

PhysConvex: Physics-Informed Dynamic Convex Fields for Reconstruction and Simulation

Dan Wang^{1*}, Xinrui Cui^{2*†}, Serge Belongie³, Ravi Ramamoorthi¹

¹ UC San Diego ² University of North Texas ³ University of Copenhagen

*Equal Contribution †Corresponding Author Project Page

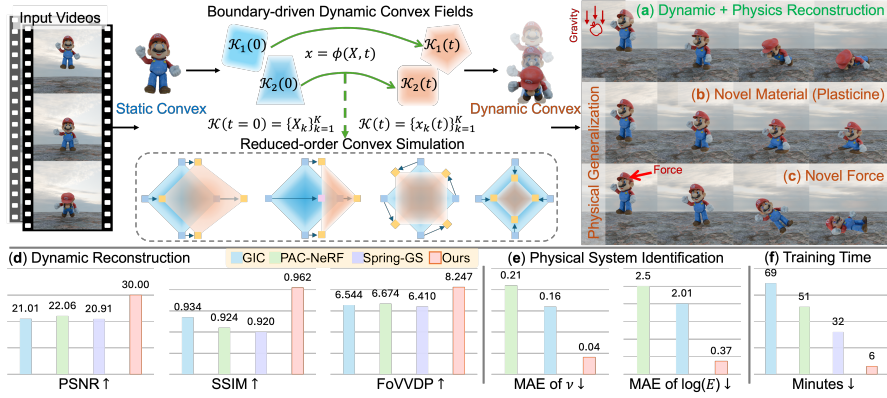


Fig. 1: PhysConvex introduces boundary-driven dynamic convex fields integrated with mesh-free reduced-order convex simulation for dynamic reconstruction, system identification(a), physical generalization(b,c). It recovers appearance, geometry, and physics from videos, improving dynamic and physical reconstruction(d,e), training efficiency(f).

Abstract. Reconstructing deformable objects from video requires a 4D representation that should preserve geometry, explain motion through physics, and generalize to future or new physical conditions. Existing dynamic NeRF and Gaussian methods achieve strong view synthesis, yet their voxel or ellipsoidal primitives are primarily designed for appearance rendering and typically driven by centers or predefined particle bindings, limiting physically meaningful non-uniform deformation and sharp boundary evolution. We present **PhysConvex**, a physics-informed dynamic convex field for video-based reconstruction, physical system identification, and simulation. PhysConvex represents a dynamic object as material-space deformable convex primitives whose boundary is advected by physical dynamics. The boundary-driven convex expresses non-uniform deformation and evolving active supports while serving simultaneously as a rendering element, deformation carrier, and physical support where mass, elastic response, forces, and contacts are evaluated. We further introduce a mesh-free reduced-order convex simulator in which neural skinning modes define physics-based deformation bases directly over deformable convex supports. Experiments show improved dynamic and physical reconstruction, efficiency, future prediction, and generalization to changed materials, forces, and boundary conditions.

Keywords: Dynamic reconstruction · Physical simulation · 4D Convex

1 Introduction

Videos capture not only how objects look, but also how they move, deform, and respond to the physical world. A dynamic 4D representation should preserve appearance, geometry, and physical behavior: it can be rendered from novel views, explained through physical properties, and simulated under new forces, contacts, or material conditions (Fig. 1). This is especially challenging for deformable objects, where appearance, geometry, and physical properties are entangled in video observations. Classical physics-based simulation [2, 15, 28, 35, 47] provides physically plausible dynamics, but typically assumes known geometry, meshing, and material parameters, limiting its real-world applicability where geometries and materials are often unknown. Alternatively, purely neural approaches [5, 17, 30, 31, 39, 42, 44, 45] directly model dynamics through learned networks from data, disregarding underlying physical laws and thereby limiting generalization and interpretability.

Recent advances [9, 23, 29, 43, 48] bridge this gap by integrating neural radiance fields with differentiable physical simulation. These methods make important progress toward video-based physics and dynamic reconstruction, but most inherit a mismatch between rendering and simulation. They still rely on primitives (voxels in NeRF-based methods [29], ellipsoids in Gaussian-based methods [6, 9, 48]) that were designed for static appearance rendering efficiency rather than dynamic physical deformation, resulting in several limitations in dynamic scenarios (Fig. 2): First, in hybrid Eulerian-Lagrangian frameworks [43], center-driven dynamics updates primitives using central particles or pooled neighbors, lacking spatial sensitivity to non-uniform deformation; Second, predefined particle-primitive bindings [7, 48] neglect material and geometric relations, resulting in inaccurate dynamic coupling; Third, existing NeRF and Gaussian representations integrated with grid-based MPM simulation [22] struggle to capture hard or evolving boundaries, hindering physically meaningful surface deformation; Finally, approximating Gaussian decomposition by symmetric, rounded ellipsoids limits the modeling of anisotropic geometric structures or nonlinear materials, while also producing poor spatial coverage near planar or angular regions due to inherent sphere-packing and vacuum-gap issues [11, 18].

Our key insight is that *the rendering primitive should also be the material support for physical deformation and simulation*. Based on this principle, we introduce **PhysConvex**, a physics-informed dynamic convex field for video-based reconstruction, physical system identification, and future-state simulation (Fig. 1). PhysConvex represents a dynamic object as a set of material-space deformable 4D convex primitives. Unlike a Gaussian primitive, which is mainly controlled by a center and covariance, each PhysConvex primitive is defined by a deformable convex hull governed by continuum mechanics to model dynamics.

We introduce a **boundary-driven dynamic convex representation** (Sec. 4.1) that gives each primitive spatially adaptive deformation capacity, rather than forcing it to move as a center-driven kernel (Fig. 2). We represent convex boundary deformation explicitly as vertex dynamics of convex hulls, or implicitly as the surface dynamics of half-space support functions: *Vertex dynamics* advect

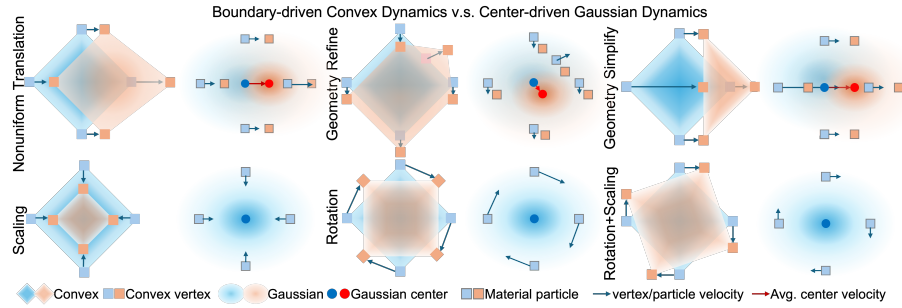


Fig. 2: Center-driven Gaussian dynamics have limited flexibility for non-uniform deformation and evolving boundaries, while boundary-driven convex dynamics enable spatially adaptive vertex advection, explicit surface evolution, topological change.

individual convex vertices under Newtonian motion, providing spatial sensitivity and flexibility to non-uniform deformation; *Surface dynamics* captures evolving edges and boundaries, allowing physically meaningful surface deformation; *Convex dynamics* enable polyhedral structural modification, enhancing geometric expressiveness and adaptability for simulating anisotropic materials.

