# MORPHTWIN: DISCOVERING AND MORPHING MATERIAL RESPONSES FROM VIDEOS FOR INTERACTIVE PHYSICAL TWINS

**Anonymous authors**
Paper under double-blind review

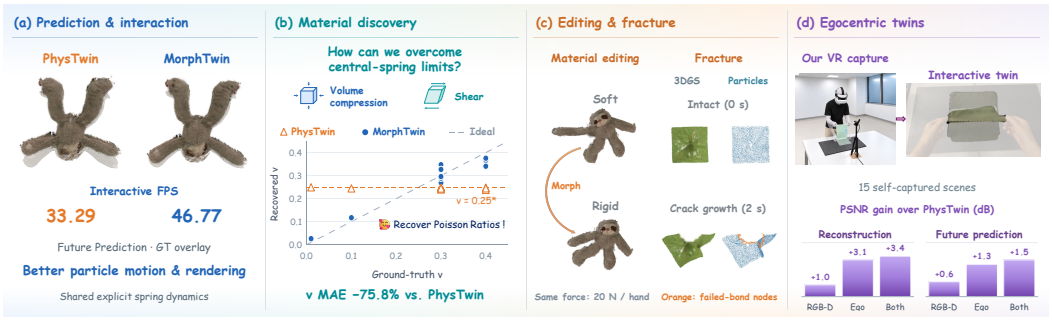


Figure 1: **MorphTwin discovers and morphs material responses from sparse videos for interactive physical twins.** We combine efficiency, accuracy, and explicit material control. ***(a) Prediction and interaction.*** MorphTwin improves future prediction and interactive throughput over PhysTwin (Jiang et al., 2025) within a shared explicit spring-graph framework. ***(b) Material discovery.*** Bulk-shear coupling limits material recovery in conventional spring models. MorphTwin relaxes this coupling for more accurate recovery. ***(c) Material morphing.*** Calibrated spring graphs support soft-to-rigid editing and fracture evolution, visualized through Gaussian rendering and simulation particles. ***(d) Downstream applications.*** Our self-captured egocentric and RGB-D observations support reconstruction and interaction, with complementary views improving visual quality.

## ABSTRACT

Reconstructing interactive physical twins from videos holds promise for virtual reality and embodied intelligence, but existing methods face a trade-off among simulation efficiency, material modeling accuracy, and explicit material control. Spring-mass models support efficient explicit simulation through simple discrete structures, providing an attractive foundation for real-time, interactive physical twins. However, their native spring parameters depend on graph structure, hindering cross-object reuse, while bulk-shear coupling in conventional central-force springs limits material expressiveness. Our key insight is to represent material behavior through standardized responses to volume and shape changes, separating a reusable material description from differences in spring geometry and connectivity across objects. Based on this insight, we propose **MorphTwin**, which augments explicit spring graphs with internally relaxable response carriers and performs offline energy calibration to compile shared material-response changes into graph-specific stiffness adjustments. This design supports material identification, interpretable editing, and cross-graph reuse while retaining efficient explicit dynamics. On synthetic scenes with known material parameters, MorphTwin reduces the mean absolute Poisson-ratio error and mean relative Young's-modulus error by 75.8% and 22.1%, respectively, compared with PhysTwin under the same inverse protocol. Real-world video experiments demonstrate improved reconstruction and future prediction and support reconstructing interactive physical twins from egocentric observations. With offline material calibration, online rollout advances the calibrated explicit spring graph without material identification or calibration optimization, preserving the real-time interaction capabilities of explicit spring-mass methods.



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.

# 1 INTRODUCTION

Reconstructing physical twins from videos aims to recover object geometry, appearance, and dynamics for simulation under novel interactions. Applications in virtual reality, embodied intelligence, and game-oriented world simulation require these twins to respond both realistically and quickly to actions such as pulling, squeezing, and grasping. Continuum approaches offer rich material models through solvers such as the Material Point Method (MPM) (Jiang et al., 2016; Xie et al., 2024; Lin et al., 2025), with methods such as GIC further recovering physically meaningful material parameters from videos (Cai et al., 2024). However, fine spatial discretization, repeated particle-grid transfers, and small time steps can make detailed MPM simulations costly for real-time interaction. Spring-mass systems offer an attractive foundation for interactive twins through their lightweight structure and explicit dynamics. PhysTwin (Jiang et al., 2025) combines these dynamics with visual reconstruction to enable real-time interaction with objects reconstructed from sparse videos, while subsequent developments further improve spring-based simulation efficiency (Jing et al., 2026; Pang et al., 2026).

This efficiency, however, does not ensure accurate or reusable material responses. Conventional spring models have limited flexibility in representing different deformation behaviors, such as how much an object contracts laterally when stretched. Moreover, fitting spring stiffnesses to observed motion yields responses that depend on node placement and spring connectivity. These graphs can vary across reconstructions. For example, material-aware graph construction may assign denser spring connections to softer regions and simpler connectivity to harder ones (Yang et al., 2026). Parameters that reproduce observed motion are therefore tied to a graph realization and do not necessarily capture material properties that remain consistent across object representations.

To serve as reusable physical assets, reconstructed physical twins must faithfully reproduce material behavior and support meaningful editing and transfer across objects. This leads to our central question: How can we extend spring-based models to recover accurate and reusable material responses from videos while preserving their efficiency for interactive simulation?

To address this question, we analyze the material expressiveness of three-dimensional spring-mass systems through elastic energy, using isotropic linear elasticity as a reference. Here, elastic energy separates into two parts: one associated with volume changes and the other with volume-preserving changes in shape. These are governed independently by the bulk and shear moduli. Accurately representing such materials therefore requires independent control of these two responses. However, an unprestressed three-dimensional central-force spring network with an isotropic response has a fixed bulk-to-shear modulus ratio under affine small strains. Within this regime, optimizing spring stiffnesses cannot remove this coupling. Improving material recovery therefore requires extending the spring model's ability to represent these two responses independently.

Our key insight is that carefully designed local spring structures can release this coupling by allowing internal nodes to move relative to their surrounding nodes. We call these structures *response carriers* and design their geometry to provide separate control over volumetric and shear responses while preserving standard central-force laws and explicit dynamics. Decoupling these responses, however, does not by itself ensure material reuse across objects: a carrier's effective response still depends on the local graph geometry. We therefore characterize carrier-augmented graphs by their relaxed elastic energies under standardized reference deformations, which provide common volumetric-shear response coordinates across different graphs. Graph-specific energy calibration then compiles shared material-response changes into native spring parameters, allowing different structures to realize material responses with consistent mechanical meaning.

Building on this formulation, we introduce ***MorphTwin***, a physical-twin framework for material discovery, editing, and cross-object reuse from videos. MorphTwin fits the carrier-augmented simulator to observed deformations to identify material responses in the shared response space. These responses can then be edited or transferred across objects through target-graph calibration, while online rollout remains an explicit spring simulation requiring no additional material identification or calibration optimization at inference time.

Our main contributions are summarized as follows:

**1) A physical-twin framework for material discovery and morphing from videos.** MorphTwin builds interactive physical twins from sparse RGB-D videos, combining efficiency, accuracy, and explicit material control to enable editing and cross-object reuse.

**2) Decoupling volumetric and shear responses in central-force spring dynamics.** We derive the effective elastic response of spring carriers with internal dynamic degrees of freedom and design local configurations for independent volumetric and shear control within standard central-force dynamics.

**3) Energy-aligned graph-specific material calibration.** We calibrate carrier-augmented graphs against shared reference-deformation energies, mapping material-level targets to native spring parameters with consistent mechanical meaning across structures.

**4) Experimental validation.** Controlled experiments demonstrate more accurate material recovery through separate bulk-shear control and better preservation of material responses across objects and discretizations. On the PhysTwin dataset, MorphTwin improves geometric accuracy and rendering quality over PhysTwin in reconstruction and future prediction while retaining efficient explicit simulation. We also collect 15 interaction sequences and temporally align their egocentric stereo and RGB-D observations. Evaluation demonstrates consistent rendering gains across single-view and sparse-view inputs, supporting egocentric reconstruction and interaction.

## 2 Related Work

**Video-based physical twins and interactive dynamics.** Recent methods reconstruct deformable physical twins from visual observations (Li et al., 2023; Cai et al., 2024; Jiang et al., 2025; Rho et al., 2026; Kim et al., 2026; Lee et al., 2026). PhysTwin (Jiang et al., 2025) combines Gaussian appearance with spring-mass dynamics. PGND (Zhang et al., 2025) and DeformMaster (Li et al., 2026) improve prediction through learned dynamics and residual corrections, while Boba (Pang et al., 2026) accelerates simulation with compact models and batched execution. Yet accurate motion fitting and efficient rollout do not ensure physically meaningful, reusable material parameters. MorphTwin identifies interpretable material responses while keeping material calibration offline and online simulation explicit for efficient interactive rollout.

**Material representations and cross-graph reuse.** Spring parameters depend on graph geometry and connectivity (Lloyd et al., 2008). MatPhys (Yang et al., 2026) learns shared material embeddings, while NeuSpring (Xu et al., 2026) models spring properties with neural fields. However, consistent parameter prediction does not guarantee consistent mechanical responses across graphs. Central springs also couple bulk and shear in unstressed, isotropic 3D networks under affine small strains. MorphTwin uses internally relaxable response carriers for separate bulk and shear control within standard central-force dynamics. Energy alignment maps shared response targets to graph-specific spring parameters for interpretable recovery, editing, and reuse across objects and discretizations. The appendix compares representative methods by their inputs, intermediate processing, and outputs.

## 3 Problem Formulation

**Problem setting.** To address our central question, we aim to combine accurate material modeling and efficient simulation with reuse across reconstructions. Continuum methods such as Finite Element Method (FEM) (Zienkiewicz et al., 2013) and MPM (Jiang et al., 2016) provide expressive constitutive models but can incur substantial computational costs at fine resolutions. Spring-mass systems support efficient explicit simulation, yet their material behavior depends on graph structure and can lack independent bulk-shear control. Given video interaction $\mathcal{V}$ and its Spring-Mass graph $\mathcal{G} = (\mathcal{P}, \mathcal{S})$, where $\mathcal{P}$ and $\mathcal{S}$ denote physical nodes and central-spring connections, we seek shared material-response coordinates that support accurate material identification and cross-graph reuse while retaining efficient explicit spring dynamics.

**Discretization dependence.** For an unstressed spring graph $\mathcal{G}$, the affine elastic energy under small strain $\boldsymbol{\epsilon}$ is $\mathcal{E}^{\text{aff}}(\boldsymbol{\epsilon}) = \frac{1}{2}\sum_{e\in\mathcal{S}} Y_e L_e (\mathbf{n}_e^\top \boldsymbol{\epsilon} \mathbf{n}_e)^2$, where $Y_e$, $L_e$, and $\mathbf{n}_e$ denote the spring coefficient, reference length, and unit direction of edge $e$. The same spring coefficients can produce different responses when geometry or connectivity changes. Material reuse requires aligning mechanical responses across graphs, with spring parameters calibrated per realization.

**Bulk-shear coupling of central springs.** Discretization dependence is accompanied by a limitation in material expressiveness. Poisson's ratio describes transverse contraction relative to axial extension under uniaxial tension, linking lateral deformation to volume change. In the unstressed, isotropic,

affine small-strain limit of a three-dimensional central-force network, directional averaging yields an effective elastic energy density. Comparing it with isotropic linear elasticity gives

$$W_{\text{central}} = \frac{c}{30}\left[(\operatorname{tr}\boldsymbol{\epsilon})^2 + 2\boldsymbol{\epsilon}:\boldsymbol{\epsilon}\right], \qquad W_{\text{iso}} = \frac{\lambda}{2}(\operatorname{tr}\boldsymbol{\epsilon})^2 + \mu\boldsymbol{\epsilon}:\boldsymbol{\epsilon} = \frac{\lambda}{2}\left[(\operatorname{tr}\boldsymbol{\epsilon})^2 + 2\frac{\mu}{\lambda}\boldsymbol{\epsilon}:\boldsymbol{\epsilon}\right], \tag{1}$$

where $\boldsymbol{\epsilon}$ is the small-strain tensor, $c$ denotes the effective network stiffness scale, and $\lambda, \mu$ are the Lamé parameters. Matching the coefficients of the two independent quadratic strain invariants yields

$$\lambda = \mu = \frac{c}{15}, \qquad \kappa = \lambda + \frac{2}{3}\mu = \frac{5}{3}\mu, \qquad \nu = \frac{3\kappa - 2\mu}{2(3\kappa + \mu)} = \frac{1}{4}, \tag{2}$$

where $\kappa, \nu$ are the bulk modulus and Poisson's ratio. This fixed ratio imposes the same relative transverse deformation across materials. Even with accurately fitted axial stiffness, the model cannot reproduce correct lateral deformation and volume change for materials with $\nu \neq 1/4$. Within this limit, learning additional edgewise stiffnesses cannot remove this coupling. MorphTwin therefore introduces internally relaxable response carriers for separate bulk and shear control, with graph-specific energy calibration for material reuse. These designs support expressive, reusable material responses while retaining efficient explicit spring dynamics.

## 4 METHODOLOGY

Figure 2 summarizes our pipeline. We first reconstruct interaction-specific physical twins (Sec. 4.1), then introduce response carriers and graph-specific energy calibration to identify shared material responses (Secs. 4.2-4.3), which support material editing and cross-object reuse (Sec. 4.4).

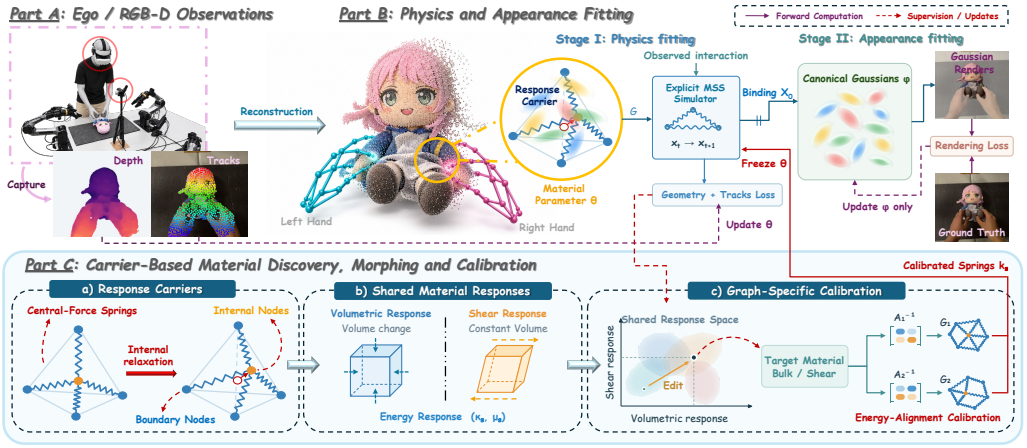


Figure 2: **Overview of MorphTwin.** *Part A:* Sparse-view observations provide geometry and motion cues. *Part B:* Physics fitting identifies material responses from deformations, followed by Gaussian appearance optimization with physics fixed. *Part C:* Relaxable response carriers enable separate bulk and shear control. Energy alignment maps shared material-response targets to graph-specific spring parameters, supporting interpretable material editing and cross-graph reuse. Offline material calibration preserves efficient online simulation through explicit spring dynamics.

