MonoPhysics: Estimating Geometry, Appearance, and Physical Parameters from Monocular Videos

Daniel Rho[1] **Jun Myeong Choi**[1] **Matthew Thornton**[1] **Biswadip Dey**[2] **Roni Sengupta**[1]

[1]University of North Carolina at Chapel Hill [2]Meta

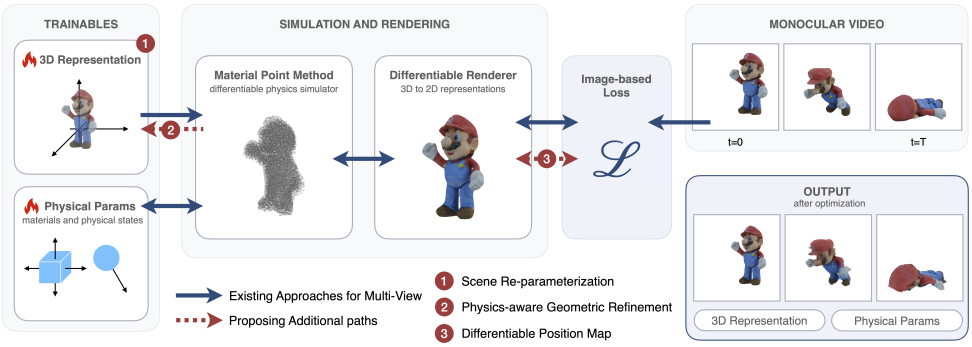


Figure 1: Overview of *MonoPhysics*. A trainable 3D representation and physical parameters are simulated and rendered, and supervised by image-based losses $\mathcal{L}$ against a monocular video. Solid blue arrows indicate optimization paths shared with existing multi-view methods, while red dashed arrows indicate the additional paths proposed in this work. *MonoPhysics* introduces three contributions, numbered in the figure: scene re-parameterization, physics-aware geometric refinement, and differentiable position map.

Abstract

Existing inverse physics methods recover physical parameters from multi-view videos, where geometric constraints across views resolve scale and 3D structure. In monocular settings, however, such constraints are absent, leading to severe scale ambiguity, inaccurate geometry, and weak coupling between appearance optimization and physical simulation. To address these challenges, we propose MonoPhysics, a framework for monocular inverse physics estimation of deformable objects that jointly optimizes geometry, appearance, and physical parameters using a differentiable simulator and 3D Gaussian Splatting. Our key contribution is removing the multi-view capture requirement of existing methods, a necessary step toward handling in-the-wild video. MonoPhysics introduces three visual-physical bridges: scene re-parameterization, physics-aware geometry refinement, and a differentiable position map. We evaluate on Vid2Sim, real-world captures, and a new dataset of elastic and plasticine objects that we introduce. MonoPhysics outperforms monocular baselines in future prediction and recovers Young's modulus on Vid2Sim with accuracy comparable to a multi-view baseline. Code and data are available at `https://daniel03c1.github.io/MonoPhysics/`.

1 Introduction

Recovering the physical properties of objects from video, such as material stiffness and deformation behavior, is important for robotic manipulation of deformable objects, as well as for building digital twins and virtual environments [10]. Recent advances in differentiable physics simulation [9, 22] and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

neural 3D representations [19, 13] have enabled *inverse physics*: recovering physical parameters by optimizing through differentiable simulations to match observed motion. Existing inverse physics methods [16, 33, 1, 32, 2] typically rely on synchronized multi-view captures, where geometric constraints across views resolve 3D structure and absolute scale. However, multi-view rigs are unavailable for the most common sources of dynamic video, including most in-the-wild captures. Removing the multi-view capture requirement is therefore an important step toward physical system identification from such videos.

Multi-view methods [16, 33, 1, 32, 2] rely on triangulation to resolve 3D structure and absolute scale before estimating physical parameters. The physics optimization therefore operates on geometry already constrained by visual evidence.

The monocular setting is significantly more challenging, as geometry must be inferred without multi-view constraints and is inherently ambiguous in scale. While recent 3D foundation models [27] can recover plausible geometry from a single image and provide a strong starting point, their outputs remain unreliable in occluded regions and lack absolute scale, both critical for physical simulation. Prior inverse physics methods typically decouple geometry reconstruction and physical parameter estimation, assuming a reliable 3D representation can be obtained from multi-view inputs [16, 33, 1, 32]. In the monocular setting, this decoupling breaks down, often requiring joint optimization of geometry, appearance, and physical parameters.

We present *MonoPhysics*, a framework for monocular inverse physics of deformable objects based on differentiable MPM simulation and 3D Gaussian Splatting. It is initialized from a 3D foundation model [27] and jointly refined under physical and visual supervision. Our key contribution is removing the multi-view capture requirement for accurate inverse physics, a step toward in-the-wild use. We use 3D Gaussians for both rendering and simulation: each Gaussian serves as a rendering primitive and as a material point for simulation. Thus, the visual and physical sides of the pipeline are coupled through differentiable rendering and simulation. In the monocular setting, we identify several connections between them that remain weak or missing, preventing visual cues from fully guiding the underlying physical parameters.

In this work, we propose three components that bridge the visual and physical sides of the pipeline. First, scene re-parameterization separates global scale from geometry. Since per-particle position updates produce only local corrections, a single learnable scalar helps aggregate a coherent global scale signal across all particles. Second, physics-aware geometry refinement dynamically computes and optimizes per-particle volumes, and uses both visual and physical importance to guide Gaussian relocation during optimization. Third, a differentiable position map provides gradients on the rendered positions of Gaussians, enabling direct position-based supervision that standard 3DGS rendering cannot provide. We optimize geometry, appearance, and physical parameters using a combination of rendering, optical flow, and silhouette losses, together with a particle distribution regularizer.

We evaluate MonoPhysics on the Vid2Sim benchmark [2], our new synthetic dataset of elastic and plasticine objects, and real captures from SpringGaus [33], comparing against existing methods [2, 1, 33, 16]. Our approach achieves the best future prediction and the lowest Young's modulus error among monocular methods on all three datasets. Notably, using only a single camera, it reaches Young's modulus accuracy on Vid2Sim comparable to that of a multi-view method.

2 Related Works

2.1 Inverse Physics from Video

Estimating physics from video sequences has been studied extensively [16, 20]. Differentiable simulation frameworks [9, 22] enable gradient-based optimization of physical parameters by differentiating through the simulation. This allows observed object behavior to directly supervise material properties, initial velocities, and other physical parameters. PAC-NeRF [16] combines differentiable MPM simulation with NeRF to optimize geometry and physical parameters from multi-view video. SpringGaus [33] integrates a spring-mass model into 3D Gaussian Splatting for reconstruction and simulation of elastic objects. GIC [1] applies point-wise losses on particles from a learned 4D representation to estimate physical parameters. Vid2Sim [2] uses feed-forward prediction for initialization, followed by lightweight optimization with a mesh-free differentiable simulator [21]. MASIV [32] introduces learnable neural constitutive models [18] that remove the need for hand-crafted constitu-

tive laws, enabling material-agnostic system identification. These methods benefit from multi-view constraints or direct 4D guidance, which reduces the problem to estimating physical parameters from already resolved geometry. In the monocular setting, however, these constraints are absent.

2.2 Monocular and Sparse-View Inverse Physics Estimation

Monocular video poses unique challenges for inverse physics. Without multi-view constraints, scale is ambiguous, geometry is inaccurate, and appearance provides only limited supervision. Prior monocular or limited-view methods partially address these issues. ProJo4D [26] improves parameter estimation and prediction in sparse-view settings, but still relies on multi-view input. NeuPhysics [25] learns a 4D representation from monocular video and estimates physical parameters from the reconstructed trajectories. However, it separates representation learning from parameter estimation and thus assumes accurate 4D reconstruction. PPR [31] resolves scale for articulated bodies, but relies on priors that do not apply to general deformable objects. Gao *et al.* [6] estimate the physical parameters of thin deformable objects from monocular video, assuming full surface visibility in the first frame. FluidNexus [5] reconstructs and predicts fluid dynamics from a single video, but is limited to fluids. In contrast, our framework directly addresses monocular-specific challenges, such as ambiguous scale and inaccurate geometry, using physics-derived signals rather than object-specific priors or multi-view constraints.

