HermiteTwin: Bicubic Hermite Shell Digital Twins for Contact-Rich Deformable Manipulation

Anonymous authors
Paper under double-blind review

Abstract

While existing methods for reconstructing deformable object manipulation have made impressive progress, they typically represent a thin object with a single particle set that must serve two purposes at once: modeling bending, and resolving contact with the robot. To bridge this gap, we present HermiteTwin, a framework that reconstructs a physical digital twin of a thin deformable object from multiview video of a robot manipulating it. Our key idea is to represent deformation and contact separately, each at its own resolution. The object is simulated as a bicubic Hermite shell, a coarse grid of C^1 elements whose nodal positions and derivatives define a smooth surface with Kirchhoff–Love bending, while the gripper acts through dense samples in material space that are captured, clamped, and released, and whose attachment forces reach the shell through the Hermite basis. This decoupling keeps the physical state compact while contact stays fine-grained. We identify effective material and contact parameters with CMA-ES followed by adjoint-based refinement, and drive a reconstructed Gaussian appearance model with the simulated shell. In experiments on real cloth and foam manipulation, HermiteTwin outperforms the state-of-the-art spring–mass twin across reconstruction, resimulation, and future prediction.

1 Introduction

Robots that fold, lift, or spread cloth would benefit from a simulator that behaves like the specific object in front of them. Physical digital twins reconstructed from video provide such simulators: they recover the geometry, dynamics, and appearance of an object from a recorded interaction, and can then resimulate that interaction, predict its continuation, or render the outcome of new actions (Jiang et al., 2025). Simulations of this kind are already used to evaluate robot policies without physical trials (Zhang et al., 2025). How faithful a twin can be depends on two modeling choices: the physical representation of the object, and the way contact with the robot enters that representation.

Thin deformable objects make both choices difficult. Cloth resists stretching but bends easily, so its shape is dominated by curvature, and a gripper acts on it through a small patch of material that is captured, clamped, and later released. PhysTwin (Jiang et al., 2025) handles both with one spring–mass system, whose particles represent the object and also connect it to the gripper. We present HermiteTwin, a digital twin for thin deformable objects that represents deformation and contact separately, each at its own resolution (Figure 1). Its physical state is a coarse bicubic Hermite shell (Ni et al., 2024), a smooth (C^1) surface on which Kirchhoff–Love bending is defined directly and has its own stiffness. Contact, in contrast, is resolved in material space: the gripper captures, clamps, and releases dense material samples, each of which acts on the shell through the Hermite basis. The grasp can therefore be resolved finely while the shell itself stays coarse.

From multiview video of a robot interaction, we reconstruct the object's rest shape and a Gaussian appearance model (Kerbl et al., 2023), identify effective material and contact parameters by inverse simulation, and render the simulated shell by moving the Gaussians with it. Grasp events make the identification objective nonconvex and only piecewise smooth, so

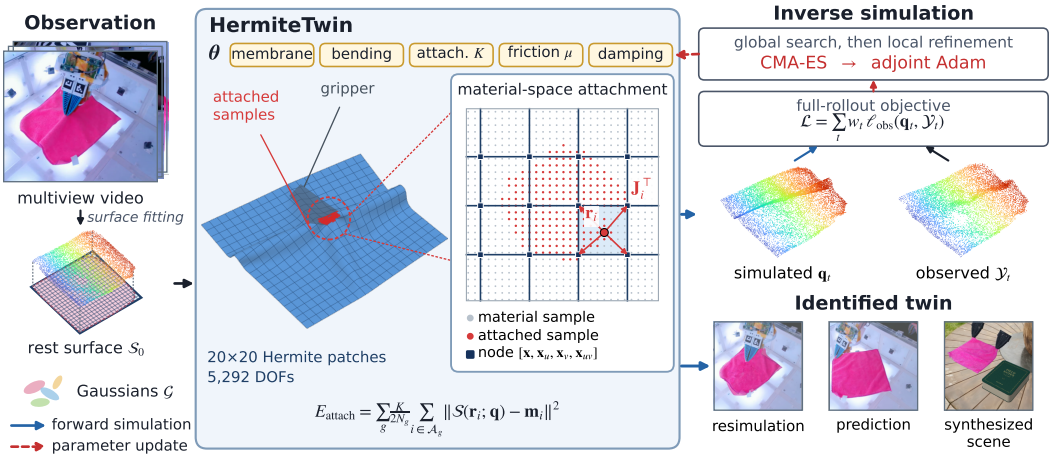


Figure 1: Overview of HermiteTwin. From multiview video of a robot manipulating cloth, we fit a rest surface $\mathcal{S}_0$ to the first-frame object points and reconstruct a Gaussian appearance model $\mathcal{G}$. HermiteTwin simulates the cloth as a coarse bicubic Hermite shell whose nodes store positions and derivatives. The gripper captures dense samples in material space (red); each attached sample $\boldsymbol{r}_i$ lies in one element, and its attachment residual reaches the 16 nodal vectors of that element through the Hermite weights $\boldsymbol{J}_i^\top$ (inset). The effective parameters $\boldsymbol{\theta}$ are identified by minimizing a full-rollout observation loss with CMA-ES followed by adjoint-based Adam. The identified twin resimulates the recorded motion, predicts held-out frames, and drives the Gaussian appearance in synthesized scenes.

we combine global search with CMA-ES (Hansen & Ostermeier, 2001) and local refinement with adjoint gradients.

