate 3D Gaussian splats, providing a differentiable representation for simulation. The material type and initial material properties inferred from the large visual foundation model are then assigned to the 3D Gaussian splats of the simulated object, ensuring that the simulation starts with realistic physical characteristics. To refine these parameters automatically, we use optical flow as guidance within the video diffusion framework, integrated with a differentiable Material Point Method (MPM). Unlike render loss or SDS loss, which are limited to small and stable motions or are computationally intensive, optical flow loss offers a memory-efficient alternative that effectively captures large and complex motions. This enables our method to optimize the material properties to reflect the interactions modeled in the generated video, allowing for simulations that are not only visually plausible but also physically accurate. Our approach supports a diverse range of material behaviors, thus enhancing the flexibility and precision of simulations to represent real-world scenarios comprehensively. In summary, our contributions are as follows:

- We present a novel multi-modal approach to infer 3D Gaussian splats along with the material type and initial material parameters to ensure accurate initial simulation.
- We introduce the guidance of optical flow to optimize material parameters through video diffusion models and differentiable MPM.
- Extensive experiments show that our approach achieves physically realistic 4D dynamic scene simulations on both synthetic and real-world scenes.

2. Related Work

2.1. Dynamic 3D Reconstruction

Dynamic 3D reconstruction has become a fundamental aspect of computer vision, enabling detailed representations of real-world scenes and objects for downstream tasks.

Neural Radiance Fields (NeRF) [30] pioneered high-fidelity 3D reconstruction [15, 25–27, 31, 39, 41, 47] by learning continuous volumetric scene functions from posed images, achieving impressive visual quality. However, NeRF’s high computational demands and reliance on multi-view input limit its effectiveness for dynamic scenes. To enhance real-time reconstruction and efficiency, methods leveraging Gaussian splats [7, 9, 19, 38] for representing surfaces in a differentiable and memory-efficient manner have been developed. Specifically, 3D Gaussian Splatting (3DGS) [19] and PGSR [6] are designed for multi-view images with known poses, enabling high-quality static scene reconstruction. For scenarios involving 4D dynamic scene videos, 4D Gaussian Splatting (4DGS) [5, 43] and Grid4D [18] extend the Gaussian splatting concept to accommodate time-varying data. When only a single image is available, Splatt3R [36] adapts Gaussian splatting to reconstruct scenes without requiring camera parameters or depth information. These Gaussian splatting methods enable continuous representations adaptable to various input types, which is crucial for creating detailed and versatile scene representations, providing the foundation for accurate physics-based simulations in complex environments.

2.2. Physics-Based Motion Generation

Physics-based motion generation encompasses methods that integrate physical principles to simulate realistic object interactions. Differentiable simulation frameworks, such as the differentiable Material Point Method (MPM) [17], play a key role by allowing gradient-based optimization for motion generation tasks. These frameworks enable the fine-tuning of physical parameters and enhance simulations with realistic material behavior.

PAC-NeRF [22] combines NeRF with a differentiable MPM to simulate the physical properties of different materials, providing a degree of flexibility through its particle-based representation. However, its limited rendering fidelity restricts its effectiveness for high-complexity scenes. PhysGaussian [44] improves visual quality by coupling 3D Gaussian splatting with MPM, but it requires manual parameter setting, which limits its adaptability for diverse, real-world applications. Therefore, some existing methods [16, 23, 49] utilize video diffusion priors to estimate

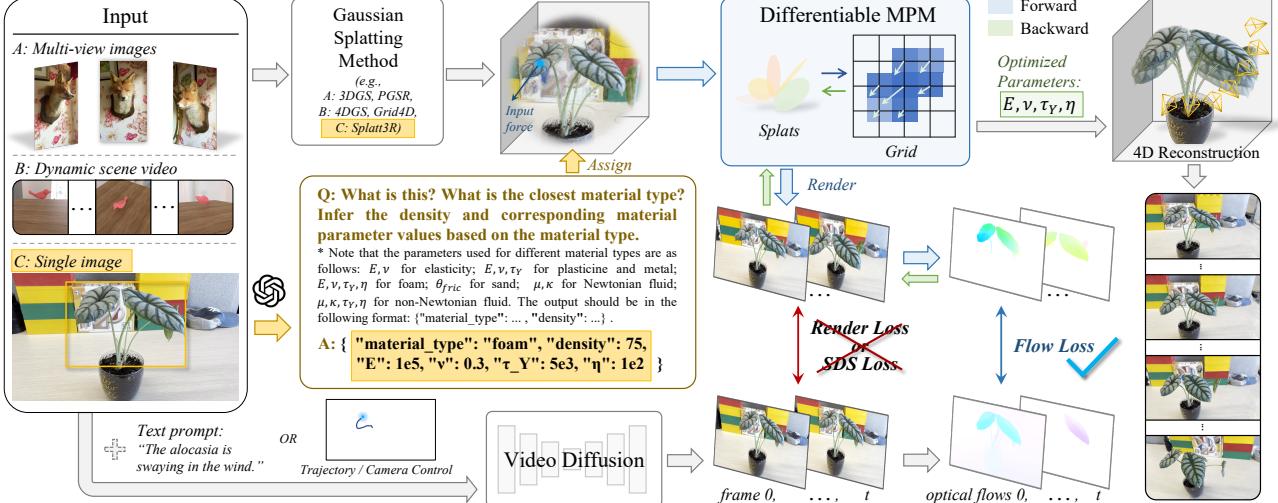


Figure 1. Overview of our proposed pipeline for 4D dynamic physical scene simulation. The process begins with 3D scene reconstruction using Gaussian splatting methods for different input types (multi-view images, dynamic video, and single image). Initial material properties are inferred through multi-modal foundation models and assigned to the reconstructed scene. The material parameters are optimized using optical flow guidance within a video diffusion framework, integrated with a differentiable MPM to ensure physically realistic simulation.

the elasticity parameters of objects. While these approaches incorporate physics to some extent, they are typically constrained to learning only elasticity, limiting their ability to handle more diverse material behaviors like plasticity or viscosity. Furthermore, GIC [4] leverages Gaussian representations for explicit shape modeling and continuum mechanics for implicit shape inference, but it requires multi-view input and known camera poses, making it less suited for real-world dynamic video data where geometry and material types are unknown.

These limitations highlight the need for automatic adjustment of material properties and handling of various material types. Our approach integrates optical flow-guided parameter refinement within a video diffusion framework, enhanced through differentiable MPM, to capture complex object interactions and adapt to diverse material types.

2.3. Video Generation Models

Video generation models have shown significant potential for simulating dynamic scenes due to their ability to create visually compelling sequences based on learned priors. These models [3, 21, 28, 33, 46, 48] take diverse prompts (e.g., text, image, trajectory, camera control, brush, and etc.) and are trained on extensive datasets to implicitly capture relationships between object appearance and motion.

To effectively utilize video generation results as guidance for physical-based simulation, Score Distillation Sampling (SDS) [34] is commonly used to align generated content with target properties by distilling scores from large pre-trained models. However, SDS loss requires substantial computational resources and high memory usage. On the other hand, perceptual loss is used to maintain co-

herence and stability between video frames [49], ensuring that the generated motion remains smooth. This constraint forces the model to limit generated movements to small-scale changes, making it unsuitable for simulations involving large or rapid motions. Our approach addresses these limitations by leveraging video diffusion models to perform optical flow-guided refinement.

3. Method

In this paper, we aim to automatically estimate material properties to enable high-fidelity 4D scene reconstruction from the input image(s) through physical simulation. The overview of our pipeline is illustrated in Fig. 1. Given input image(s) I , which can be multi-view frames, video frames, or a single frame, our pipeline starts with inferring the 3D scene $\{\mathcal{G}_p\}$ using Gaussian splatting models. Meanwhile, we utilize GPT-4 with text prompts to identify the material type M , density ρ , and the initial material parameters M_{p0} for the object from the input image. With the 3D Gaussian splats $\{\mathcal{G}_p\}$ and the material properties, we simulate the object movements under the differentiable MPM with an applied input force. Over t timesteps, we render the scene from a certain viewpoint $\mathbf{v} = [\mathbf{p}, \mathbf{d}]$ to generate several frames $\{\hat{I}_0, \dots, \hat{I}_t\}$. We then employ the video diffusion model, along with the input image and related prompts, to produce a sequence of video frames $\{I_0, \dots, I_t\}$ that capture the object motions. Distinguishing our approach from previous methods, we use the optical flows derived from the video frames, rather than the frames themselves, to guide the optimization of material parameters. We ultimately obtain the optimized material parameters M_p and facilitate 4D reconstruction across various viewpoints.