2.3 Physics-Informed Scene Understanding

Physical constraints and priors have been used to improve static scene reconstruction and infer object structure. PhyRecon [23] integrates differentiable rendering with particle-based physics simulation to produce physically stable static scene reconstructions. Guo *et al.* [7] optimize for physical compatibility to obtain plausible static objects from a single image. PhySIC [30] uses physical contact constraints for human-scene alignment from a single image. BrickGPT [24] uses physical stability constraints to produce buildable brick structures. TopoGaussian [29] and Structure from Collision [12] estimate interior structure from multi-view input. However, these methods either operate on static scenes or require multi-view input. Closer to our setting, Physics-Informed Deformable Gaussian Splatting [8] uses physics as regularization for monocular 4D reconstruction and learns time-varying material parameters, but does not perform full inverse estimation with forward prediction. In contrast, our work uses physics as a corrective signal specifically for monocular inverse estimation of dynamic deformable objects.

3 Method

We first formulate the problem overview (Sec. 3.1). Our three contributions are bridges between the visual and physical sides of the pipeline (Sec. 1): a global scale scalar (Sec. 3.2), physics-aware geometry refinement (Sec. 3.3), and a differentiable position map (Sec. 3.4). Loss functions and the full optimization procedure follow in Sec. 3.5.

3.1 Problem Formulation and Overview

The input is a monocular RGB video of a deformable object interacting with its environment, e.g., an object falling onto a table. We assume known camera parameters (intrinsics and extrinsics), a known ground plane, and a known constitutive model (e.g., Neo-Hookean elasticity), consistent with prior inverse physics methods [16, 33, 1]. We also require a foreground alpha mask for each frame. When ground-truth masks are unavailable, an off-the-shelf segmentation method can be used. Our goal is to jointly recover the object's geometry, appearance, and material parameters from these inputs. For 3D representation, we use 3D Gaussian Splatting (3DGS) [13], which integrates with particle-based simulations such as the Material Point Method (MPM). Each Gaussian has position $\mathbf{x}_i$, color $\mathbf{c}_i$, opacity o_i, and covariance $\mathbf{\Sigma}_i$ for appearance, and physical volume V_i. The initial physical state is represented by the initial velocity $\mathbf{v}_0$.

We use an off-the-shelf monocular 3D reconstruction method [27] to produce an initial set of 3D Gaussians from the first frame, then sample MPM particles from them. We use a differentiable MPM simulator [9] to advance them through time under the current physical parameters, producing deformed representations at each frame. At each frame, the color of every pixel C is rendered via 3D

Gaussian Splatting using alpha compositing [13]:

$$C = c_{bg} \prod_i (1 - \alpha_i) + \sum_i \alpha_i \, \mathbf{c}_i \prod_{j<i} (1 - \alpha_j), \tag{1}$$

where $\mathbf{c}_i$ is the color of Gaussian i, α_i is the product of the learned opacity and the Gaussian response of particle i at the pixel, and c_{bg} is the background color of that pixel. Losses are backpropagated through the differentiable simulation to update all parameters jointly, as detailed in Sec. 3.5.

3.2 Scale Estimation through Reparameterization

Monocular 3D reconstruction recovers scene geometry only up to an unknown scale factor, as points along the same projection ray are indistinguishable from the camera's perspective. For physics, however, absolute scale governs world-space positions, velocities, contact timing, volumes, and stress magnitudes, so all downstream physical quantities depend on it. Scale error therefore propagates into every estimated parameter.

Physical dynamics can resolve this ambiguity. However, naively optimizing per-particle positions cannot recover the global scale. Each particle's gradient mixes local position corrections with the global scale signal, so independent per-particle updates produce local adjustments rather than a coherent change in the object's distance from the camera (Fig. 2).

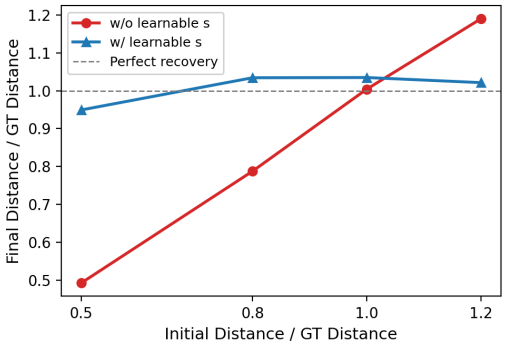


Figure 2: Effect of the learnable scale factor s on the recovered center-of-mass distance across different initial scales. Without s, the optimization cannot correct errors in the initial scale estimate, whereas with s, our method recovers a near-correct scale across tested initializations.

Existing methods store particles directly in world space [16, 33, 1, 32]. We instead maintain particles in camera space and introduce a single learnable scalar s that controls the global scale. Scaling in camera space moves particles along their projection rays, leaving 2D projections unchanged but changing all physical quantities. Since physics simulation requires world-space coordinates, we transform using the known camera extrinsics:

$$\mathbf{x}_i^w = \mathbf{R}\,(s \cdot \mathbf{x}_i) + \mathbf{t}, \tag{2}$$

where $\mathbf{R}, \mathbf{t}$ are the camera-to-world rotation and translation, $\mathbf{x}_i$ is the camera-space position. Because every particle's physics depends on s, image-space losses backpropagated through the differentiable simulation aggregate a coherent global signal for scale from all particles and all frames. This gives s a clean optimization path that per-particle degrees of freedom cannot provide (Fig. 2).

3.3 Physics-Aware Geometry Refinement

Accurately estimating geometry from a single camera is difficult: unobserved regions remain inherently ambiguous even with 3D geometric priors from foundation models. Our key idea is that a differentiable physics simulation (MPM) provides complementary signals to refine geometry, since physical behavior constrains regions that visual observation alone cannot resolve. In neural field-based representations [16, 12, 11], geometry can be refined by optimizing continuous density or occupancy fields. Particle-based representations lack this continuous structure, so geometry is instead captured through per-particle volumes.

Per-particle physical volume. Existing works that incorporate physics into 3DGS representations [1, 32] assign fixed per-particle volumes, computed once at initialization. In standard MPM initialization, these volumes are chosen to satisfy a local partition-of-unity over the simulation grid: the total volume contributed by particles to each grid node matches the cell volume Δx^3. More specifically, the total volume contributed to each grid node equals the cell volume Δx^3:

$$\sum_i V_i w_{i\ell} = \Delta x^3, \tag{3}$$

where Δx denotes cell edge size, and $w_{i\ell}$ is the contribution of particle i to grid node ℓ. During initialization, these per-particle physical volumes are determined. However, keeping volumes fixed throughout optimization prevents the representation from adapting its geometry and total volume.

We instead make volumes adaptive by maintaining the partition-of-unity (Eq. (3)), inducing volumes from the current particle distribution at each step. We compute them by iteratively refining per-particle volume to meet the partition of unity for T iterations. Further details are provided in the supplementary.

Physics-informed particle management. Adaptive volumes alone cannot add particles where geometry is under-resolved, nor remove particles wasted on negligible regions. We therefore complement adaptive volumes with a particle management scheme that relocates particles during optimization. Prior visual-only methods select particles using visual importance only [13, 15, 14]. We propose using physical importance in addition to visual importance. We define a visual importance p_i^{vis} from the rendering footprint, a physical importance p_i^{phys} from the physical volume, and the overall importance p_i as the average of the two:

$$p_i \propto \tfrac{1}{2}p_i^{vis} + \tfrac{1}{2}p_i^{phys}, \quad p_i^{vis} \propto \det(\mathbf{\Sigma}_i)^{1/2} \cdot o_i, \quad p_i^{phys} \propto V_i, \tag{4}$$

where $\mathbf{\Sigma}_i$ is the 3D covariance matrix and o_i is the learned opacity of particle i. 3DGS-MCMC [14] relocates low-opacity Gaussians to high-opacity ones and adjusts their opacity and scale so that the rendered image is approximately preserved. We reuse this relocation step but select which particles to remove and split using the importance in Eq. (4) instead of opacity alone. Specifically, we periodically remove particles with the lowest visual importance p_i^{vis} or physical importance p_i^{phys}, and replace them by splitting those with the highest overall importance p_i. This steers particle redistribution toward regions that are under-resolved either visually or physically.

