GENTWIN: PART-AWARE GENERALIZED DYNAMICS MODELING WITH NEURAL ANISOTROPIC SPRINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning physically realistic object dynamics from real-world interactions is important for modeling the physical world. Existing methods typically adopt category-specific formulations for articulated and deformable dynamics, limiting their generalization to heterogeneous objects with both rigid and deformable parts. To address this, we introduce **GenTwin**, a part-aware framework that models generalized dynamics from real-world interaction observations within a common spring-mass representation, decomposing object dynamics into intra- and inter-part dynamics. For intra-part dynamics, we introduce Part-Adaptive Spring Configuration to adapt stiffness and topology to heterogeneous part rigidity using material priors and motion-derived rigidity. For inter-part dynamics, we connect neighboring parts with Neural Anisotropic Spring Connections, whose part-pair Neural Oriented Stiffness Fields parameterize continuous directional stiffness to capture relative motions from flexible limb swinging to constrained articulation. Experiments on 18 real-world scenarios spanning deformable, articulated, and heterogeneous objects demonstrate consistently improved geometric accuracy, reducing future prediction Chamfer distance by approximately 28.6%, 27.3%, and 50.0%, respectively, over the best competing category-specific methods.

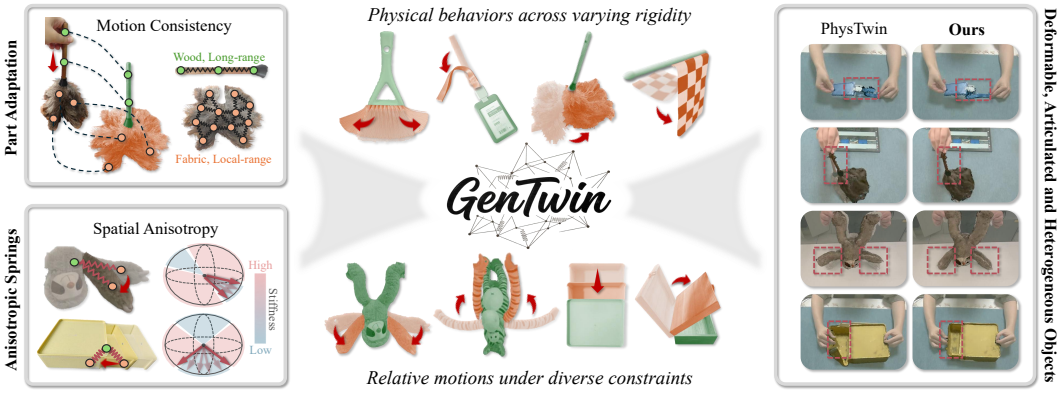
1 INTRODUCTION

Object dynamics describe how objects move and deform under external forces (Arriola-Rios et al., 2020). Learning these dynamics from real-world interactions supports applications in robotic manipulation (Ai et al., 2025), data generation (Lin et al., 2026), and virtual reality (Jiang et al., 2024).

To learn object dynamics from observations, existing methods typically adopt distinct formulations for articulated and deformable objects. Articulated methods parameterize the motion of rigid parts using explicit joints and axes (Liu et al., 2025c;b; Shen et al., 2026), whereas deformable methods model continuous deformation using physics-based simulators or learned dynamics models (Zhong et al., 2024; Jiang et al., 2025; Chen et al., 2026; Zhang et al., 2024). However, real-world objects often consist of both rigid and deformable parts, which can be categorized as *heterogeneous objects* (Kou & Tan, 2007). For example, a rigid card holder retains its shape while its soft lanyard bends during interaction, requiring joint modeling of rigid motion, flexible deformation and inter-part relative motion. Such coupled behaviors challenge category-specific formulations and raise a central question: *Can we achieve generalized dynamics modeling across deformable, articulated, and heterogeneous objects within a common physical representation?*

To address this question, we observe that dynamics across these categories arise from both deformation within parts and relative motion between parts. This motivates a part-aware decomposition into intra- and inter-part dynamics, with two key challenges (Fig. 1(b)). ***First,*** *intra-part dynamics should capture physical behaviors across varying rigidity.* Existing approaches use denser and stiffer springs for near-rigid approximation (Zhang et al., 2025b) or prescribed constitutive models for different materials (Chen et al., 2026), but such hand-crafted designs lack adaptation to heterogeneous part rigidity. ***Second,*** *inter-part dynamics should capture relative motions under diverse constraints.* Existing methods model strongly constrained articulated motion and more flexible inter-part coupling separately through explicit joints (Liu et al., 2025c) and deformable dynamics (Jiang et al., 2025), limiting their ability to accommodate diverse inter-part constraints.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Designs of GenTwin (b) Intra- and Inter-Part Challenges (c) Generalized Dynamics

Figure 1: **GenTwin for Generalized Dynamics Modeling.** (a) GenTwin adapts intra-part spring configurations using material and motion cues (Sec. 3.2), and captures spatial anisotropy in inter-part relative motion with neural anisotropic springs (Sec. 3.3). (b) These designs address physical behaviors across varying rigidity and relative motions under diverse constraints. (c) GenTwin outperforms PhysTwin across deformable, articulated, and heterogeneous objects.

To address these challenges, we introduce **GenTwin**, a part-aware framework for generalized dynamics modeling within a common spring-mass representation, guided by two key insights (Fig. 1(a)). ***First,*** *intra-part rigidity manifests through motion consistency.* We therefore estimate part rigidity from Kabsch-aligned motion residuals and combine it with part-level material priors to construct the **Part-Adaptive Spring Configuration**, adapting intra-part stiffness and topology accordingly. ***Second,*** *inter-part relative motion exhibits spatial anisotropy.* Accordingly, we introduce **Neural Anisotropic Spring Connections**, where a part-pair Neural Oriented Stiffness Field predicts continuous directional stiffness to capture diverse constraints from flexible coupling to constrained articulation. Finally, the resulting spring configurations are optimized through zero-order search followed by differentiable refinement of intra-part stiffness and inter-part stiffness anisotropy.

Our contributions can be summarized as follows:

(1) We propose **GenTwin**, a part-aware framework for generalized dynamics modeling across deformable, articulated, and heterogeneous objects within a common spring-mass representation.

(2) For intra-part dynamics, we introduce a Part-Adaptive Spring Configuration that combines material priors and motion-derived rigidity to adapt spring stiffness and topology to varying rigidity.

(3) For inter-part dynamics, we introduce Neural Anisotropic Spring Connections, using Neural Oriented Stiffness Fields to capture relative motion constraints through continuous directional stiffness.

(4) Experiments on 18 real-world scenarios demonstrate superior geometric accuracy over category-specific baselines in both resimulation and future prediction.

2 Related Work

Object Dynamics Modeling from Real-World Observations. Recent advances in 3D Gaussian Splatting (Kerbl et al., 2023; Wu et al., 2024; Deng et al., 2026a) enable high-fidelity reconstruction, while physics-aware extensions model object dynamics (Xie et al., 2024; Lin et al., 2025). Existing methods typically adopt category-specific formulations for deformable and articulated objects. For deformable objects, physics-based methods (Zhong et al., 2024; Jiang et al., 2025; Chen et al., 2026) use spring-mass models or differentiable continuum simulators, learning-based methods (Zhang et al., 2024; 2025a; Huang et al., 2026a) learn dynamics from interactions, and hybrid methods (Patel et al., 2026; Li et al., 2026; Deng et al., 2026c) augment simulators with learned residuals. Articulated methods (Liu et al., 2025c;b; Shen et al., 2026) recover rigid parts and parameterize their relative motion through explicit articulation models. However, these category-specific formulations and dependence on training distributions limit their generalization to objects with varying intra-part rigidity and diverse inter-part constraints.

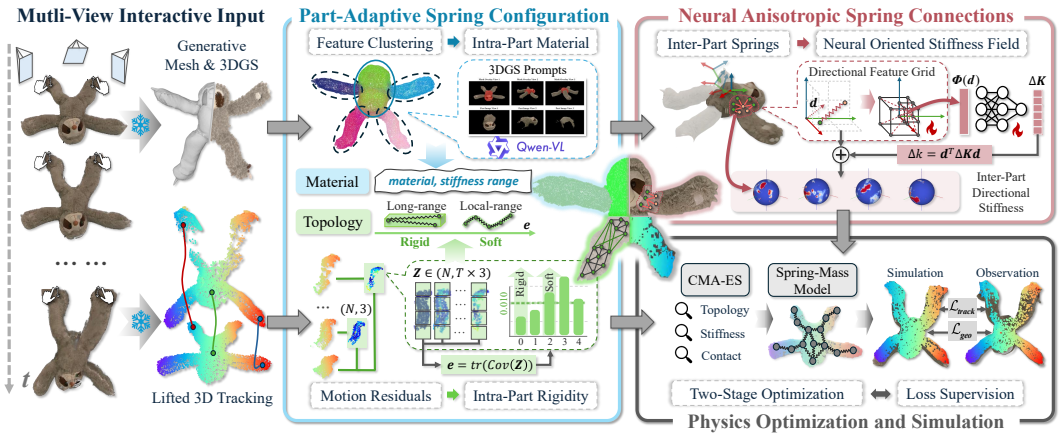


Figure 2: **Overview of GenTwin.** We decompose generalized dynamics into two aspects: (i) intra-part dynamics, where we adapt spring configurations to varying part rigidity using material priors and motion residuals; (ii) inter-part dynamics, where we connect neighboring parts with Neural Anisotropic Spring Connections to capture relative motions from flexible coupling to articulation.

Physical Property Modeling toward Generalized Dynamics. Physical properties provide useful priors for object dynamics modeling (Wang et al., 2026; Zhan et al., 2026; Garcia et al., 2025). Vision-language approaches (Zhai et al., 2024; Shuai et al., 2025; Le et al., 2026) infer material properties over reconstructed 3D representations, while feed-forward methods (Le et al., 2025; Huang et al., 2026b; Dagli et al., 2026) predict simulation-ready material fields from physically annotated assets. However, they rely largely on static visual or semantic cues rather than observed dynamics. Recent spring-mass methods further incorporate such priors into dynamics modeling from interactions: NeuSpring (Xu et al., 2026) handles heterogeneous deformation through piecewise topology, while MatPhys (Yang et al., 2026) uses semantic parts and MLLM material priors to predict spring parameters. Both remain focused on deformable dynamics, without explicitly modeling near-rigid intra-part behavior or anisotropic inter-part motion. More recently, GDGen (Li et al., 2025) further models soft, near-rigid, and articulated dynamics with anisotropic elasticity, but relies on prescribed trajectories in idealized simulations and restricts anisotropy to three orthotropic directions.

