ReconPhys: Reconstruct Appearance and Physical Attributes from Single Video

Boyuan Wang[1,2,3], Xiaofeng Wang[4,5], Yongkang Li[4], Zheng Zhu[4✉], Yifan Chang[1,2,3], Angen Ye[1], Guosheng Zhao[1,2,3], Chaojun Ni[4], Guan Huang[4], Yijie Ren[1,3], Yueqi Duan[5], and Xingang Wang[1,3✉]

[1] Institute of Automation, Chinese Academy of Sciences, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[3] Luoyang Institute for Robot and Intelligent Equipment, China
[4] GigaAI, China
[5] Tsinghua University, China

Abstract. Reconstructing non-rigid objects with physical plausibility remains a significant challenge. Existing approaches leverage differentiable rendering for per-scene optimization, recovering geometry and dynamics but requiring expensive tuning or manual annotation, which limits practicality and generalizability. To address this, we propose ReconPhys, the first feedforward framework that jointly learns physical attribute estimation and 3D Gaussian Splatting reconstruction from a single monocular video. Our method employs a dual-branch architecture trained via a self-supervised strategy, eliminating the need for ground-truth physics labels. Given a video sequence, ReconPhys simultaneously infers geometry, appearance, and physical attributes. Experiments on a large-scale synthetic dataset demonstrate superior performance: our method achieves 21.64 PSNR in future prediction compared to 13.27 by state-of-the-art optimization baselines, while reducing Chamfer Distance from 0.349 to 0.004. Crucially, ReconPhys enables fast inference ($<$1 second) versus hours required by existing methods, facilitating rapid generation of simulation-ready assets for robotics and graphics. The code is available at `https://github.com/chuanshuogushi/ReconPhys`.

Keywords: 3D Reconstruction · Dynamic Reconstruction · Physical Modeling

1 Introduction

Non-rigid object reconstruction lies at the intersection of computer vision, graphics, and physics-based simulation, with applications ranging from virtual reality and robotics to digital content creation [1, 4, 11, 17, 21, 22, 31, 36, 43, 45]. Unlike

✉ Corresponding authors. zhengzhu@ieee.org, xingang.wang@ia.ac.cn.

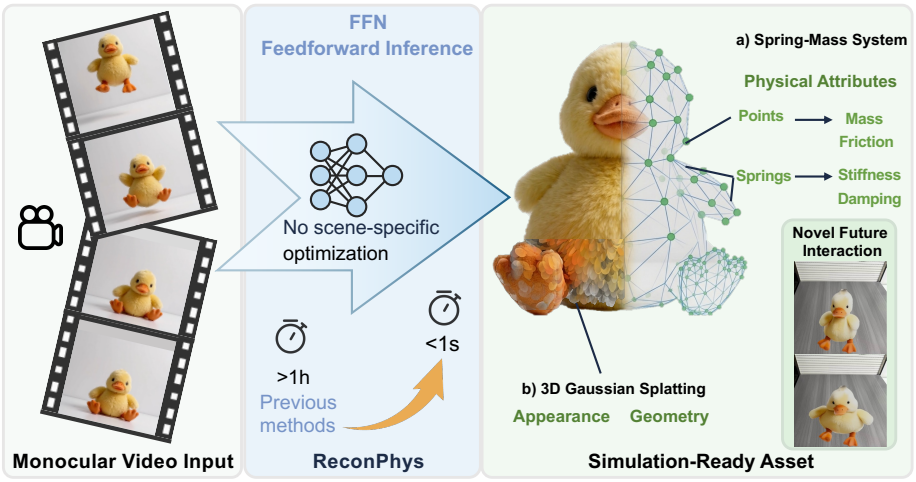

Fig. 1: ReconPhys predicts physical attributes using a feedforward neural network (FFN). This approach obviates the need for scene-specific optimization and requires only a monocular video of the object as input. Within one second, it produces 3DGS assets with binded physical attributes. This capability offers a promising and efficient pathway for generating simulation-ready data for non-rigid manipulation, enabling the rapid transfer of real-world objects into simulation environments without human intervention.

rigid bodies, non-rigid objects undergo complex deformations under external forces or internal dynamics, making their 3D recovery from 2D observations a highly ill-posed and challenging inverse problem. Accurately capturing not only their geometry and appearance but also their underlying physical behavior is crucial for enabling realistic interaction and simulation in downstream tasks. As illustrated in Figure 1, our framework directly addresses this need: by processing a single monocular video, it generates simulation-ready 3DGS assets with bound physical attributes within one second, eliminating the need for per-scene optimization.

Recent advances in dynamic scene reconstruction have been driven by Neural Radiance Fields (NeRF) [24] and, more recently, 3D Gaussian Splatting (3DGS) [13]. Early dynamic NeRF variants, such as D-NeRF [27] and Nerfies [25], model temporal changes via canonical-space deformation or scene flow, achieving impressive view synthesis quality but often incurring high training and rendering costs that limit interactive usage. The emergence of 3DGS has revolutionized this field by offering explicit representations with real-time differentiable rendering. Dynamic 3DGS methods, including deformation-based approaches [22,41] and explicit per-timestep modeling [20,36], can render photorealistic dynamic videos at high speed. However, despite their visual fidelity, these methods primarily optimize geometry and appearance without explicitly recovering physically meaningful parameters. Consequently, their predictions may be

visually plausible yet physically inconsistent under unseen forces or interactions, motivating the integration of physics-aware modeling into dynamic reconstruction.