### 4.1 PHYSICAL TWIN RECONSTRUCTION

Given sparse-view observations $\mathcal{V}$, we follow PhysTwin (Jiang et al., 2025) to initialize geometry and motion cues and construct an interaction-specific spring-mass graph $\mathcal{G} = (\mathcal{P}, \mathcal{S})$. Specifically, Grounded-SAM 2 (Liu et al., 2024; Ravi et al., 2025) provides object masks, CoTracker3 (Karaev et al., 2025) supplies point tracks that are lifted to 3D using depth, and TRELLIS (Xiang et al., 2025) provides a shape prior aligned to the initial observations. Spatial-neighbor connections define the spring topology. Graph sampling and connectivity may differ across interactions. We denote the dense backbone stiffnesses by $\mathbf{b}$ and other interaction-specific physical parameters by $\boldsymbol{\eta}$.

Given spring stiffnesses $\mathbf{Y}$, node $i$ evolves according to $m_i\ddot{\mathbf{x}}_i^t = \mathbf{f}_i^{\text{spr}}(\mathbf{X}^t;\mathbf{Y}) + \mathbf{f}_i^{\text{damp}}(\mathbf{X}^t,\dot{\mathbf{X}}^t;\boldsymbol{\eta}) + \mathbf{f}_i^{\text{int}}(\mathbf{X}^t;\boldsymbol{\eta}) + m_i\mathbf{g}$. Here, $\mathbf{X}^t$ stacks all node positions, while $m_i$, $\mathbf{g}$, and the three force terms

denote node mass, gravity, and elastic, damping, and interaction forces, respectively. Starting from the observed initial state, we explicitly integrate with time step $\Delta t$: $\mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i^t + \Delta t\, \mathbf{f}_i^t / m_i$ and $\mathbf{x}_i^{t+\Delta t} = \mathbf{x}_i^t + \Delta t\, \mathbf{v}_i^{t+\Delta t}$, yielding the differentiable trajectory $\widehat{\mathbf{X}} = \mathrm{Simulate}(\mathcal{G}, \mathbf{Y}, \boldsymbol{\eta})$.

We supervise $\widehat{\mathbf{X}}$ with the recovered 3D motion and optimize node-bound canonical 3D Gaussians (Kerbl et al., 2023) for appearance with physics fixed. We follow PhysTwin (Jiang et al., 2025) for the remaining reconstruction procedure, while the appearance binding used in our experiments is described separately in Appendix C.6. Because native spring parameters depend on graph realization, we next represent material behavior through graph-independent response coordinates.

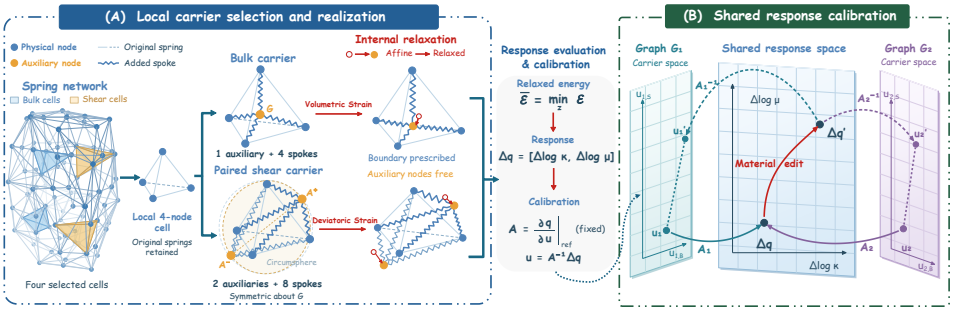


Figure 3: **Response-carrier realization and graph-specific calibration.** *Part (A):* Local four-node cells selected from the reconstructed spring graph are augmented with bulk and shear carriers. Under standardized volumetric and deviatoric probes, auxiliary nodes relax and the resulting energies define the graph response $\mathbf{q} = [\log \kappa, \log \mu]^\top$. *Part (B):* Each graph is locally calibrated by $\mathbf{A} = \partial \mathbf{q} / \partial \mathbf{u}|_{\mathrm{ref}}$. The shared coordinate $\Delta \mathbf{q}$ denotes a response change relative to that graph's own calibration reference and is mapped to carrier controls by $\mathbf{u} = \mathbf{A}^{-1} \Delta \mathbf{q}$.

### 4.2 Response-Carrier-Based Material Representation

In numerical simulation of elastic materials, the hybrid model (Golec et al., 2020) controls material responses using a prescribed lattice of regular cubic cells supplemented with a volumetric energy. Video-based reconstruction, however, produces spring graphs whose node placement and connectivity follow the observed object geometry, rather than a regular lattice. Imposing this cell structure, for example through uniform voxelization, introduces a geometry-efficiency trade-off: coarse grids can lose boundary detail and thin structures, while finer grids require more simulation nodes and springs.

MorphTwin introduces the required response freedom directly within the reconstructed graph by augmenting selected four-node cells with relaxable internal nodes, preserving the existing geometric discretization without requiring a regular volumetric mesh (Fig. 3(A)). A bulk carrier adds a centroid auxiliary node, while a paired shear carrier adds two symmetric auxiliary nodes on the circumsphere. The original springs are retained, and all auxiliary nodes follow the same explicit central-force dynamics, enabling separate volumetric and shear control through internal relaxation.

Each edge has a graph-specific base stiffness $Y_e(\mathbf{b}, \mathbf{0})$, with carrier spokes following the local backbone scale. A fixed label $\mathbf{w}_e \in \{(0,0), (1,0), (0,1)\}$ identifies backbone, bulk-carrier, or shear-carrier edges, while $\mathbf{u} = (u_B, u_S)^\top$ applies the corresponding log-stiffness offset:

$$\log Y_e(\mathbf{b}, \mathbf{u}) = \log Y_e(\mathbf{b}, \mathbf{0}) + \mathbf{w}_e^\top \mathbf{u}. \tag{3}$$

Thus, $\mathbf{b}$ remains jointly optimized, while $u_B$ and $u_S$ control the bulk-shear adjustments.

To measure the mechanical effect of these controls, we prescribe a small strain $\mathbf{E}$ on the physical nodes and relax the auxiliary degrees of freedom, giving the reduced energy derived in Appendix B.2:

$$\overline{\mathcal{E}}(\mathbf{E}; \mathbf{b}, \mathbf{u}) = \min_{\mathbf{z}} \frac{1}{2} \sum_{e=(i,j)\in\mathcal{S}} Y_e(\mathbf{b}, \mathbf{u}) L_e \left[ \mathbf{n}_e^\top \mathbf{E} \mathbf{n}_e + \frac{\mathbf{n}_e^\top (\mathbf{z}_j - \mathbf{z}_i)}{L_e} \right]^2. \tag{4}$$

Using the $1 + 5$ volumetric-deviatoric decomposition of symmetric 3D strain (Appendix B.3), we probe $\mathbf{E}_{\mathrm{vol}} = \mathbf{I}/\sqrt{3}$ and an orthonormal basis $\{\mathbf{E}_{\mathrm{dev}}^{(a)}\}_{a=1}^5$:

$$\kappa(\mathbf{b}, \mathbf{u}) = \frac{2}{3} \overline{\mathcal{E}}(\mathbf{E}_{\mathrm{vol}}; \mathbf{b}, \mathbf{u}), \quad \mu(\mathbf{b}, \mathbf{u}) = \frac{1}{5} \sum_{a=1}^{5} \overline{\mathcal{E}}(\mathbf{E}_{\mathrm{dev}}^{(a)}; \mathbf{b}, \mathbf{u}), \quad \mathbf{q}(\mathbf{b}, \mathbf{u}) = \begin{bmatrix} \log \kappa(\mathbf{b}, \mathbf{u}) \\ \log \mu(\mathbf{b}, \mathbf{u}) \end{bmatrix}. \tag{5}$$

Here, $\kappa$ and $\mu$ denote graph-integrated bulk and shear responses; for a physical reference volume $V$, the corresponding continuum moduli are $\kappa/V$ and $\mu/V$. Since $V$ is fixed for a given graph, it cancels in logarithmic response differences between material states, making these relative changes comparable across graphs.

### 4.3 Cross-Graph Energy-Response Alignment and Material Identification

Section 4.2 defines, for each graph, a response map from native carrier controls $\mathbf{u}$ to $\mathbf{q}(\mathbf{b}, \mathbf{u})$ using the same standardized probes. Because this map depends on graph geometry and carrier realization, the same $\mathbf{u}$ can induce different response changes. As illustrated in Fig. 3(B), we therefore calibrate each graph to realize a shared relative response change through graph-specific carrier controls.

**Graph-specific response calibration.** For each graph, we linearize this response map once at the reference state $(\mathbf{b}^0, \mathbf{u} = \mathbf{0}, \mathbf{q}^0 = \mathbf{q}(\mathbf{b}^0, \mathbf{0}))$:

$$\mathbf{A} = \left.\frac{\partial \mathbf{q}}{\partial \mathbf{u}}\right|_{(\mathbf{b}^0, \mathbf{0})} = \begin{bmatrix} \partial_{u_B} \log \kappa & \partial_{u_S} \log \kappa \\ \partial_{u_B} \log \mu & \partial_{u_S} \log \mu \end{bmatrix}, \quad \Delta\mathbf{q} := \mathbf{q}(\mathbf{b}^0, \mathbf{u}) - \mathbf{q}^0 \approx \mathbf{A}\mathbf{u}, \quad \mathbf{u} = \mathbf{A}^{-1}\Delta\mathbf{q}. \tag{6}$$

Thus, $\mathbf{A}$ maps local carrier changes to response changes, while $\mathbf{A}^{-1}$ converts a shared relative response change $\Delta\mathbf{q}$ into the control required by that graph. The reference response $\mathbf{q}^0$ is graph-specific and is not required to coincide across graphs. We compute $\mathbf{A}$ once and keep it fixed.

**Joint material identification.** We optimize a single shared response change $\Delta\mathbf{q}$ across interactions, while each graph realizes it through $\mathbf{u} = \mathbf{A}^{-1}\Delta\mathbf{q}$ together with its graph-specific $\mathbf{b}$ and $\boldsymbol{\eta}$:

$$\min_{\Delta\mathbf{q}, \mathbf{b}, \boldsymbol{\eta}} \mathcal{L}_{\text{motion}} \left( \widehat{\mathbf{X}} \left( \mathbf{b}, \mathbf{A}^{-1}\Delta\mathbf{q}, \boldsymbol{\eta} \right), \mathcal{V} \right). \tag{7}$$

Carrier geometry, edge-family assignments, the reference response $\mathbf{q}^0$, and $\mathbf{A}$ remain fixed; $\mathbf{u}$ is derived at each forward simulation instead of optimized independently.

### 4.4 Material Editing, Reuse, and Future Rollout

**Material editing and reuse.** Local stiffness-scale edits act through $\mathbf{b}$, while bulk-shear edits act through the shared relative response change $\Delta\mathbf{q}$. For an edited response change $\Delta\mathbf{q}'$, a target graph realizes it as $\mathbf{u}' = \mathbf{A}^{-1}\Delta\mathbf{q}'$. A target object therefore needs only one calibration $\mathbf{A}$; the same calibrated bulk-shear response change can then be applied relative to its own reference state without copying source spring or carrier parameters.

**Future rollout and damage.** For a new initial state and interaction, we freeze the identified or edited material and roll out the explicit Spring-Mass simulator without future fitting. Connections fail irreversibly when their tensile strain exceeds $\epsilon_c$; surviving spring and carrier parameters remain fixed, and Gaussian bindings follow surviving local connectivity. Without recalibration, the effective response evolves directly from connectivity loss.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets, baselines, and evaluation.** We evaluate MorphTwin on the same 20 PhysTwin scenes used by DeformMaster (Li et al., 2026), 14 simulated Spring-Gaus (Zhong et al., 2024) development scenes with known material parameters, and 15 self-captured sequences with aligned egocentric stereo and RGB-D observations. Real-world sequences assess reconstruction and future prediction on observed and subsequent frames, while Spring-Gaus provides ground truth for material recovery. Baselines include PhysTwin (Jiang et al., 2025), DeformMaster, PGND (Zhang et al., 2025), Boba (Pang et al., 2026), and MatPhys (Yang et al., 2026). We use released checkpoints, pretrained Gaussian assets, and inference code for DeformMaster, Boba, and MatPhys, and train and evaluate PGND under a matched protocol for fair comparison. Following DeformMaster, we compute PSNR (Hore & Ziou, 2010), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) using foreground masks. We measure geometry and motion with Chamfer distance (Wu et al., 2021) and tracking error, and material recovery with relative Young's-modulus error and absolute Poisson-ratio error. Dataset, implementation, and evaluation details are provided in Appendix E.

### 5.2 RECONSTRUCTION, FUTURE PREDICTION, AND EFFICIENCY

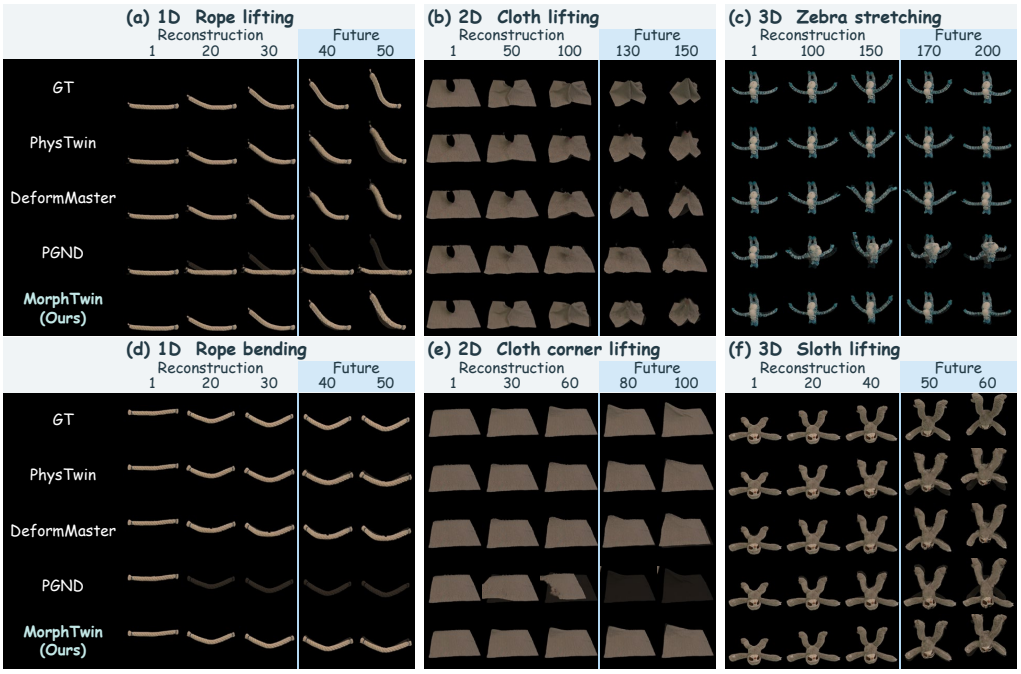


Figure 4: Reconstruction and future prediction on PhysTwin cases, two each for 1D curves, 2D surfaces, and 3D solids. Each includes reconstruction and future frames. GT appears separately and is overlaid semi-transparently on predictions to highlight motion and deformation discrepancies.