To integrate convex primitives into physics-based dynamics, we develop a **mesh-free reduced-order convex simulation** (Sec. 4.2). Dynamic convexes are advected under continuum mechanics, using a continuous implicit neural skinning field as physics-informed, shape- and material-aware deformation modes, with time-varying handle transformations as reduced degrees of freedom under Newtonian dynamics [33]. The neural skinning field captures low-dimensional physically plausible deformation modes directly over the material-space convex supports without requiring a mesh, background grid, or manually specified primitive-particle correspondence. PhysConvex therefore learns a physically grounded deformable convex field for dynamics and physics reconstruction via *differentiable rendering and simulation* (Sect. 4.3). Once optimized, the same representation can be rendered from novel views, rolled forward for future prediction, or simulated under changed material, force, and boundary conditions.

In PhysConvex, our dynamic 4D convex is simultaneously a rendering element, a deformation carrier, and a physical support. For rendering, the deformed boundary induces a smooth convex occupancy that can be splatted differentially. For dynamics, the same material-space support allows each primitive to translate, rotate, shear, stretch, and change its active supporting faces over time, and defines where mass, elastic response, forces, and contacts are evaluated. This shared representation avoids predefined bindings between rendering primitives and simulation particles, and provides compact coverage of volumetric regions, flat surfaces, and angular structures. It extends geometric advantages of static convex splatting [19] to deformable, physically grounded dynamic reconstruction.

Our contributions are: **(1)** We propose **PhysConvex**, a dynamic convex field representation for video-based reconstruction and physically grounded simulation of deformable objects. **(2)** We introduce a **boundary-driven dynamic convex representation** whose vertex and surface boundaries, rather than only

its center, are advected by a reduced deformation field. This enables non-uniform deformation, evolving active supports, and sharper geometric structures than center-driven Gaussian or voxel primitives. **(3)** We formulate a **mesh-free reduced-order convex simulation** framework in which neural skinning modes define a physical deformation basis over material-space convex supports. The same convex primitives are used for rendering, deformation, mass, elastic energy, external forces, and contact, avoiding predefined particle-primitive bindings. **(4)** We demonstrate that PhysConvex improves dynamic reconstruction and physical system identification with better training efficiency than physics-integrated NeRF and Gaussian baselines, while supporting efficient future-state prediction and generalization to changed materials, forces, and boundary conditions.

2 Related Work

Physics-Based Simulation. Classical physics simulation relies on mesh-based FEM [1, 4, 12, 35], which requires explicit meshing and linear shape functions, making it inflexible for complex or evolving geometries. Mesh-free methods such as MPM [21, 22, 26, 46] and SPH [14, 27, 34] alleviate meshing but still depend on background grids or neighborhood connectivity, limiting efficiency and stability for highly nonlinear dynamics. Recent efforts, such as Spring-Gaus [48], employ mass-spring systems for elastic behavior, but remain restricted in generalizing to diverse materials. Model Order Reduction approaches [2, 8, 10, 33, 47] offer efficient, mesh-free alternatives by learning reduced-order subspaces that enable accurate, real-time simulation of complex deformations. Inspired by these advances, our method develops a reduced-order convex simulation model that integrates vertex- and surface-based dynamics with differentiable MOR simulation and reconstruction for geometry, appearance, and physical properties.

Video-based Physical Reconstruction and Simulation. Integrating physical simulation with neural 3D reconstruction has gained traction in works [9, 23, 29, 43, 48], integrating mesh-free simulation with NeRF or 3D Gaussian Splatting. However, most existing frameworks rely on neural primitives, such as voxels in NeRF [29] or ellipsoids in 3D Gaussian Splatting [23, 43, 48], combined with physics-based simulation. However, these primitives are primarily optimized for efficient rendering rather than modeling physically consistent deformation, resulting in several key limitations: lack of spatial deformation flexibility within primitives; difficulty modeling sharp or evolving boundaries; restricted shape expressiveness due to fixed-shape Gaussian or voxel kernels; and inaccurate binding between Eulerian rendering primitives and Lagrangian dynamics particles. To address this gap, we introduce PhysConvex, a unified framework for visual rendering and physical simulation. Inspired by 3DCS [19], PhysConvex presents a physical dynamic convex representation to reconstruct and simulate radiance fields governed by continuum mechanics.

3 Preliminaries

Continuum Mechanics. Continuum mechanics describes the motion of deformable objects through a time-dependent deformation map $\mathbf{x} = \phi(\mathbf{X}, t)$, which maps

material coordinates $\mathbf{X}$ in the undeformed reference domain Ω^0 to world coordinates $\mathbf{x}$ in the deformed configuration Ω^t at time t . The evolution of ϕ is governed by the conservation of mass and momentum:

$$\int_{\Omega^t} \rho(\mathbf{x}, t) \equiv \int_{\Omega^0} \rho(\phi^{-1}(\mathbf{x}, t), 0), \quad \rho(\mathbf{x}, t) \dot{\mathbf{v}}(\mathbf{x}, t) = \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \mathbf{f}^{\text{ext}}, \quad (1)$$

where $\rho(\mathbf{x}, t)$ is the density field, $\dot{\mathbf{v}}(\mathbf{x}, t) = \frac{\partial^2 \phi}{\partial t^2}$ is the acceleration, and $\mathbf{f}^{\text{ext}}$ denotes the external force. The internal force is given by the Cauchy stress tensor $\boldsymbol{\sigma}(\mathbf{x}, t)$, which encodes local deformation behavior. The Jacobian of the deformation map, $\mathbf{F}(\mathbf{X}, t) = \nabla_{\mathbf{X}} \phi(\mathbf{X}, t)$, is the deformation gradient, representing local stretch, rotation, and shear, and is central to elasticity. The constitutive model defines the relationship between $\mathbf{F}$, the strain energy density function $\Psi(\mathbf{F})$, and the stress tensor $\boldsymbol{\sigma}$, parameterized by Young’s modulus E and Poisson’s ratio ν .