3 Method: GenTwin

GenTwin models generalized object dynamics from real-world interaction observations within a common part-aware spring-mass representation (Sec. 3.1). For intra-part dynamics, part-level material recognition and motion-derived rigidity jointly configure part-specific spring stiffness ranges and topology (Sec. 3.2). For inter-part dynamics, Neural Anisotropic Spring Connections link neighboring parts and model relative-motion constraints through continuous directional stiffness (Sec. 3.3). Finally, the resulting physical model is identified from observed interactions through a two-stage optimization scheme, consisting of zero-order search followed by differentiable refinement (Sec. 3.4). Fig. 2 provides an overview of the GenTwin framework.

3.1 Problem Formulation

Given synchronized multi-view RGB-D videos captured during physical interaction, our goal is to learn physically realistic object dynamics of the manipulated object. Building on the data preprocessing pipeline of PhysTwin (Jiang et al., 2025), we obtain the initial object geometry $\mathcal{P}^0$, partial observations $\mathcal{O}^t$, lifted 3D tracks $\mathcal{Q}^t = \{\mathbf{q}_i^t\}$, and controller trajectories $\mathcal{C}^t = \{\mathbf{c}_j^t\}$ using foundation models (Chang et al., 2026; Karaev et al., 2025), where $t = 0, \ldots, T-1$ indexes the interaction frames. Further implementation details are provided in Appendix A.3.

We represent the object as a part-aware spring graph $\mathcal{G}(\boldsymbol{\eta}) = (\mathcal{V}, \mathcal{E}_{\text{intra}} \cup \mathcal{E}_{\text{inter}})$, where $\mathcal{V}$ contains particle nodes initialized from $\mathcal{P}^0$, and $\mathcal{E}_{\text{intra}}$ and $\mathcal{E}_{\text{inter}}$ denote spring edges within and across parts, respectively. Each particle i has part label $p(i)$, mass m_i, position $\mathbf{x}_i^t$, and velocity $\mathbf{v}_i^t$. We define the system state as $\mathbf{S}^t = (\mathbf{X}^t, \mathbf{V}^t)$, collecting the positions and velocities of all particles.

We seek to optimize the graph topology parameters $\boldsymbol{\eta}$, physical parameters $\boldsymbol{\alpha}$, and neural stiffness-field parameters $\boldsymbol{\Phi}$ from observed interactions. These parameters govern the simulated dynamics under the controller trajectories $\{\mathcal{C}^t\}$, while $\mathcal{L}_{\text{obs}}$ denotes the observation loss accumulated over the observed frames $\mathcal{T}_{\text{obs}}$:

$$\mathbf{S}^{t+1} = F_{\mathcal{G}(\boldsymbol{\eta}),\boldsymbol{\alpha},\boldsymbol{\Phi}}\left(\mathbf{S}^t, \mathcal{C}^t\right), \qquad (\boldsymbol{\eta}^*, \boldsymbol{\alpha}^*, \boldsymbol{\Phi}^*) = \arg\min_{\boldsymbol{\eta},\boldsymbol{\alpha},\boldsymbol{\Phi}} \mathcal{L}_{\text{obs}}. \tag{1}$$

3.2 Intra-Part Dynamics: Part-Adaptive Spring Configuration

For generalized dynamics, intra-part behaviors vary with material and effective rigidity. GenTwin therefore derives complementary cues from appearance and observed motion to configure part-specific spring stiffness and topology, corresponding to the intra-part components of $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$.

Part-Level Material Recognition. Following prior work (Xu et al., 2026; Yang et al., 2026; 2025b), we obtain part labels by clustering PartField features (Liu et al., 2025a) on points sampled from $\mathcal{P}^0$. As feed-forward methods (Le et al., 2025; Dagli et al., 2026) do not directly predict spring coefficients, we infer material priors with a part-aware VLM pipeline. We fit a static 3D Gaussian representation $\mathcal{A}^0$ to the sparse first-frame views, transfer the part labels to the Gaussians, and render multi-view images highlighting each part. From these images, Qwen-VL (Qwen Team, 2026) identifies the object and its parts, then predicts each part's material type and plausible stiffness range. Each intra-part spring inherits its part's prior $\mathcal{M}_p = (\tau_p, k_p^0, k_p^-, k_p^+)$, where τ_p describes the material, k_p^0 initializes stiffness, and (k_p^-, k_p^+) bounds the zero-order search (Sec. 3.4). Further implementation details are provided in Appendix B.1.

Motion-Derived Rigidity Identification. Appearance-based material cues cannot fully determine part rigidity, as similar materials may deform differently under interaction. We therefore infer rigidity from motion residuals after removing the shared rigid transformation, which remain small for near-rigid parts and larger for deformable parts. Specifically, for part p at frame t, we align its valid tracked points to the reference frame using Kabsch-based rigid alignment (Kabsch, 1993):

$$(\mathbf{R}_p^t, \mathbf{b}_p^t) = \arg\min_{\mathbf{R}\in\mathrm{SO}(3),\, \mathbf{b}\in\mathbb{R}^3} \sum_{i\in\mathcal{I}_p^t} \left\|\mathbf{R}\mathbf{q}_i^t + \mathbf{b} - \mathbf{q}_i^0\right\|_2^2, \qquad \mathbf{r}_i^t = \mathbf{R}_p^t\mathbf{q}_i^t + \mathbf{b}_p^t - \mathbf{q}_i^0, \tag{2}$$

where $\mathcal{I}_p^t$ contains tracks valid at both frames and $\mathbf{r}_i^t$ denotes the residual after removing the shared rigid transformation. Inspired by motion segmentation methods using temporal variance of pairwise distances as a rigidity cue (Mateus & Horaud, 2007), we stack each point's residuals over time into $\mathbf{Z}_p \in \mathbb{R}^{N_p\times 3T_{\text{obs}}}$ and define the residual deformation variance as

$$e_p = \mathrm{tr}\left(\mathrm{Cov}(\mathbf{Z}_p)\right) = \frac{\|\mathbf{Z}_p - \mathbf{1}\bar{\mathbf{z}}_p^\top\|_F^2}{N_p - 1}, \tag{3}$$

where $\bar{\mathbf{z}}_p$ is the mean residual trajectory across points. We therefore use the motion-derived residual deformation variance as a complementary rigidity cue to adapt the spring topology of each part.

Rigidity-Adaptive Spring Topology. Conventional spring-mass models (Blickhan, 1989; Jiang et al., 2025) for deformable dynamics typically rely on local nearest-neighbor connections. For near-rigid parts, however, such local topology may still permit large-scale deformation even with dense connections and large stiffness (Zhang et al., 2025b). Guided by the motion-derived rigidity cue, we therefore augment near-rigid parts with long-range springs to better resist large-scale deformation, as illustrated in Fig. 3. Specifically, let $L_p = \max_{i,j\in\mathcal{V}_p}\|\mathbf{x}_i^0 - \mathbf{x}_j^0\|_2$ denote the spatial extent of part p. We adapt the intra-part topology as

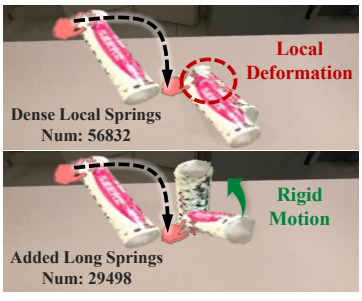


Figure 3: **Effect of Long Springs.**

$$\mathcal{E}_p^{\text{intra}} = \begin{cases} \mathcal{E}_p^{\text{local}} \cup \mathcal{E}_p^{\text{long}}, & e_p < \epsilon_{\text{rigid}}, \\ \mathcal{E}_p^{\text{local}}, & \text{otherwise}, \end{cases} \qquad \mathcal{E}_p^{\text{long}} = \left\{(i,j) : \beta^- \le \frac{\|\mathbf{x}_i^0 - \mathbf{x}_j^0\|_2}{L_p} \le \beta^+\right\}, \tag{4}$$

where β^- and β^+ define the normalized distance range for long-range spring construction.

3.3 Inter-Part Dynamics: Neural Anisotropic Spring Connections

For generalized dynamics, inter-part relative motions exhibit spatial anisotropy under diverse constraints. GenTwin therefore constructs anisotropic spring connections to configure the inter-part topology $\boldsymbol{\eta}$, with Neural Oriented Stiffness Fields $\boldsymbol{\Phi}$ parameterizing the directional stiffness $\boldsymbol{\alpha}$.

Inter-Part Spring Construction. Unlike NeuSpring (Xu et al., 2026), which shares neighborhood parameters across intra- and inter-part springs, we optimize separate inter-part parameters during the zero-order search to better capture relative-motion constraints (Sec. 3.4). We connect neighboring particles across parts and group the resulting edges by part pair as $\mathcal{E}_{pq} = \{(i,j) \in \mathcal{E}_{\text{inter}} \mid p(i) = p,\ p(j) = q\}$. Each spring is initialized with the smaller stiffness prior of its connected parts, $k^0_{pq} = \min(k^0_p, k^0_q)$. During differentiable refinement, a part-pair Neural Oriented Stiffness Field predicts its directional stiffness from observed interactions. These connections retain the flexibility of spring-based deformation while expressing articulation-like constraints without explicitly estimating joint types or articulation axes.

Neural Oriented Stiffness Field. To model the spatial anisotropy of inter-part relative motion, we model each inter-part spring group $\mathcal{E}_{pq}$ with an independent Neural Oriented Stiffness Field Φ_{pq}. Using a multi-resolution feature-grid encoder and a lightweight neural decoder (Müller, 2021; Müller et al., 2022), the field maps canonical spring directions to stiffness residuals, which are combined with k^0_{pq} through bounded decoding to obtain directional stiffness.

Before querying the Neural Oriented Stiffness Field, we first transform spring directions into a canonical frame. We choose the stiffer part of each pair as the anchor a and estimate its rotation $\mathbf{Q}^t_a$ from the reference to the current configuration using Kabsch alignment. Each spring $e = (i,j) \in \mathcal{E}_{pq}$ is then expressed in the anchor-local frame:

$$\mathbf{d}^t_e = \frac{\mathbf{x}^t_j - \mathbf{x}^t_i}{\|\mathbf{x}^t_j - \mathbf{x}^t_i\|_2}, \qquad \tilde{\mathbf{d}}^t_e = (\mathbf{Q}^t_a)^\top \mathbf{d}^t_e. \tag{5}$$

The canonical direction $\tilde{\mathbf{d}}^t_e$ serves as the field input, making stiffness predictions invariant to global rigid rotation. Given this direction, Φ_{pq} predicts six coefficients that define a symmetric residual stiffness tensor $\Delta\mathbf{K}^t_e \in \mathbb{R}^{3\times3}$. Evaluating the quadratic form of this tensor along $\tilde{\mathbf{d}}^t_e$ yields the scalar directional stiffness residual:

$$\Delta\mathbf{K}^t_e = \mathrm{Sym}\Big(\Phi_{pq}(\tilde{\mathbf{d}}^t_e)\Big), \qquad \Delta k^t_e = (\tilde{\mathbf{d}}^t_e)^\top \Delta\mathbf{K}^t_e \tilde{\mathbf{d}}^t_e, \tag{6}$$

Finally, the directional residual Δk^t_e modulates the initial stiffness k^0_{pq} through sigmoid-based bounded decoding, yielding the final directional stiffness k^t_e. Further implementation details are provided in Appendix B.4.