**Reconstruction and future prediction.** Table 1 compares reconstruction and future prediction on the same 20 *PhysTwin scenes* used by DeformMaster, using real-world recordings from three RGB-D cameras as input. MorphTwin substantially improves visual quality over PhysTwin, increasing PSNR by 3.53 dB for reconstruction and 1.99 dB for future prediction. LPIPS decreases by 22.5% and 24.8%, respectively, with improved visual fidelity extending beyond the observation interval.

Figure 4 compares 1D, 2D, and 3D cases with semi-transparent GT overlays. MorphTwin better matches GT deformation, while PhysTwin's future motion appears overly stiff. These results support separate volumetric and shape response control for prediction beyond observed interactions. Subsequent controlled experiments examine the physical meaning of the recovered material responses.

Table 1: Reconstruction and future prediction on 20 PhysTwin scenes. CD and Track are in meters, so millimeter-scale differences appear numerically small. **Best** / second-best per metric and task.

| | Reconstruction | | | | | Future Prediction | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ |
| PhysTwin (ICCV'25) (Jiang et al., 2025) | 31.12 | 0.9625 | 0.0275 | 0.00553 | 0.0094 | 25.16 | 0.9350 | 0.0610 | 0.01200 | 0.0230 |
| PGND (RSS'25) (Zhang et al., 2025) | 27.88 | 0.9486 | 0.0518 | 0.07739 | 0.0632 | 21.57 | 0.9145 | 0.1171 | 0.23661 | 0.1988 |
| Boba (ECCV'26) (Pang et al., 2026) | 31.05 | 0.9621 | 0.0322 | 0.00736 | **0.0090** | 25.92 | 0.9451 | 0.0606 | 0.01474 | 0.0223 |
| DeformMaster (Li et al., 2026) | 29.81 | 0.9534 | 0.0315 | 0.00682 | 0.0115 | 25.41 | 0.9360 | 0.0610 | 0.01140 | 0.0240 |
| MatPhys (Yang et al., 2026) | 27.39 | 0.9454 | 0.0461 | 0.01282 | 0.0211 | 23.03 | 0.9257 | 0.0855 | 0.02854 | 0.0463 |
| **MorphTwin (Ours)** | **34.65** | **0.9821** | **0.0213** | **0.00552** | 0.0092 | **27.15** | **0.9547** | **0.0459** | **0.01130** | **0.0221** |

**Sparse-view reconstruction.** Table 2 compares input-view configurations on 15 self-captured scenes. Using all three views instead of a single RGB-D view improves reconstruction PSNR by 4.75 dB for MorphTwin versus 2.36 dB for PhysTwin, highlighting gains from complementary views. With only egocentric stereo views, MorphTwin outperforms PhysTwin across all reported reconstruction and future-prediction metrics, supporting egocentric physical twins for interactive simulation. Complete reconstruction and future-prediction results appear in Appendix Figure 8.

Table 2: Observation-view ablation on 15 self-captured scenes using single RGB-D, VR stereo egocentric, and combined views under the same evaluation protocol. CD and Track are in meters.

| | | Reconstruction | | | | | Future Prediction | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Input views | Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ |
| All Views | PhysTwin | 28.74 | 0.9451 | 0.0598 | **0.00946** | **0.0172** | 25.27 | 0.9321 | 0.0850 | 0.01597 | 0.0322 |
| | **MorphTwin (Ours)** | **32.12** | **0.9652** | **0.0544** | 0.01050 | 0.0188 | **26.77** | **0.9462** | **0.0763** | **0.01509** | **0.0301** |
| Egocentric | PhysTwin | 28.06 | 0.9395 | 0.0782 | 0.00952 | 0.0178 | 25.04 | 0.9291 | 0.0971 | 0.01635 | 0.0331 |
| | **MorphTwin (Ours)** | **31.13** | **0.9531** | **0.0669** | **0.00941** | **0.0176** | **26.36** | **0.9364** | **0.0882** | **0.01508** | **0.0317** |
| Single RGB-D | PhysTwin | 26.38 | 0.9291 | 0.0887 | **0.01002** | 0.0220 | 23.80 | 0.9202 | 0.1087 | 0.01694 | 0.0376 |
| | **MorphTwin (Ours)** | **27.37** | **0.9451** | **0.0772** | 0.01078 | **0.0189** | **24.41** | **0.9288** | **0.0955** | **0.01680** | **0.0322** |

**Efficient explicit simulation.** MorphTwin calibrates volumetric and shape response targets into graph-specific spring parameters offline. Online simulation directly advances the calibrated explicit graph, without per-step material identification, energy alignment, or an additional material-prediction network. As shown in Figure 5, MorphTwin achieves higher offline and same-node interactive throughput than PhysTwin and DeformMaster on the 1D and 3D groups under the evaluated configurations. Thin-shell bending and folding make stable time integration more demanding in the 2D group. Detailed numerical results are provided in Table 7 in Appendix E.

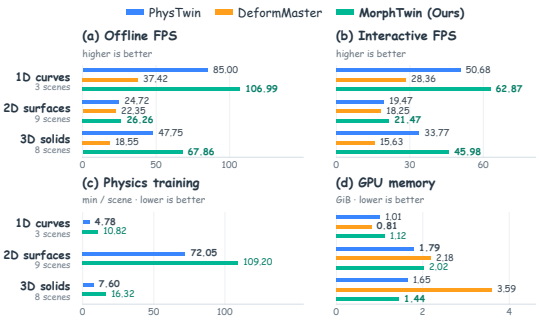


Figure 5: Physics-training cost and inference efficiency on the PhysTwin scenes.

MorphTwin realizes material responses through an explicit spring graph, making it compatible with complementary acceleration strategies developed for PhysTwin, such as Boba's compact physical models and batched simulation with shared assets (Pang et al., 2026).

### 5.3 Material Recovery on Synthetic Data with Ground-Truth Parameters

Table 3: Material recovery and controlled appearance evaluation on Spring-Gaus. **Left:** Per-scene Poisson ratios. **Right:** Material errors and reconstruction/future-prediction quality.

**Poisson-Ratio Recovery**

| Scene | $\nu_{GT}$ | PhysTwin | **MorphTwin** | Scene | $\nu_{GT}$ | PhysTwin | **MorphTwin** |
|---|---|---|---|---|---|---|---|
| apple | 0.01 | 0.2492 | **0.0260** | droplet | 0.30 | 0.2511 | **0.3268** |
| banana | 0.30 | 0.2466 | **0.2942** | duck | 0.30 | 0.2471 | **0.2693** |
| C | 0.10 | 0.2445 | **0.1171** | E | 0.40 | 0.2378 | **0.3408** |
| chess | 0.40 | 0.2478 | **0.3662** | rope | 0.30 | 0.2414 | **0.2939** |
| cream | 0.30 | 0.2489 | **0.2950** | toothpaste | 0.40 | 0.2448 | **0.3757** |
| cross | 0.30 | 0.2468 | **0.2623** | torus | 0.30 | 0.2405 | **0.2887** |
| doraemon | 0.30 | 0.2473 | **0.3475** | V | 0.30 | 0.2441 | **0.2961** |

| Metric | Spring-Gaus | PhysTwin | **MorphTwin** |
|---|---|---|---|
| **Material Recovery** | | | |
| $E$ Err. (%) ↓ | 335.29 | 18.22 | **14.20** |
| $\nu$ Err. ↓ | 0.1189 | 0.0957 | **0.0232** |
| **Reconstruction** | | | |
| PSNR ↑ | 21.10 | 22.00 | **22.18** |
| SSIM ↑ | 0.9168 | 0.9286 | **0.9296** |
| LPIPS ↓ | 0.1130 | 0.1041 | **0.1024** |
| **Future Prediction** | | | |
| PSNR ↑ | 15.80 | 16.28 | **16.35** |
| SSIM ↑ | 0.8586 | 0.8720 | **0.8746** |
| LPIPS ↓ | 0.1733 | 0.1595 | **0.1554** |

We evaluate material recovery on 14 synthetic Spring-Gaus development scenes with known Young's modulus $E$ and Poisson's ratio $\nu$. Both methods optimize graph-specific spring parameters during inverse fitting, while MorphTwin identifies shared material responses in calibrated bulk-shear coordinates. This setup evaluates material recovery under joint graph-specific fitting. For RGB-supervised Spring-Gaus, equivalent $E$ and $\nu$ are computed from its fitted spring network. Table 3 shows that MorphTwin reduces mean absolute $\nu$ error by 75.8% (0.0957 to 0.0232) and mean relative $E$ error by 22.1% (18.22% to 14.20%) compared with PhysTwin, with lower $\nu$ error on all 14 scenes. Notably, PhysTwin's estimates cluster within $0.2378$–$0.2511$ despite ground truths spanning $0.01$–$0.40$, consistent with the Cauchy restriction $\lambda = \mu$ and $\nu = 1/4$ for unstressed, isotropic 3D central-force networks under affine small strains. Separate bulk and shear energy calibration enables MorphTwin to recover Poisson ratios on both sides of $1/4$: $0.0260$ for *apple* and $0.3757$ for *toothpaste*, against ground truths of $0.01$ and $0.40$, respectively. Controlled appearance evaluation shows improved

reconstruction and future prediction. These results validate material recovery under the intended joint inverse-fitting setting and support the use of the recovered responses for editing and transfer.

### 5.4 Ablation Studies

Our ablations address three questions: *(a) Why are bulk and shear a minimal representation of isotropic linear elasticity? (b) Why focus inversion on elastic responses? (c) Why learn response coordinates and calibrate spring parameters for each target graph?*

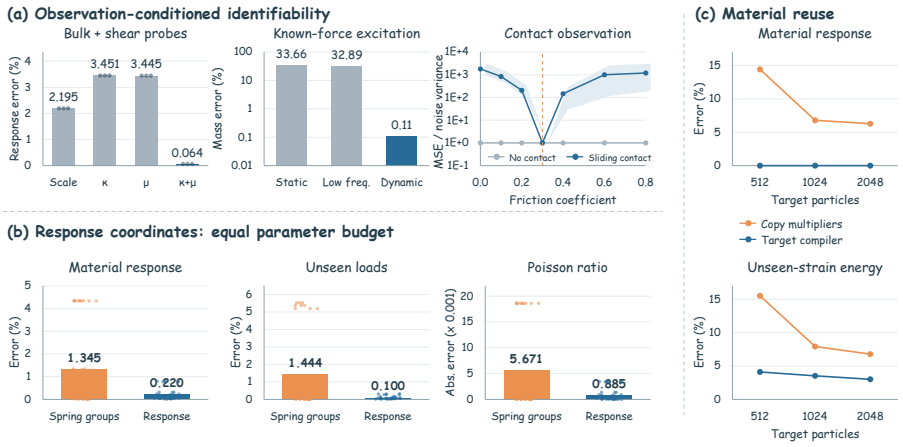


Figure 6: Ablations on material representation and reuse. (a) Bulk-shear parameterization and mass/friction observability. (b) Response, unseen-load, and Poisson-ratio errors under matched two-parameter budgets. (c) Cross-resolution response and unseen-strain energy errors.

**Material parameterization and observability.** Isotropic linear elasticity requires independent bulk and shear responses. Figure 6(a) shows that jointly optimizing $\kappa$ and $\mu$ reduces response error to 0.064%, versus 2.195–3.451% for single-parameter alternatives, supporting separate control of volume and shape changes. The same panel examines the observation requirements of additional parameters. With known forces, dynamic excitation reduces mass error from 33.66% under static loading to 0.11%. Sliding contact constrains friction, whereas no-contact observations do not. These results motivate matching learned parameters to available observations: we focus on elastic responses with fixed density, while additional parameters require suitable excitation or contact information.

**Response coordinates and graph calibration.** Under the same two-parameter budget, Figure 6(b) shows that response coordinates improve material-response fitting, unseen-load generalization, and Poisson-ratio recovery over direct spring-group optimization. Figure 6(c) shows that target-graph calibration preserves material responses across particle resolutions and reduces unseen-strain energy errors compared with copying spring multipliers, extending benefits beyond calibration probes. These results support compact, physically meaningful response coordinates while adapting their spring implementation to each graph for accurate recovery and consistent material reuse.

**Material editing, reuse, and connection failure.** Appendix E.2–E.3 further evaluates material editing, two-way material exchange, and connection failure. These experiments show that shared material responses can be realized through graph-specific spring calibration, while local surviving connectivity guides appearance updates as cracks develop. Together, they demonstrate how a reconstructed twin supports editable material behavior, cross-object reuse, and structural changes.

## 6 Conclusion, Limitations and Future Work

We presented MorphTwin for accurate material recovery, interpretable editing, and cross-object reuse from videos while retaining efficient explicit simulation. Experiments demonstrate improved reconstruction and future prediction, including from egocentric observations. These results support building physical twins with reusable, explicitly controllable material behavior. Our current formulation focuses on isotropic elasticity with fixed density. Future work will integrate feed-forward reconstruction to streamline physical-twin creation and broaden its applications.

## REFERENCES


Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gic: Gaussian-informed continuum for physical property identification and simulation. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:75035–75063, 2024.

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6178–6189, 2025.

Karolina Golec, J-F Palierne, Florence Zara, Stéphane Nicolle, and Guillaume Damiand. Hybrid 3d mass-spring system for simulation of isotropic materials with any poisson's ratio. *The Visual Computer*, 36(4):809–825, 2020.

Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pp. 2366–2369. IEEE, 2010. doi: 10.1109/ICPR.2010.579.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*, pp. 1–52, 2016.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7219–7230, 2025.

Yixiong Jing, Xingyuan Chen, Guangming Wang, Olaf Wysocki, Haibing Wu, and Brian Sheil. Physpring: Structure-preserving reduction of physics-informed twins via gnn. *arXiv preprint arXiv:2605.07687*, 2026.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. URL `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

Hyunjin Kim, Ri-Zhao Qiu, Guangqi Jiang, and Xiaolong Wang. Egophys: Learning generalizable physics models of deformable objects from egocentric video. *arXiv preprint arXiv:2606.16202*, 2026.