Despite these advances, current techniques suffer from critical limitations that hinder scalability and generalization. Bridging visual reconstruction with physical understanding is crucial for simulation-ready asset creation. Representative directions couple neural representations with differentiable physics engines, such as PAC-NeRF [17] which integrates Material Point Method (MPM) dynamics but assumes known material families, limiting adaptability to heterogeneous objects. Similarly, PhysGaussian [38] incorporates continuum mechanics into 3DGS but focuses on generative dynamics rather than inferring per-scene attributes from observations. The closest work, Spring-Gaus [45], binds a spring-mass system to 3D Gaussians for elastic objects. However, it relies on dense multi-view inputs and expensive per-scene optimization, making it impractical for general monocular deployment. This reliance on per-scene optimization not only limits practical deployment but also fails to leverage shared physical priors across object categories or materials.

To address these limitations, we propose ReconPhys, which integrates 3DGS with a differentiable spring-mass system for end-to-end inference. Unlike prior approaches that decouple visual reconstruction from physical modeling, our framework enables joint estimation of geometry, appearance, and material properties without manual annotation or per-scene optimization.

Our core insight is to replace scene-specific physics tuning with a feedforward neural estimator trained on diverse dynamic observations. By conditioning physical attributes prediction on reconstructed shape and observed motion dynamics, ReconPhys learns generalizable mappings from visual cues to material properties, allowing zero-shot transfer to unseen objects. The entire pipeline is trained self-supervised via photometric loss, leveraging differentiable rendering and physics simulation to propagate gradients from image-space errors back to physical attributes.

Extensive experiments demonstrate that ReconPhys achieves state-of-the-art performance in both dynamic reconstruction and future-state prediction. Our method not only improves visual fidelity (PSNR, LPIPS) but also yields a dramatic improvement in 3D geometric accuracy, reducing Chamfer Distance (CD) from 0.593 to 0.001 compared to 4DGS [36]. Notably, in future prediction, our approach vastly outperforms Spring-Gaus [45] across all metrics (e.g., 21.64 vs. 13.27 PSNR), confirming that embedding physical understanding directly into the reconstruction process not only enhances visual fidelity but also unlocks practical applications in robotics and simulation where realistic interaction is essential.

Our key contributions are threefold:

(1) We present the first feedforward, general-purpose, and fully differentiable framework that jointly reconstructs non-rigid object geometry, appearance, and physical attributes from a single video, enabling rapid inference without per-scene optimization;

(2) We design a novel dual-branch architecture with a self-supervised physics training strategy and contribute an automated pipeline to synthesize a large-scale dataset of deformable objects with diverse physical attributes;

(3) We demonstrate through extensive experiments that our method achieves state-of-the-art performance in both dynamic reconstruction and future-state prediction. Specifically, we yield a dramatic improvement in 3D geometric accuracy (reducing CD from 0.5932 to 0.001 compared to 4DGS) and vastly outperform Spring-Gaus in future prediction, enabling physically plausible simulation of unseen deformations.

2 Related Work

2.1 Dynamic 3DGS Reconstruction

Dynamic scene reconstruction has rapidly advanced with neural rendering, most notably Neural Radiance Fields (NeRF) [24]. Beyond the canonical static setting, dynamic NeRF variants model time-varying scenes via canonical-space deformation or scene flow, enabling monocular dynamic view synthesis [6, 19, 25–27, 30, 35]. Despite strong visual quality, NeRF-based dynamics are often computationally expensive and can be slow at render time, which limits interactive usage.

Recently, 3D Gaussian Splatting (3DGS) [13] has emerged as an explicit representation that achieves real-time differentiable rendering, inspiring a line of Dynamic 3DGS methods. Existing approaches can be broadly grouped into deformation-based and explicit modeling strategies. Deformation-based methods learn a canonical set of Gaussians and a deformation field to animate the scene, improving stability and temporal coherence [22, 41]. In contrast, explicit approaches directly optimize Gaussian positions and attributes over time, or introduce spatio-temporal parameterizations to better capture non-rigid motion [8, 10, 12, 16, 20, 34, 36, 40, 42, 44]. While these methods can render photorealistic dynamic videos at high speed, they primarily target geometric and appearance fidelity and do not explicitly recover physically meaningful parameters. As a result, their predictions may be visually plausible yet physically inconsistent under unseen forces or interactions, motivating physics-aware modeling for simulation-ready dynamic assets.

2.2 Physics-Aware Reconstruction and Physical Attributes Prediction

Bridging visual reconstruction with physical understanding is crucial for simulation-ready asset creation. A representative direction couples neural representations with differentiable physics engines, enabling gradients from image-space losses to flow back to physical parameters. PAC-NeRF [17] integrates differentiable Material Point Method (MPM) dynamics into a NeRF-like representation for geometry-agnostic system identification, but assumes known material families and typically uses coarse global parameters, limiting adaptability to heterogeneous real objects. PhysGaussian [38] incorporates continuum mechanics into

3DGS to generate physically grounded motion, yet it is primarily designed for generative dynamics rather than directly inferring per-scene physical attributes from monocular observations.

Another line of work focuses on learning-based physical modeling [7, 18, 23, 28, 29, 33]. These methods provide powerful tools for modeling dynamics, but often require either explicit state supervision, expensive per-scene optimization, or specialized simulation data.

The closest to our setting is Spring-Gaus [45], which binds a spring-mass system to 3D Gaussians for reconstructing and simulating elastic objects. However, it requires multi-view capture and relies on per-scene optimization, making it impractical for general monocular deployment and large-scale usage. Closely related is Vid2Sim [2], which also pursues video-based, physics-aware reconstruction of appearance, geometry, and physical properties. While both methods aim for generalizable physical inference from video, ReconPhys differs in three key aspects: (1) it requires only a *single monocular video* rather than multi-view inputs; (2) our pipeline is fully feed-forward with zero per-scene optimization (<1s inference), whereas Vid2Sim employs a two-stage process with scene-specific refinement; and (3) we learn physical attributes through a self-supervised loop via a differentiable spring-mass system bound to 3D Gaussians, contrasting with Vid2Sim's LBS-based reduced-order simulator. In contrast, our approach targets feedforward and monocular physical attribute inference: we combine 3DGS reconstruction with a differentiable dynamics model, and train a neural estimator to predict physical parameters directly from video, removing the need for per-scene optimization or manual physics annotation, enabling zero-shot generalization to unseen objects.