3.1. Preliminaries

3D Scene Reconstruction: Following PhysGaussian [44], our method reparameterizes the 3D scene using unstructured 3D Gaussian kernels, denoted by $\mathcal{G}_p = \{(\mathbf{x}_p, \sigma_p, \mathbf{A}_p, \mathbf{C}_p)\}_{p \in \mathcal{P}}$, where \mathbf{x}_p represents the centers of the Gaussians, σ_p the opacities, \mathbf{A}_p the covariance matrices, and \mathbf{C}_p the spherical harmonic coefficients. This explicit representation provides a differentiable and memory-efficient structure, enabling the modeling of complex scenes in various input settings.

To generate 3D Gaussian splats, we employ different methods based on the input type. For multi-view images with known camera poses, we utilize PGSR [6] to construct a static 3D representation. For dynamic scene video $\{I_0, \dots, I_t\}$, Grid4D [18] is employed to produce a time-dependent 3D representation, capturing the temporal evolution of the scene. When dealing with a single image I , Splatt3R [36] infers the scene structure using learned priors to create 3D splats.

Continuum Mechanics: Continuum mechanics describes the motion and deformation of materials using a time-dependent continuous deformation map: $\mathbf{x} = \phi(\mathbf{X}, t)$, where \mathbf{X} represents the undeformed material space and \mathbf{x} denotes the deformed space at time t . The deformation gradient $\mathbf{F}(\mathbf{X}, t) = \nabla_{\mathbf{X}}\phi(\mathbf{X}, t)$ encodes the local transformations in material, including stretching, rotation, and shear.

Our work simulates seven types of materials: *elastic, plasticine, metal, foam, sand, Newtonian fluids, and non-Newtonian fluids*. Each material type exhibits unique behaviors, necessitating different physical properties for accurate modeling. To ensure these properties are represented accurately, the evolution of ϕ must adhere to fundamental physical laws, including the conservation of mass and momentum. Conservation of mass ensures consistent density within material regions over time, while conservation of momentum is expressed through:

$$\rho(\mathbf{x}, t)\dot{\mathbf{v}}(\mathbf{x}, t) = \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \mathbf{f}^{\text{ext}}, \quad (1)$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and \mathbf{f}^{ext} denotes external forces. The deformation gradient \mathbf{F} is decomposed into elastic \mathbf{F}^E and plastic \mathbf{F}^P components to account for permanent deformations. This decomposition is essential for simulating complex material behaviors, from the elasticity of foam to the plastic flow of metals, and enables our framework to model diverse real-world materials effectively.

Material Point Method: Material Point Method (MPM) bridges the strengths of Lagrangian and Eulerian approaches by discretizing the material into particles, $\mathbf{P} = \{(\mathbf{x}_p, \mathbf{v}_p, \mathbf{F}_p)\}$, each representing a small region of the material. These particles track properties such as position \mathbf{x}_p ,

velocity \mathbf{v}_p , and deformation gradient \mathbf{F}_p . Lagrangian particles ensure mass conservation, while the Eulerian grid representation facilitates momentum conservation. The interaction between particles and the grid is handled using B-spline kernel functions. Momentum conservation is discretized over time steps, updating grid velocities and transferring them back to particles to adjust their positions:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}. \quad (2)$$

The deformation gradient \mathbf{F}_p^E is updated to track changes, with adjustments made for plasticity as needed. This combination enables the simulation of complex deformations and interactions, essential for realistic material behavior in dynamic scenes.

3.2. Material Property Initialization

To simulate an object accurately, knowing the material type is essential for applying deformations that adhere to physical laws. Additionally, the initial values of material parameters play a significant role in the subsequent optimization process. Therefore, utilizing foundation models to estimate initial material properties is critical for achieving realistic simulations. In our work, we infer initial material properties by querying GPT-4 [1] with the input image and questions. As shown in Fig. 1, the query involves identifying the closest material type M and inferring its density ρ along with the corresponding initial material parameters M_{p_0} . More details of the material parameters for seven different material types can be found in Appendix A.

3.3. Material Parameter Optimization with Optical Flow Guidance

After the initial material properties are assigned (e.g., $M_{p_0} = \{E_0, \nu_0, \tau_{Y_0}, \eta_0\}$ in Fig. 1), optimizing these parameters during the simulation is crucial for capturing realistic object interactions and ensuring adherence to physical laws. Our approach leverages optical flow to guide the optimization process dynamically, enhancing the material parameters as the simulation evolves.

Given an input image I and a set of prompts \mathcal{Q} , our method generates a video input $V = \{I_0, \dots, I_t\}$ using different video diffusion models. This flexibility allows the selection of prompts and models best suited to the simulation's needs. For instance, when only an image is provided, the Stable Video Diffusion (SVD) model [3] is used. When both image and text prompts are available, CogVideoX [46] generates the video. For scenarios involving trajectory or camera motion, MotionCtrl [42] is employed, while for more complex motions involving brush strokes, MOFA-Video [33] or the Kling 1.5 Model [20] are utilized.

Given the video input V , the optical flow $\mathbf{U}(I_t, I_{t+1})$ between consecutive frames is computed to track motion and detect discrepancies between the simulated and observed

movements. This information is used to adjust the material parameters $M_p = \{E, \nu, \tau_Y, \eta\}$, ensuring that the simulation reflects realistic behavior over time.

The optimization process begins by analyzing the predicted motion from the simulation and comparing it to the optical flow-derived motion in the input frames. A flow-based loss \mathcal{L}_{flow} is minimized to align the simulation output \hat{V} with the actual video V :

$$\mathcal{L}_{flow} = \sum_t \|\mathbf{U}(I_t, I_{t+1}) - \hat{\mathbf{U}}(\hat{I}_t, \hat{I}_{t+1})\|^2, \quad (3)$$

where $\hat{\mathbf{U}}$ denotes the flow derived from simulated frames.

This process iteratively optimizes M_p with the Moving Least Squares Material Point Method (MLS-MPM) [14]. MLS-MPM extends traditional MPM by incorporating a smoother interpolation scheme that ensures better accuracy and stability, particularly during large deformations and complex material behaviors.

Besides, unlike image-level render loss [49], which is restricted to subtle motions due to the instability and high noise in video generation outputs, or Score Distillation Sampling (SDS) loss [34], which inherently requires significant memory and becomes even more demanding when applied to video diffusion models with complex prompts, using optical flow loss as guidance presents a more efficient solution. It effectively captures motion while optimizing material parameters and conserves memory, making it a practical approach for high-fidelity simulations.

After optimizing the material parameters M_p , the updated properties are integrated into the simulation to control the motion and deformation of the 3D Gaussian splats $\{\mathcal{G}_p\}$ with the differentiable MPM framework. The final step involves rendering the simulation from new viewpoints $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$ for 4D reconstruction.

4. Experiments

In this section, we showcase the ability of our method to automatically optimize material parameters across a diverse set of material types, and evaluate its effectiveness on both synthetic and real-world datasets.

4.1. Evaluation on Synthetic Dataset

Dataset: We use the synthetic dataset introduced by PAC-NeRF [22], which features 9 objects made of different materials, including elastic (*Torus, Bird*), plasticine (*Playdoh, Cat*), sand (*Trophy*), and both Newtonian (*Droplet, Letter*) and non-Newtonian fluids (*Cream, Toothpaste*). Each object includes RGB images from 11 distinct viewpoints, with 10 to 16 frames per viewpoint. Since our work focuses on system identification and validation of our effectiveness in optimizing material parameters, we adopt the 3D Gaussian splats generated from the first frame of dynamic reconstruction by GIC [4] as our simulation objects. Besides,

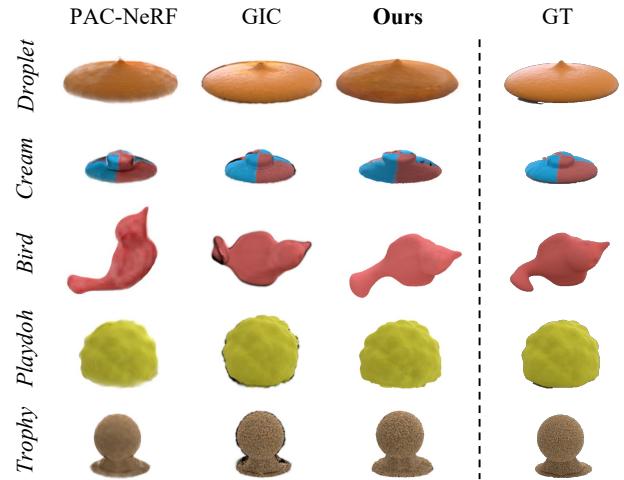


Figure 2. Qualitative results of all methods on synthetic dataset.

we choose the RGB images of a certain viewpoint from the dataset for each object instead of the video diffusion output in our pipeline for a fair comparison.