3.4 Differentiable Position Map

Aligning a rendered shape to its target requires gradients that pull misplaced particles toward the correct silhouette. Standard image-space losses provide such gradients only indirectly and only where predicted and observed shapes already overlap: rendering losses act through colors and opacities at overlapping pixels, and optical flow supplies inter-frame motion cues that still rely on a roughly aligned reference shape. Particles that are globally misaligned therefore receive no directional gradient from either loss, a limitation that becomes pronounced under large deformations, where global cues are essential.

We bridge this gap by making each pixel position a differentiable function of the Gaussians that affect that pixel's values, either color or opacity. By writing each position relative to the covering Gaussian's image-space mean, we introduce the missing gradient path that enables shape supervision. Let $\mathbf{x}_i^{2D}$ and $\mathbf{\Sigma}_i^{2D}$ denote the 2D image-space mean and covariance of Gaussian i. For pixel $\mathbf{p}$ covered by Gaussian i, we define the reparameterized position:

$$\tilde{\mathbf{p}}_i = \mathbf{x}_i^{2D} + (\mathbf{\Sigma}_i^{2D})^{1/2}\mathbf{z}, \quad \mathbf{z} = \mathrm{sg}\left((\mathbf{\Sigma}_i^{2D})^{-1/2}(\mathbf{p} - \mathbf{x}_i^{2D})\right), \tag{5}$$

where $\mathrm{sg}(\cdot)$ the stop-gradient operator. We render a position map using the same alpha-compositing formula as in Eq. (1), substituting $\tilde{\mathbf{p}}_i$ for per-Gaussian color:

$$\tilde{\mathbf{p}} = \mathbf{p}\prod_i (1-\alpha_i) + \sum_i \alpha_i\,\tilde{\mathbf{p}}_i \prod_{j<i}(1-\alpha_j), \tag{6}$$

where $\tilde{\mathbf{p}}$ is the output pixel coordinates. Numerically, $\tilde{\mathbf{p}}$ always equals the actual pixel coordinate $\mathbf{p}$. This formulation lets pixel-location losses propagate positional gradients to Gaussian parameters.

3.5 Loss Functions and Optimization

We introduce the loss terms used across the whole optimization stages.

Image losses ($\mathcal{L}_{\text{color}}$, $\mathcal{L}_\alpha$). We use standard image losses (L1 + SSIM) for $\mathcal{L}_{\text{color}}$, and L1 loss between the rendered opacity map and the ground-truth alpha mask for $\mathcal{L}_\alpha$.

Optical flow loss ($\mathcal{L}_{\text{flow}}$). Optical flows can provide which parts should move and in which direction. Several prior works have similarly used optical flow to supervise motion estimation in differentiable rendering and inverse physics pipelines [34, 8, 17]. We render predicted flow by splatting per-particle world-space velocities into the image plane using the same Gaussian rasterizer. Optical flow estimation can be noisy and uncertain, especially when physical dynamics change what is

visible between frames. We therefore use a probability-based optical flow loss [28]. The detailed parameterization is given in the supplementary. To capture both global displacement and instantaneous motion, we supervise with two complementary optical flow signals: flow from the first frame to the current frame, and flow from the previous frame to the current frame:

$$\mathcal{L}_{\text{flow}} = -\frac{1}{|\Omega_t|} \sum_{j \in \Omega_t} \Big[\log p\left(\mathbf{f}_{0\to t}(\mathbf{p}_j) \mid \hat{\mathbf{f}}_{0\to t}(\mathbf{p}_j)\right) + \log p\left(\mathbf{f}_{t\text{-}1\to t}(\mathbf{p}_j) \mid \hat{\mathbf{f}}_{t\text{-}1\to t}(\mathbf{p}_j)\right) \Big], \quad (7)$$

where $\mathbf{p}_j$ is the position of pixel j, and Ω_t is the set of ground-truth foreground pixels at frame t. $\hat{\mathbf{f}}_{i\to j}(\cdot)$ and $\mathbf{f}_{i\to j}(\cdot)$ denote the rendered and estimated optical flow respectively from frame i to frame j at pixel $\mathbf{p}$. The first term anchors global displacement to the reference frame ($t = 0$), while the second term supervises instantaneous velocity.

Silhouette loss ($\mathcal{L}_{\text{sil}}$). Using the differentiable position map (Sec. 3.4), we obtain rendered pixel positions $\tilde{\mathbf{p}}_k$ that are differentiable with respect to Gaussian parameters. We then treat each foreground region as a weighted point cloud of these positions and align it to the target using optimal transport, which provides global directional gradients even when the predicted and target silhouettes do not overlap. Let Ω_t denote the set of target foreground pixels at frame t, with per-pixel alpha values $\{a_j\}_{j\in\Omega_t}$, and let $\hat{\Omega}_t$ denote the set of active rendered foreground pixels, with rendered opacity values $\{\tilde{a}_k\}_{k\in\hat{\Omega}_t}$. Since the number of pixels and their alpha values might differ, we use unbalanced debiased Sinkhorn divergence [4] as a loss:

$$\mathcal{L}^t_{\text{sil}} = \text{SD}(\{\tilde{\mathbf{p}}_k,\, \tilde{a}_k\}_{k\in\hat{\Omega}_t},\, \{\mathbf{p}_j,\, a_j\}_{j\in\Omega_t}), \quad (8)$$

where $\text{SD}(\cdot,\cdot)$ is the debiased Sinkhorn divergence and $\mathbf{p}_j$ is the GT pixel position. The resulting gradients indicate how each rendered pixel $\tilde{\mathbf{p}}_k$ should shift spatially and adjust its opacity $\tilde{a}_k$ to match the target silhouette.

Particle distribution regularizer ($\mathcal{L}_{\text{distr}}$). With trainable particle positions, the local distribution directly affects inverse physics estimation. Particles may detach from the body and drift independently, or cluster together and leave voids in their vicinity, both of which degrade the estimation of physical parameters. To mitigate these failure modes, we regularize the spatial distribution via a K-nearest-neighbor penalty:

$$\mathcal{L}_{\text{distr}} = \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_K(i)} \max\big(0,\, \|\mathbf{x}_i - \mathbf{x}_j\| - \tau_{\max}\Delta x\big) + \max\big(0,\, \tau_{\min}\Delta x - \|\mathbf{x}_i - \mathbf{x}_j\|\big), \quad (9)$$

where $\mathcal{N}_K(i)$ are the K nearest neighbors of particle i, $\tau_{\min}$ and $\tau_{\max}$ are lower and upper bounds in the unit of simulation cell size Δx. The first term pulls a particle toward its neighbors when they drift too far apart, while the second pushes neighbors apart when they become overly clustered. Both terms vanish in well-distributed regions, avoiding over-regularization of the particle distribution and the underlying geometry. Since each Gaussian can undergo a large deformation during simulation, we apply this regularizer only at the initial configuration (frame 0), before any deformation occurs.

Optimization strategy. Starting from initial 3D geometry obtained from an image-to-3D model [27], our optimization proceeds in two stages. In the first stage, we jointly optimize all physics parameters (material parameters and initial velocity), Gaussian parameters (positions $\mathbf{x}_i$, opacity o_i, and covariance $\mathbf{\Sigma}_i$), and global scale s (Sec. 3.2) via differentiable MPM simulation, while not optimizing colors $\mathbf{c}_i$. Following GIC [1], we focus on physical dynamics in this stage and refine appearance later. Within each iteration, after backpropagating through the simulation, we take a small number of alpha-map loss steps without re-running the simulation. This keeps the visual representation consistent with the evolving geometry. We use all four losses as follows:

$$\mathcal{L} = \sum_{t=0}^{T} \left[\mathcal{L}^t_{\alpha} + \mathcal{L}^t_{\text{sil}}\right] + \sum_{t=1}^{T} \mathcal{L}^t_{\text{flow}} + \mathcal{L}_{\text{distr}}. \quad (10)$$

In the second stage, we only refine appearance, using both image $\mathcal{L}_{\text{color}}$ and alpha $\mathcal{L}_{\alpha}$ losses. Details are in the supplementary.