3.4 Physics Optimization and Simulation

Explicit Spring Dynamics. Following the NVIDIA Warp-based spring-mass simulation paradigm (Macklin, 2022; Jiang et al., 2025), a spring e with rest length ℓ_e, stiffness k_e, and axial damping coefficient γ exerts the following strain-based force along its unit direction $\mathbf{d}_e$:

$$\mathbf{f}_{i\leftarrow j} = k_e \left(\frac{\|\mathbf{x}_j - \mathbf{x}_i\|_2}{\ell_e} - 1 \right) \mathbf{d}_e + \gamma \left[(\mathbf{v}_j - \mathbf{v}_i)^\top \mathbf{d}_e \right] \mathbf{d}_e. \tag{7}$$

Together with gravity, contact, and friction forces, the stiffnesses of $\mathcal{E}_{\text{intra}}$ and $\mathcal{E}_{\text{inter}}$ govern the evolution of particle states. We then transfer particle motion to the static Gaussian $\mathcal{A}^0$ via Linear Blend Skinning (Huang et al., 2024a; Zhang et al., 2024), enabling realistic appearance rendering throughout forward simulation.

Observation Supervision. The observation loss $\mathcal{L}_{\text{obs}}$ in Sec. 3.1 follows PhysTwin (Jiang et al., 2025), combining geometric loss $\mathcal{L}_{\text{geo}}$ and tracking loss $\mathcal{L}_{\text{track}}$ over the observed frames $\mathcal{T}_{\text{obs}}$:

$$\mathcal{L}_{\text{obs}} = \lambda_{\text{geo}} \sum_{t\in\mathcal{T}_{\text{obs}}} \overrightarrow{d_{\text{CD}}}(\mathcal{O}^t, \mathbf{X}^t) + \lambda_{\text{track}} \sum_{t\in\mathcal{T}_{\text{obs}}} \frac{1}{|\mathcal{I}^t|} \sum_{i\in\mathcal{I}^t} \bar{\rho}(\mathbf{x}^t_i - \mathbf{q}^t_i). \tag{8}$$

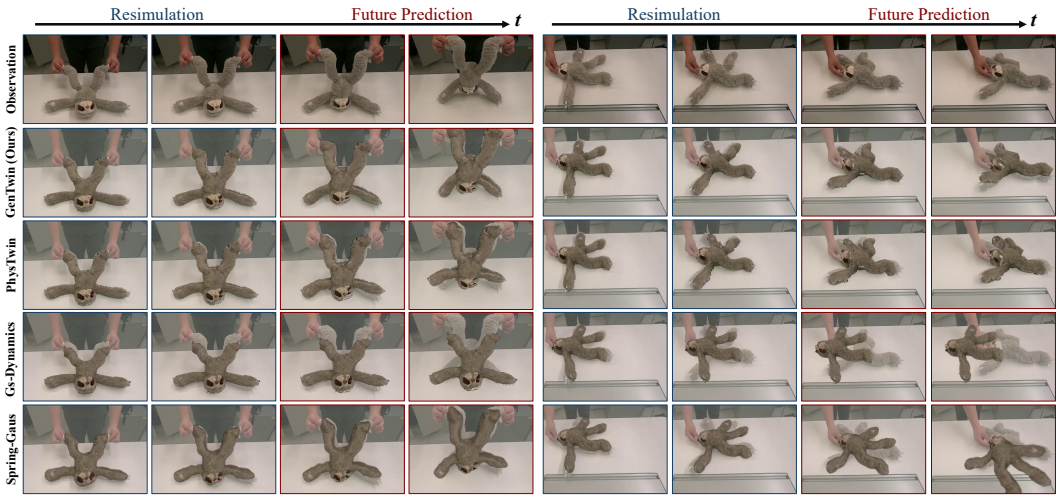


Figure 4: **Qualitative comparisons on deformable objects.** Compared with PhysTwin, GenTwin better captures the relative motion between the limbs and body through anisotropic inter-part modeling, producing limb positions and orientations that more closely match the observations.

Here $d_{\mathrm{CD}}^{\rightarrow}$ denotes the one-sided squared Chamfer distance from the partial observation to the simulated particles, and $\bar{\rho}$ is the coordinate-averaged smooth-L_1 loss over valid tracks $\mathcal{I}^t$.

Two-Stage Optimization. Following PhysTwin (Jiang et al., 2025), we adopt a two-stage optimization scheme. First, CMA-ES (Hansen, 2006) jointly searches the graph topology $\boldsymbol{\eta}$ and coarse physical parameters $\boldsymbol{\alpha}$, using part-specific stiffness ranges for intra-part springs and separate neighborhood parameters for $\mathcal{E}_{\mathrm{intra}}$ and $\mathcal{E}_{\mathrm{inter}}$. The resulting graph is then fixed during differentiable Warp simulation, where intra-part stiffness and inter-part directional stiffness parameterized by the Neural Oriented Stiffness Fields $\boldsymbol{\Phi}$ are jointly refined through gradient-based optimization.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate GenTwin on 18 real-world interaction scenarios covering diverse dynamics: 7 multi-part deformable scenarios from PhysTwin (Jiang et al., 2025), 5 self-captured articulated scenarios involving drawers and cases, and 6 self-captured heterogeneous scenarios composed of parts with different materials. Following the PhysTwin capture setup, each scenario comprises RGB-D streams from three viewpoints over 1–10 seconds, temporally split into training and test frames at a 7:3 ratio, with only the training frames used for dynamics modeling.

Baselines. For deformable and heterogeneous scenarios, we primarily compare with PhysTwin (Jiang et al., 2025), which identifies spring-mass dynamics from real-world interaction observations, along with Spring-Gaus (Zhong et al., 2024) based on spring-mass simulation and GS-Dynamics (Zhang et al., 2024) based on learned graph dynamics. For articulated scenarios, we compare with ArtGS (Liu et al., 2025c) and GaussianArt (Shen et al., 2026). We train them on sparse three-view images from the first and last training frames and obtain future predictions by temporal extrapolation. See Appendix C.2 for more details.

Metrics. Following the evaluation protocol of PhysTwin (Jiang et al., 2025), we evaluate 3D geometric and motion accuracy using Chamfer distance (CD) and tracking error (Track), and 2D rendering quality using IoU, PSNR, SSIM, and LPIPS. Results on the training frames measure reconstruction and resimulation, while those on the subsequent test frames measure future prediction.

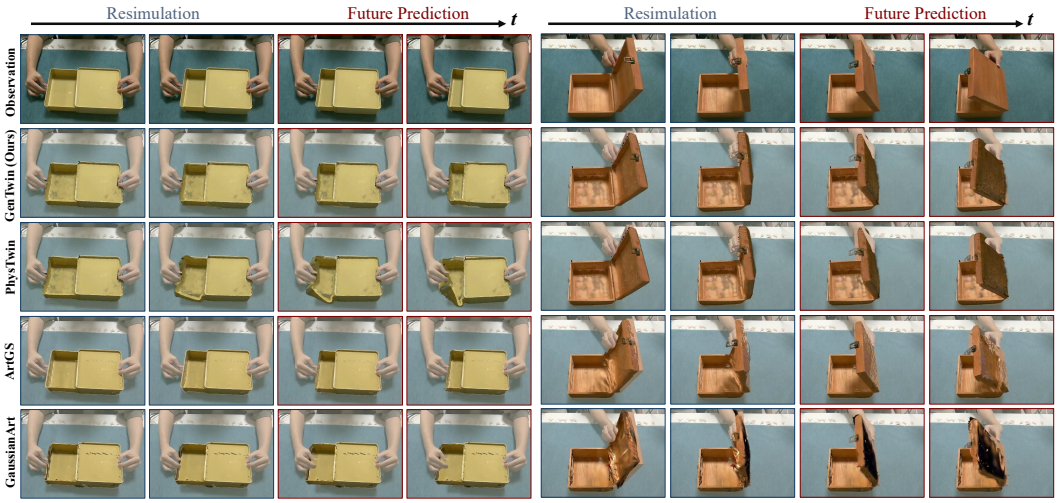


Figure 5: **Qualitative comparisons on articulated objects.** Compared with PhysTwin, GenTwin better preserves part rigidity during hinge motion. With sparse state views, ArtGS and GaussianArt exhibit incorrect Gaussian trajectories at intermediate states, producing visible motion artifacts.

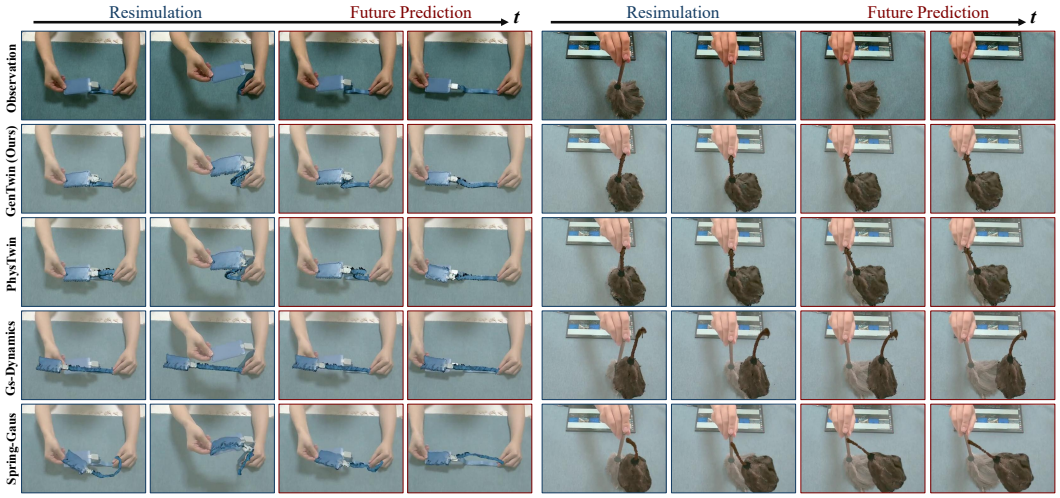


Figure 6: **Qualitative comparisons on heterogeneous objects.** Compared with PhysTwin, GenTwin better captures both the flexible lanyard motion of the card holder and the rigid handle motion of the feather duster, demonstrating its adaptability to heterogeneous materials.