3 Method

3.1 Problem Formulation

Given a monocular video $\mathcal{V}$ capturing an object $\mathcal{O}$ undergoing gravitational drop, collision, and rebound dynamics, we aim to recover its intrinsic physical attributes and 3D visual representation solely from observed motion. The object is modeled as a deformable body governed by a spring-mass system. Let $\{\mathbf{x}_i\}_{i=1}^{N_A}$ denote the N_A mass points, whose pairwise connectivity is determined once via K-nearest neighbors (KNN) on the initial shape and remains fixed thereafter. The physical attributes are defined as $\mathbf{p} = (\{m_i\}, \{k_{ij}\}, \{d_{ij}\}, f)$, where m_i is the mass of point i, $k_{ij} \geq 0$ and $d_{ij} \geq 0$ are the stiffness and damping coefficients of the spring connecting points i and j, and $f \geq 0$ is the global friction coefficient between the object and the ground.

The geometry and appearance of the object are represented by a 3DGS representation comprising N Gaussian kernels. Crucially, every Gaussian kernel $\mathbf{g}_j$ is bound to the spring-mass system, such that its 3D center $\boldsymbol{\mu}_j$ is fully determined by the simulated mass-point states. Each kernel is further characterized by orientation $\boldsymbol{\theta}_j$, RGB color $\mathbf{c}_j$, scale $\boldsymbol{\sigma}_j$, and opacity α_j.

Our feedforward prediction model $\mathcal{M}$ takes $\mathcal{V}$ as input and jointly regresses both components:

$$(\hat{\mathbf{p}}, \hat{\mathbf{g}}) = \mathcal{M}(\mathcal{V}), \tag{1}$$

enabling holistic inference of appearance, geometry, and physics from monocular observations. The initial mass point configuration $\{\mathbf{x}_i^0\}_{i=1}^{N_A}$ is derived from the first frame of $\mathcal{V}$, establishing the reference state for physical simulation.

3.2 Physical Dynamics Model

3D Spring-Mass System. We model elastic object dynamics using a 3D spring-mass system composed of N_A anchor points $\mathcal{A} = \{\mathbf{x}_i\}_{i=1}^{N_A}$, where each anchor $\mathbf{x}_i \in \mathbb{R}^3$ has mass m_i and velocity $\mathbf{v}_i$. Following Spring-Gaus [45], we establish spring connections between each anchor and its K nearest neighbors through K-nearest neighbors (KNN) search:

$$\mathcal{L} = \{l_{i,j}\}_{i=1,j=1}^{N_A,K} = \text{knn}(\mathcal{A}, \mathcal{A}, K), \tag{2}$$

where $l_{i,j}$ denotes the rest length (initial distance) between anchors $\mathbf{x}_i$ and $\mathbf{x}_{i,j}$. Each spring is characterized by stiffness $k_{i,j}$ and damping coefficient $d_{i,j}$.

The net force acting on anchor $\mathbf{x}_i^t$ at timestep t combines spring forces, damping forces, and gravity:

$$\mathbf{F}_i^t = \sum_{j=1}^{K} \mathbf{F}_{i,j}^{k,t} + \sum_{j=1}^{K} \mathbf{F}_{i,j}^{d,t} + m_i \mathbf{g}_{\text{grav}}, \tag{3}$$

where the nonlinear spring force follows a generalized Hooke's law with exponent p_k:

$$\mathbf{F}_{i,j}^{k,t} = -k_{i,j} \left(\|\mathbf{x}_i^t - \mathbf{x}_{i,j}^t\| - l_{i,j} \right)^{p_k} \cdot \frac{\mathbf{x}_i^t - \mathbf{x}_{i,j}^t}{\|\mathbf{x}_i^t - \mathbf{x}_{i,j}^t\|}, \tag{4}$$

and the damping force is:

$$\mathbf{F}_{i,j}^{d,t} = -d_{i,j} \left(\mathbf{v}_i^t - \mathbf{v}_{i,j}^t \right) \cdot \frac{\mathbf{x}_i^t - \mathbf{x}_{i,j}^t}{\|\mathbf{x}_i^t - \mathbf{x}_{i,j}^t\|} \cdot \frac{\mathbf{x}_i^t - \mathbf{x}_{i,j}^t}{\|\mathbf{x}_i^t - \mathbf{x}_{i,j}^t\|}. \tag{5}$$

Numerical Integration. Anchor positions and velocities are updated via semi-implicit Euler integration with timestep Δt:

$$\hat{\mathbf{v}}_i^{t+1} = \mathbf{v}_i^t + \frac{\mathbf{F}_i^t}{m_i} \Delta t, \tag{6}$$

$$\hat{\mathbf{x}}_i^{t+1} = \mathbf{x}_i^t + \hat{\mathbf{v}}_i^{t+1} \Delta t, \tag{7}$$

followed by boundary condition enforcement $\mathcal{B}(\cdot)$ to model ground collisions:

$$(\mathbf{x}_i^{t+1}, \mathbf{v}_i^{t+1}) = \mathcal{B}(\hat{\mathbf{x}}_i^{t+1}, \hat{\mathbf{v}}_i^{t+1}). \tag{8}$$

Binding to 3D Gaussian Splatting. To bridge physical simulation with visual representation, we introduce a two-stage binding mechanism between the spring-mass system and 3DGS:

1) Anchor Sampling: Given N Gaussian centers $\mathcal{X} = \{\boldsymbol{\mu}_i\}_{i=1}^{N}$, we perform volume sampling to generate N_A anchor points ($N_A \ll N$ for efficiency):

$$\mathcal{A} = \mathcal{V}_{\text{vol}}(\mathcal{X}), \tag{9}$$

where $\mathcal{V}_{\text{vol}}(\cdot)$ is a sampling function that distributes anchors throughout the object volume rather than just on the surface, ensuring stable simulation.