Table 1: **Evaluation on the Vid2Sim dataset.** We report rendering quality metrics for future frame prediction and material parameter estimation errors. Monocular results are median [Q1, Q3]. Multi-view results, denoted by $*$, are taken from Vid2Sim [2] and reported as single values. Bold marks the best monocular method in each column, excluding *Oracle Midpoint*.

		Future Frame Rendering			Physical Parameter Estimation	
	views	**PSNR** ↑	**LPIPS** ↓	**SSIM** ↑	**MAE** $\log E$ ↓	**MAE** ν ↓
Vid2Sim*	12	25.07	-	0.95	0.51	0.06
Oracle Midpoint					0.57 [0.25, 0.77]	0.03 [0.02, 0.07]
PAC-NeRF	1	17.9 [16.6, 18.8]	0.15 [0.14, 0.19]	0.91 [0.89, 0.93]	2.07 [0.97, 2.83]	0.14 [0.08, 0.36]
SpringGaus	1	18.9 [16.4, 20.6]	0.16 [0.10, 0.20]	0.91 [0.89, 0.94]	–	–
GIC	1	18.7 [17.5, 20.3]	0.16 [0.14, 0.18]	0.92 [0.90, 0.93]	0.48 [0.25, 0.77]	**0.07** [0.03, 0.10]
Ours	1	**22.4** [20.5, 23.6]	**0.09** [0.07, 0.13]	**0.94** [0.92, 0.95]	**0.45** [0.21, 0.80]	0.10 [0.06, 0.13]

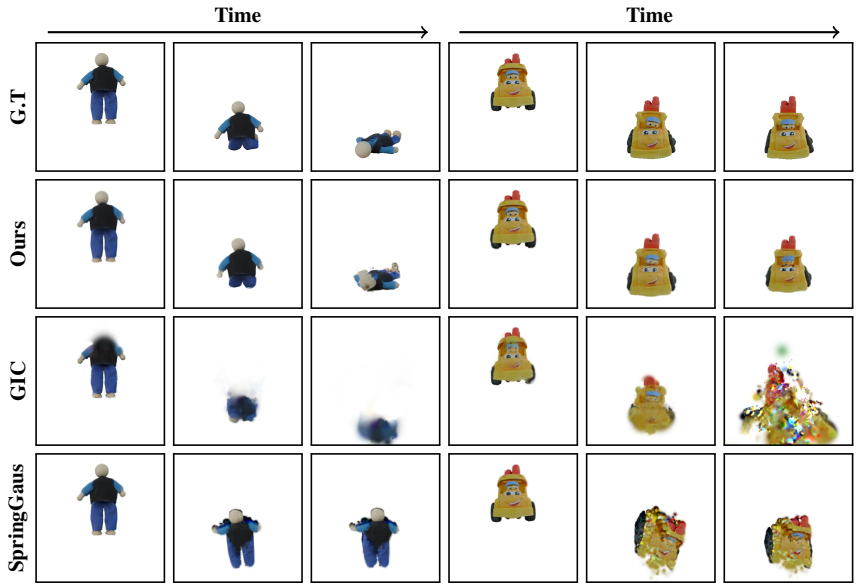


Figure 3: **Qualitative comparison on Vid2Sim dataset.** We compare the rendered image trajectory over time for SpringGaus [33], GIC [1], and our method.

4 Experiments

4.1 Experimental Settings

Baselines. We compare our method against existing inverse physics methods: PAC-NeRF [16], SpringGaus [33], GIC [1], and Vid2Sim [2]. Since Vid2Sim requires multiple views and cannot operate on monocular video input, we report only its multi-view results, taken from the original paper. We additionally report *Oracle Midpoint*, which ignores the input video and always predicts the midpoint of each parameter's sampling range. Since it has access to the sampling range, which is unavailable to the other methods, it serves only as a reference. Outperforming it indicates that a method recovers information from the video beyond what the range alone provides.

Datasets. We evaluate on two synthetic datasets and one real-world dataset. First, we use the Vid2Sim [2] dataset, which contains 12 elastic objects with no initial velocity. We run each scene independently from 5 different cameras, treating each camera as a separate monocular input (60 runs per method in total). Second, we introduce a new synthetic dataset built from Google Scanned Objects (GSO, CC-BY 4.0) [3], comprising 5 elastic (Neo-Hookean) and 5 plasticine objects. Each scene has randomly sampled material parameters, initial velocity, object size, and camera distance. As with Vid2Sim, each scene is run from 5 different cameras, giving 25 runs per material. For real-world evaluation, we use the real captures released with SpringGaus [33] (5 scenes with 3 cameras each). Since the real captures have no ground-truth material parameters, we report only the rendering quality of future predictions (Tab. 3).

Table 2: **Evaluation on our synthetic dataset.** We evaluate 3D trajectory accuracy and rendering quality on future frames. We also evaluate physical parameter estimation. Values are medians. Due to space, the full results with median [Q1, Q3] are in the supplementary material. Bold marks the best method in each column, excluding *Oracle Midpoint*.

	3D Trajectory		Future Frame Rendering			Physical Parameter Estimation			
	CD ↓	EMD ↓	PSNR ↑	LPIPS ↓	SSIM ↑	MAE $\mathbf{v}_0$ ↓	MAE $\log E$ ↓	MAE ν ↓	MAE $\log \sigma_y$ ↓
Elastic									
Oracle Midpoint						0.15	1.11	0.20	-
PAC-NeRF	438	0.74	17.4	0.19	0.92	**0.04**	1.33	**0.06**	-
SpringGaus	693	0.97	16.7	0.24	0.91	0.16	-	-	-
GIC	421	0.75	18.5	0.18	0.92	0.09	0.66	0.21	-
Ours	**24**	**0.26**	**21.1**	**0.12**	**0.93**	0.09	**0.31**	0.07	-
Plasticine									
Oracle Midpoint						0.22	0.32	0.01	0.81
PAC-NeRF	247	0.62	19.8	0.14	0.94	0.14	2.55	0.17	0.57
SpringGaus	450	0.67	17.3	0.20	0.92	0.13	-	-	-
GIC	333	0.65	22.0	0.14	0.94	**0.12**	0.95	0.07	0.34
Ours	**19**	**0.24**	**23.4**	**0.09**	**0.95**	0.14	**0.40**	**0.05**	**0.33**

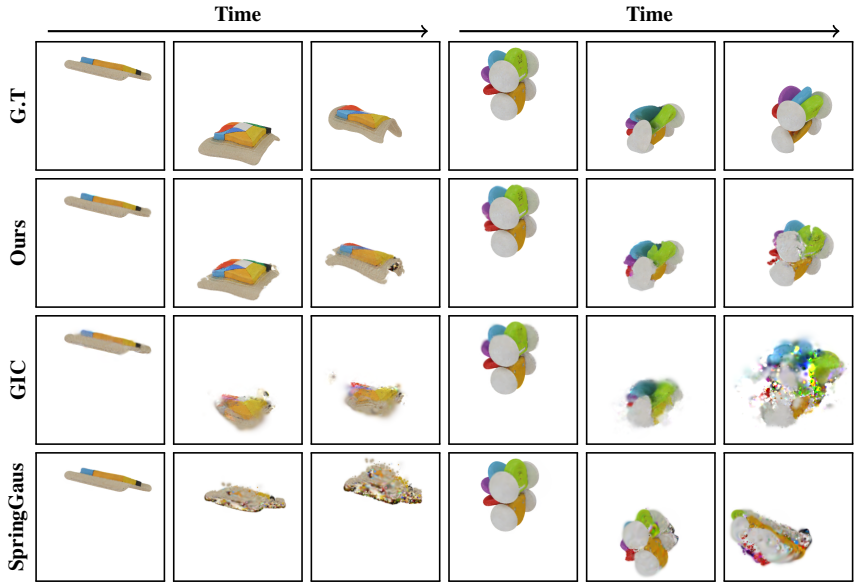


Figure 4: **Qualitative comparison on our synthetic dataset.** We compare the rendered image trajectory over time for SpringGaus [33], GIC [1], and our method. (Left: elastic. Right: plasticine.)