Baselines: For comparing system identification performance, we evaluate our method employing PAC-NeRF [22] and GIC [4] as baselines. PAC-NeRF is designed for system identification tasks and combines neural representations with differentiable physics for material parameter estimation. GIC focuses on dynamic 3D reconstructions using Gaussian splatting to capture shape and motion. Both baselines represent state-of-the-art approaches in this domain, offering a robust comparison for assessing our method’s effectiveness. For a fair comparison, all baselines and our method optimize parameters starting from the same initial settings introduced in PAC-NeRF.

Metric: We evaluate 1) Relative Error ($RE = |M_p - \hat{M}_p|/M_p$) comparing predicted versus ground-truth material parameters and 2) End-Point Error (EPE) [10] quantifying optical flow deviations to measure the system identification performance across various material types.

Analysis: We report the RE and EPE for system identification performance across various material parameters on the synthetic dataset (Tab. 4) for PAC-NeRF, GIC, and our PhysFlow. Our method consistently achieves lower RE and EPE than PAC-NeRF in most cases and remains competitive with GIC, demonstrating its effectiveness in predicting material properties. Specifically, for *Droplet* and *Letter*, our approach yields the lowest RE and EPE, indicating strong performance in *Newtonian fluids*. For *non-Newtonian fluids* like *Toothpaste*, it also delivers competitive results, maintaining low shear and bulk modulus errors. For *Trophy*, which represents *sand* (i.e., granular material), our method achieves the lowest error in friction angle estimation, highlighting its precision in granular material properties. While system identification remains challenging for complex materials, our approach performs comparably to

Method	Physical-realism \uparrow	Photo-realism \uparrow
PhysGaussian [44]	2.67	2.91
PhysDreamer [49]	2.58	2.91
Physics3D [23]	2.64	2.98
Ours	3.44	3.53

Table 2. Human evaluation on real-world dataset.

PhysGaussian	PhysDreamer	Physics3D	Ours
3.70	3.87	4.41	3.08

Table 3. Evaluation metric (ECMS \downarrow) on real-world dataset.

baseline methods. As shown in Fig. 2, our method also produces deformations that closely match the ground truth and preserves sharper textures across different material types.

4.2. Evaluation on Real-world Dataset

Dataset: Our real-world evaluation includes a comprehensive set of scenes to cover diverse material types. We use four publicly available scenes from PhysDreamer [49] (*Alocasia*, *Carnation*, *Hat*, and *Telephone*), along with additional scenes from other sources: *Fox* from Instant-NGP [32], *Plane* from NeRFStudio [37], and *Kitchen* from Mip-NeRF 360 [2]. To ensure coverage of all seven material types, we introduce two additional self-collected scenes, *Jam* and *Sandcastle*.

Baselines: For real-world simulation, we compare our method with the following baselines: PhysGaussian [44], PhysDreamer [49], and Physics3D [23]. PhysGaussian combines 3D Gaussian splatting with differentiable MPM for high-fidelity rendering and accurate physical simulations but requires manual parameter settings. PhysDreamer and Physics3D leverage priors from video diffusion models to estimate material properties, but are limited to elastic materials. To evaluate our method, we apply larger forces than previous baselines. For material settings, we use GPT-queried materials for PhysGaussian, maintain elasticity for PhysDreamer, and incorporate both elasticity and viscoelasticity for Physics3D. PhysGen [24] is not included in this comparison due to its reliance on perceptual information (e.g., albedo), which is unavailable in real-world scenarios. Instead, we provide a separate evaluation on its synthetic dataset, with additional implementation details in Appendix C.3.

Metric: To evaluate performance, we conduct a human evaluation where 32 participants rate 36 videos generated by different methods, presented in random order. Each video is rated on a 5-point Likert scale, from 1 (strongly disagree) to 5 (strongly agree), based on two criteria: **physical-realism** and **photo-realism**. Participants independently assess each video along these two dimensions to provide a comprehensive evaluation of simulation realism. Furthermore, to assess motion quality, we consider the *Energy Minimization Principle* for physically plausible motion and

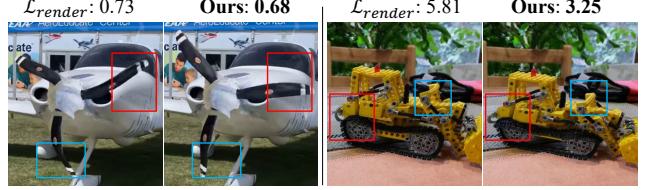


Figure 3. Comparisons of $\mathcal{L}_{\text{render}}$ and ours ($\mathcal{L}_{\text{flow}}$) with ECMS \downarrow . compute the Energy-Constrained Motion Score (ECMS):

$$E = \sum_t \|\mathbf{v}_{t+1} - \mathbf{v}_t\|^2 + \|\nabla^2 \mathbf{v}_t\|^2 + \alpha \frac{1}{\sum_t \|\mathbf{v}_t\|} \quad (4)$$

where \mathbf{v}_t denotes velocity and $\alpha = 0.1$.

Analysis: To thoroughly evaluate robustness, we applied large input forces in several scenes to induce substantial object deformations and motions, challenging the capacity to accurately simulate intense physical dynamics. As shown in Tab. 2, our method achieves the highest average scores for both physical-realism and photo-realism, significantly outperforming the baseline methods on the real-world dataset. Specifically, our method is the only method achieving scores above 3, while all baseline methods fall below this threshold. And our method achieves the lowest ECMS as shown in Tab. 3, indicating more plausible motion. The results underscore the limitations of existing methods in capturing complex physical interactions under challenging scenarios.

The qualitative results in Fig. 5 further illustrate the advantages of our method. Even under high-force conditions, our approach produces realistic and stable simulations, capturing both material deformation and dynamic interactions with greater fidelity compared to the baselines. These results highlight the robustness and adaptability of our method for scenes involving diverse materials and large motions, demonstrating the effectiveness of integrating multi-modal foundation models with optical flow-guided optimization.

Additional implementation details are provided in Appendix B, and more visual comparisons are available in Appendix C.2.

4.3. Ablation Study

We conduct a comprehensive ablation study to evaluate the contribution of key components in our method, using both synthetic and real-world datasets to provide a thorough understanding of their impacts on performance.

Effectiveness of Optical Flow Guidance: We evaluate the impact of different loss functions on the synthetic dataset to highlight the significance of our optical flow guidance. Specifically, we compare the render loss $\mathcal{L}_{\text{render}} = \sum_t \lambda L_1(I_t, \hat{I}_t) + (1 - \lambda) L_{D-\text{SSIM}}(I_t, \hat{I}_t)$ [49], the SDS loss (with text prompt) \mathcal{L}_{sds} [34], and our proposed flow loss $\mathcal{L}_{\text{flow}}$. As shown in Tab. 4, optical flow guidance

Object	PAC-NeRF [22]	GIC [4]	Ours (\mathcal{L}_{flow})	\mathcal{L}_{render}	\mathcal{L}_{sds}
Droplet	0.06 / 0.28	0.41 / 0.42	0.02 / 0.08	0.48 / 0.63	0.98 / 1.01
Letter	<u>0.26 / 0.11</u>	0.02 / 0.07	0.46 / 0.13	0.58 / 0.20	0.70 / 0.18
Cream	3.04 / 0.61	0.21 / 0.20	0.54 / 0.64	<u>0.45 / 0.30</u>	1.27 / 2.76
Toothpaste	0.42 / 0.15	0.11 / 0.07	<u>0.24 / 0.16</u>	0.30 / 0.20	<u>0.24 / 0.19</u>
Torus	<u>0.06 / 0.52</u>	0.01 / 0.24	0.51 / 1.04	0.41 / 1.64	0.57 / 1.58
Bird	<u>0.08 / 0.62</u>	0.04 / 0.36	0.31 / 0.69	0.37 / 0.88	0.76 / 3.07
Playdoh	0.37 / 0.55	0.10 / 0.16	<u>0.15 / 0.29</u>	0.54 / 4.02	0.65 / 4.05
Cat	<u>0.31 / 0.34</u>	0.02 / 0.08	0.34 / 0.24	0.37 / 0.94	0.85 / 0.70
Trophy	0.10 / 3.30	<u>0.05 / 1.85</u>	0.01 / 1.33	0.17 / 3.15	0.30 / 3.58
Avg.	0.52 / 0.72	0.11 / 0.38	0.29 / 0.51	0.41 / 1.33	0.70 / 1.90