Model Order Reduction. Model order reduction (MOR) provides a compact, mesh-free alternative with efficient numerical integration and accurate modeling of nonlinear deformation. In MOR, the deformation map is parameterized as $\mathbf{x} = \phi(\mathbf{X}, \mathbf{z}(t))$, where $\mathbf{z}(t)$ is a time-varying low-dimensional vector of reduced degrees of freedom (DOFs). Using implicit time integration, the simulation is formulated as an optimization problem:

$$\mathbf{z}_{t+1} = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{z} - \tilde{\mathbf{z}}_t\|_{\mathbf{M}}^2 + \Delta t^2 \mathbf{E}_{\text{potential}}(\mathbf{z}), \quad (2)$$

where $\mathbf{E}_{\text{potential}}(\mathbf{z})$ denotes the total potential energy of the simulated object from internal and external forces, Δt is the timestep, $\tilde{\mathbf{z}}_t$ is the first-order predictor, and $\|\cdot\|_{\mathbf{M}}^2$ is the mass-matrix-weighted norm.

3D Convex Splatting. 3D Convex Splatting (3DCS) [19] represents radiance fields using smooth convex primitives, offering greater geometric expressiveness than Gaussians and enabling efficient modeling of scenes with sharp edges and dense volumes using fewer elements. A 3D convex is defined as the convex hull of a point set $\mathcal{K} = \{\mathbf{x}_k\}_{k=1}^K$, where the points $\{\mathbf{x}_k\}$ define the hull of the shape, but are not necessarily explicit polyhedral vertices. A convex radiance field consists of N convex polytopes $\mathcal{C} = \{\mathcal{K}_n, \mathbf{c}_n, \sigma_n, \alpha_n, \beta_n\}_{n=1}^N$, where $\mathbf{c}_n$ and σ_n denote color and opacity, and α_n and β_n control smoothness and sharpness, respectively. Here, α governs the softness of vertices and edges, and β modulates the radiance transition from diffuse to dense regions.

4 Method

Given posed videos of a deformable object, our goal is to reconstruct a dynamic 4D representation that can be rendered from novel views, explained by physical parameters, and simulated beyond the observed sequence. PhysConvex is built on a design principle: the rendering primitive should also be the material support for deformation and simulation. Instead of attaching visual primitives to external particles or moving each primitive only by its center, we represent the dynamic object as material-space deformable convex primitives whose boundaries are advected by a reduced physical deformation model.

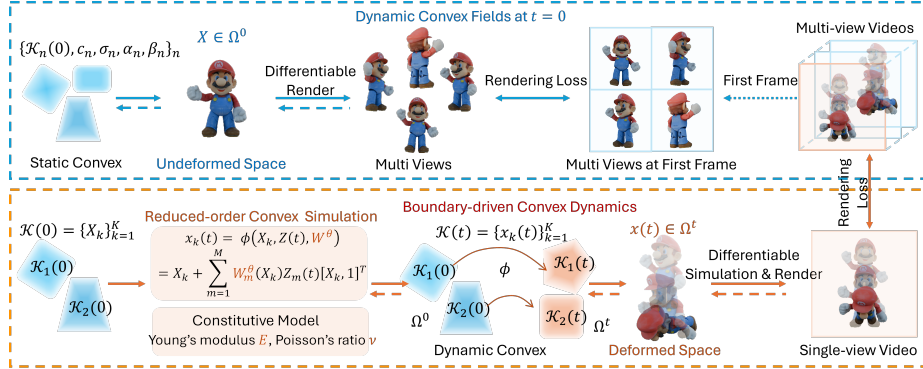


Fig. 3: PhysConvex proposes differentiable boundary-driven dynamic convex field (Sec. 4.1) and mesh-free reduced-order convex simulation (Sec. 4.2), jointly reconstructing geometry, appearance, and physical properties of dynamic objects (Sec. 4.3).

Method Overview. PhysConvex introduces boundary-driven dynamic convex fields (Sec. 4.1) integrated with physical reduced-order convex simulation (Sec. 4.2) to achieve differentiable physically-consistent 3D reconstruction and dynamic modeling (Sec. 4.3). PhysConvex is trained by enforcing rendered pixels to match the observed video in two stages (Fig. 3): (1) An undeformed convex field is reconstructed over the first multi-view frame; (2) The reduced-order convex simulation advects our boundary-driven dynamic convex field under continuum mechanics, and physical properties can be optimized through joint differential simulation and rendering using single-view video supervision.

4.1 Boundary-driven Dynamic Convex Fields

A dynamic object can be approximated as a collection of time-varying convex polytopes, forming a piecewise representation of dynamic geometry. By extending with an additional time variable $t \in \mathbb{R}^+$, we parameterize our dynamic convex field as a set of time-dependent N convex polytopes:

$$\mathcal{C}(t) = \{\mathcal{K}_n(t), \mathbf{c}_n, \sigma_n, \alpha_n, \beta_n\}_{n=1}^N, \quad (3)$$

where $\mathcal{K}_n(t)$ denotes the n -th dynamic convex at time t , and $\{\mathbf{c}_n, \sigma_n, \alpha_n, \beta_n\}$ are its time-invariant color, opacity, smoothness, and sharpness parameters.

We represent boundary-driven convex deformation explicitly as vertex dynamics of convex hulls, or implicitly as surface dynamics of half-space support functions, enabling spatially adaptive vertex deformation, physically meaningful surface evolution, and geometry modification through polyhedral structures (Fig. 2). Explicit vertex dynamics naturally support physical simulation, while implicit surface dynamics enable differentiable rendering.

4.2 Physical Reduced-order Convex Simulation

Convex Vertex Dynamics. A dynamic convex is defined as a 3D time-varying point set $\mathcal{K}(t) = \{\mathbf{x}_k(t)\}_{k=1}^K$ whose convex hull represents the primitive shape. Here, K is an upper bound on shape capacity rather than a fixed number of active

hull vertices. During simulation, these points evolve under physical dynamics, allowing each convex primitive to adapt its shape, anisotropy, and boundary structure, thereby supporting spatially flexible, non-uniform deformation. Even with a constant K , the effective convex geometry can adapt through changes in active hull supports, enabling implicit refinement or simplification (Fig. 2.a-2,3). The undeformed convex hull is denoted as $\mathcal{K}(t=0) = \{\mathbf{X}_k\}_{k=1}^K$.