Metrics. We report three groups of metrics. For future prediction, we use PSNR, SSIM, and LPIPS, evaluated from the same camera viewpoint. For geometric accuracy, we use Chamfer Distance (CD) and Earth Mover's Distance (EMD). CD is computed as the bidirectional Chamfer distance between the predicted Gaussian positions and the ground-truth point cloud. Both CD and EMD are averaged over the future frames. For physical parameter estimation, we report the mean absolute error (MAE), following the standard evaluation protocol [16, 33, 1]. Since Young's modulus E and density cannot be identified independently from motion, all methods use a known density, following prior work. Therefore, the MAE of $\log E$ should be interpreted as accuracy under a known density. Because several metrics (CD in particular) are strongly skewed, we report the median and interquartile range. Due to space, Tab. 2 reports medians only, and its full results are in the supplementary material.

4.2 Comparison

Vid2Sim benchmark. Tab. 1 and Fig. 3 show results on Vid2Sim. Among monocular methods, ours achieves the best rendering quality on all three metrics and the lowest MAE $\log E$. Notably, our MAE $\log E$ is on par with multi-view Vid2Sim using 12 cameras (0.45 vs. 0.51). For ν, GIC achieves a slightly lower MAE than ours, and no monocular method outperforms *Oracle Midpoint*.

Table 3: **Evaluation on real captures from SpringGaus [33].** Values are median [Q1, Q3].

	PSNR ↑	LPIPS ↓	SSIM ↑
SpringGaus	28.94 [26.28, 29.91]	0.0128 [0.0109, 0.0183]	0.9952 [0.9934, 0.9958]
GIC	25.89 [25.16, 26.38]	0.0255 [0.0234, 0.0282]	0.9922 [0.9914, 0.9928]
Ours	**32.33** [31.46, 33.38]	**0.0095** [0.0084, 0.0102]	**0.9957** [0.9950, 0.9962]

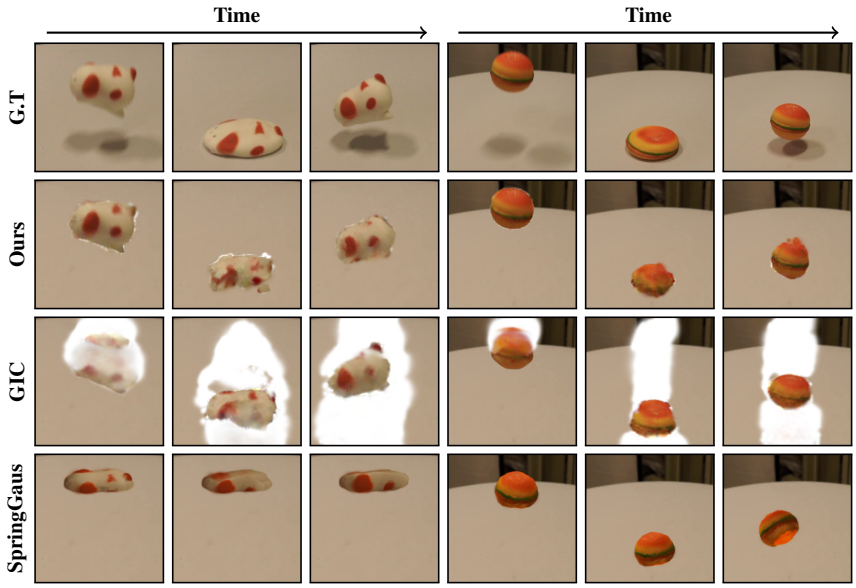


Figure 5: **Qualitative comparison on SpringGaus real-world dataset.** We compare the rendered image trajectory over time for SpringGaus [33], GIC [1], and our method.

Table 4: **Ablation study on loss components.** Values are median [Q1, Q3].

$\mathcal{L}_\alpha$	$\mathcal{L}_{\text{distr}}$	$\mathcal{L}_{\text{flow}}$	$\mathcal{L}_{\text{sil}}$	CD ↓	PSNR ↑	MAE $\log E$ ↓	MAE $\log \sigma_y$ ↓
✓	–	–	–	78 [33, 294]	21.8 [20.3, 23.9]	0.34 [0.13, 0.82]	0.45 [0.30, 0.98]
✓	✓	–	–	64 [14, 183]	22.2 [19.8, 28.1]	**0.23** [0.13, 1.02]	0.27 [0.11, 0.62]
✓	✓	✓	–	71 [32, 153]	22.9 [20.8, 24.9]	0.50 [0.17, 1.18]	**0.21** [0.16, 0.53]
✓	✓	–	✓	26 [11, 72]	22.9 [21.4, 24.4]	0.36 [0.17, 0.64]	0.52 [0.28, 0.85]
✓	✓	✓	✓	**19** [7, 72]	**23.4** [22.4, 25.4]	0.40 [0.14, 0.81]	0.33 [0.19, 0.50]

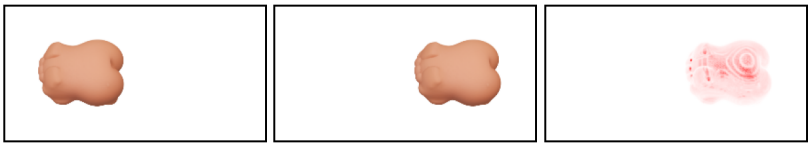
Our synthetic benchmark. Tab. 2 and Fig. 4 provide results on our synthetic dataset. Our method achieves the lowest CD and EMD for both material types, indicating more accurate geometry recovery, and the best rendering quality on all three metrics. We attribute these improvements to resolving scale ambiguity and jointly refining geometry with physical parameters. As shown in Fig. 4, our reconstructions follow the observed deformation and maintain realistic contact with the ground, whereas the baselines often produce incorrectly scaled or distorted shapes.

For material parameters, our method achieves the lowest MAE on Young's modulus for both materials, and on Poisson's ratio and yield stress for plasticine. However, yield stress is the only plasticine material parameter for which any method outperforms *Oracle Midpoint*. SpringGaus is excluded from the material parameter comparison because it uses a spring-mass simulator whose parameters are not directly comparable to those of the other methods. Initial-velocity errors are similar across methods, except that PAC-NeRF achieves notably lower error on elastic objects.

Real-world captures. Tab. 3 and Fig. 5 show results on the real captures from SpringGaus [33]. Our method predicts future frames more accurately than the baselines on all three rendering metrics, showing that it also works on real captures and not only on synthetic data.

4.3 Ablation Study

Loss Functions. Tab. 4 ablates the contribution of each loss from Sec. 3.5 on the plasticine subset of our synthetic dataset. Adding the distribution regularizer to the baseline reduces CD and lowers both



(a) Target frame (b) Rendered image (c) Position gradient map

Figure 7: Effect of the differentiable position map. The per-Gaussian position gradient $-\partial\mathcal{L}_{\text{sil}}/\partial\mathbf{x}_i$ points left (red), the correct direction toward the target shape (Sec. 3.4).

material parameter errors, consistent with its role in preventing disconnected particles from simulating independently and distorting future predictions. However, without positional guidance from optical flow or silhouette supervision, CD remains high. Optical flow supervision alone does not reduce CD. In contrast, silhouette supervision alone (Sec. 3.4) yields the largest improvement in trajectory and geometry (CD from 64 to 26), reflecting its role as a global shape-alignment signal. Combining both losses achieves the lowest CD and the highest PSNR, indicating that flow and silhouette supervision are complementary: flow anchors the physical trajectory, while silhouette supervision aligns the object shape.

Geometry Refinement. We first illustrate how physics-aware geometry refinement improves the reconstruction quality of the initial geometry in the first frame. To isolate geometric accuracy from scale ambiguity, we align both the initial and refined geometry to the true global scene scale before comparing them against the ground-truth point cloud using CD. Tab. 5 shows that refinement reduces not only the initial CD but also the future-prediction CD and the MAE of $\log E$. Fig. 6 visualizes the per-region geometric error before (*top row*) and after (*bottom row*) refinement, showing reduced error across multiple views.