2) Position Interpolation: After simulating anchor dynamics to timestep $t+1$, Gaussian centers are updated via Inverse Distance Weighting (IDW) interpolation using their n_b nearest anchors:

$$\boldsymbol{\mu}_i^{t+1} = \frac{\sum_{j=1}^{n_b} \mathbf{x}_{i,j}^{t+1} \cdot (1/r_{i,j}^{p_b})}{\sum_{j=1}^{n_b} (1/r_{i,j}^{p_b})}, \tag{10}$$

where $r_{i,j} = \|\boldsymbol{\mu}_i^0 - \mathbf{x}_{i,j}^0\|$ is the initial distance between Gaussian center $\boldsymbol{\mu}_i$ and anchor $\mathbf{x}_{i,j}$, and $p_b > 0$ controls distance falloff. This binding preserves local deformation patterns while enabling efficient simulation with sparse anchors.

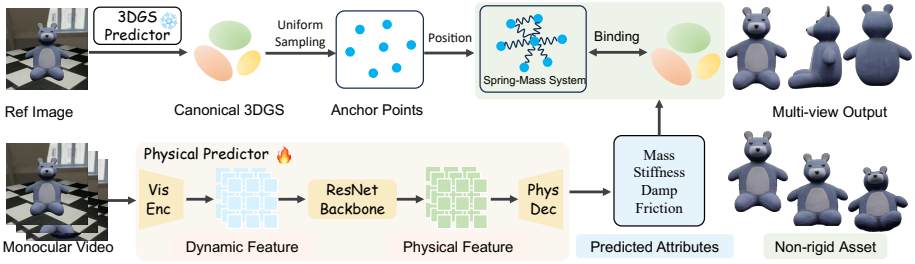

Fig. 2: Overview of our framework. Given an input image, a 3DGS predictor reconstructs the 3D object and samples anchor points to build a spring–mass system. Meanwhile, a physics predictor estimates physical attributes from videos to parameterize the system, producing simulatable 3D objects.

3.3 Model Architecture

As illustrated in Fig. 2, ReconPhys contains two components: a 3DGS predictor that provides a high-fidelity canonical representation for appearance and geometry, and a physical predictor that estimates compact physical attributes of a spring-mass system. Given a monocular video $\mathcal{V} = \{I_t\}_{t=1}^{T}$, the 3DGS predictor outputs a canonical set of Gaussians $\hat{\mathbf{g}}^0$ (including centers, scale, rotation, color, and opacity), while the physical predictor outputs $\hat{\mathbf{p}} = (\hat{m}, \hat{k}, \hat{d}, \hat{f})$ corresponding to mass, stiffness, damping, and friction. For consistency with

$\mathbf{p} = (\{m_i\}, \{k_{ij}\}, \{d_{ij}\}, f)$, we use shared parameters across points/springs, i.e., $\hat{m}_i = \hat{m}$, $\hat{k}_{ij} = \hat{k}$, and $\hat{d}_{ij} = \hat{d}$. Importantly, we directly use the off-the-shelf pretrained weights of the 3DGS predictor and keep it *frozen* throughout training; thus, our learning focuses on identifying physical attributes and their simulation-consistent deformations.

The physical predictor employs an InternViT-based vision encoder to extract dynamic features from each frame of $\mathcal{V}$. These per-frame features are subsequently processed by a ResNet [15] backbone integrated with a self-attention mechanism [32] to aggregate spatio-temporal context, yielding a compact physical feature representation. Finally, a dedicated MLP-based physical decoder maps these features to the estimated physical attributes $\hat{\mathbf{p}}$. These attributes parameterize a differentiable spring-mass simulator that evolves anchor states over time. The simulated anchor motion is then transferred to the Gaussian representation via the binding/interpolation mechanism in Sec. 3.2, yielding a time-varying 3DGS $\hat{\mathbf{g}}^t$ whose Gaussian centers $\boldsymbol{\mu}^t$ follow the simulated deformation. This design establishes a tight coupling *during training*: the video reconstruction error supervises physical attributes through a differentiable simulation–rendering loop. At inference time, the predicted $\hat{\mathbf{p}}$ can be exported as a simulation-ready asset, while $\hat{\mathbf{g}}^0$ serves as a stable canonical visual representation.

3.4 Differentiable Simulation–Rendering Loop

To learn $\hat{\mathbf{p}}$ without ground-truth physical annotations, we build an end-to-end differentiable computation graph that connects pixel-level reconstruction errors to physical attributes as shown in Fig. 3. In the forward pass, the simulator takes $\hat{\mathbf{p}}$ and recursively predicts anchor states $\{\mathbf{x}_i^t, \mathbf{v}_i^t\}$ using differentiable force computation and numerical integration. The anchor displacements are propagated to Gaussian centers through the binding rule, producing deformed centers $\boldsymbol{\mu}^t$ and hence a deformed 3DGS $\hat{\mathbf{g}}^t$. A differentiable 3DGS renderer then generates the predicted frame $\hat{I}_t$ from $\hat{\mathbf{g}}^t$.

In the backward pass, the reconstruction loss provides gradients from $\hat{I}_t$ to $\boldsymbol{\mu}^t$ through the renderer. Since $\boldsymbol{\mu}^t$ is deterministically induced by anchor states via binding, gradients further propagate to $\mathbf{x}_i^t$. Because the simulator is differentiable, gradients traverse the unrolled dynamics and reach the physical parameters $\hat{\mathbf{p}}$, thereby updating the physical predictor. Notably, since the 3DGS predictor is frozen, this gradient pathway is dedicated to identifying physically meaningful attributes that best explain the observed motion, rather than improving the underlying 3DGS reconstruction.

