NEURALCUT: DATA-FREE LEARNING OF NEURAL REDUCED-ORDER CUT SIMULATION

Anonymous authors
Paper under double-blind review

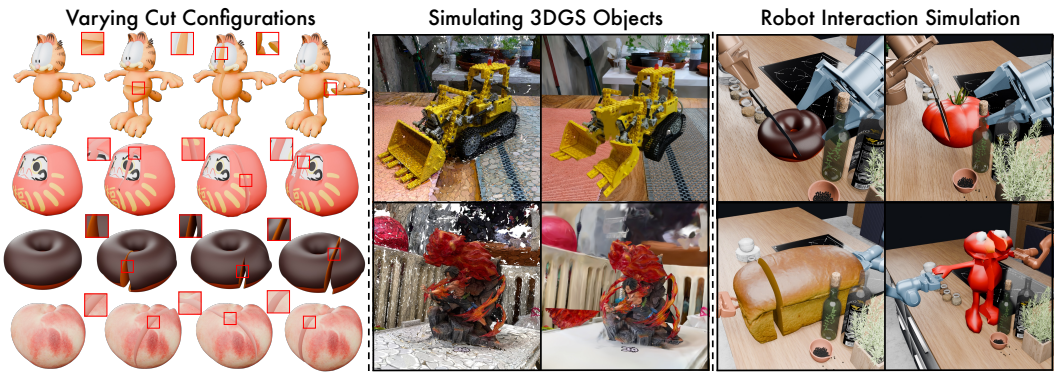

Figure 1: NeuralCut is a data-free learning framework for real-time reduced-order cut simulation of deformable 3D objects. Left: we show cut simulation under varying cut configurations. Middle: we apply our method to objects represented by 3D Gaussian Splatting (3DGS). Right: we demonstrate the application of our method to robot interaction simulation.

ABSTRACT

Cutting deformable objects is common in daily life, but the resulting displacement discontinuities pose significant challenges for reduced-order simulation. We introduce NeuralCut, a novel data-free learning framework for reduced-order cut simulation of deformable 3D objects. NeuralCut uses neural skinning to represent full-space deformation in a compact subspace. To represent cut-induced displacement discontinuities, we develop a geometry-derived lifting-and-restriction formulation that augments spatial coordinates with a cut-dependent scalar lifting coordinate. This coordinate separates opposite sides of the cut while smoothly vanishing at its boundary, enabling discontinuous deformation across the cut and continuous deformation at the boundary. We learn the skinning fields through physics-informed self-supervision without costly full-space simulation trajectories. Our training objective minimizes expected potential energy estimated by cut-aware Monte Carlo importance sampling, promotes distinct weights across the cut through a cross-side jump constraint, and encourages orthogonality among the skinning weight functions. Experiments demonstrate real-time reduced-order cut simulation across diverse 3D objects and cut configurations, supporting both meshes and Gaussian splats. We further demonstrate NeuralCut in robot interaction simulation. Moreover, coupling NeuralCut with a transformer-based model enables cross-object generalization of cut-conditioned neural skinning.

1 INTRODUCTION

Cutting deformable objects is widespread in everyday life and plays an important role in food processing, surgical manipulation, and manufacturing. Accurately simulating the resulting dynamics across diverse 3D objects would benefit robotic manipulation, virtual prototyping, and digital content creation. To support these applications, simulation methods must capture cut-induced displacement discontinuities under varying cut configurations while remaining efficient.

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.

Reduced-order modeling (ROM) (Barbič & James, 2005) accelerates deformable simulation by evolving low-dimensional subspace coordinates and reconstructing full-space deformation through a subspace mapping. To overcome the discretization dependence of classical ROMs, neural ROM methods (Chen et al., 2023b; Chang et al., 2023) use continuous neural fields to parameterize the subspace mapping, but require precomputed simulation trajectories for training. Simplicits (Modi et al., 2024) instead introduces a data-free formulation that adopts Linear Blend Skinning (LBS) to construct the deformation subspace mapping and learns continuous neural skinning weight functions through self-supervised, physics-informed optimization. Despite these advances, continuous subspace mappings cannot represent cut-induced displacement discontinuities. Recently, WindLifter (Chang et al., 2025b) augments spatial coordinates with the generalized winding number to represent such discontinuities over 2D domains. Nevertheless, reduced-order simulation of cut 3D objects remains an open challenge. These limitations raise a central question: *How can we develop a discretization-agnostic neural reduced-order representation for 3D cut simulation that captures cut-induced displacement discontinuities and can be learned in a data-free manner?*