Sec. 3.3	**Init CD** ↓	**Future CD** ↓	**MAE log E** ↓
-	10.79	223	1.10
✓	4.69	78	0.55

Table 5: Effect of physics-aware geometric refinement on a representative scene.

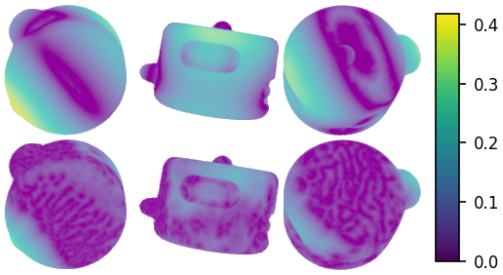


Figure 6: Per-region geometric error before (*top row*) and after (*bottom row*) physics-aware refinement, projected onto the ground-truth mesh from multiple views.

Differentiable Position Map. To visualize the gradients provided by the differentiable position map, Fig. 7 renders the positional gradient $-\partial\mathcal{L}_{\text{sil}}/\partial\mathbf{x}_i$ of each Gaussian as an image, alongside the target and the current rendering. The gradients point toward the target shape (leftward, shown in red). Further details are provided in the supplementary material.

5 Conclusion

We present *MonoPhysics*, a framework for monocular inverse physics of general deformable objects that jointly recovers geometry, appearance, and material parameters from a single-camera video. Our key insight is that, without multi-view constraints, several paths between the visual and physical sides of the inverse physics pipeline become weak or missing. Thus, we propose three visual-physical bridges: scene re-parameterization, physics-aware particle management, and a differentiable position map. Together, these bridges yield more accurate future prediction than prior monocular methods.

Our method also has limitations that point to promising future directions. First, our refinement reduces but does not eliminate geometric errors, and regions where neither visual nor physical signals are informative remain difficult. Second, following prior work, our pipeline assumes known camera parameters, a known ground plane, and a known constitutive model, which limits its applicability to fully in-the-wild videos. Third, all evaluated sequences are gravity-driven falls onto a flat plane, and other scenarios remain untested. Because a ballistic trajectory under known gravity can make absolute scale observable, scale may be more difficult to recover in these other scenarios. Relaxing these assumptions is a natural next step toward monocular inverse physics in the wild.

Acknowledgments and Disclosure of Funding

This work is supported by a National Institute of Health (NIH) NIBIB project #R21EB035832 and National Science Foundation (NSF) CAREER Award #2543161.

References

[1] Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gic: Gaussian-informed continuum for physical property identification and simulation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 75035–75063. Curran Associates, Inc., 2024.

[2] Chuhao Chen, Zhiyang Dou, Chen Wang, Yiming Huang, Anjun Chen, Qiao Feng, Jiatao Gu, and Lingjie Liu. Vid2sim: Generalizable, video-based reconstruction of appearance, geometry and physics for mesh-free simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26545–26555, June 2025.

[3] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022.

[4] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2681–2690. PMLR, 16–18 Apr 2019.

[5] Yue Gao, Hong-Xing Yu, Bo Zhu, and Jiajun Wu. Fluidnexus: 3d fluid reconstruction and prediction from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

[6] Zhiyuan Gao, Jiageng Mao, Hong-Xing Yu, Haozhe Lou, Emily Yue ting Jia, Jernej Barbic, Jiajun Wu, and Yue Wang. Seeing the wind from a falling leaf. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] Minghao Guo, Bohan Wang, Pingchuan Ma, Tianyuan Zhang, Crystal Elaine Owens, Chuang Gan, Joshua B. Tenenbaum, Kaiming He, and Wojciech Matusik. Physically compatible 3d object modeling from a single image. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[8] Haoqin Hong, Ding Fan, Fubin Dou, Zhi-Li Zhou, Haoran Sun, Congcong Zhu, and Jingrun Chen. Physics-informed deformable gaussian splatting: Towards unified constitutive laws for time-evolving material field. *arXiv preprint arXiv:2511.06299*, 2025.

[9] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. Difftaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations*, 2020.

[10] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *ICCV*, 2025.

[11] Takuhiro Kaneko. Improving physics-augmented continuum neural radiance field-based geometry-agnostic system identification with lagrangian particle optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5480, June 2024.

[12] Takuhiro Kaneko. Structure from collision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16314–16324, June 2025.

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[14] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 80965–80986. Curran Associates, Inc., 2024.

[15] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, June 2024.

[16] 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. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] Zhuoman Liu, Weicai Ye, Yan Luximon, Pengfei Wan, and Di Zhang. Unleashing the potential of multi-modal foundation models and video diffusion for 4d dynamic physical scene simulation. *CVPR*, 2025.

[18] Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B. Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. Learning neural constitutive laws from motion observations for generalizable PDE dynamics. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23279–23300. PMLR, 23–29 Jul 2023.

[19] 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. *Commun. ACM*, 65(1):99–106, December 2021.

[20] Himangi Mittal, Peiye Zhuang, Hsin-Ying Lee, and Shubham Tulsiani. Uniphy: Learning a unified constitutive model for inverse physics simulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16208–16218, 2025.

[21] Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David I. W. Levin. Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Trans. Graph.*, 43(4), July 2024.

[22] J. Krishna Murthy, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jérôme Parent-Lévesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. In *International Conference on Learning Representations*, 2021.

[23] Junfeng Ni, Yixin Chen, Bohan Jing, Nan Jiang, Bin Wang, Bo Dai, Puhao Li, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Phyrecon: Physically plausible neural scene reconstruction. 2024.

[24] Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. Generating physically stable and buildable brick structures from text. In *ICCV*, 2025.

[25] Yi-Ling Qiao, Alexander Gao, and Ming C. Lin. Neuphysics: Editable neural geometry and physics from monocular videos. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[26] Daniel Rho, Jun Myeong Choi, Biswadip Dey, and Roni Sengupta. Projo4d: Progressive joint optimization for sparse-view inverse physics estimation. *Transactions on Machine Learning Research*, 05 2026.

[27] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images. 2025.

[28] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 36–54, Cham, 2025. Springer Nature Switzerland.

[29] Xiaoyu Xiong, Changyu Hu, Chunru Lin, Pingchuan Ma, Chuang Gan, and Tao Du. Topogaussian: Inferring internal topology structures from visual clues. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Pradyumna Yalandur Muralidhar, Yuxuan Xue, Xianghui Xie, Margaret Kostyrko, and Gerard Pons-Moll. Physic: Physically plausible 3d human-scene interaction and contact from a single image. 2025.

[31] Gengshan Yang, Shuo Yang, John Z. Zhang, Zachary Manchester, and Deva Ramanan. Physically plausible reconstruction from monocular videos. In *ICCV*, 2023.

[32] Yizhou Zhao, Haoyu Chen, Chunjiang Liu, Zhenyang Li, Charles Herrmann, Junhwa Hur, Yinxiao Li, Ming-Hsuan Yang, Bhiksha Raj, and Min Xu. Toward material-agnostic system identification from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5944–5956, October 2025.

[33] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 407–423, Cham, 2025. Springer Nature Switzerland.

[34] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. MotionGS: Exploring explicit motion guidance for deformable 3d gaussian splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

A Implementation Details

Optimization. We optimize the physical dynamics parameters for 250 iterations, using the same schedule for all constitutive models. The number of simulated frames per iteration starts at 4 and increases linearly to the full sequence over the first 150 iterations. The remaining iterations use the full sequence. Within each iteration, we also optimize appearance-related parameters (excluding position and physical parameters) for 100 steps on randomly sampled frames. After dynamics optimization, we refine appearance for an additional 5,000 iterations. For real-world captures, all methods (including ours) are run for a total of 500 iterations, following the SpringGaus configuration. For our method, we use the same curriculum as above, reaching the full sequence after 150 iterations.

Loss Hyperparameters. As shown in Eq. (10), all loss terms are combined with unit weights. For the particle distribution regularizer $\mathcal{L}_{\text{distr}}$, we use $K = 3$ nearest neighbors per particle and set the lower and upper bounds to $\tau_{\min} = 0.3$ and $\tau_{\max} = 0.8$, respectively. These bounds leave sufficient room for adaptive geometry while preventing overly aggressive regularization.

