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

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	$rotation\ scale = -10 \rightarrow -5$	0.8→1
Kitchen	$(0, 0, 0.1)$	1
Jam	$(0.2, 0, 0) \rightarrow (0.1, 0.2, 0)$	2→1
Sandcastle	$release\ n_{layer} = 50$	-

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

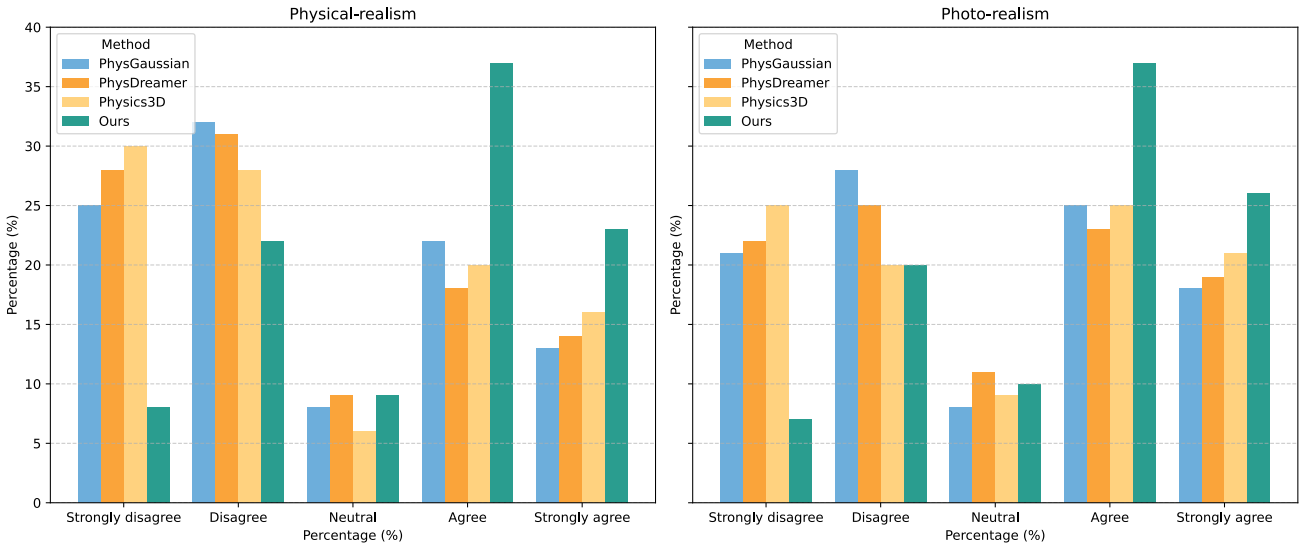

Figure 6. Human evaluation score distribution. The chart shows the score distribution for physical-realism and photo-realism based on human evaluations. Our method significantly outperforms the baseline methods across both metrics. The average score for our method approaches "Agree" for both criteria,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

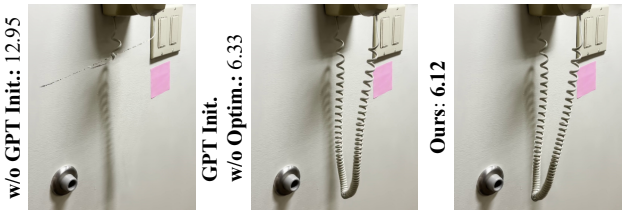

Figure 7. Ablations of GPT initialization with ECMS↓.

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.