3.5 Self-Supervised Training with Self-Forcing

We train ReconPhys in a self-supervised manner using only monocular videos. Given $\mathcal{V}$, the model predicts $\hat{\mathbf{p}} = \mathcal{M}_p(\mathcal{V})$ and performs an autoregressive simulation rollout to obtain anchor trajectories, which deform the canonical Gaussians and produce rendered frames $\{\hat{I}_t\}$. To reduce the training–inference mismatch,

we adopt Self Forcing [9]: at each step, the simulator advances from the model's own previous predicted state instead of relying on recovered or proxy ground-truth states. This encourages long-horizon stability and improves robustness to accumulated errors.

Autoregressive simulation may lead to unstable gradients when backpropagating through long unrolled trajectories. We therefore apply truncated backpropagation by detaching the input state of each rollout step before computing $\mathbf{x}_{t+1}$. This prevents gradient explosion or vanishing while still allowing supervision signals from rendered frames to update $\hat{\mathbf{p}}$. During rollout, we update Gaussian centers according to the simulated deformation while keeping appearance-related attributes (rotation, color, scale, opacity) fixed, consistent with using a frozen canonical 3DGS. The training objective is a photometric reconstruction loss:

$$\mathcal{L} = \sum_{t=1}^{T} \|I_t - \hat{I}_t\|_2^2, \tag{11}$$

which is fully differentiable with respect to $\hat{\mathbf{p}}$. Minimizing Eq. (11) enables the model to identify physically plausible parameters that reproduce the observed deformation dynamics without any physical labels.

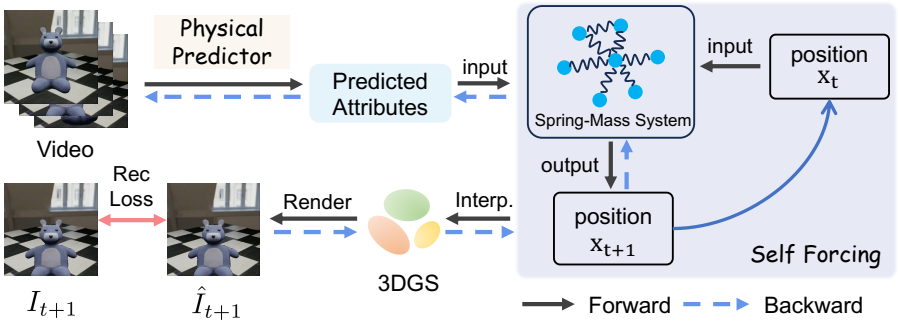

Fig. 3: Training pipeline of ReconPhys with self-forcing. The physics predictor estimates physical parameters from input videos to drive a spring–mass system for motion prediction. The predicted state is rendered and supervised by a reconstruction loss, with a self-forcing feedback loop for temporal prediction.

4 Experiment

4.1 Synthetic Data Pipeline

Learning-based physical reconstruction often suffers from a scarcity of large-scale datasets pairing 3D assets with ground-truth physical attributes. To address this,

we establish an automated pipeline to synthesize a dataset comprising 3DGS assets and their corresponding dynamics.

Semantic Selection and Asset Generation. We source 3D assets from the Objaverse-XL collection [5]. To ensure suitability for non-rigid dynamics, we employ Qwen3-8B [39] to filter candidates based on semantic labels, resulting in 500 eligible objects. For each object, we render four orthogonal views and utilize TRELLIS [37] to reconstruct high-fidelity 3DGS representations, serving as the initial geometric kernels.

Consistent Anchor Sampling. To ensure the physical representation is reproducible and comparable across runs, we enforce consistency in anchor point configuration. Unlike methods that rely on random uniform sampling, we generate the sampling seed via a hash code derived from the object's unique identifier. This guarantees that each object yields an identical set of N_A anchor points, rendering the predicted physical attributes physically interpretable.

Physical Parameterization and Simulation. To facilitate generalization across materials, we bind each 3DGS asset to a spring-mass system and sample physical configurations $\mathbf{p} = (m, k, d, f)$ from continuous distributions: mass $m \in [0.2, 6.0]$, stiffness $k \in [10, 1200.0]$, damping $d \in [0.1, 5.0]$, and friction $f \in [0.0, 1.0]$. This ensures that our dataset covers a continuous distribution of physical behaviors rather than discrete categories. For each $(\mathbf{g}, \mathbf{p})$ pair, we simulate a 30-frame free-fall trajectory under gravity, including ground collision. The resulting motion is rendered into monocular videos $\mathcal{V} = \{I_t\}_{t=0}^{T}$ at 512×512 resolution, forming a dataset of $(\mathcal{V}, \mathbf{g}, \mathbf{p})$ triplets.

4.2 Experimental Setup

We construct our dataset from the 496 synthesized objects, splitting them into 450 for training and 46 for testing to evaluate cross-object generalization. Each training object is instantiated with 10 distinct physical samples, while each test object has 2 physical samples reserved for evaluation. The data is rendered as 30-frame monocular videos at 512×512 resolution from four orthogonal views. Following standard protocols [45], the initial 20 frames are used for dynamic reconstruction, and the subsequent 10 frames serve as ground truth for future prediction.

Our method is designed for *single-view generalization*. During both training and testing, the model inputs a single-view 20-frame sequence. In contrast, baseline methods [36, 45] follow their original protocols, performing per-scene optimization using 4-view 20-frame inputs for each test object independently. We evaluate visual fidelity using PSNR, SSIM, and LPIPS, and assess 3D geometry accuracy via Chamfer Distance (CD) and Earth Mover's Distance (EMD). For physical disentanglement, we additionally report the Mean Absolute Error (MAE) between predicted and ground-truth physical parameters. All metrics for our method are computed on the held-out single view, while baselines are evaluated on the views used for their optimization.