Learning Rates. We use the Adam optimizer with fixed learning rates throughout optimization and apply the same hyperparameters across all synthetic scenes. The global scale s and the initial velocity $\mathbf{v}_0$ both use a learning rate of 0.02. For material parameters, we use 0.1 for Young's modulus E, 0.01 for Poisson's ratio ν, and 0.05 for yield stress σ_y. For Gaussian features, we use 0.0002 for position and (visual) Gaussian scale, and 0.01 for color and opacity. For real-world captures, we instead use 0.05 for $\mathbf{v}_0$ and ν, 0.01 for s, and 0.0001 for position and Gaussian scale, keeping all other learning rates unchanged.

Computational Resources and Cost. All experiments are conducted on a single NVIDIA RTX A6000 GPU. Optimization takes approximately 1–2 hours per run for both our method and the baselines.

B Physics-Aware Geometry Refinement

Per-particle physical volume. The detailed algorithm is described in algorithm 1. In our implementation, we run the iteration for 5 steps. Given that 50 to 200 simulation steps are typically required between two consecutive frames in our datasets, this costs less than $1/10$ of a single forward simulation pass for one frame.

Particle management. Every 10 iterations, we identify the bottom 1% of Gaussians by either visual importance p_i^{vis} or physical importance p_i^{phys}. We replace them by deterministically selecting the same number of Gaussians with the highest overall importance p_i, splitting each in half, and relocating the resulting copies to the positions of the removed ones. For relocation, we use the algorithm proposed in 3DGS-MCMC [14].

C Differentiable Position Map

We expand on the position-map reparameterization introduced in Sec. 3.4, showing why standard 3DGS rendering does not provide pixel-coordinate gradients and how the proposed reparameterization enables them.

Standard 3DGS rendering provides no positional gradient from pixel-coordinate losses. In Eq. (1), the rendered color at pixel $\mathbf{p}$ is a function of the per-Gaussian quantities $\{\mathbf{c}_i, o_i, \mathbf{x}_i^{2D}, \mathbf{\Sigma}_i^{2D}\}$, while $\mathbf{p}$ itself is a fixed integer pixel index. Therefore, for any loss $\mathcal{L}(\mathbf{p})$ defined on pixel *coordinates* rather than rendered colors,

$$\frac{\partial \mathcal{L}(\mathbf{p})}{\partial \mathbf{x}_i^{2D}} = 0. \tag{11}$$

In other words, standard rasterization provides no gradients from a pixel-coordinate loss to any Gaussian's image-space position.

Algorithm 1 Iterative Volume Estimation

Require: Positions $\{\mathbf{x}_i\}_{i=1}^N$, grid spacing Δx, iterations T
Notation: $i \in \{1, \dots, N\}$: particle index; ℓ: grid node index
$w_{i\ell} \geq 0$: B-spline weight from particle i to node ℓ; Δx^3: cell volume
Initialization
$n_\ell \leftarrow \sum_i w_{i\ell}$ ▷ weighted particle count at node ℓ
$\hat{n}_i \leftarrow \sum_\ell w_{i\ell}\, n_\ell$ ▷ weighted sum of node occupancies over all nodes influencing particle i
$V_i \leftarrow \min\left(\Delta x^3/\hat{n}_i,\ \Delta x^3\right)$ ▷ initial volume: one cell shared among $\hat{n}_i$ particles
Iterative refinement
for $t = 1, \dots, T$ **do**
  $\tilde{V}_\ell \leftarrow \sum_i w_{i\ell}\, V_i$ ▷ total volume attributed to node ℓ
  $c_\ell \leftarrow \min\left(\Delta x^3/\tilde{V}_\ell,\ 1\right)$ ▷ correction: scale down overfull nodes, leave underfull ones
  $V_i \leftarrow \min\left(V_i \cdot \sum_\ell w_{i\ell}\, c_\ell,\ \Delta x^3\right)$ ▷ update V_i by gathering node corrections
end for
return $\{V_i\}_{i=1}^N$

The differentiable position map provides positional gradients from pixel-coordinate losses. For a pixel $\mathbf{p}$ covered by Gaussian i, the differentiable position from Eq. (5) is

$$\tilde{\mathbf{p}}_i = \mathbf{x}_i^{2D} + (\boldsymbol{\Sigma}_i^{2D})^{1/2}\mathbf{z}, \qquad \mathbf{z} = \mathrm{sg}\Big((\boldsymbol{\Sigma}_i^{2D})^{-1/2}(\mathbf{p} - \mathbf{x}_i^{2D})\Big).$$

In the forward pass, $(\boldsymbol{\Sigma}_i^{2D})^{1/2}\mathbf{z} = \mathbf{p} - \mathbf{x}_i^{2D}$, so $\tilde{\mathbf{p}}_i = \mathbf{p}$ numerically. That is, the rendered position map equals the pixel grid. However, because $\mathbf{z}$ is treated as a constant under the stop-gradient,

$$\frac{\partial \tilde{\mathbf{p}}_i}{\partial \mathbf{x}_i^{2D}} = \mathbf{I}. \tag{12}$$

A pixel-coordinate loss applied to $\tilde{\mathbf{p}}_i$ therefore passes its gradient directly to the 2D mean of the covering Gaussian, from which it propagates through the projection and the differentiable simulation back to the Gaussians' 3D positions and physical state.

D Dataset Details

Scene composition. Our synthetic dataset is built from 10 source meshes drawn from Google Scanned Objects (GSO) [3]: 5 objects simulated with an elastic (Neo-Hookean) constitutive model and 5 simulated with a plasticine constitutive model. For each scene, the material parameters (Young's modulus E, Poisson's ratio ν, and yield stress σ_y for plasticine) and the initial velocity are independently sampled at random. For elastic objects, we sample $\log_{10} E$ from $(3.5, 7.0)$ and ν from $(0, 0.49)$. For plasticine objects, we sample $\log_{10} E$ from $(4.0, 5.5)$, ν from $(0.2, 0.45)$, and $\log_{10} \sigma_y$ from $(0.1, 1.5)$. All scenes share a fixed density of $1000\,\mathrm{kg/m^3}$, gravity of $9.81\,\mathrm{m/s^2}$, and a frictionless ground plane at $z = 0$. Each sequence is simulated with MPM at 20 fps.

Scenes. Tab. 6 lists the 10 source GSO objects and the constitutive model used for each.

Constitutive models. Both material models use the same conversion from E and ν to the Lamé parameters:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{E\,\nu}{(1+\nu)(1-2\nu)}. \tag{13}$$

Neo-Hookean (elastic). We use the compressible Neo-Hookean Kirchhoff stress:

$$\boldsymbol{\tau} = \mu\,\mathbf{F}\mathbf{F}^\top + \big(\lambda \ln J - \mu\big)\mathbf{I}, \qquad J = \det \mathbf{F}, \tag{14}$$

where $\mathbf{F}$ is the deformation gradient.

Plasticine. The plasticine model combines a Fixed Corotated elastic stress with a von Mises return mapping in principal log-strain space. Using the polar decomposition $\mathbf{F} = \mathbf{R}\mathbf{S}$, the elastic Kirchhoff stress is

$$\boldsymbol{\tau} = 2\mu\,(\mathbf{F} - \mathbf{R})\mathbf{F}^\top + \lambda\, J\,(J-1)\,\mathbf{I}. \tag{15}$$

Table 6: Source GSO objects and their assigned constitutive model.

Object	Constitutive model
BIRD_RATTLE	Neo-Hookean
CHICKEN_NESTING	Neo-Hookean
PEEKABOO_ROLLER	Neo-Hookean
TWISTED_PUZZLE	Neo-Hookean
WHALE_WHISTLE_6PCS_SET	Neo-Hookean
BABY_CAR	Plasticine
COAST_GUARD_BOAT	Plasticine
LACING_SHEEP	Plasticine
MINI_EXCAVATOR	Plasticine
OWL_SORTER	Plasticine

Plasticity is enforced by a return mapping on the trial elastic deformation gradient. Given the SVD $\mathbf{F}^{\text{trial}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, we compute the principal log strains $\boldsymbol{\varepsilon} = \ln \boldsymbol{\Sigma}$ and their deviatoric part $\hat{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\varepsilon})\,\mathbf{I}$. The von Mises yield indicator is

$$y = \|\hat{\boldsymbol{\varepsilon}}\| - \frac{\sigma_y}{2\mu}. \tag{16}$$

If $y > 0$, we project $\boldsymbol{\varepsilon} \leftarrow \boldsymbol{\varepsilon} - y\,\hat{\boldsymbol{\varepsilon}}/\|\hat{\boldsymbol{\varepsilon}}\|$ and update $\mathbf{F} \leftarrow \mathbf{U}\exp(\boldsymbol{\varepsilon})\mathbf{V}^\top$. Otherwise, $\mathbf{F}$ is left unchanged.