We evaluate HermiteTwin on real robot manipulation from the Deform360 dataset (Li et al., 2026), comparing it with PhysTwin and the learned point-based simulator GS-Dynamics (Zhang et al., 2024). HermiteTwin reproduces the observed motion more accurately in 3D and in images and predicts the continuation of the recorded interactions more accurately. On a diagnostic interaction, transferring our grasp model to PhysTwin's particles does not recover the observed fold, which suggests that the shell representation, not only the grasp model, contributes to the difference. The identified twins can also be driven by new actions, such as draping the cloth over a book or placing it inside a box. In summary, our contributions are:

- a physical digital twin for thin deformable objects built on a bicubic Hermite shell, a compact C^1 state with explicit Kirchhoff–Love bending;
- a material-space grasp model that captures, clamps, and releases dense material samples and couples them to the shell, so that the grasp is resolved finely on a coarse shell;
- an identification pipeline combining CMA-ES with adjoint refinement, and an evaluation on real manipulation data showing that HermiteTwin resimulates observed motion and predicts future frames more accurately than spring–mass and learned baselines.

2 Related Work

Physical digital twins of deformable objects. PhysTwin (Jiang et al., 2025) reconstructs a spring–mass model and Gaussian appearance from sparse-view videos and evaluates reconstruction, future prediction, and generalization to new interactions. Gaussian-splatting simulations of soft-body interactions have also been used to evaluate robot policies (Zhang et al., 2025). Learning-based simulators predict the motion of particles or Gaussians directly: GS-Dynamics (Zhang et al., 2024) and ParticleFormer (Huang et al., 2025) learn

point-based dynamics, and SoMA (Huang et al., 2026) learns an action-conditioned Gaussian simulator for robotic manipulation. PhysReal (Deng et al., 2026) combines analytical and neural constitutive models. PhysHanDI (Lee et al., 2026) drives a spring–mass object with dense hand vertices through virtual springs and assumes that the hand–object contact topology stays fixed within a sequence; our grasp samples are instead captured and released as the gripper closes and opens. These methods address a broad range of objects. We focus on thin sheets that can be parameterized over a single rectangle, for which a shell state and material-space contact are natural.

Gaussian appearance for deformable objects. 3D Gaussian Splatting (Kerbl et al., 2023) represents scenes with anisotropic primitives that can be rendered efficiently and differentiably. PhysAvatar (Zheng et al., 2024) and Gaussian Garments (Rong et al., 2024) couple simulated garment geometry with appearance reconstructed from multiview video, and GeoSplatting (Ye et al., 2025) uses explicit geometry to guide physically based inverse rendering. Cloth-Splatting (Longhini et al., 2024) estimates cloth states from RGB supervision, and GauSim (Shao et al., 2024) simulates elastic objects with continuum-inspired Gaussian kernels. We keep appearance separate from mechanics: the Gaussians are transported by the simulated shell, and no image loss enters parameter identification, so the same appearance model can be driven by different simulators and by new actions.

Thin shells and differentiable simulation. Bicubic Hermite elements (Ni et al., 2024) provide C^1 continuity through explicit derivative degrees of freedom and model Kirchhoff–Love thin shells on coarse quadrilateral grids. B-spline shells also provide C^1 continuity and embed a contact mesh whose resolution can be chosen independently of the shell (Meng et al., 2026). We use a Hermite shell as the state of a digital twin and apply separate material-space sampling to robot grasps, including capture and release. Collisions are handled with incremental potential contact (Li et al., 2020). Differentiable simulators such as DiffTaichi (Hu et al., 2020) enable gradient-based identification, but grasp events make our objective only piecewise smooth. We therefore combine derivative-free search (Hansen & Ostermeier, 2001) with adjoint gradients computed for a fixed contact configuration.

3 HermiteTwin

3.1 Problem Setup

We are given calibrated multiview videos of a robot manipulating an object and the gripper trajectory $\boldsymbol{a}_{0:T}$ with its clamp and release schedule. From these inputs we build a twin

$$\mathcal{T} = (\mathcal{S}_0, \boldsymbol{\theta}, \mathcal{A}, \mathcal{G}), \tag{1}$$

which consists of a rest surface $\mathcal{S}_0$ in material coordinates, effective material and contact parameters $\boldsymbol{\theta}$, a grasp model $\mathcal{A}$ that maps gripper motion to forces on the material, and a Gaussian appearance model $\mathcal{G}$ (Figure 1). We reconstruct $\mathcal{S}_0$ and $\mathcal{G}$ from the first frame and identify $\boldsymbol{\theta}$ by inverse simulation. Given an action sequence, the twin produces a physical trajectory $\boldsymbol{q}_{0:T}$ and rendered images. As in PhysTwin, we identify one twin per recorded interaction. The parameters are inferred from video without force measurements, so we treat them as effective values: stiffness, rest shape, friction, and attachment can partially compensate for one another.