Regarding implementation, we adopt InternViT-300M [3] as the visual backbone for spatio-temporal feature extraction. The model is trained for 100K iter-

ations with a batch size of 8 on 8 RTX 4090 GPUs. For baseline comparisons, we strictly adhere to their official implementations and optimization protocols.

Fig. 4: Visualization of different methods. Our method produces more stable dynamics and realistic future states across different objects, while baseline methods either generate unstable deformations or cannot perform future prediction.

4.3 Cross-Object Generalization

We evaluate the model's generalization capability on unseen geometries. As presented in Table 1 and visually corroborated by Figure 4, our method consistently outperforms baselines on synthesized objects, achieving superior fidelity in both dynamic reconstruction and future prediction. Notably, it attains a PSNR gain of +8.37dB over Spring-Gaus [45] in prediction tasks, demonstrating that joint physical-visual learning facilitates robust extrapolation to novel objects. In contrast to 4DGS [36], which lacks explicit physical priors and cannot extrapolate

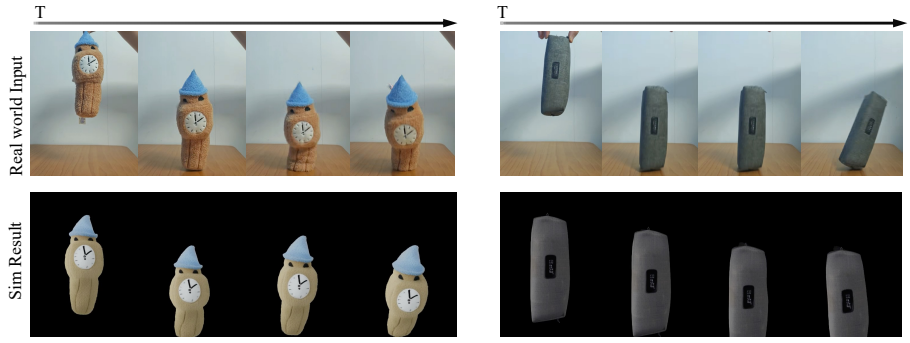

Fig. 5: Visualization of real-world non-rigid assets. Two objects are shown dropping and deforming over time T, with real-world input (top) and corresponding simulation results (bottom).

Table 1: Cross-object generalization on 46 unseen objects. Our method achieves superior future prediction through physics-aware generalization, while maintaining competitive reconstruction quality. Best results in **bold**.

Method	Dynamic Reconstruction					Future Prediction					Time
	PSNR↑	SSIM↑	LPIPS↓	CD↓	EMD↓	PSNR↑	SSIM↑	LPIPS↓	CD↓	EMD↓	
4DGS [36]	30.33	**0.983**	0.0392	0.593	1.5010	-	-	-	-	-	>1h
Spring-Gaus [45]	22.26	0.874	0.1517	0.466	0.582	13.27	0.745	0.2856	0.349	0.424	>1h
Ours	**33.84**	0.953	**0.0366**	**0.001**	**0.007**	**21.64**	**0.907**	**0.0876**	**0.004**	**0.017**	**<1s**

over time, our approach maintains competitive reconstruction quality while enabling predictive simulation. Moreover, it delivers exceptional efficiency, reducing processing time from over 1 hour to under 1 second.

4.4 Real-World Generalization

To assess cross-domain generalization beyond our synthetic training distribution, we evaluate on the real-world dataset provided by Spring-Gaus [45]. Table 2 reports quantitative results on dynamic reconstruction and future prediction. Although metrics slightly decrease compared to our synthetic benchmark, primarily due to domain shift and unseen viewpoints in real-world captures, ReconPhys consistently outperforms Spring-Gaus and maintains robust reconstruction quality. These results confirm the practical feasibility of our feed-forward paradigm under distribution shift and demonstrate that learned spatiotemporal priors transfer reasonably to real monocular observations with novel camera conditions.

Regarding out-of-distribution generalization, the spring-mass dynamics prior constrains our representation capacity for complex physical phenomena beyond the training distribution (e.g., plastic flow or fracture). Nevertheless, our real-world evaluation indicates that ReconPhys preserves reasonable performance on

unseen scenarios with novel viewpoints, suggesting effective generalization across varying observation conditions rather than memorizing synthetic rendering artifacts. Because our self-forcing training paradigm requires no physical ground-truth labels, future performance can be further improved by scaling training on diverse real-world videos. Figure 5 provides additional qualitative evidence that our model captures physics-consistent deformations on real non-rigid assets without per-scene optimization.

Table 2: Real-world quantitative evaluation on the Spring-Gaus benchmark.

Method	Dynamic Reconstruction			Future Prediction		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Spring-Gaus [45]	22.12	0.865	0.148	13.44	0.736	0.278
ReconPhys (Ours)	**28.53**	**0.882**	**0.112**	**19.45**	**0.857**	**0.126**

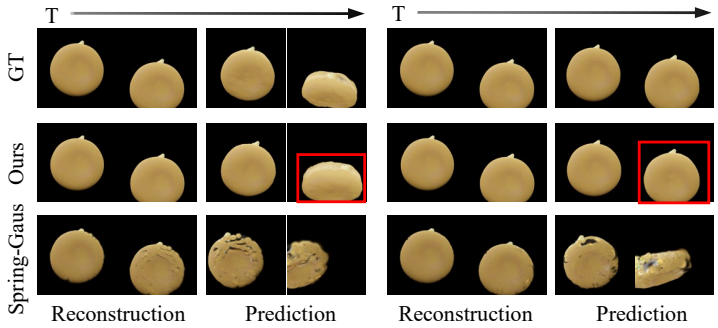

Fig. 6: For the same object with different physical attributes, our method produces more accurate deformation predictions compared to Spring-Gaus.