4.2 Experimental Results

Generalized Dynamics Modeling from Interactions. To assess whether GenTwin can model generalized dynamics across object categories, we report its resimulation and future prediction results in Table 1. GenTwin achieves the lowest geometric errors across all three categories, reducing future prediction CD by approximately 28.6%, 27.3%, and 50.0% relative to the second-best results on deformable, articulated, and heterogeneous objects, respectively. It also remains competitive in rendering quality, trailing articulation-specific methods only for articulated resimulation.

Qualitative results further support our two key designs. On deformable and articulated objects (Figs. 4 and 5), Neural Anisotropic Spring Connections capture flexible limb swinging and constrained articulated motion, whereas PhysTwin exhibits motion misalignment and undesired local deformation. Articulated methods reconstruct endpoint states well but produce inaccurate intermediate trajectories and visible artifacts under sparse two-state observations. On heterogeneous objects (Fig. 6), Part-Adaptive Spring Configuration captures heterogeneous part behaviors, modeling flexible lanyard deformation with rigid card motion, as well as rigid handle translation with deformable

Table 1: **Quantitative results across object categories.** We evaluate the reconstruction & resimulation and future prediction on deformable, articulated, and heterogeneous objects. Best and second-best results are shown in bold and underlined, respectively.

Method	Reconstruction & Resimulation						Future Prediction					
	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
Deformable objects (7 scenarios)												
GS-Dynamics	0.014	0.024	0.727	26.670	0.949	0.047	0.042	0.079	0.435	21.761	0.919	0.102
Spring-Gaus	0.046	0.055	0.624	24.006	0.931	0.082	0.158	0.168	0.347	21.294	0.916	0.118
PhysTwin	0.005	**0.007**	0.844	27.311	0.949	0.025	0.014	0.023	0.698	24.318	0.936	0.057
GenTwin (Ours)	**0.004**	**0.007**	**0.865**	**27.687**	**0.951**	**0.023**	**0.010**	**0.020**	**0.745**	**25.097**	**0.941**	**0.048**
Articulated objects (5 scenarios)												
GS-Dynamics	0.014	0.023	0.818	24.833	0.942	0.063	0.029	0.063	0.676	21.705	0.924	0.094
Spring-Gaus	0.010	0.016	0.862	25.436	0.945	0.065	0.028	0.044	0.786	23.617	0.944	0.075
ArtGS	0.015	0.014	0.875	**28.314**	**0.955**	0.041	0.018	0.021	0.824	26.435	**0.952**	0.046
GaussianArt	0.014	0.013	0.806	27.536	0.953	0.043	0.016	0.023	0.773	25.032	0.948	0.055
PhysTwin	0.008	0.013	0.899	26.407	0.948	0.044	0.011	0.025	0.859	24.242	0.945	0.055
GenTwin (Ours)	**0.007**	**0.009**	**0.917**	27.769	0.952	**0.038**	**0.008**	**0.012**	**0.906**	**26.524**	**0.952**	**0.044**
Heterogeneous objects (6 scenarios)												
GS-Dynamics	0.023	0.030	0.604	24.978	0.966	0.045	0.032	0.042	0.539	24.186	0.965	0.047
Spring-Gaus	0.063	0.084	0.273	22.125	0.953	0.081	0.085	0.099	0.277	22.708	0.957	0.072
PhysTwin	0.008	0.013	0.746	29.383	0.982	0.017	0.018	0.028	0.567	26.339	0.975	0.031
GenTwin (Ours)	**0.007**	**0.011**	**0.772**	**29.905**	**0.983**	**0.015**	**0.009**	**0.017**	**0.670**	**27.998**	**0.978**	**0.024**

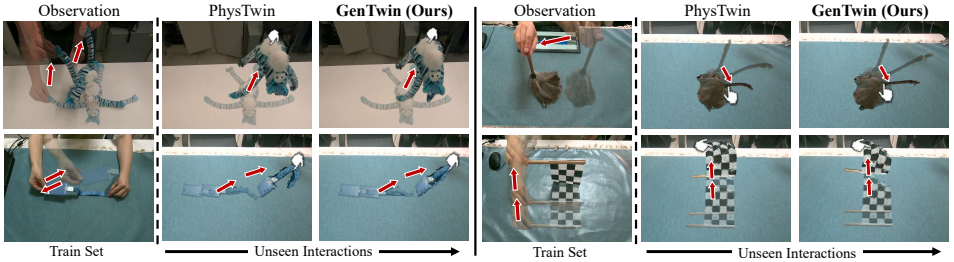


Figure 7: **Generalization to unseen interactions.** Across four unseen interactions, GenTwin produces more physically plausible predictions than PhysTwin, avoiding unrealistic behaviors such as stiff toy arms, bent rigid handle, overly rigid lanyard, and unnatural flag draping.

feather motion. Together, these results demonstrate generalized dynamics modeling within a common physical representation across varying intra-part rigidity and inter-part constraints.

Generalization to Unseen Interactions. Dynamics learned from a single interaction may not fully characterize an object under novel manipulations. We therefore evaluate the learned object dynamics under unseen interactions on four representative scenarios: *double_lift_zebra*, *single_push_duster_03*, *double_lift_card_01*, and *single_lift_flag_03*. As shown in Fig. 7, GenTwin produces more physically plausible dynamics than PhysTwin, preserving realistic inter-part motion and the contrasting rigid and deformable behaviors within heterogeneous objects even under unseen interactions. Following PhysWorld (Yang et al., 2025b), further quantitative results are provided in Appendix D.3.

Physics-informed Novel Demonstration Generation. Existing 3DGS-based methods generate novel demonstrations mainly through object trajectory transfer (Yang et al., 2025a; Deng et al., 2026b), while DeformGen (Lin et al., 2026) introduces physics-based augmentation for deformable objects. Building on GenTwin's generalized dynamics representation, we enable physics-informed novel demonstration generation across diverse object categories, as shown in Fig. 8.

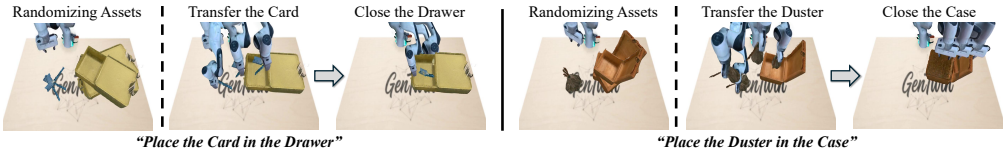


Figure 8: **Physics-informed novel demonstration generation.** Building on RoboSplat (Yang et al., 2025a), GenTwin enables physically plausible embodied demonstration generation across object categories. We show two representative tasks involving heterogeneous and articulated objects.

Table 2: **Quantitative comparisons with ablation variants.** Ablation of the Part-Adaptive Spring Configuration and Neural Anisotropic Spring Connections on the 11 self-captured scenarios.

Variant	Reconstruction & Resimulation						Future Prediction					
	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
w/o material	**0.007**	0.011	0.820	28.684	0.968	**0.026**	0.010	0.017	0.761	27.097	0.966	0.035
w/ dense local	0.008	0.012	0.814	28.420	0.967	0.028	0.011	0.019	0.750	26.778	0.965	0.036
w/o anisotropy	**0.007**	0.011	0.829	28.471	0.967	0.027	0.011	0.020	0.737	26.225	0.963	0.039
Full model	**0.007**	**0.010**	**0.838**	**28.934**	**0.969**	**0.026**	**0.009**	**0.015**	**0.777**	**27.328**	**0.967**	**0.033**

4.3 Ablation Studies

We perform ablation studies on our two key designs in Secs. 3.2 and 3.3. Quantitative results on 11 self-captured scenarios are reported in Table 2, with representative comparisons shown in Fig. 9. Additional ablation results on the PhysTwin benchmark and comparisons with feed-forward methods (Dagli et al., 2026; Le et al., 2025) are provided in Appendix D.

Part-Adaptive Spring Configuration. As shown in Table 2, removing the part-level material prior (*w/o material*) or replacing rigidity-adaptive long-range springs with denser local springs (*w/ dense local*) consistently degrades performance. Figs. 9(a,b) further illustrate these effects: without material priors, the lanyard is excessively stretched, while dense local connections fail to preserve the rigidity of the feather-duster handle under pressing.

Neural Anisotropic Spring Connections. As shown in Table 2, replacing our neural anisotropic inter-part springs with standard isotropic springs (*w/o anisotropy*) consistently degrades performance, with a larger drop in future prediction. Figs. 9(c,d) further illustrate that isotropic stiffness fails to preserve directional inter-part constraints, producing overly stiff limb motion and inhibited drawer translation.

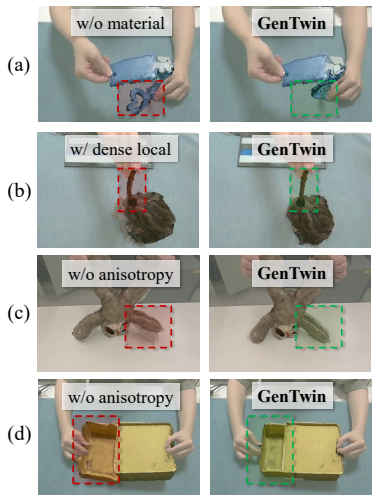


Figure 9: **Qualitative comparisons with ablation variants.**

5 Conclusion

In this work, we introduce GenTwin, a part-aware framework for modeling generalized dynamics from real-world interaction videos across deformable, articulated, and heterogeneous objects. We decompose generalized dynamics into intra- and inter-part dynamics, introducing Part-Adaptive Spring Configuration to adapt spring stiffness and topology to varying intra-part rigidity, and Neural Anisotropic Spring Connections with Neural Oriented Stiffness Fields to model diverse inter-part relative motions through continuous directional stiffness. Experiments on public and self-captured scenarios demonstrate improved geometric accuracy across all three object categories and physically plausible generalization to unseen interactions. We further demonstrate physics-informed novel demonstration generation for manipulation data synthesis across diverse object types.