Table 4. System identification performance (RE↓ / EPE↓) on the synthetic dataset.

w/o GPT Init.	GPT Init. w/o Optim.	\mathcal{L}_{render}	Ours
3.91	3.23	3.21	3.08

Table 5. Ablation results (ECMS↓) on real-world dataset.

achieves lower RE and EPE across most material parameters, particularly viscosity η , bulk modulus κ , and yield stress τ_Y (detailed comparisons for each parameter are provided in Appendix C.4), leading to more accurate material parameter optimization. These results demonstrate that optical flow guidance not only improves parameter estimation but also ensures more realistic motion in system identification compared to other guidance losses. Besides, Fig. 3 and Tab. 5 show that \mathcal{L}_{flow} performs better under large force applications and motions on real-world scenes.

Effect of Physics Reasoning with Foundation Model:

To evaluate the impact of using a foundation model (e.g., GPT-4) for physics reasoning, we conduct simulations on the real-world dataset with four configurations: (1) manually defined initial material properties without optimization, (2) use manually defined material properties as initialization followed by optimization, (3) use foundation model-inferred properties without optimization, (4) initialize material properties with foundation model-inferred values followed by optimization. As shown in Fig. 4 and Tab. 5, simulations initialized with manually defined values produce motions that partially reflect the effects of the input force but lack sufficient fidelity. Further optimization of these manually set values results in overly rigid or tense parameters, leading to less realistic outcomes. In contrast, simulations using foundation model-inferred values without additional optimization provide a more balanced starting point. By initializing with foundation model predictions and optimizing, our method achieves significantly more accurate material dynamics and visually convincing simulations.

In addition to parameter initialization, using a foundation model to identify material type and density ensures more realistic simulations. Accurate material classification directly affects an object’s physical response, as shown in Fig. 4 (a) and (b), where incorrect material properties lead to unrealistic outcomes. Leveraging the foundation model predictions enhances simulation fidelity, allowing for more accurate and consistent object behavior in complex scenes.

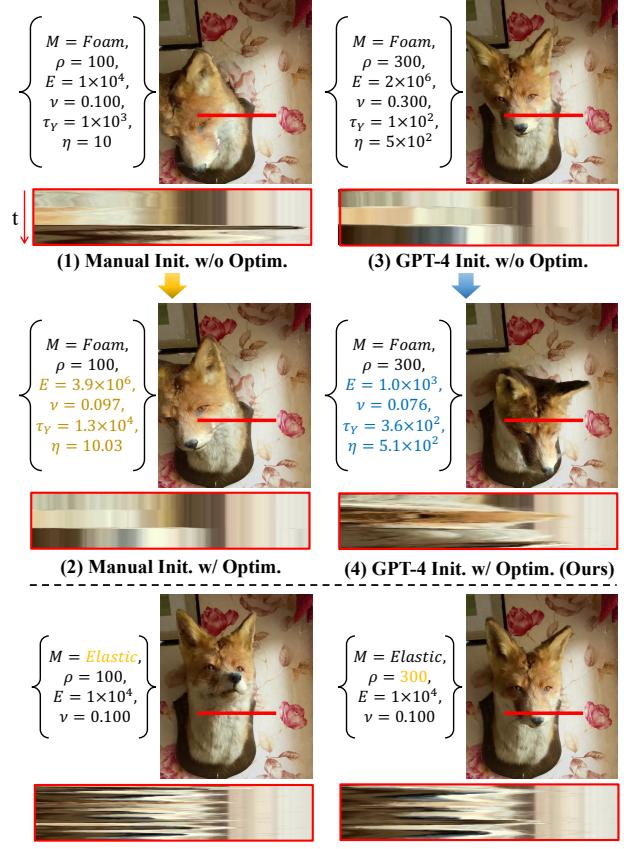


Figure 4. Ablations on physics reasoning, showing material values, timestep 30 frame, and deformation frequency.

5. Conclusion

In this paper, we present a novel approach leveraging multi-modal foundation models and video diffusion with optical flow guidance for 4D dynamic scene simulation. Our method optimizes material parameters using optical flow guidance and integrates inferred properties for realistic simulations across diverse material types. Through comprehensive experiments on synthetic and real-world datasets, we demonstrate that our method outperforms existing baselines in accuracy and adaptability, showing its potential for enhancing physical realism in dynamic simulations.

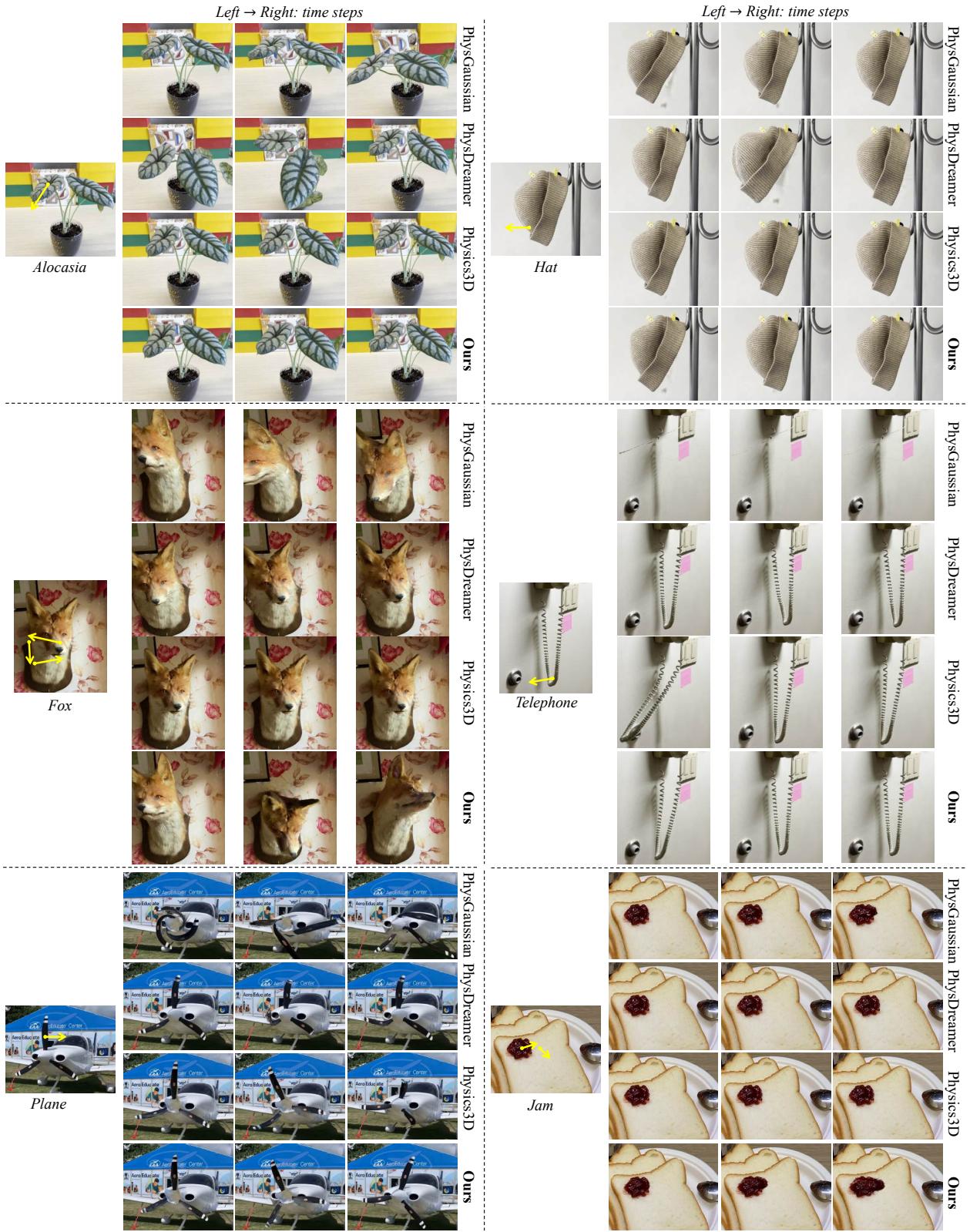


Figure 5. Qualitative results of all methods on real-world dataset. The yellow arrows show the input force for the simulated objects.