Jihyun Lee, Changmin Lee, Donghwan Kim, and Tae-Kyun Kim. Physhandi: Physics-based reconstruction of hand-deformable object interactions. *arXiv preprint arXiv:2605.09538*, 2026.

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.

Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. *arXiv preprint arXiv:2303.05512*, 2023.

Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9080–9090, 2025.

Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. *arXiv preprint arXiv:2501.18982*, 2025.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 38–55. Springer, 2024.

Bryn A Lloyd, Sercan Kirac, Gábor Székely, and Matthias Harders. Identification of dynamic mass spring parameters for deformable body simulation. In *Eurographics (Short Papers)*, pp. 131–134, 2008.

Yihan Pang, Hanxiao Jiang, Sushant Kondguli, Sarita Adve, and Shenlong Wang. Boba: Batched simulation for physics-based gaussian digital twins. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

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 (ICLR)*, volume 2025, pp. 28085–28128, 2025.

Junuthula Narasimha Reddy. *An introduction to continuum mechanics*. Cambridge university press, 2013.

Daniel Rho, Jun Myeong Choi, Matthew Thornton, Biswadip Dey, and Roni Sengupta. Monophysics: Estimating geometry, appearance, and physical parameters from monocular videos. *arXiv preprint arXiv:2605.30320*, 2026.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Balanced chamfer distance as a comprehensive metric for point cloud completion. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:29088–29100, 2021.

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 *Proceedings of the 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389–4398, 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 (AAAI)*, volume 40, pp. 11361–11369, 2026.

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.

Jiahao Zhan, Zizhang Li, Hong-Xing Yu, and Jiajun Wu. Perpetualwonder: Long-horizon action-conditioned 4d scene generation. *arXiv preprint arXiv:2602.04876*, 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*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 388–406. Springer, 2024.

Boming Zhao, Yuan Li, Ziyu Sun, Lin Zeng, Yujun Shen, Rui Ma, Yinda Zhang, Hujun Bao, and Zhaopeng Cui. Gaussianprediction: Dynamic 3d gaussian prediction for motion extrapolation and free view synthesis. In *Proceedings of the ACM SIGGRAPH Conference Papers*, pp. 1–12, 2024.

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

Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2013.

## APPENDIX

**Input–processing–output comparison.** To clarify the motivation behind MorphTwin, we first compare representative approaches through their *input - intermediate processing - output* pathways. We distinguish recovering physical models from observed interactions, predicting future motion, and generating plausible dynamics from static inputs. We then examine how these approaches represent material behavior and whether that representation supports reuse across different discretizations.

**Physical dynamics from static assets and single images.** PhysGaussian (Xie et al., 2024) animates static Gaussian assets with prescribed material parameters, while PhysDreamer (Zhang et al., 2024) and OmniPhysGS (Lin et al., 2025) infer material behavior through generated motion or video-diffusion guidance. PhysGen3D (Chen et al., 2025) infers geometry and physical properties from a single image, while WonderPlay (Li et al., 2025) and PerpetualWonder (Zhan et al., 2026) combine physical simulation with generative refinement for action-conditioned dynamics.

These methods enable convenient asset creation and plausible motion, but static appearance alone leaves target-specific material responses underconstrained. This motivates identifying material properties from observed deformation and interaction, through which material behavior becomes observable, so that reconstructed physical twins reproduce the target object's dynamics.

**Video-based physical twins and interactive dynamics.** PAC-NeRF (Li et al., 2023) couples multi-view videos with a hybrid Eulerian-Lagrangian radiance representation and differentiable MPM to recover geometry and physical parameters. GIC (Cai et al., 2024) reconstructs dynamic Gaussians from multi-view videos, converts them into a filled continuum, and uses surface and mask supervision to identify constitutive parameters for simulation. MonoPhysics (Rho et al., 2026) jointly estimates geometry, appearance, and physical parameters from monocular video through differentiable MPM and Gaussian rendering, using scale alignment and physics-aware geometry refinement to reduce reconstruction ambiguities. Spring-based methods provide an efficient alternative. Spring-Gaus (Zhong et al., 2024) fits spring-connected simulation anchors and Gaussian appearance to multi-view videos, producing elastic assets for future prediction and resimulation. PhysTwin (Jiang et al., 2025) combines geometry completion, inverse spring-model fitting, and Gaussian reconstruction from sparse-view RGB-D interaction videos to produce real-time interactive twins. PhysHanDI (Lee et al., 2026) reconstructs dense hands from sparse-view RGB-D videos, fits spring-based object dynamics under hand-induced forces, and refines the hands through inverse physics for coherent reconstruction and prediction. Given existing Gaussian twins, Boba (Pang et al., 2026) combines compact spring surrogates with optimized deformation, rendering, and batched execution to accelerate independent rollouts. Continuum formulations offer rich constitutive behavior at potentially higher simulation cost, while spring-based models encode material behavior through parameters whose effects depend on graph geometry and connectivity.

For isotropic elasticity, MorphTwin combines three advantages: (i) separate bulk and shear control through internally relaxable response carriers; (ii) interpretable material identification, editing, and cross-graph reuse through energy-aligned, graph-specific calibration; and (iii) efficient explicit online dynamics without repeated material identification or calibration optimization.

**Learned dynamics and reusable material representations.** GaussianPrediction (Zhao et al., 2024) reconstructs deforming Gaussians from videos, distills motion into keypoint trajectories, and predicts their continuation with a graph convolutional network for future novel-view rendering, without identifying an explicit constitutive model. PGND (Zhang et al., 2025) learns particle-grid velocity fields from sparse-view RGB-D interactions and couples predicted trajectories with Gaussian rendering for action-conditioned prediction and planning. DeformMaster (Li et al., 2026) combines MPM, neural residuals, compliant hand actuation, and spatially varying constitutive experts to reconstruct interactive models from real videos, supporting novel actions, material-parameter variation, and novel-view rendering. These approaches represent part or all of the predicted motion through learned trajectories, state transitions, or residual corrections.

Other methods learn parameters for explicit simulation. MatPhys (Yang et al., 2026) combines part-aware reconstruction, a shared material codebook, and motion and local-geometry features from monocular video to predict simulation-ready spring parameters, improving cross-scene consistency. EgoPhys (Kim et al., 2026) distills inverse-physics solutions into a codebook to predict dense spring stiffness fields from egocentric RGB video, avoiding per-spring test-time optimization for unseen objects. NeuSpring (Xu et al., 2026) fits piecewise topologies and canonical-coordinate neural spring fields from interaction videos, capturing heterogeneous properties and spatial correlations for reconstruction and prediction. These methods improve parameter estimation and sharing, but preserving mechanical responses across graphs additionally requires accounting for how geometry and connectivity affect the resulting material behavior (Lloyd et al., 2008). Moreover, unstressed, isotropic three-dimensional central-force networks couple bulk and shear under affine small strains.

Inspired by the hybrid mass-spring model in the numerical simulation literature (Golec et al., 2020), whose formulation focuses on regular 3D cubic lattices, we revisit material-response expressiveness from first principles while preserving the mesh-based explicit dynamics that enable interactive video-based physical twins. We introduce internally relaxable response carriers and energy-based material calibration to enable independent bulk and shear control, interpretable material identification, and consistent editing and reuse across spring graphs, while retaining efficient online simulation.

**Appendix overview.** The Appendix provides an AI use statement (Sec. A), theoretical analysis and derivations (Sec. B), optimization and inference details (Sec. C), dataset and capture details (Sec. D), experimental protocols and additional results (Sec. E), and an overview of the interactive interface and operating workflow (Sec. F). We provide three supplementary videos demonstrating MorphTwin's reconstruction, interactive simulation, and material control (Table 4).

Table 4: Supplementary videos and their purposes.

| Video | Purpose |
|---|---|
| **Video 1:** Dataset comparisons | Compare reconstruction and future prediction with PhysTwin on self-captured scenes. |
| **Video 2:** Interactive UI demo | Demonstrate interactive manipulation, material morphing, and connection failure. |
| **Video 3:** Material morphing and reuse | Examine material recovery, interaction stability, and calibrated material reuse across graphs. |

## A AI USE STATEMENT

For manuscript preparation, we used AI writing assistants to improve the grammar, wording, readability, and internal consistency of author-written text. All AI-assisted edits were manually reviewed and revised by the authors. We also used AI agents to assist with processing our self-captured egocentric stereo recordings from VR devices and RGB-D videos from depth cameras, primarily for keypoint annotation and temporal alignment. Every processing stage underwent rigorous and systematic human review. The authors checked the resulting annotations and temporal correspondences and corrected them where necessary before use.

# B Theoretical Analysis and Derivations

## B.1 Bulk-shear coupling of affine central-force networks

We provide the derivation of the bulk-shear coupling stated in Eq. 1. Consider an unstressed three-dimensional central-force network under a symmetric small strain $\boldsymbol{\epsilon}$. For an edge with reference direction $\mathbf{n}$, the first-order axial strain is $\mathbf{n}^\top \boldsymbol{\epsilon} \mathbf{n}$. Under an isotropic effective response, the stiffness-weighted edge directions satisfy the standard fourth-order directional average

$$\langle n_i n_j n_k n_l \rangle = \frac{1}{15} \left( \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} \right). \tag{8}$$

Here and below, $\mathbf{A} : \mathbf{B} = \mathrm{tr}(\mathbf{A}^\top \mathbf{B}) = \sum_{i,j} A_{ij} B_{ij}$ denotes the double contraction. Contracting the directional average with $\epsilon_{ij}\epsilon_{kl}$ therefore gives

$$\left\langle (\mathbf{n}^\top \boldsymbol{\epsilon} \mathbf{n})^2 \right\rangle = \frac{1}{15} \left[ (\mathrm{tr}\, \boldsymbol{\epsilon})^2 + 2\boldsymbol{\epsilon} : \boldsymbol{\epsilon} \right]. \tag{9}$$

Absorbing the network stiffness density into a scalar $c$ gives

$$W_{\mathrm{central}}(\boldsymbol{\epsilon}) = \frac{c}{30} \left[ (\mathrm{tr}\, \boldsymbol{\epsilon})^2 + 2\boldsymbol{\epsilon} : \boldsymbol{\epsilon} \right]. \tag{10}$$

To characterize which isotropic material responses this spring network can represent, we compare its effective energy with the general small-strain energy of an isotropic linear-elastic solid. The latter admits two independent elastic parameters, whereas Eq. (10) contains only the single network stiffness scale $c$. The isotropic constitutive law is $\boldsymbol{\sigma} = \lambda\, \mathrm{tr}(\boldsymbol{\epsilon})\mathbf{I} + 2\mu\boldsymbol{\epsilon}$, which gives the standard strain-energy density (Reddy, 2013)

$$W_{\mathrm{iso}}(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\epsilon} = \frac{\lambda}{2}(\mathrm{tr}\, \boldsymbol{\epsilon})^2 + \mu\, \boldsymbol{\epsilon} : \boldsymbol{\epsilon}. \tag{11}$$

Requiring the two quadratic energies to agree for arbitrary symmetric small strains allows their coefficients to be matched independently, giving $\lambda = \mu = c/15$. Substituting this relation into the standard modulus conversions used in Eq. 2 recovers $\kappa = 5\mu/3$ and $\nu = 1/4$. Thus, the isotropic central-force network spans only a restricted one-parameter subset of isotropic linear elasticity: changing the spring stiffness scale changes the overall rigidity but not the relative volumetric and shear response. In particular, it cannot independently control $\kappa$ and $\mu$ within the unstressed, isotropic, affine small-strain regime considered here. Nonuniform edge weights may in general introduce anisotropy; the restriction above concerns the isotropic effective response.

## B.2 Relaxed energy of response carriers

We next derive the relaxed elastic energy in Eq. 4. Here, $\mathbf{E}$ denotes a prescribed symmetric small-strain probe applied affinely to the original physical nodes; it is kept generic in the derivation, while the volumetric probe $\mathbf{E}_{\mathrm{vol}} = \mathbf{I}/\sqrt{3}$ and the five orthonormal trace-free probes $\{\mathbf{E}_{\mathrm{dev}}^{(a)}\}_{a=1}^{5}$ used in Eq. 5 are particular choices of $\mathbf{E}$. For a spring $e = (i,j) \in \mathcal{S}$, let its reference edge vector be $\mathbf{d}_e = \mathbf{x}_j - \mathbf{x}_i = L_e \mathbf{n}_e$. In addition to the affine displacement induced by $\mathbf{E}$, the relaxable auxiliary nodes may undergo internal displacements $\mathbf{z}_i$, whereas the original physical nodes have $\mathbf{z}_i = \mathbf{0}$. The first-order change of the edge vector is $\delta\mathbf{d}_e = \mathbf{E}\mathbf{d}_e + (\mathbf{z}_j - \mathbf{z}_i)$, giving the first-order axial strain

$$\frac{\delta L_e}{L_e} = \mathbf{n}_e^\top \mathbf{E} \mathbf{n}_e + \frac{\mathbf{n}_e^\top (\mathbf{z}_j - \mathbf{z}_i)}{L_e}. \tag{12}$$

For a fixed internal relaxation $\mathbf{z}$, the corresponding quadratic central-spring energy is

$$\mathcal{E}(\mathbf{E}, \mathbf{z}; \mathbf{b}, \mathbf{u}) = \frac{1}{2} \sum_{e=(i,j)\in\mathcal{S}} Y_e(\mathbf{b}, \mathbf{u}) L_e \left[ \mathbf{n}_e^\top \mathbf{E} \mathbf{n}_e + \frac{\mathbf{n}_e^\top (\mathbf{z}_j - \mathbf{z}_i)}{L_e} \right]^2. \tag{13}$$

The physical nodes are constrained to the prescribed affine deformation, while the auxiliary nodes are free to relax. Eliminating these internal degrees of freedom gives the relaxed energy $\overline{\mathcal{E}}(\mathbf{E}; \mathbf{b}, \mathbf{u}) = \min_{\mathbf{z}} \mathcal{E}(\mathbf{E}, \mathbf{z}; \mathbf{b}, \mathbf{u})$, which is the quantity used in Eq. 4. Evaluating this same function at $\mathbf{E}_{\mathrm{vol}}$ and $\mathbf{E}_{\mathrm{dev}}^{(a)}$ yields the volumetric and deviatoric energies used to define $\kappa(\mathbf{b}, \mathbf{u})$ and $\mu(\mathbf{b}, \mathbf{u})$ in Eq. 5.