AI USE STATEMENT

In this work, we used generative AI tools for text polishing, formatting, and editing the manuscript to improve readability. We have not used generative AI tools for generating synthetic datasets, developing theoretical models, formulating claims or proofs, designing research methodology, implementing methods, or data analysis, and all other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for proofreading and stylistic improvements of the text. We have reviewed all AI-assisted work. Specifically, all AI-edited text was manually reviewed by the authors to ensure technical precision, domain accuracy, and consistency with our research findings. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We support reproducibility by detailing the preprocessing pipeline for our self-captured dataset in Appendix A.3 and the implementation of GenTwin in Appendix B. Appendix C.2 reports the configurations used for all baseline methods, while Appendix C.3 specifies the evaluation protocols and metrics. Both the data and code will be released.

REFERENCES

Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Tobias Pfaff, Cheston Tan, Henrik I Christensen, Hao Su, Jiajun Wu, and Yunzhu Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10(106):eadt1497, 2025.

Veronica E Arriola-Rios, Puren Guler, Fanny Ficuciello, Danica Kragic, Bruno Siciliano, and Jeremy L Wyatt. Modeling of deformable objects for robotic manipulation: A tutorial and review. *Frontiers in Robotics and AI*, 7:82, 2020.

Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22 (11-12):1217–1227, 1989.

Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation. In *International Conference on Learning Representations*, volume 2026, pp. 90387–90401, 2026.

Yunuo Chen, Yafei Hu, Lingfeng Sun, Tushar Kusnur, Laura Herlant, and Chenfanfu Jiang. Empm: Embodied mpm for modeling and simulation of deformable objects. *IEEE Robotics and Automation Letters*, 2026.

Rishit Dagli, Donglai Xiang, Vismay Modi, Charles Loop, Clement Fuji Tsang, Anka He Chen, Anita Hu, Gavriel State, David Levin I.W., and Maria Shugrina. VoMP: Predicting volumetric mechanical property fields. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yinan Deng, Jianyu Dou, Jiahui Wang, Jingyu Zhao, Yi Yang, and Yufeng Yue. Pamosplat: Part-aware motion-guided gaussian splatting for dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026a.

Yinan Deng, Kejia Hu, Ye Chen, Jianyu Dou, Jiahui Wang, Jingyu Zhao, Haojia Ao, Yi Yang, and Yufeng Yue. Video2robo: 3dgs-based synthetic data from one video enables scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6695–6705, 2026b.

Yinan Deng, Jianqiao Song, Yisi Zhang, Yuhan Wang, Jiahui Wang, and Yufeng Yue. Physreal: Learning real-world deformable object physics via hybrid constitutive modeling. *arXiv preprint arXiv:2609.07532*, 2026c.

Alejandro Castaneda Garcia, Jan Warchocki, Jan Van Gemert, Daan Brinks, and Nergis Tomen. Learning physics from video: Unsupervised physical parameter estimation for continuous dynamical systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27924–27933, 2025.

Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pp. 75–102, 2006.

Mu Huang, Hui Wang, Kerui Ren, Linning Xu, Yunsong Zhou, Mulin Yu, Bo Dai, and Jiangmiao Pang. Soma: A real-to-sim neural simulator for robotic soft-body manipulation. *arXiv preprint arXiv:2602.02402*, 2026a.

Qilin Huang, Quynh Anh Huynh, Long Le, Chen Wang, Chuhao Chen, Ryan Lucas, Eric Eaton, and Lingjie Liu. Unipixie: Unified and probabilistic 3d physics learning via flow matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19907–19916, June 2026b.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4220–4230. IEEE, 2024a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7219–7230. IEEE, 2025.

Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, 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.

Wolfgang Kabsch. Automatic processing of rotation diffraction data from crystals of initially unknown symmetry and cell constants. *Applied Crystallography*, 26(6):795–800, 1993.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Chanho Kim, Suhas V Sumukh, and Li Fuxin. Learning a particle dynamics model with real-world videos. *arXiv preprint arXiv:2605.23845*, 2026.

XY Kou and Sooi Thor Tan. Heterogeneous object modeling: A review. *Computer-Aided Design*, 39(4):284–301, 2007.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. Cowtracker: Tracking by warping instead of correlation. *arXiv preprint arXiv:2602.04877*, 2026.

Hoang Le, Joonwoo Kwon, Elkhan Ismayilzada, Yufei Zhang, and Zijun Cui. Siphy: Single-image physical property reasoning. *arXiv preprint arXiv:2607.22355*, 2026.

Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. Pixie: Fast and generalizable supervised learning of 3d physics from pixels. *arXiv preprint arXiv:2508.17437*, 2025.

Can Li, Zhoujian Li, Ren Li, Jie Gu, Lei Lei, Jingmin Chen, and Lei Sun. Deformmaster: An interactive physics-neural world model for deformable objects from videos. *arXiv preprint arXiv:2605.09586*, 2026.

Yichen Li, Zhiyi Li, Brandon Feng, Dinghuai Zhang, and Antonio Torralba. Generalized dynamics generation towards scannable physical world model. *arXiv preprint arXiv:2510.15041*, 2025.

Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. In *International Conference on Learning Representations*, volume 2025, pp. 70160–70187, 2025.

Zili Lin, Wenyao Zhang, Yuyang Zhang, Zekun Qi, Junyan Lin, Hanxin Zhu, Jiaolong Yang, Zhibo Chen, Yao Mu, Xiaokang Yang, et al. Deformgen: Dynamics-based topology augmentation for deformable manipulation policy learning. *arXiv preprint arXiv:2606.25939*, 2026.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. PARTFIELD: Learning 3d feature fields for part segmentation and beyond. *arXiv preprint arXiv:2504.11451*, 2025a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Yu Liu, Baoxiong Jia, Ruijie Lu, Chuyue Gan, Huayu Chen, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Videoartgs: Building digital twins of articulated objects from monocular video. *arXiv preprint arXiv:2509.17647*, 2025b.

Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Building interactable replicas of complex articulated objects via gaussian splatting. In *International Conference on Learning Representations*, volume 2025, pp. 78444–78467, 2025c.

Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/NVIDIA/warp, March 2022. NVIDIA GPU Technology Conference (GTC).

Diana Mateus and Radu Horaud. Spectral methods for 3-d motion segmentation of sparse scene-flow. In *2007 IEEE Workshop on Motion and Video Computing (WMVC'07)*, pp. 14–14. IEEE, 2007.

Thomas Müller. tiny-cuda-nn, 4 2021. URL https://github.com/NVlabs/tiny-cuda-nn.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Shivansh Patel, Kaifeng Zhang, Sanjay Pokkali, Svetlana Lazebnik, and Yunzhu Li. Learning physics-guided residual dynamics for deformable object simulation. *arXiv preprint arXiv:2607.13451*, 2026.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Licheng Shen, Saining Zhang, Honghan Li, Peilin Yang, Zihao Huang, Zongzheng Zhang, and Hao Zhao. Gaussianart: Unified modeling of geometry and motion for articulated objects. In *2026 International Conference on 3D Vision (3DV)*, pp. 384–395. IEEE, 2026.

Yinghao Shuai, Ran Yu, Yuantao Chen, Zijian Jiang, Xiaowei Song, Nan Wang, Jv Zheng, Jianzhu Ma, Meng Yang, Zhicheng Wang, et al. Pugs: Zero-shot physical understanding with gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4478–4485. IEEE, 2025.

Boyuan Wang, Xiaofeng Wang, Yongkang Li, Zheng Zhu, Yifan Chang, Angen Ye, Guosheng Zhao, Chaojun Ni, Guan Huang, Yijie Ren, et al. Reconphys: Reconstruct appearance and physical attributes from single video. In *European Conference on Computer Vision*, pp. 492–509. Springer, 2026.

Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5249–5260. IEEE, 2025.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320, June 2024.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389–4398. IEEE, 2024.

Qingshan Xu, Jiao Liu, Shangshu Yu, Yuxuan Wang, Yuan Zhou, Junbao Zhou, Jiequan Cui, Yew-Soon Ong, and Hanwang Zhang. Neuspring: Neural spring fields for reconstruction and simulation of deformable objects from videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11361–11369, 2026.

Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025a.

Yang Yang, Yiyan Wang, Zheming Liu, and Naoya Iwamoto. Matphys: Learning material-aware physics parameters for deformable object simulation from videos. *arXiv preprint arXiv:2605.19386*, 2026.

Yu Yang, Zhilu Zhang, Xiang Zhang, Yihan Zeng, Hui Li, and Wangmeng Zuo. Physworld: From real videos to world models of deformable objects via physics-aware demonstration synthesis. *arXiv preprint arXiv:2510.21447*, 2025b.

Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28296–28305. IEEE, 2024.

Guanqi Zhan, Xianzheng Ma, Weidi Xie, and Andrew Zisserman. Inferring dynamic physical properties from video foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5214–5224, 2026.

Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. *arXiv preprint arXiv:2506.15680*, 2025a.

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. *arXiv preprint arXiv:2511.04665*, 2025b.

Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. *arXiv preprint arXiv:2410.18912*, 2024.

Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In *European Conference on Computer Vision*, pp. 407–423. Springer, 2024.

APPENDIX

A Data Collection and Processing **15**

A.1 Capture Setup 15

A.2 Objects and Interactions 15

A.3 Data Preprocessing 16

B Implementation Details **17**

B.1 Part-Level Material Recognition 17

B.2 Motion-Derived Rigidity Identification 18

B.3 Rigidity-Adaptive Spring Topology 18

B.4 Neural Oriented Stiffness Field 18

B.5 Physics Optimization and Simulation 19

C Experimental Details **19**

C.1 Training Configuration 19

C.2 Baselines 19

C.3 Evaluation Metrics 20

D Additional Experimental Results **20**

D.1 Ablations on the PhysTwin Benchmark 20

D.2 Alternative Material Priors 21

D.3 Generalization to Unseen Interactions 22

D.4 Additional Qualitative Comparisons 22

A Data Collection and Processing

A.1 Capture Setup

We capture the 11 self-captured articulated and heterogeneous interaction scenarios using three Intel RealSense D455 cameras arranged around a tabletop workspace, as shown in Fig. 10. We calibrate the three cameras and transform their depth observations into a shared coordinate system for multi-view geometry processing and 3D trajectory lifting.

For higher-quality depth acquisition, following the configuration in GS-Physics (Kim et al., 2026), we disable the D455 infrared emitters and feed the synchronized left and right infrared images into FoundationStereo (Wen et al., 2025) to estimate depth in the left infrared camera frame. The estimated depth is then aligned to the RGB frame using the calibrated camera intrinsics and extrinsics, producing smoother and cleaner depth observations than the native D455 depth.