To advect dynamic convex fields, inspired by reduced-order simulation [33], we model the vertex dynamics $\mathbf{x}_k(t)$ using spatially varying neural skinning eigenmodes W^θ as reduced deformation bases,

$$\phi(\mathbf{X}_k, \mathbf{Z}(t), W^\theta) = \mathbf{X}_k + \sum_{m=1}^M W_m^\theta(\mathbf{X}_k) \mathbf{Z}_m(t) [\mathbf{X}_k, 1]^T, \quad (4)$$

where skinning eigenmodes $W^\theta(\mathbf{X}) \in \mathbb{R}^3 \rightarrow \mathbb{R}^M$ represent physics-informed deformation bases encoding both shape and material awareness, and M is the number of reduced DOFs. The reduced bases $W^\theta(\mathbf{X})$ are represented as a small MLP-based neural field parameterized by θ , enabling smooth, mesh-free discretization for complex geometries and heterogeneous materials. The neural skinning field $W^\theta(\mathbf{X})$ is trained to learn physically significant modes by minimizing:

$$\min_{\theta} \lambda_{\text{ela}} \int_{\mathbb{R}^3} \varphi(\mathbf{X}) \Psi(\phi(\mathbf{X}, \mathbf{Z}, W^\theta)) d\mathbf{X} + \lambda_{\text{ort}} \sum_{i=1}^M \sum_{j=1}^M \int_{\mathbb{R}^3} \varphi(\mathbf{X}) (W_i^\theta W_j^\theta - \delta_{ij})^2 d\mathbf{X}, \quad (5)$$

where $\varphi(\mathbf{X})$ is a density function, $\Psi(\cdot)$ is the strain energy density function, and δ_{ij} is the Kronecker delta. $W^\theta(\mathbf{X})$ is optimized in a data-free manner via elastic energy minimization (the first term in Eq. 5) and orthogonality regularization (the second term) with randomly sampled $(\mathbf{X}, \mathbf{Z})$ pairs. Physically implausible deformations incur high elastic energy, while orthogonality constraint prevents collapse into trivial rigid modes. These objectives encourage physically plausible, low-energy deformation modes rather than memorization of object-specific trajectories. It enables $W^\theta(\mathbf{X})$ to encode low-dimensional, physically consistent deformation behaviors that are material- and shape-aware, thereby achieving mesh-free model reduction without requiring explicit geometric scaffolding.

During the simulation, $\mathbf{Z}(t) \in \mathbb{R}^{3 \times 4 \times M}$ represents time-varying reduced DOFs as handle transformation matrices, which are flattened into $\mathbf{z}(t) = \text{flat}(\mathbf{Z}(t)) \in \mathbb{R}^{12 \times M}$. $\mathbf{Z}(t)$ is initialized to zero to ensure that $\mathbf{x}_k(0) = \mathbf{X}_k$. At each discrete time step, $\mathbf{Z}(t)$ is updated via implicit time integration using the incremental potential formulation (Eq. 2 with an inertia term and a potential energy term) and solved through a standard Newton-based solver [3, 28]. This process advances the dynamic convex field under physically consistent reduced-order dynamics.

Convex Surface Dynamics. Given a dynamic convex point set, the convex hull can equivalently be expressed by H time-varying planes $\mathcal{H}(t) = \{(\mathbf{n}_h(t), d_h(t))\}_{h=1}^H$, where $\mathbf{n}_h(t)$ and $d_h(t)$ denote the normal and offset of the h -th plane, respectively. The signed distance of a point $\mathbf{x}$ to the h -th time-dependent plane is:

$$f_h(\mathbf{x}, t) = \mathbf{n}_h(t) \cdot \mathbf{x} + d_h(t). \quad (6)$$

Following the static indicator formulation in CvxNet [13], we define $\mathcal{O}(\mathbf{x}, t) : \mathbb{R}^3 \rightarrow [0, 1]$ as the time-dependent occupancy function for each dynamic convex:

$$\mathcal{O}(\mathbf{x}, t) = \text{Sigmoid}(-\beta \tau(\mathbf{x}, t)), \quad \tau(\mathbf{x}, t) = \log \left(\sum_{h=1}^H \exp(\alpha f_h(\mathbf{x}, t)) \right), \quad (7)$$

where α and β control convex smoothness and sharpness, respectively, and $\tau(\mathbf{x}, t)$ approximates the signed distance function of a dynamic convex hull.

To render deformed convex kernels, we extend the splatting from 3DCS [19] to handle dynamic scenes. PhysConvex projects 3D time-varying points to 2D, constructs their dynamic convex hulls, and rasterizes them using α -blending. Given camera pose and N dynamic convexes, the time-varying pixel color at location $\mathbf{q}$ and time t is:

$$\mathbf{C}(\mathbf{q}, t) = \sum_{i=1}^N \mathbf{c}_i \sigma_i \mathcal{O}_i(\mathbf{q}, t) \prod_{j=1}^{i-1} (1 - \sigma_j \mathcal{O}_j(\mathbf{q}, t)), \quad (8)$$

where $\mathbf{c}_i$ and σ_i denote the color and opacity of the i -th dynamic convex.

4.3 Differentiable Reconstruction and Simulation

Given posed multi-view videos of a dynamic scene, we reconstruct the dynamic object’s appearance, geometry, and physical properties through a joint differentiable reconstruction and simulation, as shown in Fig. 3.

Appearance and Geometry Optimization. To recover the undeformed convex field $\mathcal{C}(0)$, we optimize from the first multi-view frames of the input videos by minimizing:

$$\mathcal{C}^*(0) = \arg \min_{\{\mathcal{K}_n(0), \mathbf{c}_n, \sigma_n, \alpha_n, \beta_n\}} (1 - \lambda) \mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}} + \beta \mathcal{L}_m, \quad (9)$$

where $\mathcal{L}_1$ and $\mathcal{L}_{\text{D-SSIM}}$ are image reconstruction losses, and $\mathcal{L}_m$ regularizes the number of convexes following [19]. This optimization yields the undeformed convex field $\mathcal{C}^*(0) = \{\mathcal{K}_n^*(0), \mathbf{c}_n^*, \sigma_n^*, \alpha_n^*, \beta_n^*\}_{n=1}^N$ in material space Ω^0 .

Physical Model Optimization. To recover physical properties, we optimize the physical parameters $\{E, \nu\}$ with video supervision:

$$E^*, \nu^* = \arg \min_{E, \nu} \frac{1}{SP} \sum_{p=1}^P \sum_{s=1}^S \|\mathbf{C}(\mathbf{q}_p, t_s) - \mathbf{C}_{\text{gt}}(\mathbf{q}_p, t_s)\|_2^2, \quad (10)$$

where $\mathbf{C}$ and $\mathbf{C}_{\text{gt}}$ denote the rendered and ground-truth pixel colors at each time step. Since both simulation and rendering are differentiable, physical and material parameters can be optimized end-to-end from video observations.

To equip our model with generalizable physical understanding, we adopt a feed-forward physical system identification framework inspired by [9]. Given a single front-view video, a pretrained video Vision Transformer [40] extracts

motion cues and physical knowledge prior from large-scale video data. Small MLP regression heads then estimate $\{E, \nu, \theta\}$, providing an initial coarse prediction that is subsequently refined through differentiable simulation to better match reference videos. In summary, PhysConvex jointly reconstructs geometry, appearance, and physical properties of deformable objects by unifying physical boundary-driven dynamic convex representation, differentiable reduced-order convex simulation, and rendering.

5 Results

In this section, we evaluate PhysConvex on four tasks: physical system identification (Sec. 5.1), dynamic reconstruction (Sec. 5.2), physical generalization capability (Sec. 5.3), future state prediction (Sec. 5.4). We also provide efficiency comparison (Sec. 5.5), ablation study (Sec. 5.6), real-world evaluation (Sec. 5.7).