To make explicit how auxiliary relaxation modifies the response seen by the prescribed strain probe, we rewrite the spring energy in matrix form and eliminate the internal degrees of freedom to obtain a reduced effective-stiffness representation. Let $\mathbf{g}(\mathbf{E})$ collect the affine axial strains $g_e(\mathbf{E}) = \mathbf{n}_e^\top \mathbf{E}\mathbf{n}_e$, let $\mathbf{Bz}$ collect the relaxation-induced axial strains $\mathbf{n}_e^\top(\mathbf{z}_j - \mathbf{z}_i)/L_e$, and define $\mathbf{K}(\mathbf{b},\mathbf{u}) = \mathrm{diag}(Y_e(\mathbf{b},\mathbf{u})L_e)_{e\in\mathcal{S}}$. Then the unrelaxed energy can be written as

$$\mathcal{E}(\mathbf{E},\mathbf{z};\mathbf{b},\mathbf{u}) = \frac{1}{2}\left[\mathbf{g}(\mathbf{E}) + \mathbf{Bz}\right]^\top \mathbf{K}(\mathbf{b},\mathbf{u})\left[\mathbf{g}(\mathbf{E}) + \mathbf{Bz}\right]. \tag{14}$$

Minimizing the quadratic form over the auxiliary displacements $\mathbf{z}$ eliminates the internally relaxable degrees of freedom, yielding the reduced energy of the prescribed physical-node deformation:

$$\overline{\mathcal{E}}(\mathbf{E};\mathbf{b},\mathbf{u}) = \frac{1}{2}\mathbf{g}(\mathbf{E})^\top \left[\mathbf{K} - \mathbf{KB}\left(\mathbf{B}^\top\mathbf{KB}\right)^\dagger \mathbf{B}^\top\mathbf{K}\right]\mathbf{g}(\mathbf{E}), \tag{15}$$

where $\mathbf{K} = \mathbf{K}(\mathbf{b},\mathbf{u})$ for compactness and $(\cdot)^\dagger$ denotes the pseudoinverse when internal null modes are present. The bracketed operator is the effective stiffness acting on the affine edge-strain vector after the auxiliary degrees of freedom have relaxed. It shows explicitly how internal relaxation removes part of the stiffness seen by the prescribed deformation. Because this reduction depends on the carrier geometry through $\mathbf{B}$, different carrier constructions retain different amounts of energy under the volumetric and deviatoric probes, enabling separate control of the two material responses.

### B.3 Volumetric and deviatoric response modes

A symmetric $3\times 3$ strain tensor has six independent components, which decompose orthogonally into one volumetric direction and a five-dimensional trace-free deviatoric subspace. Since $\|\mathbf{I}\|_F = \sqrt{3}$, the unit volumetric probe in Eq. 5 is $\mathbf{E}_{\mathrm{vol}} = \frac{\mathbf{I}}{\sqrt{3}}$. For deviatoric subspace, we use orthonormal basis

$$\begin{aligned}
\mathbf{E}_{\mathrm{dev}}^{(1)} &= \frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0\\0&-1&0\\0&0&0\end{bmatrix}, \quad \mathbf{E}_{\mathrm{dev}}^{(2)} = \frac{1}{\sqrt{6}}\begin{bmatrix}1&0&0\\0&1&0\\0&0&-2\end{bmatrix}, \quad \mathbf{E}_{\mathrm{dev}}^{(3)} = \frac{1}{\sqrt{2}}\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix}, \\
\mathbf{E}_{\mathrm{dev}}^{(4)} &= \frac{1}{\sqrt{2}}\begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix}, \quad \mathbf{E}_{\mathrm{dev}}^{(5)} = \frac{1}{\sqrt{2}}\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix}.
\end{aligned} \tag{16}$$

These probes satisfy $\mathrm{tr}\,\mathbf{E}_{\mathrm{dev}}^{(a)} = 0$ and $\mathbf{E}_{\mathrm{dev}}^{(a)} : \mathbf{E}_{\mathrm{dev}}^{(b)} = \delta_{ab}$. For a general strain, define $\mathbf{E}_{\mathrm{dev}} = \mathbf{E} - \frac{1}{3}(\mathrm{tr}\,\mathbf{E})\mathbf{I}$. The corresponding isotropic strain-energy density is

$$W(\mathbf{E}) = \frac{\kappa}{2V}\left(\mathrm{tr}\,\mathbf{E}\right)^2 + \frac{\mu}{V}\mathbf{E}_{\mathrm{dev}} : \mathbf{E}_{\mathrm{dev}}, \tag{17}$$

where $\kappa/V$ and $\mu/V$ are the corresponding continuum bulk and shear moduli. The total elastic energy of the reference body is therefore $\mathcal{E}(\mathbf{E}) = VW(\mathbf{E}) = \frac{\kappa}{2}\left(\mathrm{tr}\,\mathbf{E}\right)^2 + \mu\,\mathbf{E}_{\mathrm{dev}} : \mathbf{E}_{\mathrm{dev}}$. For the unit volumetric probe, $\mathrm{tr}(\mathbf{E}_{\mathrm{vol}}) = \sqrt{3}$ and $\mathbf{E}_{\mathrm{dev}} = \mathbf{0}$, giving $\mathcal{E}(\mathbf{E}_{\mathrm{vol}}) = \frac{3}{2}\kappa$. Each unit deviatoric probe is trace-free and has unit Frobenius norm, giving $\mathcal{E}(\mathbf{E}_{\mathrm{dev}}^{(a)}) = \mu$. These identities give exactly the normalization factors $2/3$ and $1/5$ used in Eq. 5: the volumetric response is recovered from the relaxed total energy under the volumetric probe, while the shear response is obtained by averaging the relaxed total energies over the five deviatoric probes.

For a perfectly isotropic response, all five deviatoric probe energies are identical. For a discrete graph with residual anisotropy, averaging over any complete orthonormal basis of the five-dimensional deviatoric subspace gives a basis-independent scalar summary of its quadratic response. Although absolute graph-integrated responses contain the reference-volume factor, it is fixed for each graph and cancels in the relative logarithmic change $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}^0$. Thus, $\Delta\mathbf{q}$ measures the relative change in the corresponding continuum bulk and shear moduli from the graph's calibration reference and serves as the shared quantity in Sec. 4.3.

## C Optimization and Inference Details

This section describes the optimization procedure used in MorphTwin and provides the numerical settings required for reproduction. We focus only on the variables that are optimized in the final pipeline; the material representation and graph-specific response calibration are defined in Sec. 4.

Given an interaction observation $\mathcal{V}$ and its reconstructed Spring-Mass graph $\mathcal{G} = (\mathcal{P}, \mathcal{S})$, we initialize the graph-specific backbone parameters $\mathbf{b}$ and interaction parameters $\boldsymbol{\eta}$ from the physical reconstruction in Sec. 4.1. MorphTwin then fits the material response in the shared bulk-shear response space before optimizing appearance.

### C.1 Material Fitting

**Graph calibration.** For each reconstructed graph, we perform the response calibration described in Sec. 4.3 once at the reference material state. We denote its graph-specific reference response by $\mathbf{q}^0 = \mathbf{q}(\mathbf{b}^0, \mathbf{0})$. The fixed response Jacobian $\mathbf{A}$ converts a shared relative response change $\Delta\mathbf{q} = \mathbf{q}(\mathbf{b}^0, \mathbf{u}) - \mathbf{q}^0$ into the graph-specific carrier control through Eq. 6. The carrier geometry, edge-family assignments, $\mathbf{q}^0$, and $\mathbf{A}$ remain fixed throughout subsequent optimization.

**Physics optimization.** We optimize the material-response change $\Delta\mathbf{q}$ together with the graph-specific backbone parameters $\mathbf{b}$ and interaction parameters $\boldsymbol{\eta}$. At each forward pass, $\mathbf{u}$ is obtained from the current $\Delta\mathbf{q}$ using the fixed calibration, and the corresponding native spring stiffnesses $\mathbf{Y}$ are instantiated according to Eq. 3. The resulting trajectory $\widehat{\mathbf{X}}$ is produced by the explicit Spring-Mass simulator described in Sec. 4.1.

The final material-fitting stage is supervised by the geometry and tracking terms, $\mathcal{L}_{\text{motion}} = \mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{track}}$. The former aligns the simulated surface with the visible reconstructed geometry, while the latter supervises the recovered motion correspondences. Their weights and the remaining optimization hyperparameters are summarized in Table 5.

Importantly, the carrier control $\mathbf{u}$ is not an independent optimization variable. It is recomputed from $\Delta\mathbf{q}$ at every forward pass, so the optimization remains in the shared response space while the graph-specific calibration determines its native spring realization.

**Appearance fitting.** After physics optimization converges, we freeze $\Delta\mathbf{q}$, $\mathbf{b}$, and $\boldsymbol{\eta}$ and optimize only the canonical Gaussian appearance parameters $\varphi$. Gaussian motion is driven by the fixed simulated trajectories through the Gaussian deformation and binding described in Appendix C.6.

**Inference and material morphing.** Future prediction uses the identified material parameters directly, without additional material fitting or response calibration. For material editing, the identified response $\Delta\mathbf{q}$ is replaced by an edited response $\Delta\mathbf{q}'$, which is compiled through the same fixed graph calibration before explicit rollout. For cross-graph transfer, the target graph uses its own calibration matrix while preserving the same response-space target.

### C.2 Optimization and Inference Procedure

Algorithm 1 summarizes the complete optimization and inference procedure. Graph calibration is performed once before material fitting. During physics optimization, the material response is updated in the shared response space and converted to native spring parameters at each forward pass. Physics is then frozen for appearance optimization. The same calibrated graph is directly used for future rollout or material editing.

### C.3 Optimization Settings

Table 5 summarizes the numerical settings used for material fitting. Dataset-specific appearance-training and evaluation settings are reported separately in Appendix E.

### C.4 Carrier Construction and Calibration

Given a reconstructed physical graph $\mathcal{G}_0 = (\mathcal{P}_0, \mathcal{S}_0)$, we augment selected local four-node cells with auxiliary carrier nodes while retaining the original physical nodes and spring connectivity. For each physical node, we consider combinations formed from its $48$ nearest spatial neighbors, deduplicate identical four-node sets, and discard geometrically degenerate candidates. For a candidate $T$ with

**Algorithm 1** MorphTwin optimization and inference.

**Input:** observation $\mathcal{V}$ and reconstructed graph $\mathcal{G} = (\mathcal{P}, \mathcal{S})$
**Physical initialization**
Initialize $\mathbf{b}$ and $\boldsymbol{\eta}$ from the reconstruction in Sec. 4.1
**Graph-specific calibration**
Calibrate the response map at the reference material state
Fix the resulting response Jacobian $\mathbf{A}$
**Material fitting**
Initialize $\Delta\mathbf{q} \leftarrow \mathbf{0}$
**repeat**
  $\mathbf{u} \leftarrow \mathbf{A}^{-1}\Delta\mathbf{q}$
  $\mathbf{Y} \leftarrow \mathrm{Materialize}(\mathbf{b}, \mathbf{u})$
  $\widehat{\mathbf{X}} \leftarrow \mathrm{Simulate}(\mathcal{G}, \mathbf{Y}, \boldsymbol{\eta})$
  Compute $\mathcal{L}_{\mathrm{motion}}(\widehat{\mathbf{X}}, \mathcal{V})$
  Update $\Delta\mathbf{q}, \mathbf{b}, \boldsymbol{\eta}$
**until** convergence
**Appearance fitting**
Freeze $\Delta\mathbf{q}, \mathbf{b}, \boldsymbol{\eta}$
Optimize Gaussian appearance $\varphi$
**Inference / material morphing**
Choose response $\Delta\mathbf{q}'$ ▷ $\Delta\mathbf{q}' = \Delta\mathbf{q}$ for prediction
$\mathbf{u}' \leftarrow \mathbf{A}^{-1}\Delta\mathbf{q}'$
$\mathbf{Y}' \leftarrow \mathrm{Materialize}(\mathbf{b}, \mathbf{u}')$
$\widehat{\mathbf{X}}' \leftarrow \mathrm{Simulate}(\mathcal{G}, \mathbf{Y}', \boldsymbol{\eta})$

Table 5: Optimization settings used for MorphTwin material fitting. The epoch count corresponds to the released solid-object training protocol.

| Setting | Value | Setting | Value |
|---|---|---|---|
| Optimizer | Adam | Material-fitting epochs | 188 |
| Shared-response learning rate | $10^{-3}$ | Local-parameter learning rate | $10^{-3}$ |
| Adam $(\beta_1, \beta_2)$ | $(0.9, 0.99)$ | Adam $\epsilon$ | $10^{-8}$ |
| $\lambda_{\mathrm{geo}}$ | 1 | $\lambda_{\mathrm{track}}$ | 1 |
| Video rate | 30 Hz | Simulation substep $\Delta t$ | $5 \times 10^{-5}$ s |
| Learning-rate decay | None | Substeps per frame transition | 667 |

centroid $\mathbf{c}_T$, we measure its departure from an isotropic local configuration using

$$\mathbf{S}_T = \sum_{i \in T} (\mathbf{x}_i^0 - \mathbf{c}_T)(\mathbf{x}_i^0 - \mathbf{c}_T)^\top, \qquad s_T = \frac{\left\| \mathbf{S}_T - \frac{\mathrm{tr}(\mathbf{S}_T)}{3}\mathbf{I} \right\|_F}{\mathrm{tr}(\mathbf{S}_T)/\sqrt{3}}. \tag{18}$$

We use $s_T \leq 0.10$ for the base candidate pool and $s_T \leq 0.20$ for the extended pool. A bulk carrier places one auxiliary node at $\mathbf{c}_T$ and connects it to the four physical nodes. The basic shear construction places a symmetric pair of auxiliary nodes on the circumsphere of the same cell, with each auxiliary node connected to all four physical nodes, as illustrated in Fig. 3. The shear orientation is selected deterministically from 192 spherical direction candidates after screening for geometric locality, non-degeneracy, and numerical conditioning. Only candidates passing these checks are retained for deployment.

After carrier deployment, we evaluate the augmented graph using the standardized volumetric and deviatoric probes defined in Eq. 5. The resulting bulk-shear response is calibrated to the shared response coordinates using the graph-specific Jacobian $A$ in Eq. 6. We verify that $A$ has full rank and record its singular values and spectral condition number. The latter characterizes the local numerical sensitivity of the response-to-carrier mapping around the calibration state, rather than the accuracy of finite-amplitude material edits.

### C.5 Auxiliary Dynamics

Auxiliary carrier nodes have positive mass and participate directly in the explicit dynamics. To keep the total model mass unchanged when auxiliary degrees of freedom are introduced, we redistribute part of the original physical-node mass to nearby auxiliary nodes. Let $t_{i,a} \geq 0$ denote the amount of mass transferred from physical node $i \in \mathcal{P}_0$ to auxiliary node $a \in \mathcal{P}_{\mathrm{a}}$. The deployed masses are

$$m_i = m_i^0 - \sum_{a \in \mathcal{P}_{\mathrm{a}}} t_{i,a}, \qquad m_a = \sum_{i \in \mathcal{P}_0} t_{i,a}. \tag{19}$$

The coefficients $t_{i,a}$ define a sparse nonnegative mass redistribution on the reference geometry. We favor transfers between spatially nearby physical and auxiliary nodes while enforcing positive deployed masses and preserving the total mass and reference first mass moment. Thus, $t_{i,a}$ only determines how the existing model mass is redistributed; it is not an additional material parameter.