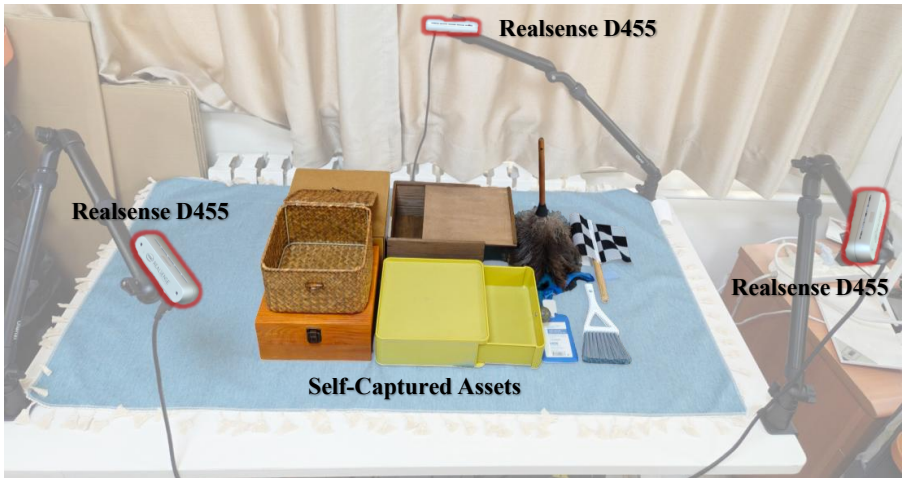


Figure 10: **Self-captured objects and multi-view capture setup.** Three Intel RealSense D455 cameras observe the tabletop workspace from left, right, and elevated oblique views. The center shows the articulated and heterogeneous objects used in our 11 self-captured scenarios.

A.2 Objects and Interactions

Deformable Objects. The seven PhysTwin (Jiang et al., 2025) scenarios involve sloth and zebra toys under single- and double-hand lifting, stretching, and pushing.

Articulated Objects. The five self-captured articulated scenarios involve closing drawers and folding cases, with translational and rotational motion between rigid components.

Heterogeneous Objects. The six self-captured heterogeneous scenarios cover card holders with lanyards, flags with supporting sticks, a feather duster, and a broom. These materially heterogeneous objects combine rigid components with flexible lanyards, cloth, feathers, or bristles.

Table 3: **Deformable object scenarios.** Seven deformable-object scenarios adopted from the PhysTwin benchmark.

Object	Scenario
Sloth	`double_lift_sloth`
Sloth	`double_stretch_sloth`
Sloth	`single_lift_sloth`
Sloth	`single_push_sloth`
Zebra	`double_lift_zebra`
Zebra	`double_stretch_zebra`
Zebra	`single_lift_zebra`

Table 4: **Articulated object scenarios.** Five self-captured articulated-object scenarios involving drawers and cases.

Object	Scenario
Drawer	`double_close_drawer_10`
Drawer	`double_close_drawer_11`
Case	`single_fold_case_03`
Case	`single_fold_case_08`
Case	`single_fold_case_09`

Table 5: **Heterogeneous object scenarios.** Six self-captured heterogeneous scenarios featuring objects composed of parts with different materials.

Object	Scenario
Card	`double_lift_card_01`
Card	`single_lift_card_05`
Flag	`single_lift_flag_03`
Flag	`single_lift_flag_05`
Duster	`single_push_duster_03`
Broom	`single_push_broom_01`

A.3 Data Preprocessing

Following the data preprocessing pipeline of PhysTwin (Jiang et al., 2025), we tailor several implementation details to our self-captured datasets. Given the synchronized multi-view RGB-D observations, the preprocessing pipeline reconstructs the initial geometry $\mathcal{P}^0$ and static Gaussian representation $\mathcal{A}^0$, extracts per-frame partial observations $\mathcal{O}^t$, lifts part-level object tracks $\mathcal{Q}^t$ into 3D, and estimates controller trajectories $\mathcal{C}^t$.

First-Frame Initialization. Following PhysTwin, we use generative reconstruction to complete missing geometry. Unlike PhysTwin, which applies TRELLIS (Xiang et al., 2025) to a single first-frame view, we use three views with ReconViaGen (Chang et al., 2026) to obtain a multi-view consistent shape prior $\mathcal{P}^0$ for surface and interior point sampling. We then initialize a static 3D Gaussian representation $\mathcal{A}^0$ from $\mathcal{P}^0$ and optimize it against the three RGB views for appearance rendering during simulation (Kerbl et al., 2023).

Object and Controller Segmentation. Following PhysTwin, we obtain temporally consistent object and hand masks for each view. Grounding DINO (Liu et al., 2023) first detects the object and hands in the initial RGB frame using text prompts. The resulting bounding boxes are then used to initialize SAM2 (Ravi et al., 2025), which segments and propagates the corresponding masks throughout the video.

Object and Controller Tracking. Following PhysTwin, we use CoTracker3 (Karaev et al., 2025) to obtain pseudo tracks for supervision. Points sampled within the first-frame masks are tracked in 2D, lifted to 3D, and filtered for inconsistent motion, yielding object tracks $\mathcal{Q}^t$ and controller trajectories $\mathcal{C}^t$. For weakly textured or thin structures in our self-captured data, such as the card in Fig. 11, CoTracker3 may suffer from track loss. We therefore use CowTracker (Lai et al., 2026) for dense pixel tracking and convert its outputs to the same format for more reliable 3D trajectories.

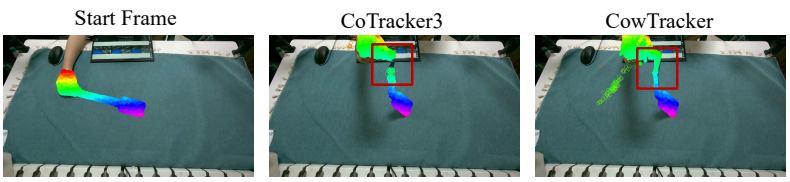


Figure 11: **CoTracker3 vs. CowTracker.** CowTracker better preserves tracks on weakly textured or thin structures, while occasional drift in hand tracks can be removed by mask filtering.

In addition to the neighborhood motion-consistency and controller-track filtering in PhysTwin (Jiang et al., 2025), we further apply a part-aware cross-part intrusion filter. A track is truncated once it enters a local neighborhood dominated by valid tracks from another part, suppressing cross-part jumps caused by depth and mask errors.

B Implementation Details

B.1 Part-Level Material Recognition

Part-Level Visual Stitches. We cluster PartField features (Liu et al., 2025a) on 100,000 points uniformly sampled from $\mathcal{P}^0$ and transfer the resulting part labels to the static Gaussians in $\mathcal{A}^0$ by nearest-neighbor assignment. For each part, we render a *Part-Level Visual Stitch* composed of two rows: the top row highlights the target part in semi-transparent red from three viewpoints, while the bottom row shows enlarged close-up views. Representative examples are shown in Fig. 12.

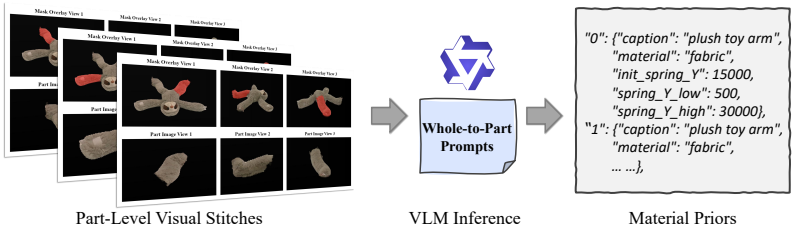


Figure 12: **Part-Level Material Recognition.** Part-level visual stitches are fed to Qwen-VL (Qwen Team, 2026) with whole-to-part prompting to infer part-level material and stiffness priors.

VLM Inference. The 3DGS rendered Part-Level Visual Stitches are then fed to Qwen-VL (Qwen Team, 2026) with whole-to-part prompting, which first identifies the whole object and then predicts each part's caption, material type, and stiffness parameters. The output fields `init_spring_Y`, `spring_Y_low`, and `spring_Y_high` specify k_p^0, k_p^-, and k_p^+ for the intra-part springs of part p, respectively. The whole-to-part prompts are provided below.

Whole-Object Recognition Prompt.

Focus ONLY on the top row of the image which shows the full object from multiple viewpoints.
Identify what this main whole object is (e.g., 'plush zebra toy', 'wooden dining chair', 'ceramic teapot').
Reply with ONLY the concise name of the whole object in lower case, and absolutely no other words, punctuation, or formatting.

In the following, {`global_object_name`} is replaced with the whole-object recognition result.

Part-Level Material and Stiffness Prompt.

You are a world-class Physical Material Expert specializing in system identification for physical simulations.
CONFIRMED GLOBAL CONTEXT: The primary object being analyzed in this scene has been identified as: {`global_object_name`}.
Image Structure:
The top row ("Mask Overlay View X") shows the FULL '{`global_object_name`}' with ONE candidate part highlighted in semi-transparent red.
The bottom row ("Part Image View X") shows a zoomed-in, cropped close-up of ONLY that highlighted part.
Task Instructions: Follow this step-by-step reasoning process to determine the physics parameters for the highlighted part:
Step 1 [Part Identification]: Identify what specific sub-part of the '{`global_object_name`}' is highlighted in red. Combine them into a precise caption (e.g., '{`global_object_name`} leg', '{`global_object_name`} backrest', '{`global_object_name`} handle'). Do NOT use vague terms like 'part' or 'limb'.

Step 2 [Material Analysis]: Inspect the surface texture, specular reflections, edges, and visual appearance in both top and bottom rows. Select the SINGLE most accurate material from the library: {foam, fabric, leather, rubber, plastic, paper, wood, metal, glass, ceramic}.
Step 3 [Stiffness Calibration]: Determine the spring stiffness coefficient (`init_spring_Y`) reflecting bending/deformation compliance based on the geometry and material.