Google Scanned Object Dataset. Following [9], we use 12 high-quality 3D meshes from Google Scanned Objects (GSO) [16], featuring complex geometries and detailed textures. Objects are simulated using FEM to obtain physically accurate reference dynamics. The animations are rendered from 12 viewpoints at a resolution of 448×448 for 24 frames. The first 16 frames are used as observations, and the remaining 8 frames serve as ground-truth references for future state prediction. Following [9], Young’s modulus and Poisson’s ratio are randomly sampled from $E \in [10^4, 10^6]$, $\nu \in [0.2, 0.5]$.

Real-world Dataset. We evaluate PhysConvex on real-world data following SpringGaus [48]. Real-world dataset includes both static scenes and dynamic multi-view videos. Static scenes: Each object is placed on a table and captured with 50–70 multi-view images from the upper hemisphere. Camera poses are recovered using COLMAP [36, 37]. Dynamic multi-view videos: Synchronized RGB videos are captured from three viewpoints at a resolution of 1920×1080 . Intrinsic and extrinsic parameters are calibrated using a checkerboard. Object masks for all scenes are obtained using the Segment Anything Model (SAM) [25].

Baselines. (1) *PAC-NeRF* [29] jointly reconstructed dynamic NeRFs and a simulatable model using MPM [22]. (2) *Spring-GS* [48] models elastic objects with spring-mass 3D Gaussians and learnable spring stiffness parameters, limited to elastic dynamics. (3) *GIC* [6] integrates 3D Gaussians with MPM simulation, enhancing shape reconstruction via a motion-factorized network and coarse-to-fine filling strategy. (4) *Vid2Sim* [9] is a feed-forward Gaussian-based model with reduced-order simulation by utilizing pre-trained VideoMAE [40] and Large Multi-view Gaussian Model (LGM) [38] as backbones for reconstructing physical parameters and 3D Gaussians (opacity, RGB, μ , Σ).

Metrics. For *physical parameters estimation*, we report Mean Absolute Error (MAE) of $\log(E)$ and ν under Neo-Hookean model (excluding Spring-GS). For *dynamic reconstruction and future prediction*, we measure PSNR [20], SSIM [41], and video perceptual loss (FoVVD) [32].

Implementation details. Each convex shape is initialized with $K = 6$ points and spherical harmonics of degree 3 (69 parameters per convex shape). We first train 3D convexes in the undeformed space, and then train the neural skinning

network using the data-free method. We then jointly optimize E, ν, θ for 400 iterations with learning rates $5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-7}$ using the Adam optimizer [24]. Simulation uses $M = 10$ control handles as reduced DOFs. The average training time cost per scene is about 6 min on a single Nvidia RTX A6000 (Details in Sec. 5.5). Additional results for varying K and M appear in Sec. 5.6. More details can be found in the Supplementary.

5.1 Physical System Identification

Following [6, 9, 29], we assessed our method and all baselines on physical system identification against 12 GSO test cases. As shown in Tab. 1, PhysConvex achieves the lowest error in most cases and remains competitive in the rest, validating its accuracy in recovering physical properties.

Table 1: Comparison of Mean Absolute Error on Physical System Identification.

	methods	backpack	bell	blocks	bus	cream	elephant	grandpa	leather	lion	mario	sofa	turtle	Mean Error ↓
$\log(E)$	PAC-NeRF	3.28	1.08	4.02	3.30	3.22	3.05	2.99	1.20	2.34	3.37	0.20	1.94	2.50
	Vid2Sim	0.69	0.15	0.54	0.26	0.95	0.18	1.07	0.73	0.48	0.50	0.18	0.44	0.51
	GIC	1.16	2.87	2.12	1.93	2.13	1.53	0.42	3.45	2.85	1.82	0.65	3.18	2.01
	Ours	0.32	0.34	0.10	0.28	0.34	0.60	0.97	0.18	0.23	0.54	0.01	0.56	0.37
ν	PAC-NeRF	0.21	0.23	0.33	0.16	0.12	0.06	0.36	0.26	0.14	0.33	0.30	0.01	0.21
	Vid2Sim	0.10	0.10	0.07	0.06	0.11	0.05	0.02	0.06	0.05	0.06	0.08	0.02	0.06
	GIC	0.11	0.24	0.29	0.18	0.08	0.24	0.26	0.02	0.14	0.22	0.01	0.08	0.16
	Ours	0.06	0.09	0.02	0.02	0.05	0.08	0.03	0.02	0.03	0.12	0.02	0.01	0.04



Fig. 4: Comparison with baseline [6, 9, 29, 48] on dynamic reconstruction and future state prediction. Our method preserves high-quality geometry and appearance while producing physically plausible dynamics.

5.2 Dynamic Reconstruction

Our method and baselines for dynamic reconstruction were evaluated using a set of 12 GSO test cases following [6, 9, 29, 48]. For fair comparison, we use Vid2Sim’s variant that trains 3D Gaussians from scratch without initializing 3D Gaussians from pre-trained Large Multi-view Gaussian Model (LGM) [38] predictions. As shown in Tab. 2 and Fig. 4, PhysConvex achieves superior reconstruction quality and physical realism. Prior methods that optimize voxels in NeRF or ellipsoids in 3DGS often produce blurred textures or inaccurate dynamics, whereas our reduced-order, boundary-driven convex fields preserve sharp details and dynamic fidelity. Moreover, unlike grid-based MPM or spring-mass elastic dynamics, which are susceptible to numerical artifacts and oscillations in large-scale elastodynamics of complex geometries, our mesh- and grid-free reduced-space solver yields efficient, accurate dynamics.

Table 2: Quantitative Comparison on Dynamic Reconstruction (Best).