3.2 Bicubic Hermite Shell

We parameterize the object by material coordinates $(u, v) \in [0, 1]^2$ on a structured grid of bicubic Hermite elements. The coordinates come from the minimum-area rectangle around the first-frame object points projected onto their fitted plane, and the sheet rests flat on this rectangle. Each node stores a position together with its first and mixed derivatives,

$$\boldsymbol{q}_{ij} = [\boldsymbol{x}_{ij},\ \boldsymbol{x}_{u,ij},\ \boldsymbol{x}_{v,ij},\ \boldsymbol{x}_{uv,ij}] \in \mathbb{R}^{12}. \tag{2}$$

Within an element, the surface is

$$\mathcal{S}(u, v; \boldsymbol{q}) = \sum_{a=0}^{1} \sum_{b=0}^{1} \boldsymbol{H}_a(u)^\top \boldsymbol{Q}_{ab}\, \boldsymbol{H}_b(v), \tag{3}$$

where $\boldsymbol{H}$ denotes the cubic Hermite basis and $\boldsymbol{Q}_{ab}$ collects the position, tangent, and twist vectors of the element's corner nodes. Neighboring elements share nodal derivatives, so the surface is C^1 across element boundaries. The surface has continuous tangents and normals, and its curvature within each element defines the bending energy. The elastic energy combines a membrane term and a Kirchhoff–Love bending term,

$$E_{\mathrm{el}}(\boldsymbol{q};\boldsymbol{\theta}) = s_{\mathrm{mem}}\, E_{\mathrm{mem}}(\boldsymbol{q}; E, \nu, h) + s_{\mathrm{bend}}\, E_{\mathrm{bend}}(\boldsymbol{q}; E, \nu, h), \tag{4}$$

with nominal Young's modulus E, Poisson ratio ν, thickness h, and a St. Venant–Kirchhoff constitutive law. The scales s_{mem} and s_{bend} are part of $\boldsymbol{\theta}$ and adjust the membrane and bending stiffness.

Time integration. We advance the shell implicitly by minimizing an incremental potential at each step,

$$\boldsymbol{q}_{t+1} = \arg\min_{\boldsymbol{q}} \; \frac{1}{2\Delta t^2} \left\| \boldsymbol{q} - \hat{\boldsymbol{q}}_{t+1} \right\|^2_{\boldsymbol{M}} + E_{\mathrm{el}}(\boldsymbol{q};\boldsymbol{\theta}) + E_{\mathrm{attach}}(\boldsymbol{q};\mathcal{M}_t) + E_{\mathrm{IPC}}(\boldsymbol{q}) + E_{\mathrm{fric}}(\boldsymbol{q}), \tag{5}$$

where $\hat{\boldsymbol{q}}_{t+1}$ is the inertial prediction, $\boldsymbol{M}$ is the mass matrix, E_{IPC} is the barrier of incremental potential contact (Li et al., 2020) that prevents interpenetration with the gripper and the table, and E_{fric} is a lagged friction potential; we omit the damping term for brevity. Each step is solved with Newton's method and a line search. The attachment energy E_{attach} couples the shell to the gripper and is defined next.

3.3 Material-Space Grasp Coupling

Dense material samples. We model the grasp with dense samples on a fixed grid in material space. Sample i has a persistent material coordinate $\boldsymbol{r}_i = (u_i, v_i)$ and the physical position

$$\boldsymbol{x}_i(\boldsymbol{q}) = \mathcal{S}(\boldsymbol{r}_i; \boldsymbol{q}). \tag{6}$$

Samples are tied to material coordinates rather than snapped to Hermite nodes, and they are far denser than the nodes. Each sample is interpolated from the 16 nodal vectors (positions, tangents, and twists) of its element, so contact can be resolved densely around a fold without adding a particle for every sample. Contact resolution is thereby decoupled from state resolution: where a grasp-induced fold spans only a few elements, the attachment targets still cover it densely.

Capture, clamp, and release. Each sample is free, captured, or clamped. A free sample is captured when a thin box around a gripper pad sweeps over it during a substep. The capture records the sample's material coordinate, the gripper side that captured it, and an attachment target $\boldsymbol{m}_i$ expressed in the gripper frame. Before clamp, the target may relax tangentially toward the sample; at clamp, it is frozen in the gripper frame and moves rigidly with the gripper. Release, or moving out of the pad region before clamp, returns the sample to the free state, from which it can be captured again. Because the grasp is defined on material identities, a new grasp acquires whatever material lies under the gripper at that time rather than a fixed set of nodes.