During online simulation, auxiliary and original physical nodes are advanced together by the explicit spring-mass integrator in Sec. 4.1. Auxiliary nodes are therefore not relaxed to a static equilibrium at every simulation step. The minimization in Appendix B.2 is used only to define the static bulk-shear response for material calibration. At finite loading frequencies, auxiliary inertia and damping may additionally affect the dynamic response; accordingly, we do not assume that the calibrated $(\kappa, \mu)$ fully characterize a frequency-independent dynamic modulus.

### C.6 Gaussian Deformation and Binding

For our experiments, Gaussian appearance is driven by the trajectories of the original physical nodes $P_0$; auxiliary carrier nodes participate in dynamics but are not used as appearance-binding supports. For each reference Gaussian center $\mathbf{g}_j^0$, we fix its $K = 16$ nearest physical nodes in the reference configuration and use normalized inverse-distance weights $(\|\mathbf{g}_j^0 - \mathbf{x}_i^0\|_2 + 10^{-6})^{-1}$. The neighborhood and weights remain fixed over time. From the weighted reference covariance $\mathbf{C}_j$ and reference-to-current cross-covariance $\mathbf{H}_j^t$, we estimate a proper local rotation $\mathbf{R}_j^t$ and define

$$\mathbf{F}_j^t = \mathbf{H}_j^t \mathbf{C}_j^+ + \mathbf{R}_j^t \left(\mathbf{I} - \mathbf{C}_j \mathbf{C}_j^+\right), \tag{20}$$

where $\mathbf{C}_j^+$ retains eigendirections above $10^{-6}$ of the largest eigenvalue, while $\mathbf{R}_j^t(\mathbf{I} - \mathbf{C}_j \mathbf{C}_j^+)$ completes the weakly constrained complementary subspace.

The same $\mathbf{F}_j^t$ transports both the Gaussian center and covariance from the reference state, allowing local translation, rotation, stretching, compression, and shear. Appearance optimization may additionally learn a time-shared reference-space center offset whose per-coordinate magnitude is bounded by $2\%$ of the reference Gaussian bounding-box diagonal; it does not introduce independent per-frame motion. Each frame is computed directly from the reference state.

PhysTwin instead blends local rigid transformations and updates Gaussian orientation through blended rotations, without directly transporting covariance by a local affine deformation gradient. Our binding uses $\mathbf{F}_j^t$ to transport both Gaussian centers and covariances, allowing local affine deformation beyond rigid motion. The two pipelines therefore share the same physics-then-appearance optimization strategy but differ in their appearance-deformation model.

## D Dataset Details

**Capture setup and raw recording.** We build a first-person interaction capture system centered on a Meta Quest 3 headset and a data-processing workstation, together with an external Orbbec DaBai RGB-D camera that provides the main third-person RGB-D view. On the headset, stereo RGB images are acquired through the camera interface, while the device depth and motion-tracking interfaces record depth maps, head pose, articulated hand joints, and body pose. We additionally store camera intrinsics and extrinsics together with timestamps for all recorded streams. During capture, an operator manipulates deformable objects on a tabletop experimental platform equipped with a robotic arm through grasping, squeezing, and stretching interactions, and the acquisition program packages the image and tracking streams into multi-track recording files. The raw egocentric RGB streams are recorded at 60 fps with a per-eye resolution of $1280 \times 960$, while headset depth is recorded at $320 \times 320$ with an effective sampling rate of approximately 4.9 Hz in the current collection.

**Post-processing and aligned observations.** During post-processing, we first retain raw recordings with a duration of at least 20 s. We then remove non-interaction segments before and after the manipulation interval and extract the resulting interaction sequences for subsequent processing and evaluation. Subsequently, we decode the recorded streams, match observations by timestamp, verify pose validity, and transform all measurements into consistent camera and world coordinate systems. Depth observations are back-projected into 3D and registered to the corresponding RGB images, producing colored point clouds in both the world coordinate frame and the egocentric camera frame for subsequent object segmentation, deformation tracking, and physical modeling. The Meta Quest 3 streams and the main DaBai RGB-D stream are temporally aligned in post-processing, although hardware-level synchronization is not assumed. For the final experiments, all sequences are processed at $640 \times 360$ with a nominal frame rate of 30 fps. AI agents assist with keypoint annotation and temporal alignment, with every processing stage undergoing systematic manual review and correction where necessary. The resulting aligned multi-view observations support physical-twin reconstruction and future-prediction evaluation using the egocentric views, the RGB-D view, or their combination.

Table 6: Overview of the 15 interaction sequences.

| Object | Morphology | Manipulation | Frames | Duration (s) |
|---|---|---|---|---|
| Fabric Sheet (Trial 1) | Thin-shell | Bimanual | 196 | 6.53 |
| Fabric Sheet (Trial 2) | Thin-shell | Bimanual | 324 | 10.80 |
| Black Fabric | Thin-shell | Bimanual | 361 | 12.03 |
| Mesh Fabric | Thin-shell | Bimanual | 496 | 16.53 |
| Cushion (Trial 1) | Volumetric | Bimanual | 118 | 3.93 |
| Cushion (Trial 2) | Volumetric | Bimanual | 521 | 17.37 |
| Neck Pillow | Volumetric | Bimanual | 471 | 15.70 |
| Plush Toy (Trial 1) | Volumetric | Unimanual | 501 | 16.70 |
| Plush Toy (Trial 2) | Volumetric | Bimanual | 251 | 8.37 |
| Laptop | Volumetric | Bimanual | 431 | 14.37 |
| Book | Volumetric | Bimanual | 356 | 11.87 |
| Water Bottle | Volumetric | Unimanual | 176 | 5.87 |
| Sun Hat | Thin-shell | Bimanual | 695 | 23.17 |
| Headphones | Composite | Bimanual | 436 | 14.53 |
| Smartphone | Volumetric | Unimanual | 407 | 13.57 |
| Total: 15 sequences | | | 5740 | 191.33 |

**Dataset composition and visualization.** Table 6 summarizes the 15 interaction sequences used in the dataset, including object morphology, manipulation type, number of frames, and sequence duration. Figure 7 provides a corresponding visual overview, showing one representative temporally aligned sample from the three camera views for each sequence.

Sequence duration is computed as the number of output frames divided by 30, with the total duration calculated before rounding. Frame counts refer to temporal samples within a sequence rather than the sum across camera views. Each temporal sample contains one main RGB-D observation from the DaBai camera together with the left and right egocentric RGB observations from the Meta Quest 3 headset. We use an approximately 70/30 temporal observation/future split.

**Morphology and manipulation labels.** The morphology labels describe coarse object form rather than measured physical properties. *Thin-shell* denotes objects dominated by a thin deformable surface, *volumetric* denotes objects with substantial three-dimensional bulk, and *composite* denotes multi-part structures. Manipulation is categorized as *unimanual* or *bimanual* according to whether one or two hands are involved, while trial numbers distinguish separate recordings of the same object type.

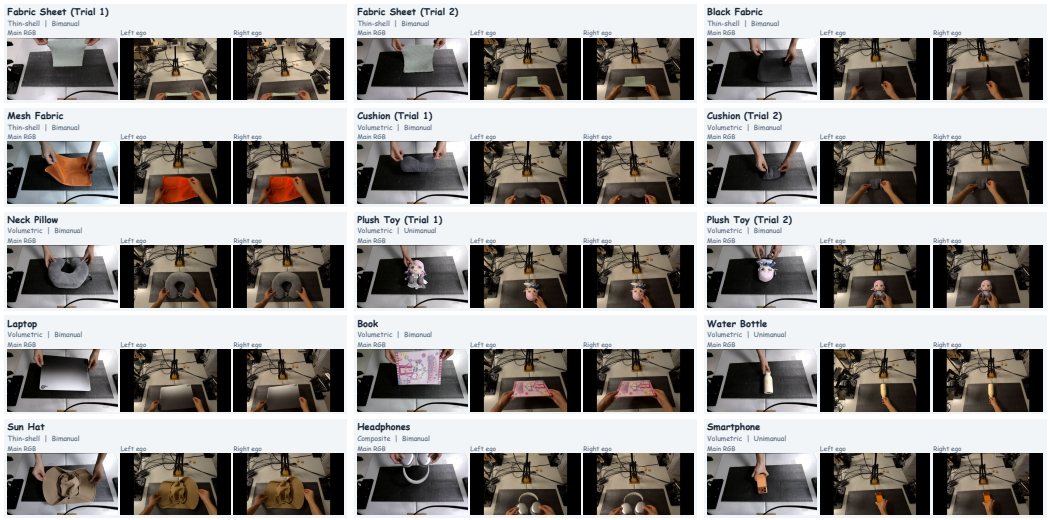


Figure 7: Visual overview of the 15 interaction sequences listed in Table 6. For each sequence, we show a representative temporally aligned sample from the main RGB view and the left and right egocentric views. The morphology and manipulation labels shown in each panel follow the terminology used in Table 6.

# E Experimental Protocol and Additional Results

## E.1 Experimental Protocol

**Real-video evaluation.** Following DeformMaster (Li et al., 2026), we select the same 20 PhysTwin interaction scenes to maintain a consistent scene set for comparison. Let $T_{\text{obs}}$ denote the observation boundary and $T$ the sequence length. Reconstruction image metrics use frames $[0, T_{\text{obs}})$, whereas geometric and tracking metrics exclude the initial frame and use $[1, T_{\text{obs}})$. Future metrics use $[T_{\text{obs}}, T)$. MorphTwin fits physics before optimizing Gaussian appearance with the physical trajectories fixed. Future rollout uses prescribed interaction controls without fitting to future object observations.

**Synthetic material and appearance evaluation.** We use 14 simulation-generated Spring-Gaus development scenes to evaluate material recovery. Their known Young's moduli and Poisson's ratios provide ground-truth references that are unavailable for our real-world recordings. These references enable direct quantitative validation of material recovery using errors in both parameters. MorphTwin and the central-spring PhysTwin adaptation share an inverse frontend supervised by observed particle trajectories, with geometry, initial conditions, and the density assumption fixed.

As defined in Sec. 4.2 and Appendix B.3, the recovered $\kappa$ and $\mu$ are graph-integrated elastic responses, while $\kappa/V$ and $\mu/V$ are the corresponding continuum moduli. Dynamic observations constrain elastic stiffness relative to mass, so reporting an absolute Young's modulus additionally requires a density convention. We use the fixed density of $1000\,\mathrm{kg/m^3}$ for all synthetic material recovery experiments. The reported absolute Young's-modulus values are therefore conditional on this density assumption, whereas Poisson's ratio and the relative logarithmic response change $\Delta\mathbf{q}$ are independent of the common absolute stiffness-mass scale.

Methods share a canonical initialization fitted for 30,000 iterations, followed by 10,000 appearance-fitting iterations per method. Cameras other than camera 5 provide training images, while camera 5 is held out for evaluation at $512 \times 512$. Frames 0–19 are observed and frames 20–29 are future. We use the prespecified final appearance checkpoint.

**Self-captured egocentric sequences.** The current collection contains 15 sequences and 5,740 temporal samples, corresponding to 191.33 seconds at a nominal 30 fps. Each sample includes left and right egocentric views and a main RGB-D view, processed at $640 \times 360$. The streams are temporally aligned; hardware synchronization is not assumed. We use an approximately 70/30

Table 7: Physics-training cost and inference efficiency on the 20 PhysTwin scenes. Bold indicates the highest reported FPS and lowest reported memory per group.

| Scene type | Method | Training ↓ (min/scene) | Offline FPS ↑ | Interactive FPS ↑ | Memory ↓ (GiB) |
|---|---|---|---|---|---|
| **1D curves** (3 scenes) | PhysTwin | 4.78 | 85.00 | 50.68 | 1.01 |
| | DeformMaster | – | 37.42 | 28.36 | **0.81** |
| | **MorphTwin** | 10.82 | **106.99** | **62.87** | 1.12 |
| **2D surfaces** (9 scenes) | PhysTwin | 72.05 | 24.72 | 19.47 | **1.79** |
| | DeformMaster | – | 22.35 | 18.25 | 2.18 |
| | **MorphTwin** | 109.20 | **26.26** | **21.47** | 2.02 |
| **3D solids** (8 scenes) | PhysTwin | 7.60 | 47.75 | 33.77 | 1.65 |
| | DeformMaster | – | 18.55 | 15.63 | 3.59 |
| | **MorphTwin** | 16.32 | **67.86** | **45.98** | **1.44** |

temporal observation/future split, with frame counts referring to temporal samples. For each input-view configuration, geometry reconstruction and the resulting 3D motion tracks are obtained by fusing only the designated input streams: the main RGB-D and two egocentric views for *All Views*, the two egocentric views for *Egocentric*, and the main RGB-D stream for *Single RGB-D*. The same designated views supervise Gaussian appearance optimization, while PSNR, SSIM, and LPIPS are evaluated over all three views under a common protocol. Future rollout does not use future object geometry, motion, or appearance observations; it is conditioned on prescribed interaction controls calibrated from the complete interaction recording and then held fixed during prediction.

**Metric definitions.** Local PhysTwin evaluation uses camera 0, black backgrounds, hand exclusion, and AlexNet LPIPS. Chamfer distance is the directed visible-ground-truth-to- prediction nearest-neighbor $L_1$ distance in meters. Tracking error is the mean $L_2$ distance using fixed particle associations established at frame 0. Spring-Gaus appearance evaluation uses VGG LPIPS; its separate geometry diagnostics use symmetric unsquared Chamfer distance and correspondence RMSE in millimeters. These geometry metrics are not interchangeable across datasets. Material errors are $100|\hat{E} - E_{\mathrm{GT}}|/E_{\mathrm{GT}}$ and $|\hat{\nu} - \nu_{\mathrm{GT}}|$. Local aggregate scores weight scenes equally. Controlled response and cross-graph experiments are reported separately from video metrics, with their controls and error definitions specified alongside the corresponding experiments.

### E.2 Material Editing and Reuse

**Cross-interaction reuse of video-recovered physical models.** We further test whether physical behavior recovered from one interaction video can be reused for another interaction of the same object. We evaluate *Fabric Sheet (Trial 2)→Fabric Sheet (Trial 1)* and *Cushion (Trial 1)→Cushion (Trial 2)*. For each pair, we reuse the physical model optimized from the source video, including its spring network, carrier parameters, internal rest lengths, masses, damping, and contact settings. The source model is rigidly aligned to the target first frame by vertical-axis rotation and translation, without scaling or non-rigid alignment; only the control-spring rest lengths are reset to the target initial configuration. The target hand trajectory is provided as the prescribed interaction input, and the model is rolled out continuously over the full target sequence without target-specific physical optimization or a state reset at the future boundary. We compare against MorphTwin and PhysTwin models fitted locally on the target recording.