		backpack	bell	blocks	bus	cream	elephant	grandpa	leather	lion	mario	sofa	turtle	Mean $\uparrow$
FoVVD $\uparrow$	PAC-NeRF	6.043	7.473	7.001	6.540	5.991	6.791	6.626	6.485	7.006	6.876	6.543	6.711	6.674
	Spring-Gaus	5.455	6.862	6.890	6.377	5.899	6.524	6.998	5.988	6.902	6.153	6.300	6.569	6.410
	Vid2Sim w/o LGM	7.778	6.884	8.792	7.622	9.117	7.468	6.246	8.949	7.681	6.325	8.034	8.628	7.794
	GIC	6.130	6.230	6.062	6.552	5.889	6.907	6.855	6.737	6.331	6.985	7.069	6.782	6.544
	Ours	8.403	7.843	8.783	8.005	8.708	8.178	7.752	9.153	7.617	7.792	7.925	8.801	8.247
PSNR $\uparrow$	PAC-NeRF	19.37	25.00	23.36	20.72	23.24	22.27	21.63	20.85	22.66	21.01	22.49	22.19	22.06
	Spring-Gaus	17.42	20.49	22.78	20.06	23.58	21.30	21.64	18.29	21.95	20.23	21.89	21.23	20.91
	Vid2Sim w/o LGM	26.87	26.59	32.57	26.53	34.82	26.99	24.45	31.21	27.62	24.83	30.01	30.33	28.57
	GIC	18.94	19.55	18.78	20.84	23.81	21.86	20.50	21.00	19.33	21.28	24.16	22.09	21.01
	Ours	28.09	29.67	32.65	27.97	34.62	29.17	27.67	33.50	27.58	27.46	30.15	31.49	30.00
SSIM $\uparrow$	PAC-NeRF	0.887	0.956	0.940	0.908	0.893	0.922	0.939	0.932	0.936	0.921	0.926	0.923	0.924
	Spring-Gaus	0.867	0.941	0.941	0.903	0.912	0.919	0.948	0.917	0.937	0.920	0.921	0.920	0.920
	Vid2Sim w/o LGM	0.931	0.959	0.973	0.937	0.955	0.942	0.944	0.973	0.954	0.936	0.952	0.963	0.952
	GIC	0.903	0.945	0.930	0.925	0.922	0.936	0.948	0.949	0.934	0.938	0.942	0.936	0.934
	Ours	0.950	0.972	0.975	0.948	0.960	0.957	0.964	0.980	0.958	0.954	0.955	0.971	0.962

5.3 Generalization Capability

We further demonstrate the generalization of our method to novel materials, forces, and boundary conditions, showing consistent performance.

Generalization to Different Materials. PhysConvex generalizes naturally to different material behaviors and their constitutive models. We demonstrate this by simulating *Elastic*, *Plasticine*, and *Sand* materials following [6, 29]. In Fig. 5, our method captures the distinct material responses by utilizing corresponding energy densities derived for each model (see Supplementary).

Generalization to Novel Boundary and Force Conditions. PhysConvex is robust to varying environmental dynamics, seamlessly integrating into dynamic scenes with novel forces and boundary interactions. As shown in Fig. 6, our method produces stable, high-quality reconstructions and physically consistent animations under diverse constraints.



Fig. 5: Generalization to novel materials. Elastic (stiff): $E=10^7$, $\nu=0.49$. Elastic (soft): $E = 8000$, $\nu = 0.4$. Plasticine: $\tau_Y=500$. Sand: $\theta_f=10^\circ$.

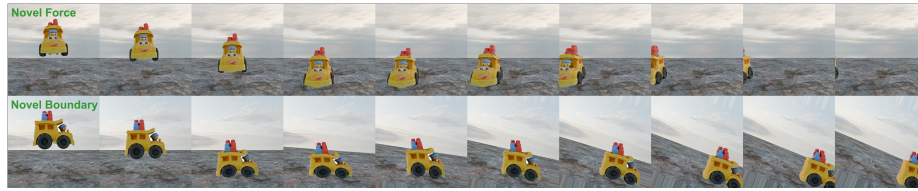


Fig. 6: Generalization to novel force and complex boundary conditions. Force: A continuous external force pushing the falling object toward the left. Boundary: An object falls and is subjected to sliding motion on a continuously moving floor.

5.4 Future State Prediction

We further evaluate temporal generalization following [48]. After reconstructing the first 16 frames, all methods predict the subsequent 8 frames. Average results across all test cases (Tab. 3 and Fig. 4) show that PhysConvex maintains the top accuracy, demonstrating improved physical consistency in temporal evolution.

Table 3: Future State Prediction.

	PAC-NeRF	Spring-Gaus	GIC	Vid2Sim	Ours
PSNR $\uparrow$	20.11	18.32	19.20	25.07	25.71
SSIM $\uparrow$	0.913	0.905	0.916	0.945	0.948
FoVVD $\uparrow$	5.948	5.443	5.702	7.770	7.683

Table 4: Training & Primitive Efficiency.

	PAC-NeRF	GIC	Spring-GS	Vid2Sim	Ours
Avg. Training Time (min)	51	69	32	8	6
Primitive Num.	-	-	22882	31363	23681

5.5 Efficiency Comparison

We report the average total training time and the average number of primitives required for different methods using their default settings (excluding the primitives from network-based models, PAC-NeRF and GIC). As shown in Tab. 4, our method is notably more efficient, achieving shorter training time and requiring fewer primitives compared to even highly efficient baselines like Spring-Gaus and Vid2Sim. This superior efficiency stems from the compact representation achieved by our convex primitives and the inherent performance gains of our reduced-order simulation combined with an implicit Euler solver. All tests were executed on a single NVIDIA-RTX A6000 GPU.

Table 5: Ablation of Our Dynamic Convex Field with Boundary-driven Kinematics via Model Order Reduction Simulation (MOR). (Best , Second , Third)

Model	3D Primitive		Kinematics		Simulation	Dynamic Reconstruction			Physics Identification	
	Gaussian	Convex	Center	Boundary	MOR	PSNR $\uparrow$	SSIM $\uparrow$	FoVVD $\uparrow$	$\log(E)$ $\downarrow$	ν $\downarrow$
A	✓		✓		✓	28.65	0.954	7.895	0.54	0.07
B		✓	✓		✓	29.16	0.958	8.173	0.57	0.05
C		✓		✓	✓	30.00	0.962	8.247	0.37	0.04

Table 6: Ablation of K of Convex Points ($M = 10$). (Best , Second , Third).

	Dynamic Reconstruction			Future State Prediction			Physics Identification		Primitives	Training Time
K	PSNR $\uparrow$	SSIM $\uparrow$	FoVVD $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	FoVVD $\uparrow$	$\log(E)$	ν	Num.	Average
K=3	24.90	0.939	6.957	19.63	0.919	6.139	0.74	0.05	11827	5min 20s
K=4	29.56	0.960	8.151	24.50	0.944	7.500	0.38	0.04	14405	5min 36s
K=5	29.88	0.961	8.235	25.09	0.946	7.688	0.36	0.04	18773	6min 5s
K=6	30.00	0.962	8.247	25.71	0.948	7.683	0.37	0.04	23681	6min 14s
K=7	30.01	0.962	8.235	24.95	0.945	7.664	0.38	0.05	29646	6min 39s
K=8	30.19	0.962	8.290	25.38	0.948	7.760	0.47	0.04	35320	7min 5s

Table 7: Ablation of the Number M of Reduced DOFs. (Best , Second , Third)

		Dynamic Reconstruction			Future State Prediction			Physics Identification		Primitives
K	M	PSNR $\uparrow$	SSIM $\uparrow$	FoVVD $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	FoVVD $\uparrow$	$\log(E)$	ν	Num.
K=4	M=8	29.47	0.959	8.152	24.55	0.944	7.605	0.351	0.053	14392
	M=10	29.56	0.960	8.151	24.50	0.944	7.500	0.376	0.044	14405
	M=12	29.60	0.960	8.206	24.47	0.943	7.558	0.510	0.043	14180
K=6	M=8	29.96	0.962	8.214	25.05	0.946	7.635	0.330	0.048	23832
	M=10	30.00	0.962	8.247	25.71	0.948	7.683	0.371	0.044	23681
	M=12	29.88	0.961	8.210	24.76	0.944	7.601	0.321	0.048	23736

5.6 Ablation Study

Boundary-driven Kinematics Ablation. We ablate the proposed boundary-driven kinematics compared with a center-driven convex variant. In Tab. 5, boundary-driven convex dynamics (Full Model **C**) consistently outperform center-driven convex dynamics (Model **B**), improving dynamic reconstruction and reducing physical parameter errors. These results demonstrate that explicitly advecting convex boundaries, rather than only moving primitive centers, captures non-uniform deformation and yields more accurate physics reconstruction.