Attachment energy. For gripper g with a nonempty set $\mathcal{A}_g$ of active samples and $N_g = |\mathcal{A}_g|$, the attachment energy is a normalized isotropic penalty,

$$E_{\mathrm{attach}}(\boldsymbol{q};\mathcal{M}) = \sum_g \frac{K}{2N_g} \sum_{i\in\mathcal{A}_g} \left\| \mathcal{S}(\boldsymbol{r}_i;\boldsymbol{q}) - \boldsymbol{m}_i \right\|^2, \tag{7}$$

where $\mathcal{M}$ denotes the active sets and targets and K is the attachment stiffness. Normalizing by N_g keeps the total stiffness of each grasp independent of the sample density. Because $\mathcal{S}$ is linear in $\boldsymbol{q}$, the Jacobian $\boldsymbol{J}_i = \partial\mathcal{S}(\boldsymbol{r}_i;\boldsymbol{q})/\partial\boldsymbol{q}$ consists of the Hermite basis weights of sample i, and the residual of each sample acts on the 16 nodal vectors of its element. For a fixed active set, E_{attach} is quadratic in $\boldsymbol{q}$ and adds a constant positive semidefinite term to the Newton system of Eq. (5). Dense contact coverage therefore adds no degrees of freedom: the physical state remains a compact Hermite surface.

3.4 Inverse Simulation

We identify the parameters by minimizing an observation loss over a full rollout,

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(\boldsymbol{\theta}) = \sum_{t \in \mathcal{O}} w_t \, \ell_{\text{obs}}(\boldsymbol{q}_t, \mathcal{Y}_t) \quad \text{s.t.} \quad \boldsymbol{q}_{t+1} = F(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t; \boldsymbol{\theta}, \boldsymbol{a}_t), \tag{8}$$

where F is one step of Eq. (5), $\mathcal{O}$ is the set of fitting frames with equal weights $w_t = 1/|\mathcal{O}|$, and $\mathcal{Y}_t$ is the set of tracked object points at frame t. The observation term is the squared bidirectional Chamfer distance

$$\ell_{\text{obs}}(\boldsymbol{q}_t, \mathcal{Y}_t) = \frac{1}{|\mathcal{Y}_t|} \sum_{\boldsymbol{y} \in \mathcal{Y}_t} \min_{\boldsymbol{x} \in \mathcal{X}_t} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \frac{1}{|\mathcal{X}_t|} \sum_{\boldsymbol{x} \in \mathcal{X}_t} \min_{\boldsymbol{y} \in \mathcal{Y}_t} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \tag{9}$$

where $\mathcal{X}_t = \mathcal{X}(\boldsymbol{q}_t)$ are simulated points at fixed material locations (Section 4.1). Rather than E, ν, and h individually, we optimize a few effective parameters in log space: the stiffness scales s_{mem} and s_{bend} of Eq. (4), the attachment stiffness K, table friction, and damping.

Capture and clamp events make $\mathcal{L}$ nonconvex and only piecewise smooth. We therefore first search the parameters with CMA-ES (Hansen & Ostermeier, 2001), replaying the same clamp and release schedule for every candidate, and then refine the best candidate with Adam using discrete-adjoint gradients. At each step, the converged Newton residual $\boldsymbol{f}_{t+1}(\boldsymbol{q}_{t+1}, \boldsymbol{q}_t; \boldsymbol{\theta}) = \mathbf{0}$ defines $\boldsymbol{q}_{t+1}$ implicitly; the adjoint solves linear systems with $\partial \boldsymbol{f}_{t+1}/\partial \boldsymbol{q}_{t+1}$ backward in time and accumulates $\mathrm{d}\mathcal{L}/\mathrm{d}\boldsymbol{\theta}$ with the contact configuration held fixed. Section 4.4 compares such gradients with finite differences when the grasp relies on frictional contact instead.

3.5 Appearance Transport and New Actions

We reconstruct Gaussians (Kerbl et al., 2023) of the object in the first frame and bind each to its 16 nearest surface points in that frame (material samples for HermiteTwin, particles for PhysTwin). Its mean and covariance follow the local rigid motion of these neighbors, estimated by Procrustes fitting as in the linear blend skinning of PhysTwin (Jiang et al., 2025). Colors, opacities, and reference anisotropies stay fixed, and no image loss is used to identify $\boldsymbol{\theta}$. The same appearance model can therefore be driven by any simulated trajectory: by either simulator in our comparison, or by new actions. For a new action, we replace the recorded gripper trajectory and continue the simulation from the identified state, which includes positions and velocities, without re-initializing the object. Embedding each Gaussian directly in the continuous Hermite surface is a natural alternative; we keep the neighbor-based rule so that both simulators share one appearance model.

4 Experiments

We evaluate how accurately HermiteTwin resimulates observed manipulation (Section 4.2), how well it predicts held-out future frames (Section 4.3), which components contribute to its accuracy (Section 4.4), and how it supports new interactions (Section 4.5).

4.1 Experimental Setup

Data. We use ten interactions from Deform360 (Li et al., 2026), a multiview visuotactile dataset of robot manipulation: eight with a red cloth and two with a white foam sheet (Appendix B). For each interaction, object points with fixed identities tracked over time serve as the observations $\mathcal{Y}_t$, together with the recorded robot motion and tactile signals. Contact onset, clamp, and release are extracted from the tactile signals and frozen before fitting; PhysTwin uses the same onset and release.

Baselines. We compare with PhysTwin (Jiang et al., 2025), keeping its spring–mass model, integrator, and optimizer. Because the gripper approaches the object from a distance in these recordings, its controller springs are connected once, to the object points near the

gripper at contact onset, and act only between onset and release. We also compare with GS-Dynamics (Zhang et al., 2024), a learned point-based simulator that predicts the motion of the object points from the gripper motion with a graph neural network. Within each interaction, all methods use the same observations and transport the same Gaussian appearance model, so differences in the rendered images reflect differences in the simulated motion.