E Full Results on Our Synthetic Dataset

Due to space, Tab. 2 in the main paper reports medians only. Tab. 7 provides the full results as median [Q1, Q3].

Table 7: **Full evaluation on our synthetic dataset.** We evaluate 3D trajectory accuracy and rendering quality on future frames. We also evaluate physical parameter estimation. Values are median [Q1, Q3]. This extends Tab. 2, which reports medians only due to space. Bold marks the best method in each column, excluding *Oracle Midpoint*.

	3D Trajectory		Future Frame Rendering			Physical Parameter Estimation			
	CD ↓	EMD ↓	PSNR ↑	LPIPS ↓	SSIM ↑	MAE $\mathbf{v}_0$ ↓	MAE $\log E$ ↓	MAE ν ↓	MAE $\log \sigma_y$ ↓
Elastic									
Oracle Midpoint						0.15 [0.14, 0.31]	1.11 [0.93, 1.23]	0.20 [0.15, 0.21]	-
PAC-NeRF	438 [199, 696]	0.74 [0.54, 0.88]	17.4 [16.5, 19.0]	0.19 [0.16, 0.22]	0.92 [0.91, 0.93]	**0.04** [0.03, 0.10]	1.33 [0.43, 2.27]	**0.06** [0.03, 0.26]	-
SpringGaus	693 [592, 814]	0.97 [0.89, 1.05]	16.7 [15.1, 18.4]	0.24 [0.20, 0.28]	0.91 [0.90, 0.92]	0.16 [0.10, 0.25]	-	-	-
GIC	421 [174, 1122]	0.75 [0.47, 1.16]	18.5 [17.7, 19.9]	0.18 [0.16, 0.23]	0.92 [0.91, 0.93]	0.09 [0.05, 0.17]	0.66 [0.30, 1.17]	0.21 [0.17, 0.29]	-
Ours	**24** [11, 77]	**0.26** [0.16, 0.38]	**21.1** [19.3, 22.1]	**0.12** [0.10, 0.15]	**0.93** [0.92, 0.94]	0.09 [0.05, 0.14]	**0.31** [0.19, 0.49]	0.07 [0.02, 0.09]	-
Plasticine									
Oracle Midpoint						0.22 [0.13, 0.23]	0.32 [0.20, 0.65]	0.01 [0.01, 0.04]	0.81 [0.37, 0.86]
PAC-NeRF	247 [135, 433]	0.62 [0.47, 0.75]	19.8 [16.8, 22.9]	0.14 [0.12, 0.20]	0.94 [0.91, 0.96]	0.14 [0.08, 0.20]	2.55 [1.10, 3.39]	0.17 [0.12, 0.24]	0.57 [0.49, 1.06]
SpringGaus	450 [223, 804]	0.67 [0.55, 0.93]	17.3 [16.4, 20.2]	0.20 [0.16, 0.26]	0.92 [0.90, 0.94]	0.13 [0.07, 0.22]	-	-	-
GIC	333 [185, 509]	0.65 [0.53, 0.75]	22.0 [19.9, 24.4]	0.14 [0.12, 0.20]	0.94 [0.92, 0.96]	**0.12** [0.07, 0.27]	0.95 [0.45, 2.06]	0.07 [0.04, 0.27]	0.34 [0.25, 0.65]
Ours	**19** [7, 72]	**0.24** [0.13, 0.43]	**23.4** [22.4, 25.4]	**0.09** [0.07, 0.11]	**0.95** [0.93, 0.96]	0.14 [0.10, 0.23]	**0.40** [0.14, 0.81]	**0.05** [0.02, 0.14]	**0.33** [0.19, 0.50]

F Details on Losses

Optical Flow Loss $\mathcal{L}_{flow}$ The probability-based optical flow loss in the main paper uses a per-pixel uncertainty σ produced by the flow estimator [28]. The full loss with explicit uncertainty is:

$$\begin{aligned}\mathcal{L}_{\text{flow}} = -\frac{1}{|\Omega_t|}\sum_{j\in\Omega_t}\Big[&\log p\Big(\mathbf{f}_{0\to t}(\mathbf{p}_j) \mid \hat{\mathbf{f}}_{0\to t}(\mathbf{p}_j),\, \sigma_{0\to t}(\mathbf{p}_j)\Big)\\ &+\log p\Big(\mathbf{f}_{t\text{-}1\to t}(\mathbf{p}_j) \mid \hat{\mathbf{f}}_{t\text{-}1\to t}(\mathbf{p}_j),\, \sigma_{t\text{-}1\to t}(\mathbf{p}_j)\Big)\Big],\end{aligned} \tag{17}$$

where $\sigma_{i\to j}(\cdot)$ is the per-pixel uncertainty from frame i to frame j at pixel $\mathbf{p}$. The likelihood downweights high-uncertainty pixels, which improves robustness when physical dynamics change visibility between frames.

G Broader Impacts

Recovering physical properties from monocular video can benefit robotic manipulation of deformable objects and the construction of digital twins and virtual environments. However, the same capability could also be misused, for example, to create physically plausible but fake videos, or digital twins of objects and scenes captured without consent. At present, our method's assumptions limit its direct application to in-the-wild captures, which reduces the immediate risk of such misuse. Nevertheless, as monocular inverse physics matures, it should be used responsibly and with caution regarding these applications.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect the paper's contributions, including (i) the MonoPhysics framework for monocular inverse physics and (ii) the proposed visual-physical coupling mechanisms (global scale, adaptive particle volume, and differentiable position map). These claims are supported by the experimental results demonstrating improved performance over prior monocular methods.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It is in Sec. 5.

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results such as theorems or proofs. The differentiable position map (Sec. 3.4) is presented as part of the method design rather than a theoretical result requiring formal assumptions or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides sufficient information to reproduce the main experimental results, including the model formulation and optimization procedure described in Sec. 3.5, as well as detailed implementation and training settings provided in Sec. A.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and data are publicly available at `https://daniel03c1.github.io/MonoPhysics/`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies sufficient training and evaluation details to understand the results, including the optimization procedure in Sec. 3.5 and detailed implementation, dataset, and experimental settings in Sec. A, Sec. D, and Sec. F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Since several metrics (CD in particular) are strongly right-skewed, all quantitative tables report the median with the interquartile range, median [Q1, Q3], instead of mean and standard deviation (Sec. 4.1). The statistics are computed over runs. The reported spread therefore captures variation across scenes, viewpoints, and the resulting initial geometry, rather than across random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: The paper provides sufficient information about the computational resources required for the experiments, including GPU-based training and simulation. Details on the hardware setup and runtime requirements are described in Sec. A, which includes the compute environment used for training and evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics. The work does not involve human subjects, does not collect or use sensitive personal data, and uses publicly available or synthetic datasets for evaluation. No ethical concerns arise from the proposed method or experiments.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss positive applications and potential misuse, such as generated videos or digital twins created without consent, in Sec. G.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, models, and codebases used in this work are properly credited and used in accordance with their respective licenses. We follow the original licenses of the Vid2Sim dataset, the SpringGaus real-world captures, Google Scanned Objects (GSO), and prior inverse physics and 3D reconstruction methods, and cite all sources appropriately in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

Justification: Our synthetic dataset is released at `https://daniel03c1.github.io/MonoPhysics/`. Its construction, including the source objects, constitutive models, and parameter sampling ranges, is documented in Sec. D.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The proposed method does not rely on LLMs for any core methodological contribution or research idea. LLMs were used only to assist with text refinement and coding.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.