Convex Representation Ablation. To isolate the effect of the convex representation from the MOR simulation framework, we compare Gaussian and convex primitives under the same reduced-order simulation setting in Tab. 5. Comparing the center-driven Gaussian variant (Model **A**) with the center-driven convex variant (Model **B**) shows that replacing Gaussians with convex primitives improves dynamic reconstruction and reduces the Poisson ratio error. This validates that convex primitives provide better volumetric coverage and geometric expressiveness than Gaussian primitives under the same reduced-order

simulation. With the full boundary-driven convex formulation (Model **C**), the performance further improves, confirming the effectiveness of our dynamic convex field for both geometry and physics identification.

DOFs M and Convex Hull K Ablation. Ablation studies validate two key components: the number of 3D points K per convex hull in our boundary-driven convex dynamics (Tab. 6) and the number of degrees of freedom (skinning handles) M in our reduced-order convex simulation (Tab. 7). Increasing K enhances the flexibility of convex topology representation and improves reconstruction quality, though it introduces higher computational cost and longer training time. However, the performance gain saturates beyond $K = 6$. Similarly, increasing M boosts deformation expressivity in simulation but is bounded by the capacity of the neural skinning network. Note that both K and M can be adjusted to accommodate more complex geometries and material deformations in different dynamic scenes. Despite the added complexity, our convex representation remains highly efficient, requiring fewer primitives and shorter training time than Gaussian-based methods (see Tab. 4), owing to compact convex coverage and reduced-order simulation.

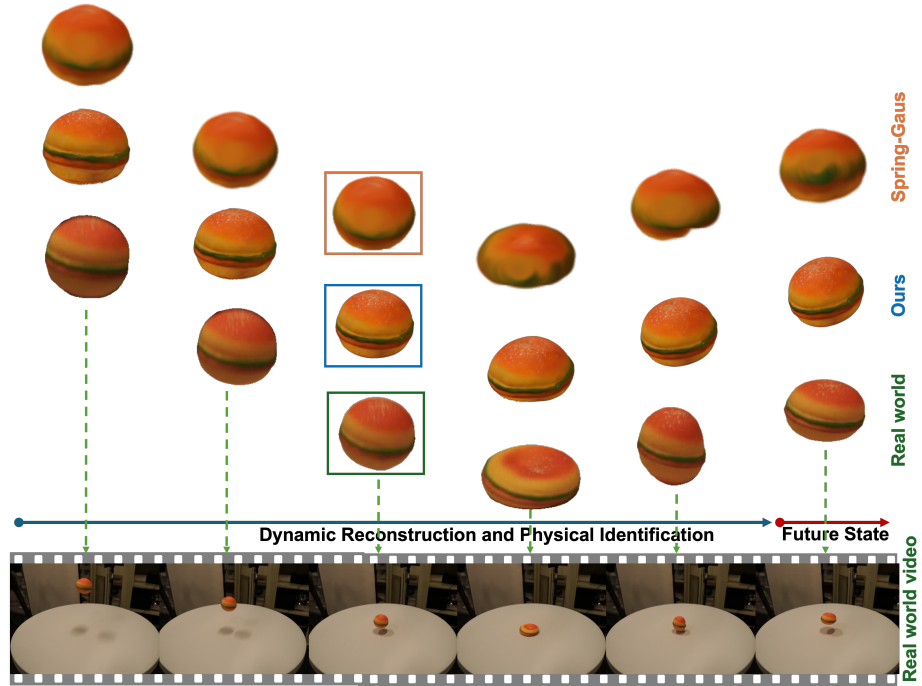


Fig. 7: Results on real-world samples (bottom: real references with background; top: zoomed-in objects without background). Compared with blurred real video observations and Spring-Gaus reconstructions, our method recovers geometry and appearance with clearer deformable details. Moreover, the reconstructed dynamics closely match the real video, exhibiting physically consistent motion and deformation.

5.7 Real-world Evaluation

Following Spring-Gaus [48], we evaluate PhysConvex on real-world dynamic reconstruction and future state prediction, as shown in Fig. 7. Reconstructing and simulating dynamic scenes from sparse viewpoints and motion-blurred videos in real-world datasets is highly challenging. To address this, we adopt the registration network in [48] to align 3D convex kernels from static scene coordinates to dynamic scene coordinates. Furthermore, our reduced-order boundary-driven convex dynamics enable efficient and physically plausible dynamic reconstruction and simulation. Compared with the blurred real-capture video and the reconstruction results of Spring-Gaus, our method recovers sharper geometry and appearance with finer details of deformable objects under real-world physics, such as the white dots on the burger object highlighted in the color boxes of Fig. 7. Moreover, our reconstructed dynamics closely match the real video, demonstrating physically consistent motion and deformation.

6 Conclusions and Future Work

We introduced PhysConvex, a physics-informed dynamic convex field for video-based reconstruction, physical system identification, and simulation of deformable objects. The key idea is to make each convex primitive a shared material-space unit for rendering, deformation, and physical simulation. By advecting convex boundaries rather than only primitive centers, PhysConvex captures non-uniform deformation, evolving active supports, sharp boundary motion, and compact anisotropic geometry. We further introduced a mesh-free reduced-order convex simulator, where neural skinning modes define compact deformation bases over convex supports. The same supports are used to evaluate mass, elastic energy, forces, and contacts, avoiding dense grids, tetrahedral meshes, and predefined particle-primitive bindings. Coupled with differentiable convex splatting, PhysConvex jointly recovers appearance, geometry, dynamics, and material parameters from videos. Experiments show improved dynamic reconstruction, physical parameter estimation, efficiency, future prediction, and generalization to changed materials, forces, and boundary conditions. Current limitations include the need for posed videos, object masks, and constrained physical settings such as gravity, boundary, and contact conditions. Future work will explore feed-forward convex initialization, contact modeling, and multi-object dynamic scene reconstruction.