Fitting. HermiteTwin and PhysTwin fit every interaction with the objective of Eq. (9), which also selects the model; each starts from its default parameters and runs CMA-ES followed by Adam refinement (Appendix A), and for PhysTwin, this objective replaces its native loss. GS-Dynamics is trained on the same fitting frames with its native loss, and we keep the checkpoint whose rollout over these frames has the lowest Chamfer distance. All methods are scored with Eq. (9) at a common set of material locations obtained from PhysTwin's particle sampling of the first frame: HermiteTwin predicts the object at these locations, PhysTwin's particles contain them, and GS-Dynamics predicts all tracked points, which include them.

Metrics. We report the objective of Eq. (9) as the 3D error (CD, in mm^2) and measure image accuracy with silhouette IoU and PSNR in three fixed camera views; for prediction, we also report SSIM and LPIPS. Renderings are compared with the recorded images on a black background, excluding gripper pixels. Metrics are averaged with equal weights over views, then frames, then interactions.

4.2 Reconstruction & Resimulation

Table 1 evaluates how well each twin resimulates the fitting frames of all ten interactions. HermiteTwin improves all three metrics on average. Figure 2 shows representative frames. The red-cloth interaction is dominated by contact and bending: the gripper pinches a corner, lifts it, and folds it back over the sheet. HermiteTwin reproduces this fold at $t = 160$ and $t = 269$, whereas both baselines leave the sheet nearly flat. This holds even though PhysTwin is given the grasp timing from the tactile signals, so that its controller springs connect only once the gripper touches the cloth (Section 4.1); the spring–mass sheet still does not form the fold. The foam sheet is much stiffer, and PhysTwin does not maintain this stiffness: its sheet deforms around the grasp instead of keeping its flat shape, whereas HermiteTwin preserves it.

Table 1: Reconstruction and resimulation. CD, three-view silhouette IoU, and PSNR over the fitting frames, averaged with equal weight over the ten interactions.

Method	CD ↓ (mm^2)	IoU ↑ (%)	PSNR ↑ (dB)
GS-Dynamics	116.95	83.24	22.160
PhysTwin	61.89	85.22	22.745
HermiteTwin (ours)	42.05	88.89	24.242

4.3 Future Prediction

On the red cloth, the models fitted on the first 300 frames in Section 4.2 continue the same rollout to the end of each recording with frozen parameters; the simulated state is carried over without resetting it to the observation, and the held-out frames never enter fitting or model selection. The robot motion and contact events of the held-out frames come from the recording, so this experiment predicts the cloth under known future control rather than the robot's actions.

Table 2 shows that HermiteTwin predicts the held-out frames more accurately in all five metrics, and Figure 3 shows one interaction over its fitting and prediction frames.

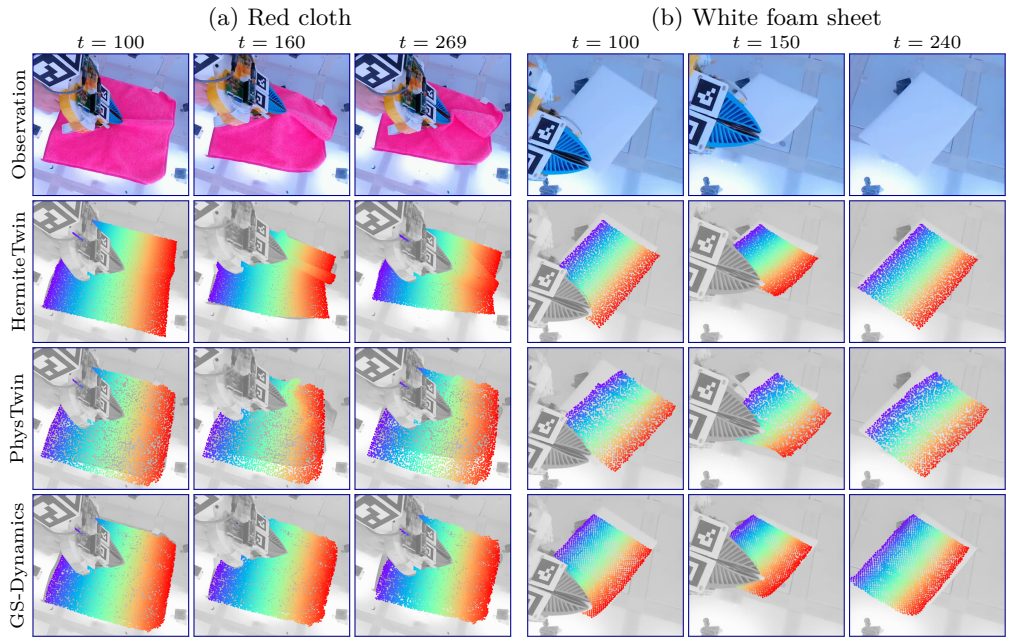


Figure 2: Qualitative reconstruction and resimulation. Observations and resimulations of the red cloth (a) and the foam sheet (b). Simulated panels show points sampled densely on the HermiteTwin surface and the simulated PhysTwin and GS-Dynamics particles, colored by their material coordinate at rest. HermiteTwin reproduces the folded cloth corner at $t = 160$ and $t = 269$.