We introduce NeuralCut, a data-free learning framework for neural reduced-order cut simulation of 3D objects (Fig. 1). Specifically, we adopt a neural skinning-based reduced-order formulation (Modi et al., 2024). For cut simulation, a requirement is that the deformation subspace mapping can represent displacement discontinuities across the cut surface while remaining continuous at the cut-surface boundary for finite cuts. However, standard neural fields conditioned solely on spatial coordinates remain continuous and cannot represent such discontinuities. We therefore introduce a geometry-derived lifting-and-restriction formulation that augments spatial coordinates with a cut-dependent scalar lifting coordinate. This coordinate takes distinct values across the cut surface and smoothly vanishes at the cut-surface boundary. When restricted to the lifted domain, a neural field conditioned on a positional encoding of the scalar lifting coordinate defines a discretization-agnostic deformation subspace mapping that can represent cut-induced displacement discontinuities.

However, learning this cut-conditioned representation without full-space simulation trajectories remains non-trivial. In prior data-free learning formulations (Modi et al., 2024), deformation representations are learned through potential-energy and orthogonality objectives estimated using uniform Monte Carlo samples over the spatial domain. Cut-conditioned learning introduces two additional challenges: First, this uniform sampling may provide insufficient coverage near the cut surface, where the prescribed cut most strongly influences the neural skinning fields. Second, although the lifted representation makes the prescribed discontinuity representable, these objectives do not enforce it, so training may collapse to a degenerate solution that remains continuous across the cut. To overcome these challenges, we introduce cut-aware importance sampling to increase sampling near the cut surface while preserving unbiased potential-energy estimation, and a cross-side jump constraint to explicitly separate the skinning weight vectors on opposite sides of the cut. Together, these mechanisms enable data-free learning of cut-conditioned neural skinning fields for reduced-order simulation under varying 3D cut configurations.

Our contributions are: (1) We present **NeuralCut**, to our knowledge the first neural reduced-order simulation framework supporting finite cuts in deformable 3D objects; (2) We introduce a **Geometry-Derived Lifting-and-Restriction Formulation** that encodes the cut surface through a scalar lifting coordinate, enabling a neural skinning-based deformation subspace mapping to represent the prescribed discontinuity while remaining continuous at the cut-surface boundary; (3) We develop a **Cut-Aware Data-Free Learning Strategy** that integrates cut-aware importance sampling for unbiased potential-energy estimation with a cross-side jump constraint to learn skinning fields with the prescribed discontinuity; (4) Experiments demonstrate real-time reduced-order cut simulation across diverse 3D shapes and cut configurations, as well as mesh-free simulation with 3D Gaussians and robot interaction simulation. A Transformer-based extension further demonstrates cross-object generalization in predicting cut-conditioned skinning weight fields.

2 Related Work

Cutting Simulation of Deformable 3D Objects. Traditional cutting simulators explicitly resolve evolving discontinuities within a spatial discretization (Wu et al., 2015). Mesh-based methods split, refine, or remesh elements along the cut (Bielser & Gross, 2000; Mor & Kanade, 2000; Bielser et al., 2003; Busaryev et al., 2013; Chen et al., 2014; Paulus et al., 2015; Areias & Rabczuk, 2017), whereas virtual-node methods duplicate intersected elements to accommodate arbitrary cuts (Molino et al., 2004; Sifakis et al., 2007). Particle-grid methods such as the material point method (MPM) avoid

remeshing (Jiang et al., 2016), but their less precise surface representation can produce numerical fractures and imprecise cuts (Su et al., 2022; Fan et al., 2025). The extended finite element method (XFEM) instead enriches a fixed background mesh with discontinuous basis functions (Moës et al., 1999; Jeřábková & Kuhlen, 2009; Kaufmann et al., 2009; Koschier et al., 2017; Ton-That et al., 2024). Simulators based on the finite element method (FEM) have also been coupled with robotic cutting (Jamdagni & Jia, 2019; Heiden et al., 2021; Beltran-Hernandez et al., 2024). Nevertheless, existing cutting simulators remain discretization-dependent and computationally expensive, leaving mesh-free, real-time reduced-order cut simulation an open challenge.