Acknowledgement: The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. GRF/PolyU 15606321 and Project No. GRF/PolyU 15607922).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. [2](#), [4](#)
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. [6](#)
- [3] 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. [1](#), [3](#), [4](#)
- [4] Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gaussian-informed continuum for physical property identification and simulation. *NeurIPS*, 2024. [1](#), [2](#), [3](#), [5](#), [7](#)
- [5] 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](#)
- [6] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arxiv preprint arxiv:2406.06521*, 2024. [2](#), [4](#)
- [7] Junyi Chen, Weicai Ye, Yifan Wang, Danpeng Chen, Di Huang, Wanli Ouyang, Guofeng Zhang, Yu Qiao, and Tong He. Gigags: Scaling up planar-based 3d gaussians for large scene surface reconstruction. *arXiv preprint arXiv:2409.06685*, 2024. [2](#)
- [8] Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dream-scene4d: Dynamic multi-object scene generation from monocular videos. *NeurIPS*, 2024. [1](#)
- [9] Xiao Cui, Weicai Ye, Yifan Wang, Guofeng Zhang, Wengang Zhou, Tong He, and Houqiang Li. Streetsurfgs: Scalable urban street surface reconstruction with planar-based gaussian splatting. *arXiv preprint arXiv:2410.04354*, 2024. [2](#)
- [10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *ICCV*, pages 2758–2766, 2015. [5](#)
- [11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. [1](#)
- [12] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Chengzhuo Tong, Peng Gao, Chunyuan Li, and Pheng-Ann Heng. Sam2point: Segment any 3d as videos in zero-shot and promptable manners. *arXiv preprint arXiv:2408.16768*, 2024. [1](#)
- [13] Warp: A high-performance python framework for gpu simulation and graphics. <https://github.com/nvidia/warp>. *NVIDIA GPU Technology Conference (GTC)*, 2022. [1](#)
- [14] 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 TOG*, 2018. [5](#)
- [15] Chenxi Huang, Yuenan Hou, Weicai Ye, Di Huang, Xiaoshui Huang, Binbin Lin, Deng Cai, and Wanli Ouyang. Nerf-det++: Incorporating semantic cues and perspective-aware depth supervision for indoor multi-view 3d detection. *arXiv preprint arXiv:2402.14464*, 2024. [2](#)
- [16] Tianyu Huang, Yihan Zeng, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv preprint arXiv:2406.01476*, 2024. [1](#), [2](#)
- [17] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM TOG*, 2015. [1](#), [2](#)
- [18] Xu Jiawei, Fan Zexin, Yang Jian, and Xie Jin. Grid4D: 4D decomposed hash encoding for high-fidelity dynamic scene rendering. *NeurIPS*, 2024. [2](#), [4](#)
- [19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. [1](#), [2](#)
- [20] KlingAI. Kling ai. <https://klingai.com/>, 2024. [4](#), [2](#)
- [21] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. *arXiv preprint arXiv:2412.04383*, 2024. [3](#)
- [22] 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. *ICLR*, 2023. [1](#), [2](#), [5](#), [7](#)
- [23] Fangfu Liu, Hanyang Wang, Shunyu Yao, Shengjun Zhang, Jie Zhou, and Yueqi Duan. Physics3d: Learning physical properties of 3d gaussians via video diffusion. *arXiv preprint arXiv:2406.04338*, 2024. [1](#), [2](#), [6](#)
- [24] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. *ECCV*, 2025. [1](#), [2](#), [6](#)
- [25] Xiangyu Liu, Weicai Ye, Chaoran Tian, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Coxgraph: multi-robot collaborative, globally consistent, online dense reconstruction system. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. [2](#)
- [26] Zhuoman Liu, Bo Yang, Yan Luximon, Ajay Kumar, and Jinxi Li. Raydf: Neural ray-surface distance fields with multi-view consistency. *NeurIPS*, 36:25238–25267, 2023.
- [27] Zhuoman Liu, Yan Luximon, Wei Lin Ng, and Eric Chung. Ddf-isn: Internal structure modeling of human head using probabilistic directed distance field. *IEEE TVCG*, 2025. [2](#)
- [28] Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical

- commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024. 3
- [29] Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quan-feng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 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. *ECCV*, 2020. 1, 2
- [31] Yuhang Ming, Weicai Ye, and Andrew Calway. idf-slam: End-to-end rgb-d slam with neural implicit mapping and deep feature tracking. *arXiv preprint arXiv:2209.07919*, 2022. 2
- [32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 6
- [33] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yingqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. *ECCV*, 2024. 3, 4
- [34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ICLR*, 2022. 1, 3, 5, 6
- [35] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *Conference on Robot Learning (CoRL)*, 2023. 1
- [36] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 2, 4
- [37] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *SIGGRAPH*, 2023. 6
- [38] Shengji Tang, Weicai Ye, Peng Ye, Weihao Lin, Yang Zhou, Tao Chen, and Wanli Ouyang. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. *arXiv preprint arXiv:2410.06245*, 2024. 2
- [39] Ziyu Tang, Weicai Ye, Yifan Wang, Di Huang, Hujun Bao, Tong He, and Guofeng Zhang. Nd-sdf: Learning normal deflection fields for high-fidelity indoor reconstruction. *arXiv preprint arXiv:2408.12598*, 2024. 2
- [40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1
- [41] Yifan Wang, Di Huang, Weicai Ye, Guofeng Zhang, Wanli Ouyang, and Tong He. Neurodin: A two-stage framework for high-fidelity neural surface reconstruction. *arXiv preprint arXiv:2408.10178*, 2024. 2
- [42] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *SIGGRAPH*, 2024. 4
- [43] 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. *CVPR*, 2024. 2
- [44] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *CVPR*, 2024. 1, 2, 4, 6
- [45] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3d dynamic scene representations for robot manipulation. *Conference on Robot Learning (CoRL)*, 2020. 1
- [46] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiaheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 1, 3, 4
- [47] Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis. In *ICCV*, 2023. 2
- [48] Weicai Ye, Chenhao Ji, Zheng Chen, Junyao Gao, Xiaoshui Huang, Song-Hai Zhang, Wanli Ouyang, Tong He, Cairong Zhao, and Guofeng Zhang. Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. *NeurIPS*, 2024. 3
- [49] 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. *ECCV*, 2024. 1, 2, 3, 5, 6
- [50] Zeshun Zong, Xuan Li, Minchen Li, Maurizio M Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. In *SIGGRAPH Asia*, 2023. 1

Unleashing the Potential of Multi-modal Foundation Models and Video Diffusion for 4D Dynamic Physical Scene Simulation

Supplementary Material

A. Material Properties

In our work, we apply the Material Point Method (MPM) to simulate seven distinct material types: *elastic*, *plasticine*, *metal*, *foam*, *sand*, *Newtonian fluid*, and *non-Newtonian fluid*. The definitions for the first five types are derived from PhysGaussian, while the latter two types follow the specifications outlined in PAC-NeRF. The material parameters for each type are detailed as follows:

- *Elastic*: Young’s modulus E and Poisson’s ratio ν .
- *Plasticine*: Young’s modulus E , Poisson’s ratio ν , and yield stress τ_Y .
- *Metal*: Young’s modulus E , Poisson’s ratio ν , and yield stress τ_Y .
- *Foam*: Young’s modulus E , Poisson’s ratio ν , and plastic viscosity η .
- *Sand*: friction angle θ_{fric} .
- *Newtonian fluid*: fluid viscosity μ and bulk modulus κ .
- *Non-Newtonian fluid*: shear modulus μ , bulk modulus κ , yield stress τ_Y , and plastic viscosity η .

For the synthetic dataset, each object’s material type is predefined, while the real-world dataset lacks this specification. To infer the material types, we leverage a large pre-trained visual foundation model (*e.g.*, GPT-4) for classification. As shown in Tab. 6, some inferred material types differ from those specified in the original dataset (*i.e.*, *Alocasia* and *Carnation* are assigned as *elastic* in PhysDreamer). This underscores the importance of leveraging a robust visual foundation model to accurately determine material types, ensuring that simulations are based on reliable initial properties. Besides, we adopt the constitutive models for different material types as detailed in [22, 44, 50], which provide a strong framework for simulating a variety of materials with distinct physical behaviors.