For *Fabric Sheet*, the transferred model remains close to the locally fitted models on the observed interval and reduces future CD and Track errors by $20.8\%$ and $13.8\%$, respectively, relative to local PhysTwin. For *Cushion*, transfer has larger modeling errors, but its future CD is $10.3\%$ lower than local PhysTwin and its future Track differs by only $0.07$ mm, while the locally fitted MorphTwin remains more accurate. The fabric case also exhibits run-to-run variation, so these numbers should be interpreted as results from the evaluated transfer run. Together with the controlled cross-graph study below, these results support our central insight: reusable physical assets can be built by expressing material behavior through shared mechanical response targets and realizing those targets with graph-specific spring parameters.

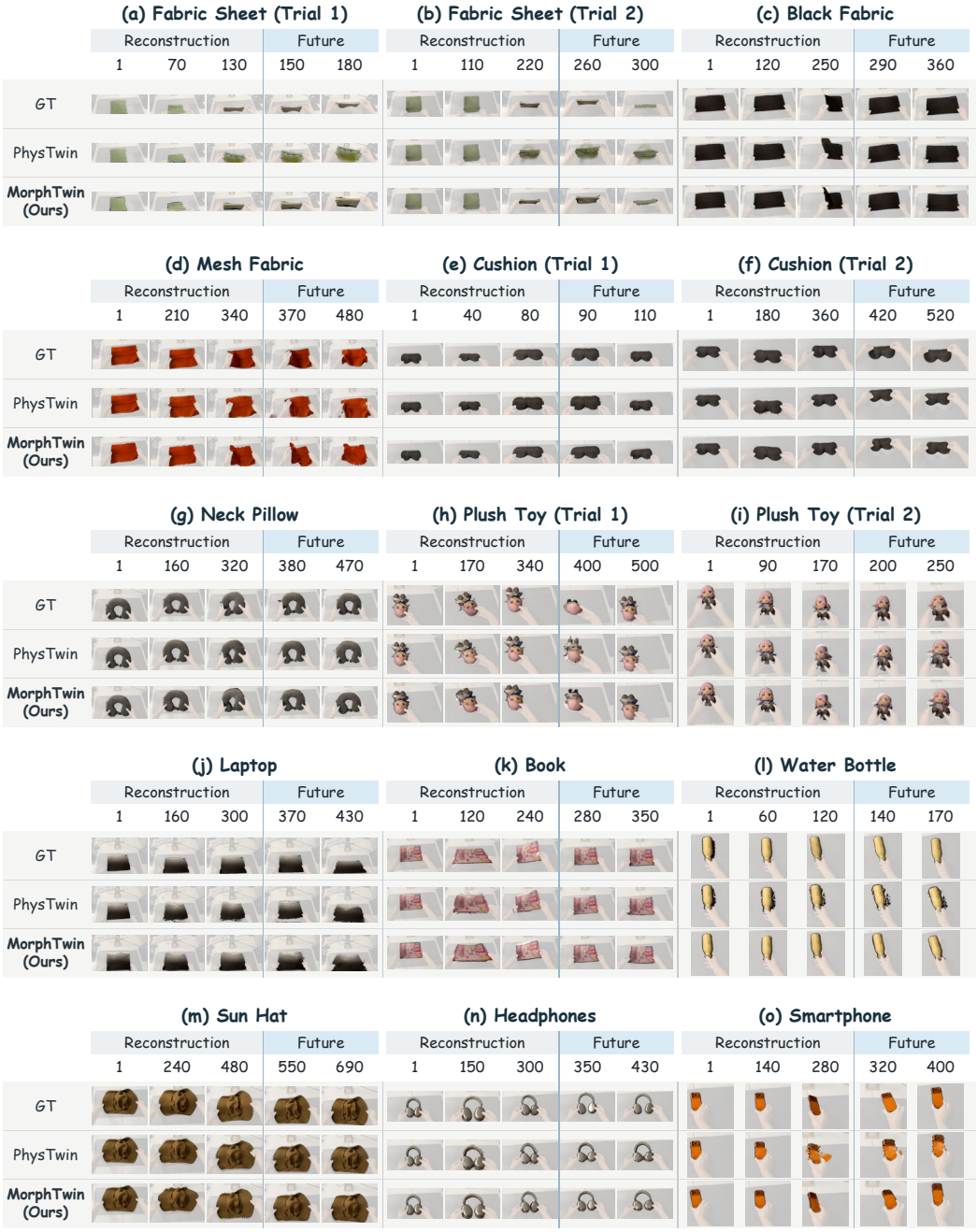


Figure 8: Reconstruction and future prediction on all 15 self-captured interaction sequences under the All Views setting. Each panel shows representative reconstruction and future frames for GT, PhysTwin, and MorphTwin (Ours). GT is displayed separately to facilitate direct comparison of object motion and deformation across methods. The corresponding per-scene quantitative results are reported in Table 8. We strongly recommend viewing *Supplementary Video-1* for a clearer comparison of motion and deformation over the full sequences.

Table 8: Per-scene comparison of PhysTwin and MorphTwin under the All Views setting on 15 self-captured scenes. CD and Track are measured in meters. Bold indicates the better result between the two methods for each scene and metric; ties at the displayed precision are both highlighted.

| | | Reconstruction | | | | | Future Prediction | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Object | Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | Track ↓ |
| Fabric Sheet (Trial 1) | PhysTwin | 25.94 | 0.9608 | 0.0481 | 0.01002 | 0.0231 | 20.85 | 0.9366 | 0.0994 | **0.03182** | 0.0629 |
| | **MorphTwin (Ours)** | **31.72** | **0.9819** | **0.0390** | **0.00948** | **0.0201** | **22.54** | **0.9526** | **0.0805** | 0.03513 | **0.0626** |
| Fabric Sheet (Trial 2) | PhysTwin | 30.28 | 0.9561 | **0.0350** | **0.00490** | **0.0095** | 23.09 | 0.9157 | 0.0955 | 0.02071 | 0.0441 |
| | **MorphTwin (Ours)** | **33.72** | **0.9778** | 0.0362 | 0.00562 | 0.0119 | **24.92** | **0.9518** | **0.0681** | **0.01806** | **0.0438** |
| Black Fabric | PhysTwin | 33.31 | 0.9300 | **0.0855** | 0.01221 | 0.0184 | 31.96 | 0.9260 | 0.0971 | 0.02013 | 0.0452 |
| | **MorphTwin (Ours)** | **37.82** | **0.9624** | 0.0935 | **0.00947** | **0.0149** | **35.70** | **0.9545** | **0.0872** | **0.01272** | **0.0318** |
| Mesh Fabric | PhysTwin | 22.65 | 0.8521 | **0.1524** | **0.01591** | **0.0333** | 22.31 | 0.8513 | **0.1785** | **0.01654** | 0.0360 |
| | **MorphTwin (Ours)** | **25.97** | **0.8929** | 0.1629 | 0.02662 | 0.0516 | **23.41** | **0.8704** | 0.1938 | 0.01764 | **0.0306** |
| Cushion (Trial 1) | PhysTwin | 36.88 | 0.9802 | 0.0285 | 0.00492 | **0.0075** | **31.95** | 0.9622 | 0.0468 | 0.01542 | 0.0239 |
| | **MorphTwin (Ours)** | **41.13** | **0.9900** | **0.0146** | **0.00485** | 0.0080 | 31.64 | **0.9689** | **0.0412** | **0.01070** | **0.0190** |
| Cushion (Trial 2) | PhysTwin | 35.71 | 0.9802 | 0.0261 | 0.00865 | 0.0144 | 30.36 | 0.9602 | 0.0692 | 0.01606 | 0.0394 |
| | **MorphTwin (Ours)** | **40.22** | **0.9893** | **0.0172** | **0.00514** | **0.0101** | **30.61** | **0.9612** | **0.0673** | **0.01281** | **0.0332** |
| Neck Pillow | PhysTwin | 27.70 | 0.9390 | 0.0620 | 0.00945 | **0.0157** | 24.97 | 0.9181 | 0.1018 | **0.01329** | **0.0253** |
| | **MorphTwin (Ours)** | **29.72** | **0.9605** | **0.0532** | **0.00925** | 0.0178 | **25.46** | **0.9317** | **0.0915** | 0.01366 | 0.0307 |
| Plush Toy (Trial 1) | PhysTwin | 26.05 | 0.9572 | **0.0389** | **0.00613** | **0.0118** | 22.97 | 0.9529 | 0.0690 | 0.01337 | 0.0256 |
| | **MorphTwin (Ours)** | **28.00** | **0.9666** | 0.0443 | 0.00615 | 0.0124 | **24.38** | **0.9601** | **0.0561** | **0.01314** | **0.0251** |
| Plush Toy (Trial 2) | PhysTwin | 28.41 | 0.9696 | 0.0266 | **0.00515** | **0.0100** | 22.89 | 0.9527 | 0.0560 | **0.01194** | 0.0361 |
| | **MorphTwin (Ours)** | **32.15** | **0.9825** | **0.0233** | 0.00552 | 0.0112 | **24.23** | **0.9580** | **0.0509** | 0.01283 | **0.0335** |
| Laptop | PhysTwin | 20.70 | 0.9036 | 0.1599 | **0.02745** | **0.0541** | 19.56 | 0.9134 | 0.1449 | 0.01796 | 0.0431 |
| | **MorphTwin (Ours)** | **24.57** | **0.9517** | **0.1099** | 0.03352 | 0.0584 | **23.03** | **0.9408** | **0.1125** | **0.01598** | **0.0395** |
| Book | PhysTwin | 23.92 | 0.9098 | **0.0729** | **0.01008** | **0.0154** | 22.14 | 0.9003 | **0.0785** | **0.01641** | **0.0242** |
| | **MorphTwin (Ours)** | **24.72** | **0.9298** | 0.0812 | 0.01344 | 0.0183 | **22.33** | **0.9102** | 0.0944 | 0.01857 | 0.0271 |
| Water Bottle | PhysTwin | 32.41 | 0.9887 | 0.0204 | **0.00502** | **0.0078** | 28.61 | 0.9826 | 0.0324 | **0.00644** | 0.0120 |
| | **MorphTwin (Ours)** | **35.02** | **0.9932** | **0.0128** | 0.00508 | 0.0080 | **30.07** | **0.9880** | **0.0209** | 0.00652 | **0.0117** |
| Sun Hat | PhysTwin | 26.00 | 0.8901 | **0.0925** | **0.01084** | **0.0183** | 22.94 | 0.8643 | 0.1330 | **0.01435** | 0.0295 |
| | **MorphTwin (Ours)** | **28.63** | **0.9252** | 0.0968 | 0.01146 | 0.0194 | **24.40** | **0.8889** | **0.1292** | 0.01487 | **0.0291** |
| Headphones | PhysTwin | 28.94 | 0.9721 | 0.0245 | **0.00588** | **0.0091** | 25.66 | 0.9628 | 0.0356 | **0.00984** | **0.0132** |
| | **MorphTwin (Ours)** | **32.05** | **0.9822** | **0.0180** | 0.00597 | **0.0091** | **27.04** | **0.9689** | **0.0296** | 0.01000 | 0.0135 |
| Smartphone | PhysTwin | 32.16 | 0.9870 | 0.0245 | **0.00524** | **0.0099** | 28.84 | 0.9820 | 0.0368 | 0.01530 | 0.0217 |
| | **MorphTwin (Ours)** | **36.40** | **0.9924** | **0.0133** | 0.00587 | 0.0109 | **31.81** | **0.9873** | **0.0208** | **0.01370** | **0.0197** |

Table 9: Cross-interaction reuse of video-recovered physical models. CD and Track are reported in millimeters (mm). Transfer reuses the source-video physical model without target-specific physical fitting during rollout.

| Method | Modeling CD ↓ | Modeling Track ↓ | Future CD ↓ | Future Track ↓ |
|---|---|---|---|---|
| *Fabric Sheet (Trial 2) → Fabric Sheet (Trial 1)* | | | | |
| MorphTwin transfer | 10.05 | 20.62 | **25.19** | **54.19** |
| MorphTwin local | **9.48** | **20.05** | 35.13 | 62.56 |
| PhysTwin local | 10.02 | 23.06 | 31.82 | 62.88 |
| *Cushion (Trial 1) → Cushion (Trial 2)* | | | | |
| MorphTwin transfer | 10.39 | 21.62 | 14.41 | 39.52 |
| MorphTwin local | **5.14** | **10.14** | **12.81** | **33.21** |
| PhysTwin local | 8.65 | 14.41 | 16.06 | 39.45 |

**Shared material responses across different graphs.** Material reuse in our formulation means preserving the same calibrated relative mechanical-response change while adapting its discrete spring realization to each graph. We evaluate this principle using *apple*, *E*, and *rope*, with source graphs containing 2,048 particles and target graphs containing 512, 1,024, or 2,048 particles. Three Poisson-ratio settings (0.15, 0.25, and 0.35) yield 72 non-identity transfers per method.

All graphs use unit total mass, so the reported elastic scales are mass-normalized responses rather than Pa-valued continuum moduli. In this controlled experiment, the transfer target is the source graph's measured mass-normalized bulk–shear response. Each target graph realizes this response through a target-specific nonlinear response table calibrated offline with virtual small-strain equilibrium probes; spring parameters are obtained by interpolation and stiffness rescaling without fitting target motion. This complements the local Jacobian formulation in Sec. 4.3, which parameterizes relative response changes, and directly evaluates response preservation across geometries and resolutions.

Table 10 shows that target-graph calibration reduces mean response error from $9.5173\%$ to $0.000503\%$ compared with copying spring multipliers. It also reduces energy error over 24 unseen strain directions from $10.4833\%$ to $3.6203\%$. These results support separating transferable mechanical-response targets from graph-dependent spring parameters: different geometries and discretizations can realize the same bulk–shear response through their own calibration.

Table 10: Controlled cross-graph response transfer on the *apple*, *E*, and *rope* geometries from the Spring-Gaus dataset. Three source material settings ($\nu = 0.15,\ 0.25,\ 0.35$) on 2,048-particle graphs are transferred to target graphs with 512, 1,024, or 2,048 particles. Transfer targets are the measured mass-normalized bulk-shear response values of the source graph under unit total mass. Each strategy is evaluated on the same 72 non-identity conditions. Response and energy errors are percentages; $\nu$ error is absolute. Bold indicates the lower error within each scope.

| Scope | Transfer strategy | Cases | Response error (%) ↓ | $\nu$ error ↓ | Energy error (%) ↓ |
|---|---|---|---|---|---|
| All transfers | Copy spring multipliers | 72 | 9.5173 | 0.01435 | 10.4833 |
| | Target-graph calibration | 72 | **0.000503** | $\mathbf{2.80\times10^{-7}}$ | **3.6203** |
| Cross-object | Copy spring multipliers | 54 | 9.3973 | 0.01560 | 10.3014 |
| | Target-graph calibration | 54 | **0.000482** | $\mathbf{2.70\times10^{-7}}$ | **3.5529** |
| Same object, new resolution | Copy spring multipliers | 18 | 9.8773 | 0.01059 | 11.0291 |
| | Target-graph calibration | 18 | **0.000565** | $\mathbf{3.12\times10^{-7}}$ | **3.8223** |

**Material editing under fixed interaction forces.** A reusable material representation should also support changing an object's response without reconstructing its geometry. Figure 9 demonstrates this capability under identical interaction forces. Scaling spring stiffness from $0.25\times$ to $4\times$ reduces the grip-span increase at $0.60$ s from $6.78$ to $1.47$ cm. The original setting lies between these responses, while a separate rigid-projection control preserves the grip span. These results show how one reconstructed asset can support different interaction behaviors through explicit material control, without retraining geometry or appearance.