4.5 Physical Disentanglement

To verify the model's ability to disentangle physical attributes from geometric structure, we evaluate 46 objects, each assigned two distinct physical configurations in our dataset. As summarized in Table 3, our method consistently maintains high reconstruction fidelity while accurately distinguishing between these varying physical states. The quantitative results demonstrate that, despite sharing identical underlying geometry, the model successfully captures and separates the distinct physical properties assigned to each instance, confirming its strong capability in geometry-attribute decoupling.

To quantitatively validate this disentanglement, we further evaluate the accuracy of physical property recovery by computing parameter estimation errors.

Table 3: Physical disentanglement on objects with two distinct attributes sets. Our method preserves geometric fidelity while maximizing trajectory divergence between physical states.

Reconstruction Method		Cylinder-1	Cylinder-2	Headset-1	Headset-2	Hamburger-1	Hamburger-2	Lantern-1	Lantern-2
PSNR↑	Spring-Gaus [45]	20.89	20.77	23.87	21.02	26.003	21.51	20.37	19.09
	Ours	**36.86**	**36.35**	**33.13**	**31.79**	**33.63**	**34.07**	**33.14**	**30.86**
SSIM↑	Spring-Gaus	0.8637	0.8744	0.8993	0.8744	0.9326	0.9104	0.8275	0.8169
	Ours	**0.9833**	**0.9822**	**0.9517**	**0.9435**	**0.9614**	**0.9670**	**0.9473**	**0.9327**
LPIPS↓	Spring-Gaus	0.1880	0.1616	0.1306	0.1424	0.0906	0.1077	0.1840	0.1884
	Ours	**0.0102**	**0.0233**	**0.0297**	**0.0425**	**0.0362**	**0.0339**	**0.0257**	**0.0610**
CD↓	Spring-Gaus	0.4145	0.4845	0.4848	0.4995	0.5324	0.5248	0.4452	0.4609
	Ours	**0.0003**	**0.0005**	**0.0003**	**0.0006**	**0.0003**	**0.0004**	**0.0002**	**0.0027**
EMD↓	Spring-Gaus	0.6171	0.6749	0.6328	0.6652	0.6299	0.6377	0.6308	0.6560
	Ours	**0.0050**	**0.0054**	**0.0032**	**0.0044**	**0.0034**	**0.0154**	**0.0048**	**0.0128**

Table 4: Comparison of physical attributes errors.

Method	Stiffness	Damp	Mass	Friction
Spring-Gaus	827.67	2.546	2.276	**1.082**
Ours	**297.3**	**1.151**	**1.337**	1.508

As detailed in Table 4, our approach not only outperforms Spring-Gaus in reconstruction quality and geometric fidelity but also achieves substantially lower errors in physical parameter estimation. This marked reduction in error confirms that our model successfully decouples visual appearance from underlying physics. Consequently, identical geometries assigned distinct material properties yield accurately differentiated physical predictions, while the visual reconstruction remains consistently high-fidelity.

Figure 6 visualizes this capability, illustrating divergent future trajectories when the same object is simulated with different predicted $\mathbf{p}$. These results validate that our self-supervised training strategy effectively learns to infer physically meaningful attributes from monocular video alone, without requiring explicit physics supervision.

4.6 Application in Robot Non-rigid Object Manipulation

This methodology demonstrates the applicability of video-based non-rigid object reconstruction to robotic deformable object manipulation. As illustrated in Fig. 7, we first capture a video sequence of a freely falling object. The target object is segmented using the Segment Anything Model (SAM) [14], and the segmented video is subsequently processed by ReconPhys to reconstruct both a 3DGS representation and a spring-mass system. Within a virtual simulation environment, configured following the framework of PhysTwin [11], controller points can be interactively manipulated via keyboard inputs to simulate robotic gripper or anthropomorphic hand interactions with the deformable object. This pipeline provides an efficient solution for rapidly acquiring high-fidelity deformable digital assets tailored for photorealistic physical simulation, thereby establishing a

practical pathway toward scalable simulation frameworks for robotic manipulation of non-rigid objects.

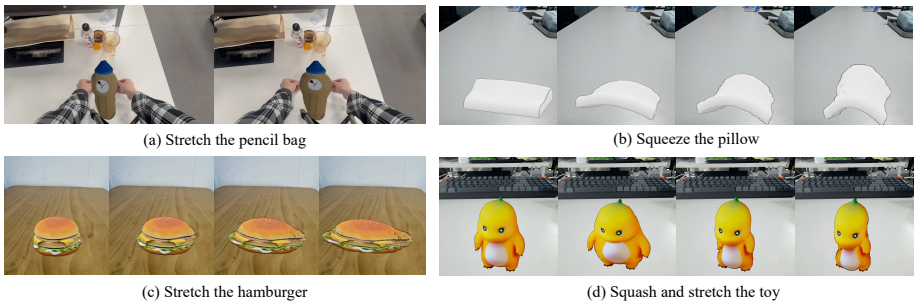

Fig. 7: Our method can be utilized in non-rigid manipulation. Four manipulation scenarios are demonstrated: (a) stretching the pencil bag, (b) squeezing the pillow, (c) stretching the hamburger, and (d) squashing and stretching the toy, showcasing the versatility of our approach across diverse deformable objects.

5 Limitations

Despite promising results, ReconPhys has several inherent limitations. Our physical dynamics use a spring-mass system with fixed connectivity after initial KNN construction, so the framework cannot represent topological changes such as tearing, fracture, or permanent self-contact splitting. The spring-mass representation limits our capacity to model complex physical phenomena beyond the training distribution, and while real-world evaluation demonstrates cross-domain generalization, absolute accuracy remains bounded by sim-to-real gaps in appearance, camera setup, and motion patterns not covered during training.