B. Implementation Details

B.1. Optimization Details

In our experiments, we implement the differentiable MPM using Warp [13]. Besides, we use RAFT [40] to compute optical flow, which serves as a critical component for guiding the optimization process. For the real-world dataset, we segment the foreground points where input forces are applied using SAM2Point [12], ensuring precise identification of the regions impacted by force. Specific forces and durations are detailed in Tab. 7. Video generation is facilitated by CogVideoX [46], which leverages a text prompt (refer to Tab. 6) combined with a selected frame from the dataset

to guide the synthesis process. All optimization tasks are performed on a single NVIDIA A800 GPU.

B.2. Baselines

PAC-NeRF/GIC: PAC-NeRF and GIC estimate material parameters from multi-view images with deforming objects, while our approach relies solely on the 3D Gaussian splats captured at the initial frame. To ensure fair comparisons, we align the simulation timestep with those methods and use the deformed Gaussian splats at each timestep as additional supervision, employing Chamfer distance [11] as the geometry loss.

PhysGaussian/PhysDreamer/Physics3D: PhysGaussian utilizes a fixed set of manually defined material properties, including material type, density, and associated parameters, without further optimization. This setup is comparable to the conditions outlined in Sec. 4.3 (1). Changes in material type, density, or further parameter optimization can lead to different simulation outcomes. To highlight the benefits of incorporating a large pre-trained visual foundation model for physics reasoning and parameter refinement, we apply the same input force to each scene and compare our results with PhysGaussian. While PhysDreamer and Physics3D include methods for parameter optimization, they are limited to the *elastic* material type. For consistency and fair comparison, we assign *elastic* as the material type for PhysDreamer and *elastic with viscoelasticity* for Physics3D. Besides, in our experiments, we apply larger forces (refer to Tab. 7) than previous baselines on the real-world dataset. The weaker performance of baselines under these conditions highlights our approach’s advantage in leveraging GPT initialization and optical flow guidance.

PhysGen [24]: Compared to other baselines, PhysGen utilizes foundation model-based physics reasoning, which removes the need for manual parameter initialization. However, its dependence on single-image input and lack of 4D reconstruction capability limit its applicability for comprehensive simulations. Additionally, PhysGen simulates objects at a fixed depth, which restricts its ability to handle complex scenarios. These limitations prevent direct comparisons with other baselines on both the synthetic and real-world datasets. Therefore, we conduct a separate evaluation of our method on the dataset proposed by PhysGen to assess our simulation performance in Sec. C.3.

Scene	Material Type	Text Prompt
Alocasia	Foam	<i>The alocasia is swaying in the wind.</i>
Carnation	Foam	<i>The carnation is swaying in the wind.</i>
Hat	Elastic	<i>The hat is given a tug.</i>
Telephone	Elastic	<i>The telephone coil is given a tug.</i>
Fox	Foam	<i>The fox is shaking its head.</i>
Plane	Metal	<i>The propeller is spinning.</i>
Kitchen	Plasticine	<i>The Lego on the table is being squeezed by a downward force.</i>
Jam	Non-Newtonian fluid	<i>The jam on the toast is being spread.</i>
Sandcastle	Sand	<i>The sandcastle on the beach is collapsing.</i>

Table 6. Material types inferred by GPT-4 and text prompts to generate guidance videos for scenes in real-world dataset.

Scene	Force (s) ((x, y, z) for most cases)	Duration (s)
Alocasia	(0.25, 0, 0)	1
Carnation	(-0.1, 0, 0)	1
Hat	(1, -2, 1)	1
Telephone	(-1, 0, 0)	2
Fox	$(0, -0.5, 0.25) \rightarrow (0, 0, -0.5) \rightarrow (0, 0.5, 0.25)$	1→1→1
Plane	<i>rotation scale = -10 → -5</i>	0.8→1
Kitchen	(0, 0, 0.1)	1
Jam	$(0.2, 0, 0) \rightarrow (0.1, 0.2, 0)$	2→1
Sandcastle	<i>release nlayer = 50</i>	-

Table 7. Simulation forces and durations on real-world dataset.

C. Additional Experimental Results

C.1. Human Evaluation

We conducted a human evaluation to assess the physical-realism and photo-realism of videos generated by our method compared to the baseline methods: PhysGaussian, PhysDreamer, and Physics3D. As illustrated in Fig. 6, our method achieves significantly higher scores in the "Agree" and "Strongly Agree" categories across both evaluation criteria. In contrast, baseline methods, such as PhysGaussian and PhysDreamer, received lower ratings, particularly in the "Neutral" and "Disagree" categories. The average score for our method surpasses 3.0, consistently approaching "Agree," reflecting its ability to simulate visually and physically plausible scenarios. This result underscores the robustness and effectiveness of our approach in addressing the challenges of material parameter optimization and dynamic scene simulation.

C.2. Additional Experimental Results of Synthetic and Real-world Datasets

We provide additional comparisons between our method and baselines on both synthetic (see Tab. 9 and Fig. 9) and real-world (see Fig. 11) datasets. These results emphasize the effectiveness of our method in handling diverse material behaviors and complex scenarios. To further enhance understanding and visualization, we offer more results in videos accessible through our project page: <https://zhuomanliu.github.io/PhysFlow>

PhysGen	PhysGaussian	Ours
0.54	2.95	0.85

Table 8. Evaluation metric (ECMS↓) on PhysGen scenes.

C.3. Additional Evaluation Using Single-view Input

In addition to experiments on synthetic and real-world datasets, we further validate our method on scenarios with single-view input. Specifically, we employ Splatt3R to reconstruct 3D Gaussian splats from a single image and utilize an inpainting technique to recover the background geometry. For simulation, we focus exclusively on the segmented foreground points, ensuring the deformation and dynamics are isolated to the target object.

To evaluate our approach, we perform experiments using videos generated by PhysGen and Kling [20], with input images sourced from the PhyGenBench [29] dataset. As shown in Fig. 10, the visual results highlight the capability of our method to capture realistic deformations and motion trajectories guided by videos from both PhysGen and Kling. We also report the quantitative results (ECMS) on PhysGen scenes in Tab. 8. These results demonstrate the adaptability and robustness of our method, even when limited to single-view inputs, effectively simulating complex interactions and maintaining high fidelity across varying scenarios.

C.4. Ablation Study

Effectiveness of Optical Flow Guidance: We performed ablations of \mathcal{L}_{flow} on the entire synthetic dataset, which includes 9 cases across various material types. We provide detailed ablation results of each object in Tab. 10 and

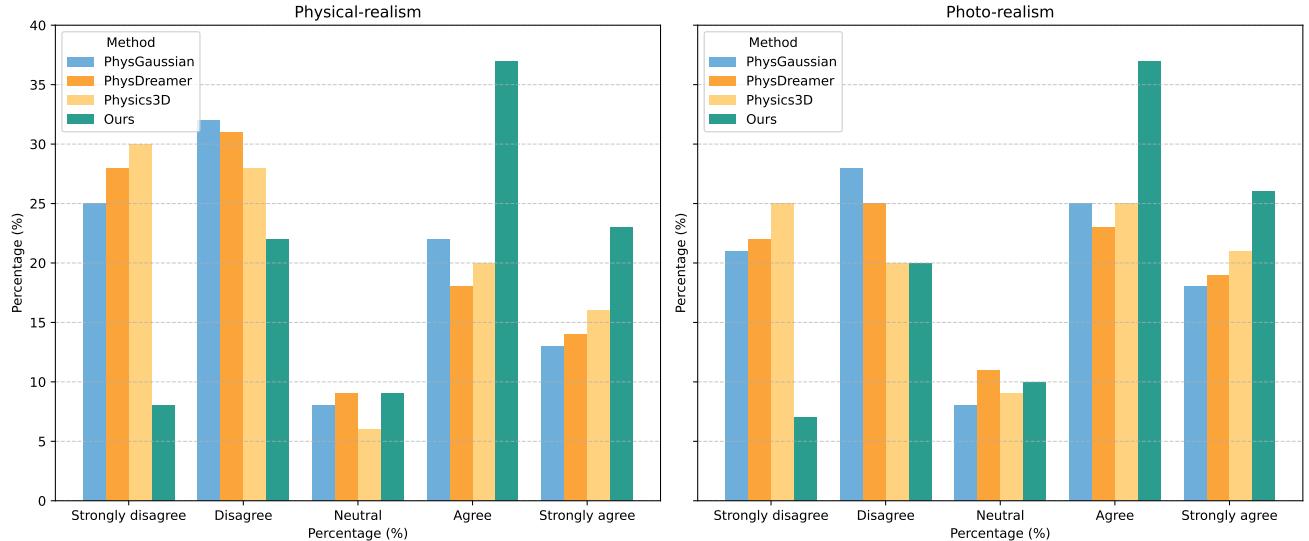


Figure 6. Human evaluation score distribution. The chart shows the score distribution for physical-realism and photo-realism based on human evaluations. Our method significantly outperforms the baseline methods across both metrics. The average score for our method approaches "Agree" for both criteria, indicating superior performance in producing realistic simulations.