Table 2: Future prediction. All methods are fitted on the first 300 frames of eight red-cloth interactions and predict the rest of each recording under the recorded robot motion; each interaction is weighted equally.

Method	CD ↓ (mm^2)	IoU ↑ (%)	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓
GS-Dynamics	552.40	71.40	17.696	0.9132	0.1291
PhysTwin	120.40	81.08	20.022	0.9316	0.0994
HermiteTwin (ours)	67.75	84.78	21.417	0.9396	0.0850

4.4 Analysis

Is the grasp model alone sufficient? Our comparisons change the physical representation and the grasp model together. To separate them, we transfer our grasp to PhysTwin: the gripper captures cloth particles, clamps them to gripper-local targets, and releases them, in place of the native controller springs, and this variant is fitted as in the main comparison (Appendix C). The transferred grasp engages, but the sheet stays laterally spread and does not form the observed fold that HermiteTwin reproduces (Figure 4). The grasp model alone therefore does not account for the difference.

Can frictional contact replace the attachment? The gripper could instead hold the sheet through IPC frictional contact alone, without attachment. We explored this route and found two problems (Appendix D). First, the grasp is hard to form and slow to simulate. Grasps at an edge or a corner form readily, but grasps that must first pinch a fold out of the flat sheet slipped or stalled in our attempts unless the jaws followed a hand-tuned closing path. Even at an edge, whether the sheet stays in the jaws depends on the friction coefficient, and each forward pass is slow at the tolerance that accurate gradients require. Second, the gradients become unreliable once the lift begins: the adjoint agrees with finite differences while the fingers close, but not after the sheet is lifted, even with tightly converged solves. Our attachment instead holds the captured samples with a smooth energy from clamp to

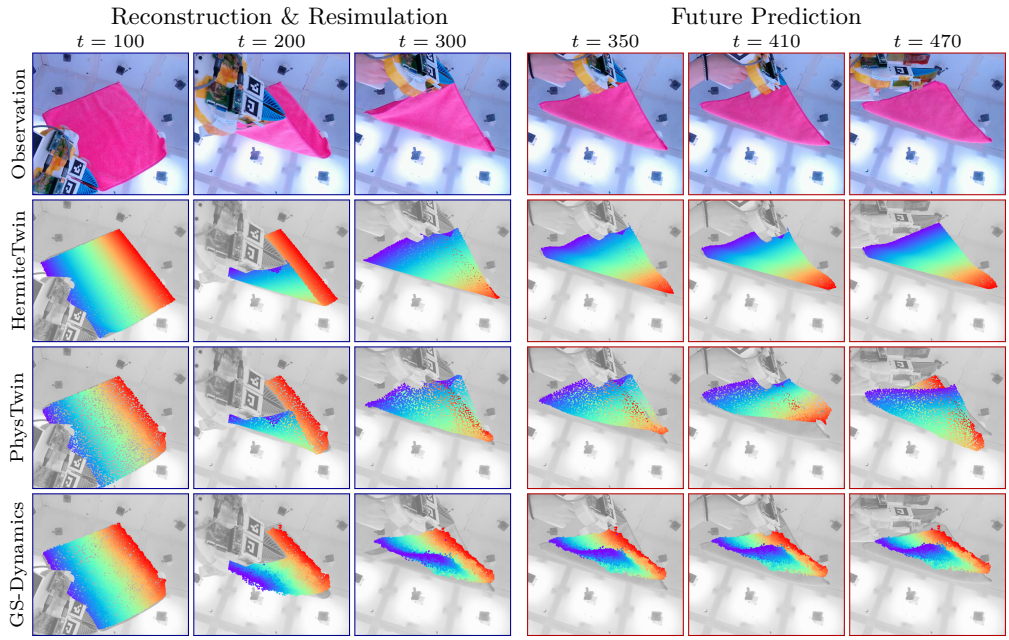
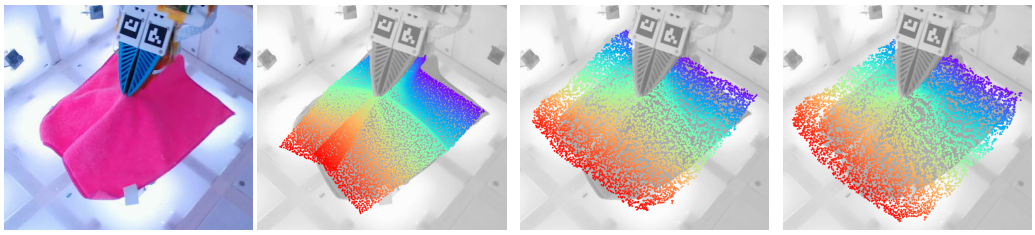


Figure 3: Reconstruction and future prediction on a red-cloth interaction. The first three columns show reconstruction and resimulation ($t = 100, 200, 300$); the last three show future prediction ($t = 350, 410, 470$), with parameters fitted on frames 1–300.