6 Conclusion

We introduced a self-supervised framework that jointly infers 3D Gaussian Splatting representations and physical attributes of deformable objects from monocular free-fall videos. By establishing end-to-end differentiable pathways between visual reconstruction loss and physical simulation, our dual-branch architecture eliminates the need for ground-truth physics labels while achieving superior cross-object generalization and physical disentanglement. Experiments demonstrate that modeling dynamics through gradient flow to physical attributes acts as an effective regularizer, yielding 8.37dB higher PSNR in future prediction than appearance-only baselines and clear trajectory divergence for objects with identical geometry but different material properties. This work bridges visual perception and physical understanding for dynamic object modeling.

References

1. Božič, A., Palafox, P., Zollhöfer, M., Thies, J., Dai, A., Nießner, M.: Neural deformation graphs for globally-consistent non-rigid reconstruction. CVPR (2021)
2. Chen, C., Dou, Z., Wang, C., Huang, Y., Chen, A., Feng, Q., Gu, J., Liu, L.: Vid2sim: Generalizable, video-based reconstruction of appearance, geometry and physics for mesh-free simulation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26545–26555 (2025)
3. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24185–24198 (2024)
4. Das, D., Wewer, C., Yunus, R., Ilg, E., Lenssen, J.E.: Neural parametric gaussians for monocular non-rigid object reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10715–10725 (2024)
5. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36**, 35799–35813 (2023)
6. Driess, D., Huang, Z., Li, Y., Tedrake, R., Toussaint, M.: Learning multi-object dynamics with compositional neural radiance fields. In: Conference on robot learning. pp. 1755–1768. PMLR (2023)
7. Du, T., Wu, K., Ma, P., Wah, S., Spielberg, A., Rus, D., Matusik, W.: Diffpd: Differentiable projective dynamics. ACM Transactions on Graphics (ToG) **41**(2), 1–21 (2021)
8. Guo, Z., Zhou, W., Li, L., Wang, M., Li, H.: Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. IEEE Transactions on Circuits and Systems for Video Technology **35**(4), 3119–3133 (2024)
9. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
10. Huang, Y.H., Sun, Y.T., Yang, Z., Lyu, X., Cao, Y.P., Qi, X.: Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4220–4230 (2024)
11. Jiang, H., Hsu, H.Y., Zhang, K., Yu, H.N., Wang, S., Li, Y.: Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. arXiv preprint arXiv:2503.17973 (2025)
12. Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H., Gao, F., Yang, Y., et al.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In: ACM SIGGRAPH 2024 conference papers. pp. 1–1 (2024)
13. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
14. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
15. Koonce, B.: Resnet 50. In: Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, pp. 63–72. Springer (2021)

16. Kratimenos, A., Lei, J., Daniilidis, K.: Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In: European Conference on Computer Vision. pp. 252–269. Springer (2024)
17. Li, X., Qiao, Y.L., Chen, P.Y., Jatavallabhula, K.M., Lin, M., Jiang, C., Gan, C.: Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. arXiv preprint arXiv:2303.05512 (2023)
18. Li, X., Hsu, K., Gu, J., Mees, O., Pertsch, K., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al.: Evaluating real-world robot manipulation policies in simulation. In: Conference on Robot Learning. pp. 3705–3728. PMLR (2025)
19. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6498–6508 (2021)
20. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21136–21145 (2024)
21. Lin, Y., Lin, C., Xu, J., Mu, Y.: Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. arXiv preprint arXiv:2501.18982 (2025)
22. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 2024 International Conference on 3D Vision (3DV). pp. 800–809. IEEE (2024)
23. Ma, P., Chen, P.Y., Deng, B., Tenenbaum, J.B., Du, T., Gan, C., Matusik, W.: Learning neural constitutive laws from motion observations for generalizable pde dynamics. In: International Conference on Machine Learning. pp. 23279–23300. PMLR (2023)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
25. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5865–5874 (2021)
26. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. arXiv preprint arXiv:2106.13228 (2021)
27. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10318–10327 (2021)
28. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics **378**, 686–707 (2019)
29. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.: Learning to simulate complex physics with graph networks. In: International conference on machine learning. pp. 8459–8468. PMLR (2020)
30. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12959–12970 (2021)
31. Tretschk, E., Kairanda, N., BR, M., Dabral, R., Kortylewski, A., Egger, B., Habermann, M., Fua, P., Theobalt, C., Golyanik, V.: State of the art in dense monocular non-rigid 3d reconstruction. In: Computer Graphics Forum. vol. 42, pp. 485–520. Wiley Online Library (2023)

32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Wang, B., Meng, X., Wang, X., Zhu, Z., Ye, A., Wang, Y., Yang, Z., Ni, C., Huang, G., Wang, X.: Embodiedreamer: Advancing real2sim2real transfer for policy training via embodied world modeling. arXiv preprint arXiv:2507.05198 (2025)
34. Wang, B., Ouyang, R., Wang, X., Zhu, Z., Zhao, G., Ni, C., Zhang, X., Huang, G., Ren, Y., Liu, L., et al.: Humandreamer-x: Photorealistic single-image human avatars reconstruction via gaussian restoration. arXiv preprint arXiv:2504.03536 (2025)
35. Wang, C., MacDonald, L.E., Jeni, L.A., Lucey, S.: Flow supervision for deformable nerf. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21128–21137 (2023)
36. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20310–20320 (2024)
37. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21469–21480 (2025)
38. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4389–4398 (2024)
39. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
40. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023)
41. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20331–20341 (2024)
42. Yu, H., Julin, J., Milacski, Z.Á., Niinuma, K., Jeni, L.A.: Cogs: Controllable gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21624–21633 (2024)
43. Zhang, M., Zhang, K., Li, Y.: Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. arXiv preprint arXiv:2410.18912 (2024)
44. Zheng, Z., Zhao, X., Zhang, H., Liu, B., Liu, Y.: Avatarrex: Real-time expressive full-body avatars. ACM Transactions on Graphics (TOG) **42**(4), 1–19 (2023)
45. Zhong, L., Yu, H.X., Wu, J., Li, Y.: Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In: European Conference on Computer Vision. pp. 407–423. Springer (2024)