Acknowledgements. This work was supported in part by ONR grant N00014-23-1-2526, IARPA via DOI/IBC contract number 140D0423C0076, gifts from Adobe, Google and Qualcomm, an NSERC Postdoctoral Fellowship, the Ronald L. Graham Chair, and the UC San Diego Center for Visual Computing. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing IARPA, ONR, DOI/IBC, or the U.S. Government.

References

1. Bargteil, A.W., Wojtan, C., Hodgins, J.K., Turk, G.: A finite element method for animating large viscoplastic flow. *ACM Transactions on Graphics (TOG)* **26**(3), 16-es (2007)
2. Benckroun, O., Zhang, J.E., Chaudhuri, S., Grinspun, E., Zhou, Y., Jacobson, A.: Fast complementary dynamics via skinning eigenmodes. *ACM Transactions on Graphics (TOG)* (2023)
3. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: Numerical optimization: theoretical and practical aspects. Springer (2006)
4. Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective dynamics: Fusing constraint projections for fast simulation. *Association for Computing Machinery* (2023)
5. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. *OpenAI Blog* (2024)
6. Cai, J., Yang, Y., Yuan, W., He, Y., Dong, Z., Bo, L., Cheng, H., Chen, Q.: Gic: Gaussian-informed continuum for physical property identification and simulation. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems* (2024)
7. Cao, J., Guan, S., Ge, Y., Li, W., Yang, X., Ma, C.: NeuMA: Neural material adaptor for visual grounding of intrinsic dynamics. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems* (2024)
8. Chang, Y., Chen, P.Y., Wang, Z., Chiaramonte, M.M., Carlberg, K., Grinspun, E.: Licrom: Linear-subspace continuous reduced order modeling with neural fields. In: *SIGGRAPH Asia 2023 Conference Papers* (2023)
9. 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 26545–26555 (2025)
10. Chen, P.Y., Xiang, J., Cho, D.H., Chang, Y., Pershing, G., Maia, H.T., Chiaramonte, M.M., Carlberg, K., Grinspun, E.: Crom: Continuous reduced-order modeling of pdes using implicit neural representations. *International Conference on Learning Representations* (2023)
11. Conway, J.H., Sloane, N.J.A.: Sphere packings, lattices and groups, vol. 290. Springer Science & Business Media (2013)
12. Cutler, B., Dorsey, J., McMillan, L., Müller, M., Jagnow, R.: A procedural approach to authoring solid models. *ACM Transactions on Graphics (TOG)* **21**(3), 302–311 (2002)
13. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnet: Learnable convex decomposition. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 31–41 (2020)
14. Desbrun, M., Gascuel, M.P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In: *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*. p. 61–76 (1996)
15. Diazi, L., Panozzo, D., Vaxman, A., Attene, M.: Constrained delaunay tetrahedrization: A robust and practical approach. *ACM Transactions on Graphics (TOG)* **42**(6) (2023)
16. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A high-quality dataset

- of 3d scanned household items. In: International Conference on Robotics and Automation (ICRA). pp. 2553–2560 (2022)
17. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4D-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In: ACM SIGGRAPH 2024 Conference Papers (2024)
 18. Gross, M., Pfister, H.: Point-based Graphics. Elsevier (2011)
 19. Held, J., Vandeghen, R., Hamdi, A., Deliege, A., Cioppa, A., Giancola, S., Vedaldi, A., Ghanem, B., Droogenbroeck, M.V.: 3D convex splatting: Radiance field rendering with 3D smooth convexes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21360–21369 (2025)
 20. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 20th International Conference on Pattern Recognition. pp. 2366–2369 (2010)
 21. Hu, Y., Anderson, L., Li, T.M., Sun, Q., Carr, N., Ragan-Kelley, J., Durand, F.: DiffTaichi: Differentiable programming for physical simulation. International Conference on Learning Representations (2020)
 22. Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: ACM SIGGRAPH 2016 Courses (2016)
 23. 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. International Conference on Computer Vision (ICCV) (2025)
 24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (2015)
 25. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3992–4003 (2023)
 26. Klár, G., Gast, T., Pradhana, A., Fu, C., Schroeder, C., Jiang, C., Teran, J.: Drucker-prager elastoplasticity for sand animation. ACM Transactions on Graphics (TOG) **35**(4), 1–12 (2016)
 27. Kugelstadt, T., Bender, J., Fernández-Fernández, J.A., Jeske, S.R., Löschner, F., Longva, A.: Fast corotated elastic sph solids with implicit zero-energy mode control. Proceedings of the ACM on Computer Graphics and Interactive Techniques **4**(3) (2021)
 28. Li, M., Ferguson, Z., Schneider, T., Langlois, T.R., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Transactions on Graphics (TOG) **39**(4) (2020)
 29. 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. International Conference on Learning Representations (2023)
 30. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4D reconstruction with dynamic 3d gaussian particle. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21136–21145 (2024)
 31. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3D gaussians: Tracking by persistent dynamic view synthesis. In: International Conference on 3D Vision (3DV). pp. 800–809 (2024)
 32. Mantiuk, R.K., Denes, G., Chapiro, A., Kaplanyan, A., Rufo, G., Bachy, R., Lian, T., Patney, A.: Fovvideovdp: A visible difference predictor for wide field-of-view video. ACM Transactions on Graphics (TOG) **40**(4) (2021)

33. Modi, V., Sharp, N., Perel, O., Sueda, S., Levin, D.I.W.: Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Transactions on Graphics (TOG)* (2024)
34. Peer, A., Gissler, C., Band, S., Teschner, M.: An implicit sph formulation for incompressible linearly elastic solids. *Computer Graphics Forum* **37**(6), 135–148 (2018)
35. Schneider, T., Dumas, J., Gao, X., Botsch, M., Panozzo, D., Zorin, D.: Poly-spline finite-element method. *ACM Transactions on Graphics (TOG)* **38**(3), 1–16 (2019)
36. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4104–4113 (2016)
37. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: *European Conference on Computer Vision*. pp. 501–518. Springer (2016)
38. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: *European Conference on Computer Vision*. pp. 1–18. Springer (2024)
39. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Proceedings of the 38th International Conference on Neural Information Processing Systems* (2024)
40. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems* (2022)
41. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004)
42. 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 20310–20320 (2024)
43. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4389–4398 (2024)
44. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *International Conference on Learning Representations* (2024)
45. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3D gaussians for high-fidelity monocular dynamic scene reconstruction. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 20331–20341 (2024)
46. Yue, Y., Smith, B., Batty, C., Zheng, C., Grinspun, E.: Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics (TOG)* **34**(5), 1–20 (2015)
47. Zhang, J.E., Bang, S., Levin, D.I.W., Jacobson, A.: Complementary dynamics. *ACM Transactions on Graphics (TOG)* (2020)
48. Zhong, L., Yu, H.X., Wu, J., Li, Y.: Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. *European Conference on Computer Vision (ECCV)* (2024)