(a) Observation (b) HermiteTwin (c) PhysTwin, native springs (d) PhysTwin + our grasp

Figure 4: Grasp transfer on one cloth interaction ($t = 160$). Our material-space grasp lifts the PhysTwin cloth near the fingertip (d) but does not recover the observed fold (a), which HermiteTwin reproduces (b). Simulated panels show points sampled densely on the HermiteTwin surface and the simulated PhysTwin particles, colored by their material coordinate at rest.

release, so retention does not depend on friction, and the grasp changes only at capture and release.

4.5 Appearance-Aware Interactions

The identified twin can be driven by actions that were never recorded. In Figure 5 (Appendix E), the twin identified on one red-cloth interaction follows the recorded motion until the grasp closes; the gripper then lifts the cloth and either drapes it over a book or lowers it into an open box, and releases it. In a third scene, a free bottle stands on the cloth during a recorded two-gripper lift and topples as the cloth rises. The book, the box, and the bottle collide with the shell, and the Gaussian appearance follows the simulated shell without retraining. These scenes have no real counterpart, so the results are qualitative.

5 Limitations

Identification and generalization. Following PhysTwin (Jiang et al., 2025), we identify each twin from a single recorded interaction, and the recovered parameters are effective values (Section 3). Our quantitative prediction therefore continues a recorded interaction, while the new actions in Figure 5 are evaluated only qualitatively. Identifying a shared object model from multiple interactions of the same object could better constrain its intrinsic properties and enable quantitative evaluation under unseen actions. In addition, the twin relies on tracked object points from calibrated multiview capture, and its accuracy is bounded by their quality; extending the pipeline to sparser observations with appropriate priors would make it applicable to less instrumented settings.

Appearance. Appearance is transported with the simulated shell rather than modeled jointly with it. The Gaussians therefore do not reproduce view-dependent shading on wrinkles, surfaces newly exposed by folding, or strain-dependent appearance, which limits the image metrics independently of the mechanics. Refining appearance jointly with the dynamics could improve the renderings, provided that the more expressive appearance model does not absorb errors of the physical model.

6 Conclusion

We presented HermiteTwin, a physical digital twin for thin deformable objects that represents deformation and contact separately: a coarse bicubic Hermite shell models the object, and dense material-space grasp samples couple it to the robot. In experiments on real robot manipulation, HermiteTwin resimulates observed motion and predicts held-out frames more accurately than spring–mass and learned baselines, and the identified twins can be driven by new actions and interact with new objects. Our analysis further suggests that the shell representation, not only the grasp model, contributes to this gain. Identifying a shared object model from multiple interactions and predicting its behavior under unseen actions are the next steps toward twins that support planning for contact-rich manipulation.

AI Use Statement

Generative AI tools were used to assist with manuscript drafting, language editing, and checking clarity and consistency. The authors take full responsibility for the final content, including all claims, results, and references.

References

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

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation, 9(2):159–195, 2001.

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

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

Suning Huang, Qianzhong Chen, Xiaohan Zhang, Jiankai Sun, and Mac Schwager. ParticleFormer: A 3d point cloud world model for multi-object, multi-material robotic manipulation. In Conference on Robot Learning, 2025.

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. arXiv preprint arXiv:2503.17973, 2025.

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

Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. Affine body dynamics: Fast, stable and intersection-free simulation of stiff materials. ACM Transactions on Graphics, 41(4), 2022.

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.

Hongyu Li, Wanjia Fu, Xiaoyan Cong, Zekun Li, Binghao Huang, Hanxiao Jiang, Xintong He, Yiqing Liang, Rao Fu, Tao Lu, Srinath Sridhar, Kevin A. Smith, George Konidaris, and Yunzhu Li. Deform360: A massive multi-view visuotactile dataset for deformable world models. arXiv preprint arXiv:2607.05390, 2026.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R. Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. Incremental potential contact: Intersection- and inversion-free, large-deformation dynamics. ACM Transactions on Graphics, 39(4), 2020.

Alberta Longhini, Marcel Büsching, Bardienus P. Duisterhof, Jens Lundell, Jeffrey Ichnowski, Mårten Björkman, and Danica Kragic. Cloth-splatting: 3d cloth state estimation from rgb supervision. In Conference on Robot Learning, 2024.

Yuqi Meng, Yihao Shi, Kemeng Huang, Zixuan Lu, Ning Guo, Taku Komura, Yin Yang, and Minchen Li. Efficient B-Spline finite elements for cloth simulation. ACM Transactions on Graphics, 45(4), 2026. doi: 10.1145/3811278.

Xingyu Ni, Xuwen Chen, Cheng Yu, Bin Wang, and Baoquan Chen. Simulating thin shells by bicubic hermite elements. Computer-Aided Design, 174:103734, 2024. doi: 10.1016/j.cad.2024.103734.

Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J. Black, Bernhard Thomaszewski, Christina Tsalicoglou, and Otmar Hilliges. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. arXiv preprint arXiv:2409.08189, 2024.

Yidi Shao, Mu Huang, Chen Change Loy, and Bo Dai. GauSim: Registering elastic objects into digital world by gaussian simulator. arXiv preprint arXiv:2412.17804, 2024.

Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. GeoSplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025.

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

Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. In Conference on Robot Learning, 2024.

Yang Zheng, Qingqing Zhao, Guandao Yang, Yifan Wang, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, and Gordon Wetzstein. PhysAvatar: Learning the physics of dressed 3d avatars from visual observations. arXiv preprint arXiv:2404.04421, 2024.

A Implementation Details

Simulation. All twins use a 20×20 element grid, with nodal derivatives scaled by the element extent, and advance each video frame in five substeps. On the red cloth, grasp samples are spaced 2 mm apart at rest, giving about 21,000 candidates for a shell with 5,292 scalar degrees of freedom. The nominal material values are $E = 100$ kPa, $\nu = 0.3$, a density of 500 $\mathrm{kg\,m^{-3}}$, and a thickness of $h = 1$ mm for the red cloth and $h = 5$ mm for the foam sheet; both stiffness scales in Eq. (4) start at one. Linear systems are solved with a preconditioned iterative solver.

Optimization. Each system first evaluates its default parameters, then runs four CMA-ES generations of five candidates; failed candidates count toward this budget. The best feasible candidate initializes ten Adam iterations; for HermiteTwin, each is one update of the five parameters, which share one Adam state. The selected model is the feasible forward run with the lowest fitting objective. In the main comparison, PhysTwin sums the spring forces on each particle in a fixed order, so repeated forward runs agree. All experiments run on NVIDIA RTX 5090 GPUs; fitting all ten interactions takes about 6.5 h for HermiteTwin and 5 h for PhysTwin.

B Dataset

We use ten interactions from Deform360 (Li et al., 2026): eight with a red cloth of about 29×29 cm and two with a white foam sheet of about 10.5×15.7 cm; the robot grasps the sheet with one or two grippers. Each interaction provides calibrated multiview video at 30 frames per second, the robot motion, and fingertip tactile signals. The tracked object points are obtained with the processing pipeline of Deform360. Contact onset, clamp, and release are detected by thresholding the tactile signals. Frame 0 initializes both systems and is not scored. The red-cloth interactions are fitted on frames 1–300, and the rest of each recording is held out for prediction; the foam interactions are fitted on frames 1–267.

C Grasp Transfer to a Mass–Spring Cloth

This study transfers the grasp model of Section 3.3 to PhysTwin on a red-cloth interaction. Both PhysTwin variants keep the spring–mass model, integrator, and optimizer of the main comparison, and both are fitted with the objective of Eq. (9) and the optimization budget used for Tables 1 and 2. The native variant drives the cloth through controller springs, as in the main comparison. The transferred-grasp variant replaces them with our grasp model, applied to the particles in place of the material samples of HermiteTwin: every cloth particle is a grasp candidate, captured when the thin box around a gripper pad sweeps over it; its gripper-local target relaxes before clamp, is frozen at clamp, and is cleared on release; and the total grasp stiffness is divided by the number of active particles, as in Eq. (7). Capture, clamp, and release therefore follow the same pad geometry and the same recorded contact events as in HermiteTwin, and the variant additionally optimizes its grasp stiffness.

The captured particles follow the gripper, but the sheet stays laterally spread instead of forming the observed fold that HermiteTwin reproduces (Figure 4).

D Grasping through Frictional Contact

We also tried to hold the sheet through contact alone, without attachment, using either penalty contact forces or IPC (Li et al., 2020) with friction between the shell and the fingers, which follow the recorded gripper motion. Both variants share two problems. First, the grasp often fails to lift the sheet. Grasps at an edge or a corner can enter the jaws directly, but grasps that must first pinch a fold out of the sheet are fragile: for one gripper of a bimanual grasp, the sheet slips or the Newton solve stalls across variations of jaw opening, pose, height, and bending stiffness, unless the jaws are pre-opened and closed along a staged, hand-tuned path that lets the sheet buckle into a load-bearing fold. Grasping through



Figure 5: Driving the identified cloth with new actions and objects. Top: draping the cloth over a book. Middle: placing the cloth inside a box. Bottom: a free bottle toppling as a recorded two-gripper lift raises the cloth.

collision also takes at least three times as long to simulate as our attachment. Second, the gradients are inaccurate: the adjoint agrees with finite differences while the fingers close, but not after the sheet is lifted. We suspect two causes. Coupling the shell to the fingers as rigid bodies (Lan et al., 2022) degrades the linear systems that the adjoint replays, and when the shell is lifted through friction alone, the contact locations change too abruptly between steps for the rollout to stay smooth in the parameters. Our attachment avoids both problems, since it holds the captured samples with a smooth energy that changes only at capture and release.

E Driving the Twin with New Actions and Objects

All scenes use twins identified on red-cloth interactions, with unchanged parameters (Figure 5). In the first two, the recorded gripper motion is kept until the grasp closes and then replaced by new motions that drape the cloth over a book or lower it into an open box; in the third, a recorded two-gripper lift is replayed with a free bottle standing on the cloth. The book and the box are fixed obstacles and the bottle is an affine body (Lan et al., 2022), all colliding with the shell through IPC. The cloth is rendered with its reconstructed Gaussians transported by the simulated shell.