Figure 7. Ablations of GPT initialization with $\text{ECMS} \downarrow$.

qualitative comparisons in Fig. 8. Our method consistently achieves the lowest relative error (RE) and ECMS across a range of material parameters compared to other loss functions. The simulations generated by our approach closely align with the ground-truth, both in terms of material deformation and shape fidelity. These results show that optical flow guidance effectively captures complex material behaviors and enables precise material parameter optimization.

Effectiveness of GPT Initialization: Along with experimental results in Sec. 4.3 and Fig. 7, GPT initialization reduces errors, and our full method with optical flow guidance achieves the lowest ECMS score and realistic motion.

D. Limitations

Our method simulates deformations on 3D Gaussian splats and renders the resulting frames without incorporating relighting effects. This limits visual realism in aspects such as dynamic shadows and specular highlights. Future work could focus on integrating relighting techniques to capture more complex lighting interactions, thereby enhancing the overall fidelity and realism of the simulated scenes.

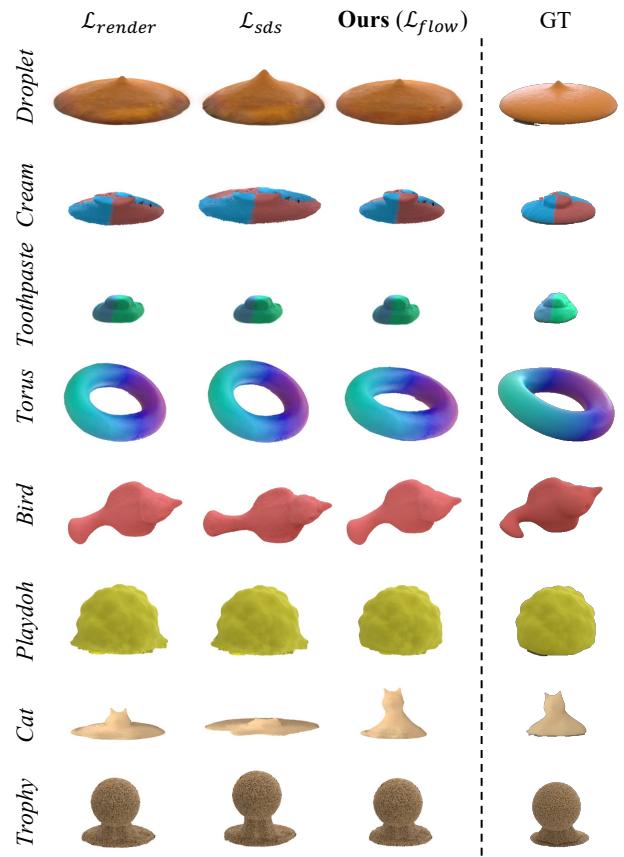


Figure 8. Qualitative results of ablation study comparing different loss functions for system identification on synthetic dataset.

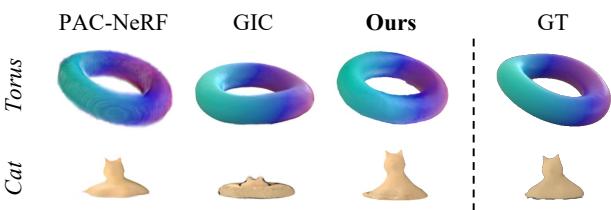


Figure 9. Qualitative results of all methods on synthetic dataset.

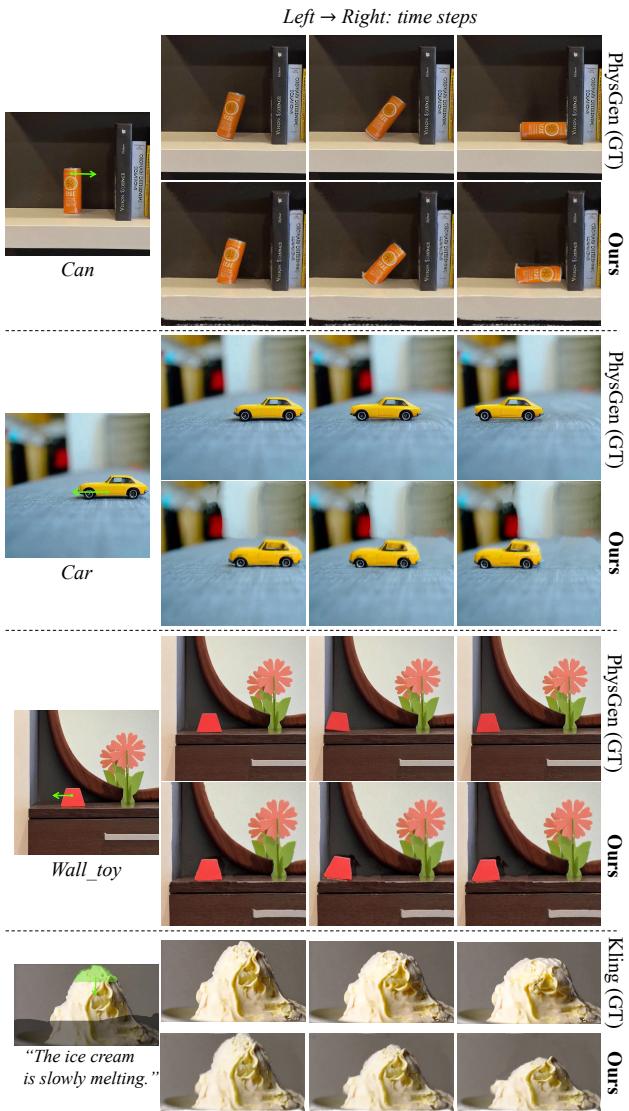


Figure 10. Qualitative results of our method using video guidance from PhysGen and Kling. The green arrows show the input force for the simulated objects. For Kling video generation, we utilize a text prompt and an input image, complemented by a motion brush (green mask) to define the motion trajectory (green arrow) and a static mask (gray mask) to restrict camera movement.