[Material Types & Stiffness Ranges]:				
	foam:	1000–10000	fabric:	1000–30000
	leather:	1000–30000	rubber:	1000–30000
	plastic:	20000–80000	paper:	20000–80000
	wood:	60000–80000	metal:	60000–100000
	glass:	60000–100000	ceramic:	60000–100000

[STRICT BOUNDARY CONSTRAINTS]:
1. `spring_Y_low` and `spring_Y_high` MUST EXACTLY match the empirical range listed above for the chosen material. Do NOT alter these boundary numbers.
2. `init_spring_Y` must be a valid integer bounded by [`spring_Y_low`, `spring_Y_high`] depending on thickness/structure.
Format Requirement: Briefly state your reasoning for Steps 1–3. At the very end of your response, output the final tuple strictly inside angle brackets on a NEW LINE:
`<caption, material, init_spring_Y, spring_Y_low-spring_Y_high>`

B.2 Motion-Derived Rigidity Identification

Kabsch alignment in Sec. 3.2 uses only track correspondences $(\mathbf{q}_i^0, \mathbf{q}_i^t)$ that are valid in both the reference and current frames. Frames with fewer than three common valid tracks retain their unaligned coordinates for residual computation. We further retain only tracks whose valid-frame ratio exceeds $\rho_{\text{valid}} = 0.8$. For the resulting N_p tracks, we initialize $\mathbf{Z}_p \in \mathbb{R}^{N_p \times 3T_{\text{obs}}}$ to zero and fill valid entries with the corresponding residuals $\mathbf{r}_i^t$. In implementation, we compute e_p by applying Principal Component Analysis (PCA) to $\mathbf{Z}_p$, with tracked points as samples and residual trajectories as features, and summing the explained variances of all principal components:

$$e_p = \sum_k \lambda_{p,k} = \operatorname{tr}(\operatorname{Cov}(\mathbf{Z}_p)) .$$

B.3 Rigidity-Adaptive Spring Topology

For long-range spring topology in Sec. 3.2, we set the normalized distance bounds to $\beta^- = 0.4$ and $\beta^+ = 0.4005$, corresponding to candidate connections within $[\beta^- L_p, \beta^+ L_p]$ for part p. We use a rigidity threshold of $\epsilon_{\text{rigid}} = 0.01$ to determine when long-range spring augmentation is applied.

B.4 Neural Oriented Stiffness Field

Directional Encoding and Field Architecture. We implement each field with tinycudann (Müller, 2021), mapping canonical directions to $[0,1]^3$ as $\mathbf{u}_e^t = \operatorname{clamp}((\tilde{\mathbf{d}}_e^t + \mathbf{1})/2, 0, 1)$. The hash-grid encoder uses 12 levels, 2 features per level, hash-table size 2^{16}, base resolution 8, and per-level scale 1.5. A two-layer MLP with 64 hidden units and ReLU activations outputs six coefficients (a, b, c, d, e, f) defining a symmetric residual tensor

$$\Delta\mathbf{K}_e^t = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}, \qquad \Delta k_e^t = (\tilde{\mathbf{d}}_e^t)^\top \Delta\mathbf{K}_e^t \tilde{\mathbf{d}}_e^t. \tag{9}$$

Zero Residual Initialization. Since the randomly initialized hash grid and MLP may produce a nonzero directional residual at the initial frame, we introduce an edge-wise offset to preserve the coarse zero-order solution:

$$o_e = -\Delta k_e^0, \qquad \Delta k_e^t \leftarrow \Delta k_e^t + o_e, \tag{10}$$

such that $\Delta k_e^0 = 0$ and the field initially recovers the coarse isotropic stiffness k_{pq}^{cma}.

Sigmoid-Based Bounded Decoding. We modulate the coarse isotropic stiffness k_{pq}^{cma} with the directional residual Δk_e^t after offset correction in bounded log space. Given the stiffness bounds (k_{pq}^-, k_{pq}^+), k_{pq}^{cma} is first encoded into the sigmoid domain as

$$\xi_{pq} = \frac{\log k_{pq}^{\mathrm{cma}} - \log k_{pq}^-}{\log k_{pq}^+ - \log k_{pq}^-}, \qquad z_{pq} = \mathrm{logit}(\mathrm{clamp}(\xi_{pq}, \varepsilon, 1-\varepsilon)) . \tag{11}$$

The final directional stiffness is then decoded as

$$k_e^t = \exp\left[\log k_{pq}^- + \sigma\left(z_{pq} + \Delta k_e^t\right)\left(\log k_{pq}^+ - \log k_{pq}^-\right)\right] . \tag{12}$$

Decoupled Gradient Propagation. We decouple the Neural Oriented Stiffness Field from the Warp-based differentiable simulation graph (Macklin, 2022). Canonical spring directions are computed without gradients, and predicted directional stiffnesses are detached before being passed to Warp. After simulation, gradients of the observation loss with respect to these stiffnesses are transferred back to the corresponding field outputs to update the field parameters, while the Warp-computed gradients of inter-part spring stiffnesses are discarded.

B.5 Physics Optimization and Simulation

Particle-to-Gaussian Motion Transfer. For realistic rendering during simulation, we animate the initial static Gaussian representation $\mathcal{A}^0$ using Linear Blend Skinning (LBS) (Huang et al., 2024a; Zhang et al., 2024) driven by the simulated particles. At each simulation step, neighboring particles define local rigid motions through their positions $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t+1}$ and local rotations $\mathbf{R}_i^t$ estimated from neighborhood deformation. For each Gaussian g, nearby particles are assigned normalized inverse-distance weights w_{gi}^t, and its center is updated incrementally as

$$\begin{aligned} w_{gi}^t &= \frac{(\|\boldsymbol{\mu}_g^t - \mathbf{x}_i^t\|_2 + \varepsilon)^{-1}}{\sum_{j\in\mathcal{N}_g^t}(\|\boldsymbol{\mu}_g^t - \mathbf{x}_j^t\|_2 + \varepsilon)^{-1}}, \\ \boldsymbol{\mu}_g^{t+1} &= \sum_{i\in\mathcal{N}_g^t} w_{gi}^t \left[\mathbf{R}_i^t(\boldsymbol{\mu}_g^t - \mathbf{x}_i^t) + \mathbf{x}_i^{t+1}\right], \end{aligned} \tag{13}$$

where $\mathcal{N}_g^t$ denotes the neighboring particles of Gaussian g at frame t. Gaussian orientations are updated using the corresponding blended local rotations, while appearance attributes are retained throughout the motion transfer.

C Experimental Details

C.1 Training Configuration

All experiments are conducted on a single NVIDIA L40 GPU using the two-stage optimization scheme described in Sec. 3.4. The zero-order stage uses CMA-ES to search over graph topology and coarse physical parameters for 20 iterations. With the resulting topology fixed, differentiable refinement jointly optimizes the physical parameters and Neural Oriented Stiffness Fields for 200 iterations using Adam with a learning rate of 10^{-3}. During forward rollout, the Warp-based simulator uses 667 substeps per video frame with a time step of 5×10^{-5} s. Across all experiments, only the training split is used for optimization, with the test split reserved for future prediction.

C.2 Baselines

Following the experimental setup of PhysTwin, we further evaluate GS-Dynamics (Zhang et al., 2024) and Spring-Gaus (Zhong et al., 2024) on our dataset under the same configurations. For articulated reconstruction baselines, we adapt ArtGS (Liu et al., 2025c) and GaussianArt (Shen et al., 2026) to our self-captured scenarios. Both methods receive three-view observations of the initial state and the final training state, as required by their two-state reconstruction pipelines. Below, we detail the training configurations and evaluation protocol of these articulated baselines.

ArtGS Training. We convert the five articulated scenarios into paired endpoint datasets. For a training split containing T_{tr} frames, the two endpoints correspond to frames 0 and $T_{\text{tr}} - 1$, each containing three calibrated RGB-D views. Following the original ArtGS pipeline (Liu et al., 2025c), single-state Gaussians are independently reconstructed at the two training endpoints and matched to initialize the coarse canonical Gaussians. ArtGS then performs a warm-up stage to predict the joint type, followed by joint optimization of the canonical Gaussians and articulation model under the predicted joint-type constraint.

GaussianArt Training. GaussianArt additionally uses depth maps and part segmentation masks at the two training endpoints. We obtain semantic maps by backprojecting depth pixels and assigning each pixel the label of its nearest object point. We then follow the original warm-up, soft-training, and hard-training stages of GaussianArt (Shen et al., 2026). For our scenarios, the part with the smallest median displacement between the two training endpoints is treated as the base. The rotational parameters of translational parts are fixed during optimization.

Articulated Baseline Evaluation. Let T denote the full sequence length and T_{tr} the number of training frames. ArtGS and GaussianArt reconstruct each frame by interpolating their learned articulation at the normalized time $s_t = t/(T_{\text{tr}}-1)$. Frames within the training interval are reconstructed by interpolating the learned articulation. Future test frames satisfy $s_t > 1$ and are predicted by extrapolating the same learned articulation beyond the final training state. For geometric and tracking evaluation, each initial reference point is associated with its nearest Gaussian center in the initial state, with this association fixed to obtain per-frame trajectories. For rendering evaluation, images are rendered from the evaluation cameras at the corresponding normalized articulation coordinates.

Fundamentally, the articulated baselines perform endpoint conditioned interpolation within the observed training interval and extrapolate their learned articulation during future prediction, whereas GenTwin performs controller conditioned forward simulation with learned object dynamics.

C.3 Evaluation Metrics

Chamfer Distance. We measure geometric error using the mean one-sided L_1 distance from visible observed points $\mathcal{O}^t_{\text{vis}} \subseteq \mathcal{O}^t$ to simulated surface particles:

$$\text{CD}^t = \frac{1}{|\mathcal{O}^t_{\text{vis}}|} \sum_{\mathbf{o} \in \mathcal{O}^t_{\text{vis}}} \min_{i \in \mathcal{V}_{\text{surf}}} \|\mathbf{o} - \mathbf{x}^t_i\|_1, \tag{14}$$

where $\mathcal{V}_{\text{surf}} \subseteq \mathcal{V}$ indexes the surface particles. This evaluation metric uses L_1 distance, whereas the geometric loss $\mathcal{L}_{\text{geo}}$ uses squared Euclidean distance.

Tracking Error. We measure motion error using the mean Euclidean distance between valid reference tracks and their associated simulated particles. Each track is assigned to its nearest particle in the initial frame, with the assignment $\pi(i)$ fixed throughout evaluation:

$$\text{Track}^t = \frac{1}{|\mathcal{I}^t|} \sum_{i \in \mathcal{I}^t} \|\mathbf{x}^t_{\pi(i)} - \mathbf{q}^t_i\|_2, \tag{15}$$

where $\mathcal{I}^t$ indexes reference tracks with finite target coordinates at frame t. Evaluation uses Euclidean distance, whereas the tracking loss uses coordinate-averaged smooth-L_1. Frames with no valid correspondences are assigned zero tracking error.

Final Reports. For both metrics, we first average per-frame values within each evaluation segment, then average the resulting scenario-level values within each object category.

D Additional Experimental Results

D.1 Ablations on the PhysTwin Benchmark

Table 6 reports additional ablation results for our *w/o material* and *w/o anisotropy* variants on the seven PhysTwin scenarios with multi-part deformable objects.

Table 6: **Ablation results on the PhysTwin benchmark.** Results are averaged over seven scenarios.