**Exchanging materials while preserving geometry.** We further exchange two material presets between the native *apple* and *torus* graphs (Figure 10). Under the unit-total-mass convention described above, the presets use mass-normalized elastic scales of $100$ and $400$ and Poisson ratios of $0.15$ and $0.35$, respectively. These scale values represent the stiffness-mass normalization used in this controlled experiment and are not continuum Young's moduli in Pa. Each object retains its geometry and connectivity. For each donor preset, its measured mass-normalized bulk-shear response change $\Delta\mathbf{q}$ defines the transfer target. The receiving graph realizes this target through its own offline response table, using interpolation and stiffness rescaling to determine its spring parameters without fitting target motion. Under fixed supports and loading within each object, exchanging the material presets produces distinct particle deformations. Across the two transfer directions, target-graph calibration reduces bulk-shear response error from $1.52\%$ and $4.90\%$ with copied multipliers to below $0.00013\%$. This illustrates our central insight: material-response targets can be shared across objects while their spring realization adapts to each object's graph.

### E.3 Structural damage and connectivity-aware appearance updates

MorphTwin realizes material responses through an explicit spring graph. This representation also provides a direct connection between structural damage and appearance updates: when springs fail, the surviving connections determine both the remaining mechanical support and the local support for Gaussian appearance transport. We examine this connection through a controlled fracture demonstration and an ablation of connectivity-aware appearance binding.

**From calibrated responses to structural damage.** Material calibration specifies the intact-graph response. To model connection failure, we introduce an activity variable $a_e \in \{0, 1\}$ for each spring:

$$U(\mathbf{x}, \mathbf{a}; \mathbf{z}) = \sum_e a_e\, U_e(\mathbf{x}; C_{\mathcal{G}}(\mathbf{z}))\,, \tag{21}$$

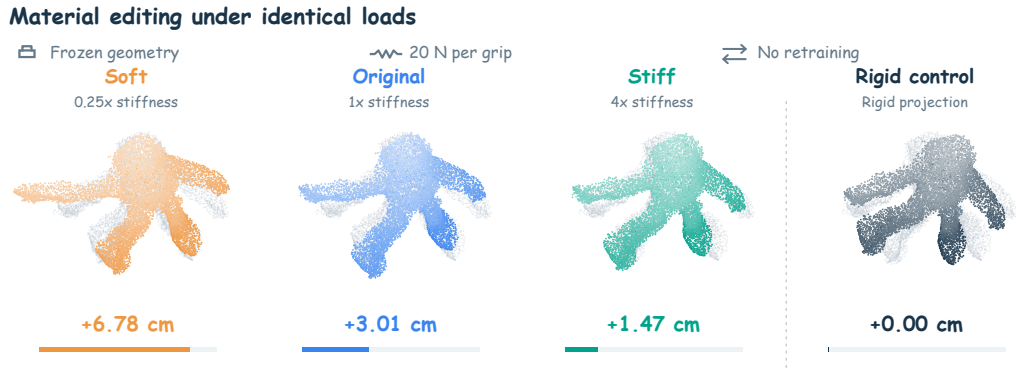


Figure 9: Material editing under identical interaction forces. From left to right, soft ($0.25\times$), original ($1\times$), and stiff ($4\times$) elastic settings produce progressively smaller deformations, illustrating direct material-response control; a rigid-projection control is shown alongside. Gray particles indicate the initial configuration, and annotations report displacement. All panels use the same reconstructed object, applied forces, camera, scale, and observation time.

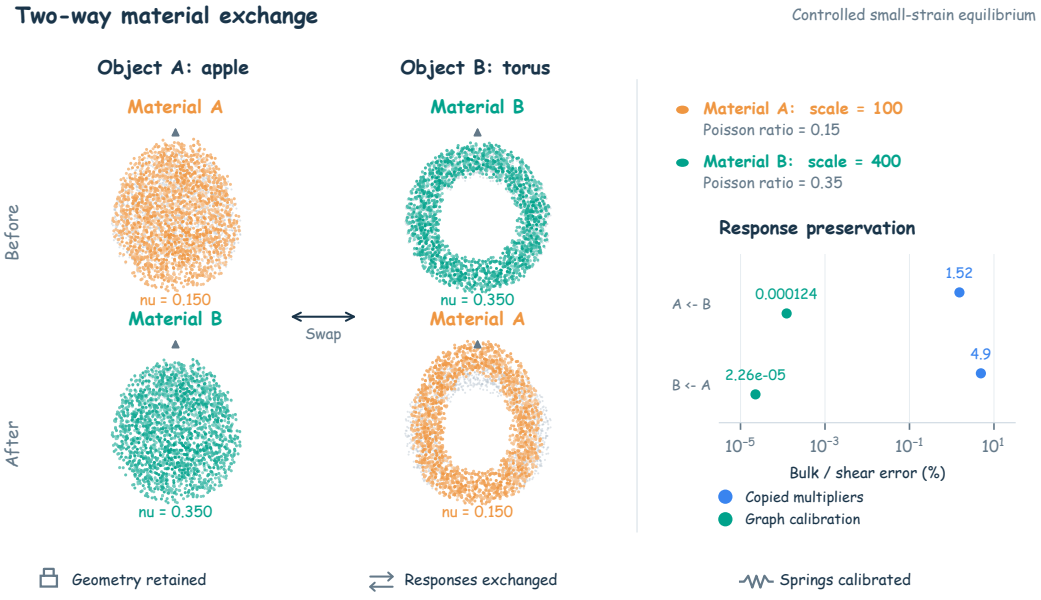


Figure 10: Two-way material-response exchange in controlled small-strain equilibrium. Each object retains its geometry, supports, and loading. Target-graph calibration transfers the donor's measured mass-normalized bulk–shear response more accurately than copying spring multipliers. Elastic scales follow the unit-total-mass convention and are not Pa-valued moduli; displacements are magnified $20\times$ for visualization.

where $\mathbf{z}$ denotes material-response coordinates and $C_{\mathcal{G}}$ maps them to graph-specific spring parameters. A failed connection contributes no elastic energy. The parameters of surviving springs remain fixed, so changes in the effective structural response arise from the loss of connections, without recalibrating the graph after each failure.

We demonstrate this behavior on the cloth twin using two opposing grasps of radius 36 mm, each moving 12 cm along a half-cosine trajectory. The controller force is capped at 40 N per grasp, with unit total mass and gravity disabled. Connections fail irreversibly when their tensile strain exceeds the prescribed threshold of $0.35$. Figure 11 shows the resulting Gaussian appearance, deformed particles, and failed connections. At 1.2 s and 2.0 s, respectively, 648 and 1,189 connections have failed, while the main connected component retains its 5,881 nodes. Local cracks therefore develop before the object separates into disconnected components, motivating appearance updates that follow local changes in connectivity.

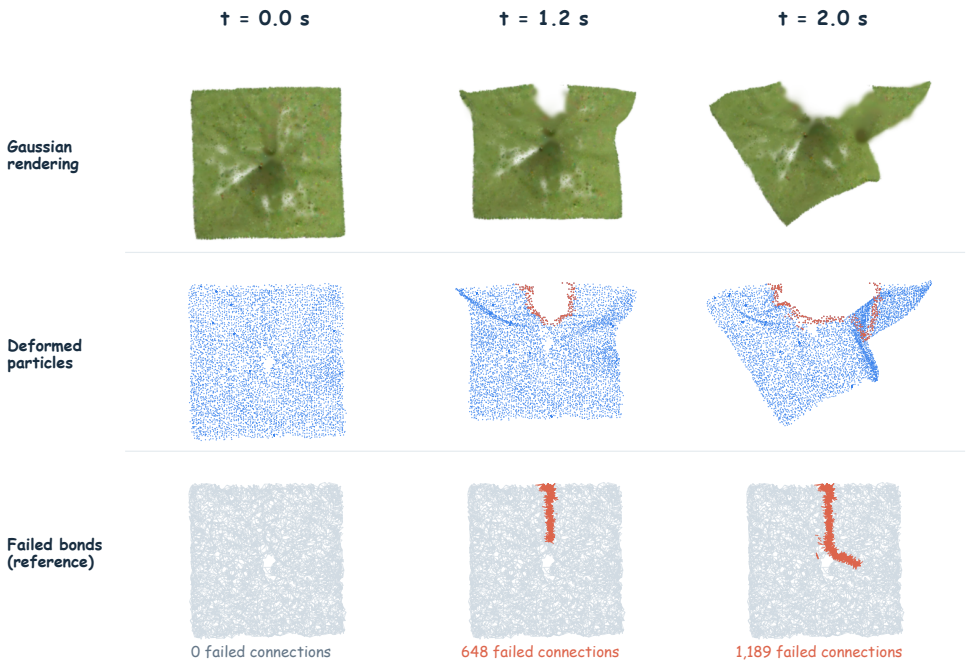


Figure 11: Progressive crack formation in the cloth twin under two opposing grasps. Columns show successive interaction stages. Top: Gaussian renderings show the evolving cloth appearance. Middle: deformed particles, with failed-connection endpoints highlighted in orange. Bottom: failed connections in reference coordinates reveal the spatial progression of damage independently of deformation. The main component remains connected as the crack opens, motivating appearance bindings that follow surviving local connections before complete separation.

**Local connectivity for appearance transport.** We isolate the effect of local connectivity by comparing three binding strategies: fixed K16 binding, which binds each appearance tracer to its 16 nearest physical nodes in the reference configuration; component-only restriction, which retains candidates in the same connected component; and restriction to surviving local connections. The evaluation uses grids with a partial slit and a remaining connection bridge. We evaluate nine combinations of three grid resolutions ($16 \times 16$, $24 \times 24$, and $32 \times 32$) and three opening widths (5, 10, and 20 mm). Each condition uses 4,096 appearance tracers and the same prescribed displacement field, with analytic tracer positions as the reference. Initial binding candidates, weights, regularization, and deformation limits are shared across methods.

Table 11 shows that restricting bindings to surviving local supports reduces mean tracer error from $0.313$ to $0.049$ mm and P95 error from $1.957$ to $0.305$ mm. Component-only restriction gives the same results as fixed binding because the grid remains globally connected. These results show that local connectivity helps avoid interpolation across an opening crack before complete separation. The

explicit graph thus serves both as the realization of calibrated material responses and as the structural support for appearance updates as connections fail.

Table 11: Controlled binding diagnostic over nine conditions. Errors are in millimeters relative to the analytic reference and are equally averaged across conditions; P95 is computed per condition.

| Binding support | Mean error ↓ | P95 error ↓ | Near-slit error ↓ |
|---|---|---|---|
| Fixed K16 | 0.313 | 1.957 | 1.940 |
| Component only | 0.313 | 1.957 | 1.940 |
| Surviving local support | **0.049** | **0.305** | $< 10^{-9}$ |

## F INTERACTIVE INTERFACE.

Figure 12 illustrates the interaction workflow demonstrated in Supplementary Video 2. Users load reconstructed twins into a shared scene and manipulate them through single, dual, or multiple grasps, with adjustable locations, radii, and force limits. They can adjust elastic scale, configure rigid and fracture behavior, and reuse interaction presets while retaining the target's reference geometry and connectivity. Running or pausing the explicit simulation lets users inspect deformation, contact, and connection failure under new manipulations.

MorphTwin's response-preserving material reuse operates in the shared volumetric-shear response space and uses target-graph energy calibration, as evaluated in our material-transfer experiments. The interface's preset-import shortcut instead matches a bulk-response scale and copies interaction settings; it is distinct from this calibrated transfer.

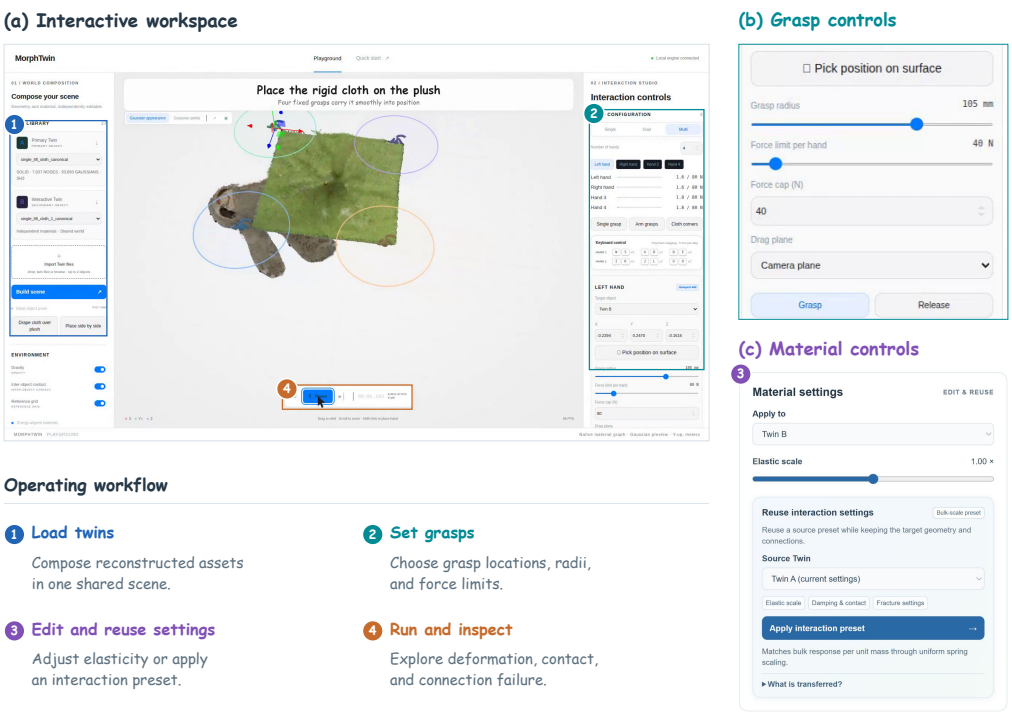


Figure 12: **MorphTwin's interactive interface for composing and manipulating reconstructed physical twins.** *Part (a):* Shared workspace; numbered markers link controls to the operating workflow. *Part (b):* Grasp controls for selecting positions, adjusting radii and force limits, and grasping or releasing objects. *Part (c):* Elastic editing and interaction-preset reuse. Workspace and grasp views are from *Supplementary Video 2*, and part (c) shows the revised material-control layout.