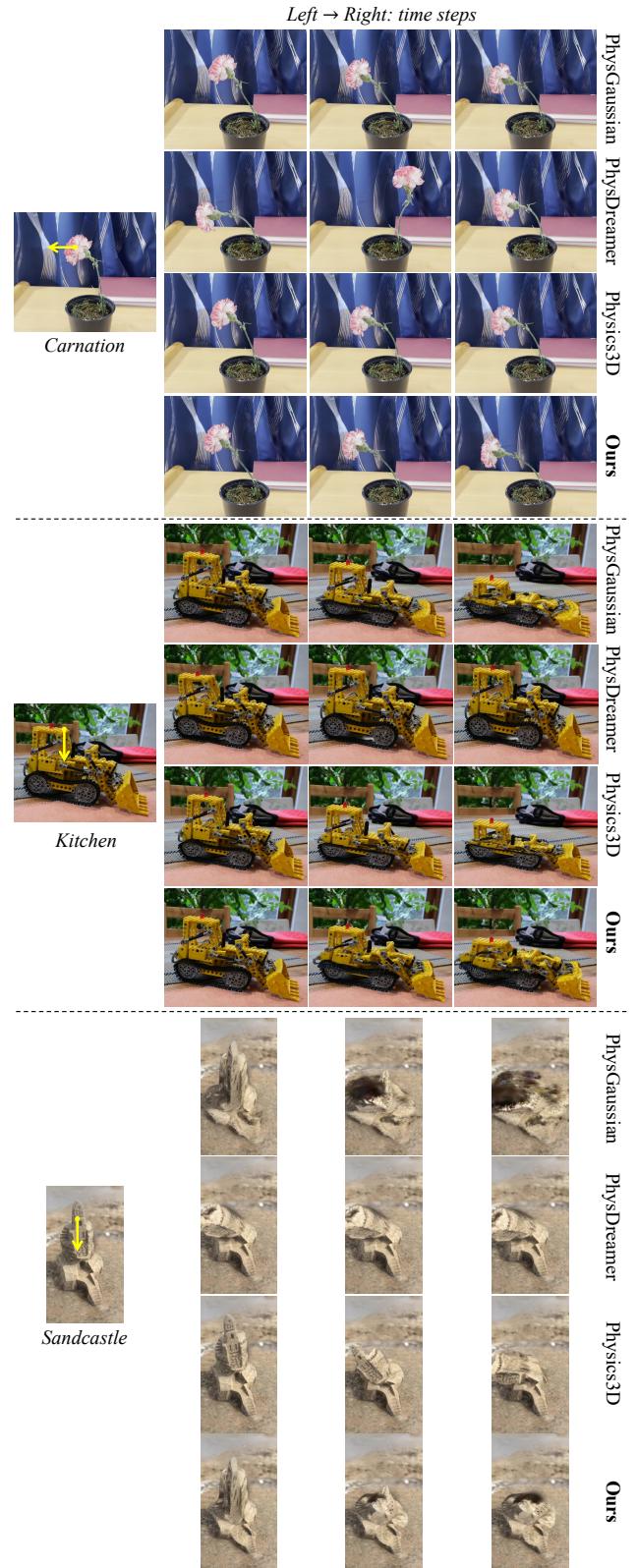


Figure 11. Qualitative results of all methods on real-world dataset. The yellow arrows show the input force for the simulated objects.

Object	\mathcal{L}_{render}	\mathcal{L}_{sds}	Ours (\mathcal{L}_{flow})
Droplet	$\delta_\mu = 0.045, \delta_\kappa = 0.080$	$\delta_\mu = \mathbf{0.005}, \delta_\kappa = 0.820$	$\delta_\mu = 0.004, \delta_\kappa = \mathbf{0.030}$
Letter	$\delta_\mu = 0.162, \delta_\kappa = 0.350$	$\delta_\mu = 0.050, \delta_\kappa = \mathbf{0.000}$	$\delta_\mu = \mathbf{0.023}, \delta_\kappa = 0.893$
Cream	$\delta_\mu = 11.100, \delta_\kappa = 0.570,$ $\delta_{\tau_Y} = 0.053, \delta_\eta = 0.440$	$\delta_\mu = \mathbf{0.030}, \delta_\kappa = \mathbf{0.480},$ $\delta_{\tau_Y} = \mathbf{0.007}, \delta_\eta = \mathbf{0.340}$	$\delta_\mu = \mathbf{0.080}, \delta_\kappa = 0.755,$ $\delta_{\tau_Y} = 0.577, \delta_\eta = 0.750$
Toothpaste	$\delta_\mu = 0.302, \delta_\kappa = 1.220,$ $\delta_{\tau_Y} = 0.140, \delta_\eta = \mathbf{0.023}$	$\delta_\mu = 0.162, \delta_\kappa = \mathbf{0.076},$ $\delta_{\tau_Y} = \mathbf{0.130}, \delta_\eta = 0.090$	$\delta_\mu = \mathbf{0.015}, \delta_\kappa = 0.136,$ $\delta_{\tau_Y} = 0.232, \delta_\eta = 0.539$
Torus	$\delta_E = 0.040, \delta_\nu = 0.073$	$\delta_E = \mathbf{0.010}, \delta_\nu = \mathbf{0.017}$	$\delta_E = 0.039, \delta_\nu = 0.989$
Bird	$\delta_E = 0.073, \delta_\nu = 0.090$	$\delta_E = \mathbf{0.027}, \delta_\nu = \mathbf{0.053}$	$\delta_E = 0.040, \delta_\nu = 0.571$
Playdoh	$\delta_E = 0.920, \delta_\nu = 0.093,$ $\delta_{\tau_Y} = 0.097$	$\delta_E = 0.210, \delta_\nu = \mathbf{0.073},$ $\delta_{\tau_Y} = \mathbf{0.013}$	$\delta_E = \mathbf{0.104}, \delta_\nu = 0.327,$ $\delta_{\tau_Y} = 0.027$
Cat	$\delta_E = 0.839, \delta_\nu = 0.023,$ $\delta_{\tau_Y} = 0.073$	$\delta_E = \mathbf{0.020}, \delta_\nu = \mathbf{0.013},$ $\delta_{\tau_Y} = \mathbf{0.023}$	$\delta_E = 0.387, \delta_\nu = 0.414,$ $\delta_{\tau_Y} = 0.221$
Trophy	$\delta_{\theta_{fric}} = 0.098$	$\delta_{\theta_{fric}} = 0.117$	$\delta_{\theta_{fric}} = \mathbf{0.013}$

Table 9. Comparisons with baselines for system identification performance on the synthetic dataset. δ_* denotes the relative error (RE) \downarrow for the material parameter $*$.

Object	\mathcal{L}_{render}	\mathcal{L}_{sds}	Ours (\mathcal{L}_{flow})
Droplet	$\delta_\mu = 0.230, \delta_\kappa = 0.731$	$\delta_\mu = 1.515, \delta_\kappa = 0.449$	$\delta_\mu = \mathbf{0.004}, \delta_\kappa = \mathbf{0.030}$
Letter	$\delta_\mu = 0.250, \delta_\kappa = 0.918$	$\delta_\mu = 0.575, \delta_\kappa = \mathbf{0.827}$	$\delta_\mu = \mathbf{0.023}, \delta_\kappa = 0.893$
Cream	$\delta_\mu = 0.160, \delta_\kappa = 0.732,$ $\delta_{\tau_Y} = \mathbf{0.070}, \delta_\eta = 0.841$	$\delta_\mu = 3.320, \delta_\kappa = \mathbf{0.004},$ $\delta_{\tau_Y} = 0.843, \delta_\eta = 0.893$	$\delta_\mu = \mathbf{0.080}, \delta_\kappa = 0.755,$ $\delta_{\tau_Y} = 0.577, \delta_\eta = \mathbf{0.750}$
Toothpaste	$\delta_\mu = 0.283, \delta_\kappa = 0.173,$ $\delta_{\tau_Y} = 0.480, \delta_\eta = 0.249$	$\delta_\mu = 0.255, \delta_\kappa = 0.141,$ $\delta_{\tau_Y} = 0.375, \delta_\eta = \mathbf{0.178}$	$\delta_\mu = \mathbf{0.015}, \delta_\kappa = \mathbf{0.136},$ $\delta_{\tau_Y} = \mathbf{0.232}, \delta_\eta = 0.539$
Torus	$\delta_E = 0.237, \delta_\nu = \mathbf{0.586}$	$\delta_E = 0.369, \delta_\nu = 0.771$	$\delta_E = \mathbf{0.039}, \delta_\nu = 0.989$
Bird	$\delta_E = 0.120, \delta_\nu = 0.616$	$\delta_E = 0.797, \delta_\nu = 0.727$	$\delta_E = \mathbf{0.040}, \delta_\nu = \mathbf{0.571}$
Playdoh	$\delta_E = 0.483, \delta_\nu = 0.196,$ $\delta_{\tau_Y} = 0.942$	$\delta_E = 0.828, \delta_\nu = \mathbf{0.165},$ $\delta_{\tau_Y} = 0.943$	$\delta_E = \mathbf{0.104}, \delta_\nu = 0.327,$ $\delta_{\tau_Y} = \mathbf{0.027}$
Cat	$\delta_E = \mathbf{0.167}, \delta_\nu = \mathbf{0.290},$ $\delta_{\tau_Y} = 0.652$	$\delta_E = 0.644, \delta_\nu = 0.623,$ $\delta_{\tau_Y} = 1.288$	$\delta_E = 0.387, \delta_\nu = 0.414,$ $\delta_{\tau_Y} = \mathbf{0.221}$
Trophy	$\delta_{\theta_{fric}} = 0.173$	$\delta_{\theta_{fric}} = 0.305$	$\delta_{\theta_{fric}} = \mathbf{0.013}$

Table 10. Ablation study of different losses for system identification performance on the synthetic dataset. δ_* denotes the relative error (RE) \downarrow for the material parameter $*$.