Variant	Reconstruction & Resimulation						Future Prediction					
	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
w/o material	0.006	0.009	0.860	27.553	0.950	0.024	0.013	0.024	0.716	24.607	0.938	0.052
w/o anisotropy	0.005	**0.007**	0.860	27.571	0.950	**0.023**	0.012	0.021	0.721	24.657	0.938	0.053
Full model	**0.004**	**0.007**	**0.865**	**27.687**	**0.951**	**0.023**	**0.010**	**0.020**	**0.745**	**25.097**	**0.941**	**0.048**

Table 7: **Quantitative comparison of alternative material priors.** We evaluate GenTwin using material priors predicted by Pixie and VoMP on the 11 self-captured scenarios.

Method	Reconstruction & Resimulation						Future Prediction					
	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓	CD↓	Track↓	IoU↑	PSNR↑	SSIM↑	LPIPS↓
Pixie	0.010	0.014	0.787	28.187	0.966	0.029	0.013	0.019	0.710	26.118	0.964	0.037
VoMP	0.009	0.013	0.797	28.428	0.968	0.027	0.013	0.020	0.688	26.164	0.964	0.039
GenTwin (Ours)	**0.007**	**0.010**	**0.838**	**28.934**	**0.969**	**0.026**	**0.009**	**0.015**	**0.777**	**27.328**	**0.967**	**0.033**

D.2 Alternative Material Priors

To evaluate alternative sources of part-level material priors, we replace the material priors from Qwen-VL with two recent feed-forward 3D physical property estimators, Pixie (Le et al., 2025) and VoMP (Dagli et al., 2026). For each method, the predicted Young's modulus distribution within each part is converted into part-level spring-stiffness priors for the same optimization pipeline as GenTwin. We evaluate both alternatives on the eleven self-captured scenarios. Quantitative results are reported in Table 7, with qualitative comparisons shown in Fig. 13.

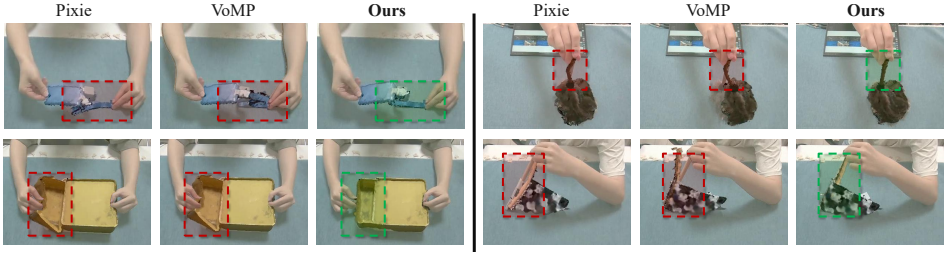


Figure 13: **Qualitative comparison of alternative material priors.** Results on four self-captured scenarios reveal discrepancies between predicted Young's modulus and observed physical behavior.

Pixie Integration. Given calibrated multi-view RGB images from the initial state, Pixie (Le et al., 2025) predicts a voxelized 3D material field with spatially varying Young's modulus. We transform the field into our world coordinate system and transfer modulus values to part-labeled physical points by nearest-neighbor interpolation. For each part, finite positive predictions are aggregated by confidence: the weighted geometric mean defines the modulus initialization, while the weighted 10th and 90th percentiles define its bounds.

VoMP Integration. Given the initial static Gaussian representation, VoMP (Dagli et al., 2026) voxelizes the geometry and aggregates multi-view DINOv2 features to predict per-voxel mechanical properties. We transfer the predicted Young's modulus values to part-labeled physical points by nearest-neighbor interpolation. For each part, the median defines the modulus initialization, while the 10th and 90th percentiles define its bounds.

Modulus to Stiffness. For our spring-mass optimization, we convert the estimated Young's modulus to spring stiffness using a fixed logarithmic mapping over $[k_-, k_+] = [1000, 100000]$:

$$g(E) = k_- + (k_+ - k_-)\,\mathrm{clip}\left(\frac{\log_{10} E - L_-}{L_+ - L_-}, 0, 1\right), \tag{16}$$

where L_- and L_+ denote the lower and upper bounds of $\log_{10} E$. The modulus initialization and bounds of each part are mapped to obtain the corresponding stiffness prior (k_p^0, k_p^-, k_p^+).

D.3 Generalization to Unseen Interactions

Following PhysWorld (Yang et al., 2025b), we evaluate rendered rollouts on VBench (Huang et al., 2024b) in aesthetic quality, motion smoothness, and subject consistency. GenTwin outperforms PhysTwin across all three dimensions (Table 8).

Table 8: **Quantitative comparison of unseen interactions.** Scores are averaged over four unseen interaction rollouts. Higher values indicate better performance for all metrics.

Method	Aesthetic Quality ↑	Motion Smoothness ↑	Subject Consistency ↑
GenTwin (Ours)	**0.4342**	**0.9972**	**0.9312**
PhysTwin	0.4163	0.9971	0.9272

D.4 Additional Qualitative Comparisons

We provide additional qualitative comparisons on the articulated objects, PhysTwin benchmark, and heterogeneous objects in Figs. 14, 15, and 16, respectively. We recommend referring to the supplementary videos for more intuitive visual comparisons.

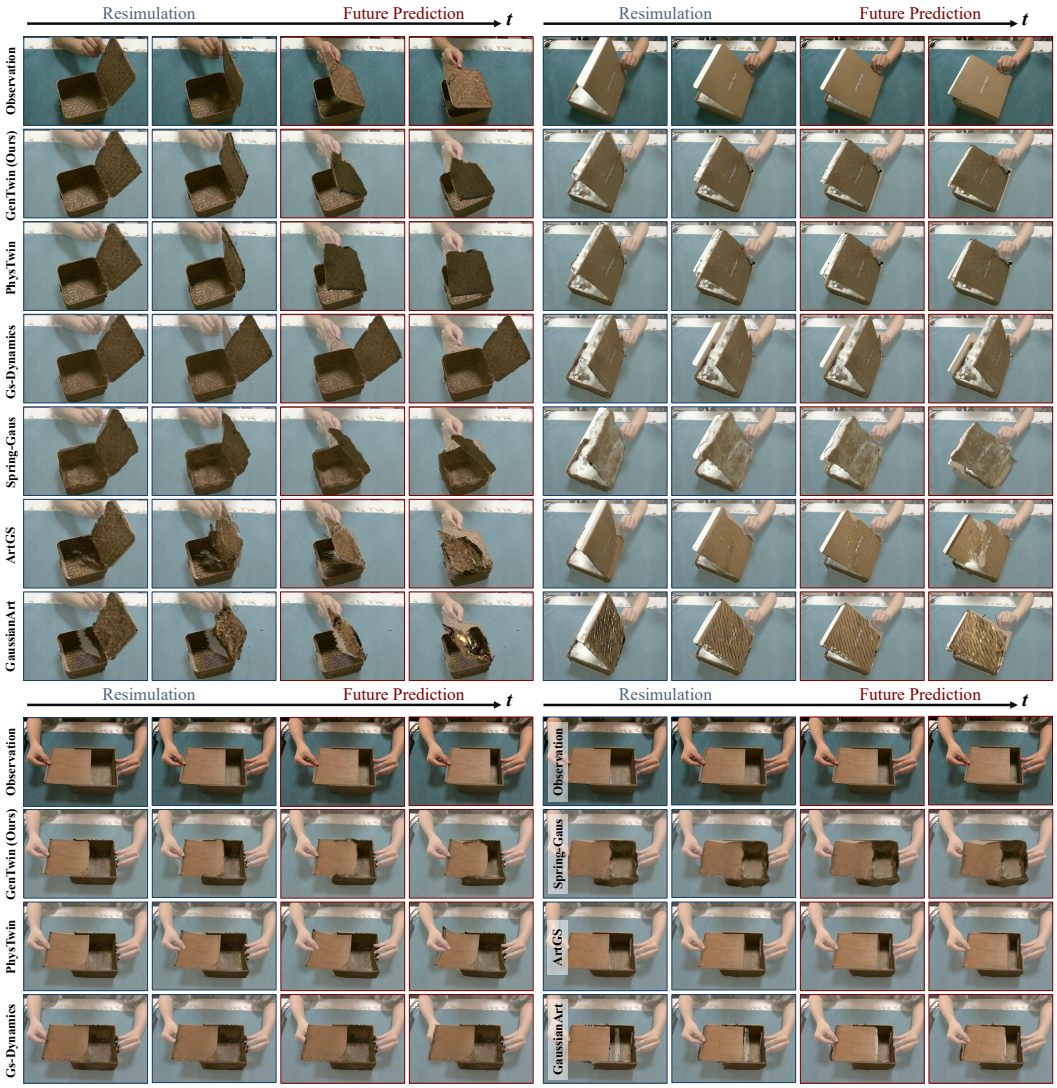


Figure 14: **Additional qualitative comparisons on articulated objects.** GS-Dynamics and Spring-Gaus are additionally included beyond the methods compared in the main paper.

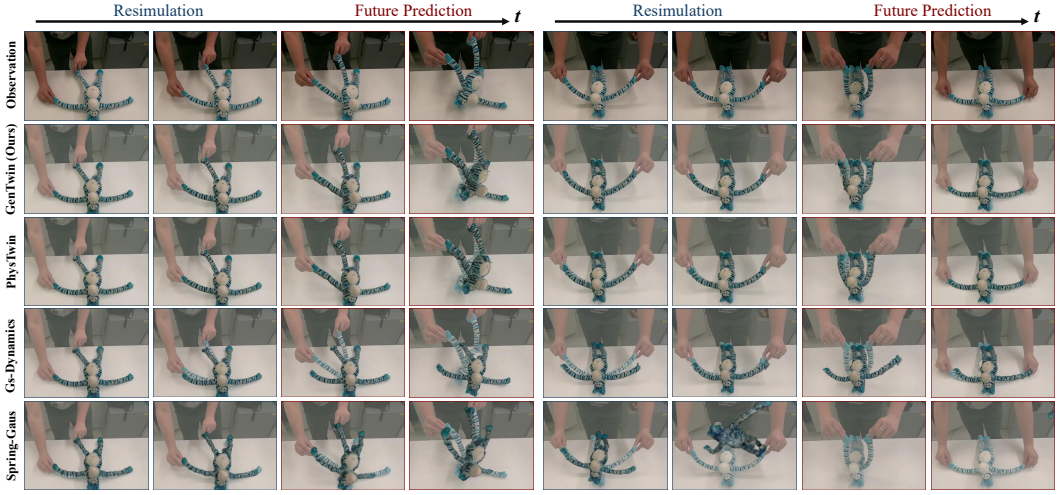


Figure 15: **Additional qualitative comparisons on the PhysTwin benchmark.**



Figure 16: **Additional qualitative comparisons on heterogeneous objects.**