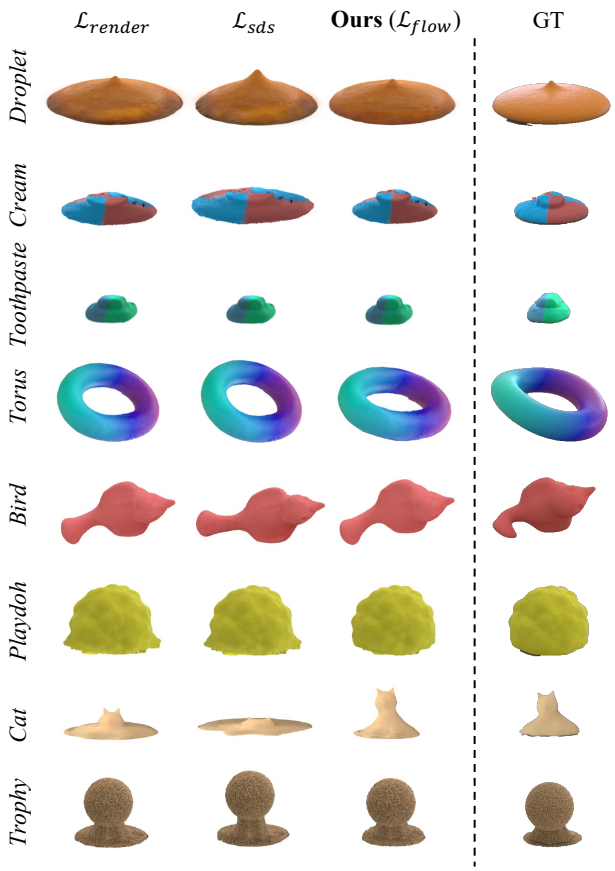

Figure 8. Qualitative results of ablation study comparing different loss functions for system identification on synthetic dataset.

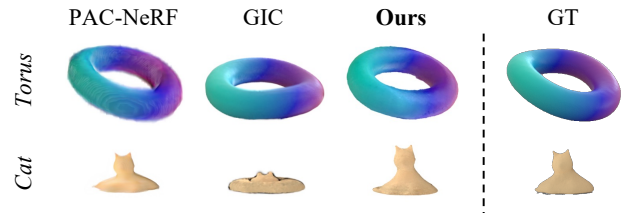

Figure 9. Qualitative results of all methods on synthetic dataset.

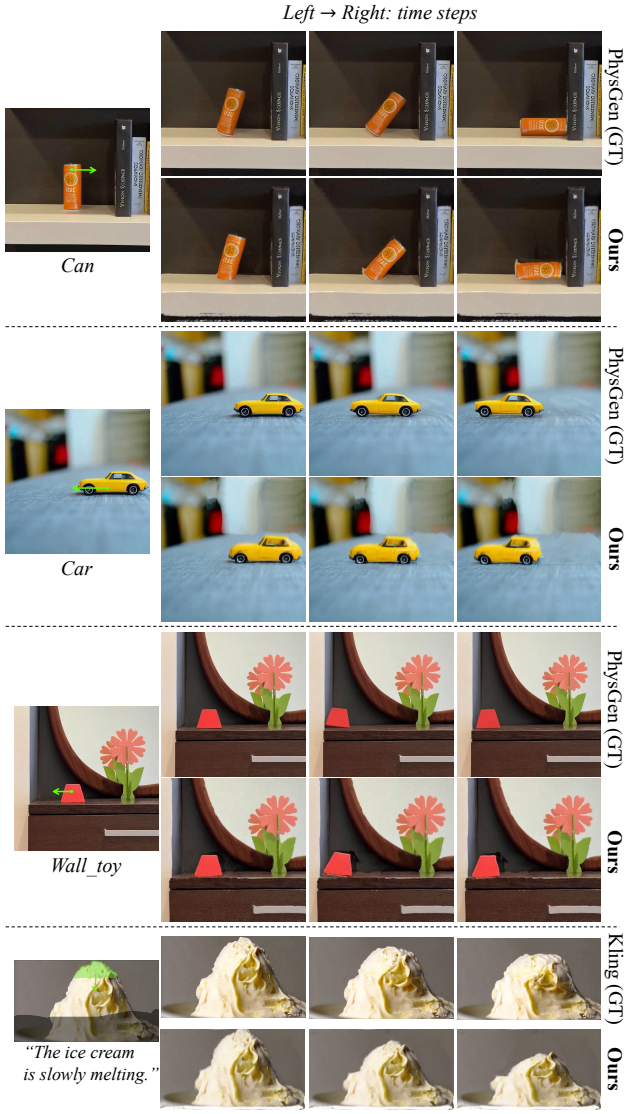

Figure 10. Qualitative results of our method using video guidance from PhysGen and Kling. The green arrows show the input force for the simulated objects. For Kling video generation, we utilize a text prompt and an input image, complemented by a motion brush (*green mask*) to define the motion trajectory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