Neural Reduced-Order Physics Simulation. Neural networks have been widely used to accelerate physics-based simulation, including deformable-body and cloth dynamics (Chen et al., 2023a; Feng et al., 2024b; Lyu et al., 2024; Bertiche et al., 2022; Kairanda et al., 2024; Zhang & Li, 2024), fluid simulation (Kim et al., 2019; Chu et al., 2022; Deng et al., 2023; Wang et al., 2024b; Jain et al., 2024; Tao et al., 2024), and collision handling (Romero et al., 2021; Yang et al., 2020; Cai et al., 2022). Our work is most closely related to neural reduced-order approaches, which parameterize reduced dynamics or deformation subspace mappings with neural networks (Holden et al., 2019; Fulton et al., 2019; Shen et al., 2021; Chen et al., 2023b; Zong et al., 2023; Chang et al., 2023). These approaches broadly comprise data-driven methods learned from precomputed simulation trajectories (Fulton et al., 2019; Shen et al., 2021; Chen et al., 2023b; Zong et al., 2023; Chang et al., 2023) and data-free methods optimized directly through physical objectives (Sharp et al., 2023; Lyu et al., 2024; Modi et al., 2024; Wang et al., 2024a; Chang et al., 2025a). Neural-field formulations provide continuous, discretization-agnostic subspace mappings, while neural skinning formulations employ Linear Blend Skinning (LBS) to construct compact deformation subspaces (Modi et al., 2024; Xiang et al., 2026; Lei et al., 2026). Recently, WindLifter (Chang et al., 2025b) augments spatial coordinates with the generalized winding number to represent cut-induced discontinuities, but is limited to 2D domains. Related lifting-based formulations encode boundary conditions (Liao et al., 2026) and gradient discontinuities (Liu et al., 2025a). Despite these advances, neural reduced-order cut simulation of deformable 3D objects remains an open challenge.

3 METHOD

NeuralCut is a data-free learning framework for neural reduced-order simulation of 3D objects under varying cut configurations. As shown in Fig. 2, we first represent full-space deformation using a few affine skinning handle transformations, defining a compact subspace for efficient implicit time integration (Sec. 3.1). To represent displacement discontinuities across finite cuts, we introduce a geometry-derived lifting-and-restriction formulation. The lifting augments spatial coordinates with a cut-dependent scalar lifting coordinate that separates opposite cut sides and smoothly vanishes at the cut-surface boundary. Restricting a continuous neural field to the resulting lifted domain yields cut-conditioned skinning weight functions that can represent discontinuities across the cut (Sec. 3.2). We learn these fields through physics-informed self-supervision. Building on elastic potential energy minimization and orthogonality among skinning weight functions, we introduce cut-aware importance sampling to concentrate Monte Carlo samples near the cut surface and a cross-side jump constraint to promote separation between skinning weight vectors across the cut (Sec. 3.3).

3.1 NEURAL SKINNING-BASED REDUCED-ORDER SIMULATION

We consider a deformable 3D object occupying a reference domain $\Omega \subset \mathbb{R}^3$ with a cut surface Γ. To model its full-space deformation within a reduced-order subspace, following Simplicits (Modi et al., 2024), we represent the deformation map $\boldsymbol{x} = \phi_\Gamma(\mathbf{X}, \mathbf{z})$ as a weighted sum of m affine transformations applied to the undeformed position:

$$\phi_\Gamma(\mathbf{X}, \mathbf{z}) = \mathbf{X} + \sum_{i=1}^{m} \boldsymbol{W}_i^\Gamma(\mathbf{X})\, \mathbf{Z}_i \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{X} \in \Omega$ and $\boldsymbol{x} \in \mathbb{R}^3$ are the undeformed and deformed positions, respectively, m is the number of skinning handles, and $\boldsymbol{W}_i^\Gamma : \Omega \to \mathbb{R}$ is the skinning weight function associated with handle i. The superscript Γ indicates that the skinning weights depend on the cut surface. Each $\mathbf{Z}_i \in \mathbb{R}^{3\times 4}$ denotes the affine transformation of handle i. Collecting these transformations as $\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_m)$, we define the reduced-order subspace coordinates as $\mathbf{z} = \mathbf{flat}(\mathbf{Z}) \in \mathbb{R}^{12m}$.

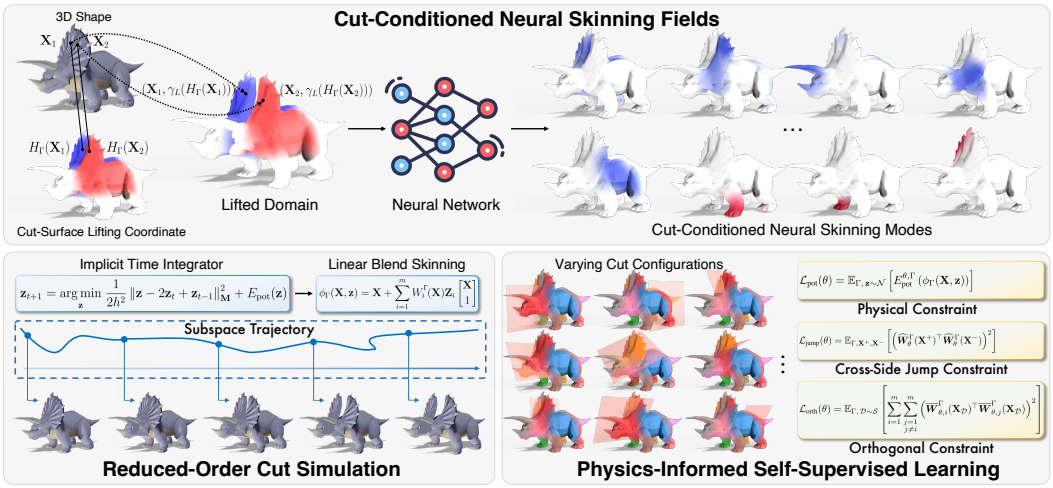

Figure 2: We augment the 3D spatial coordinates with a cut-dependent scalar lifting coordinate to represent cut-induced displacement discontinuities. The neural field maps the lifted input to cut-conditioned neural skinning modes. A physics-informed self-supervised learning strategy optimizes the neural field under physical, cross-side jump, and orthogonal constraints, enabling real-time reduced-order cut simulation under varying cut configurations.

With this skinning-based subspace representation, we update the subspace coordinates using standard implicit time integration (Li et al., 2026):

$$\mathbf{z}_{t+1} = \arg\min_{\mathbf{z}} \frac{1}{2h^2} \|\mathbf{z} - 2\mathbf{z}_t + \mathbf{z}_{t-1}\|_{\mathbf{M}}^2 + E_{\text{pot}}(\mathbf{z}), \tag{2}$$

where h is the time-step size, $\mathbf{M}$ is the reduced mass matrix, and the elastic potential energy is $E_{\text{pot}}(\mathbf{z}) := E_{\text{pot}}(\phi_\Gamma(\mathbf{X}, \mathbf{z})) = \int_\Omega \Psi(F_\Gamma(\mathbf{X}, \mathbf{z}))\, \mathrm{d}\mathbf{X}$. Here, Ψ is a hyperelastic energy density, and the deformation gradient $F_\Gamma = \partial\phi_\Gamma/\partial\mathbf{X}$ is evaluated on each cut side. This optimization can be solved using standard Newton-based methods (Nocedal & Wright, 2006). Since the dimensionality of the reduced subspace is significantly smaller than that of the full space, the optimization in Eq. (2) converges rapidly (Benchekroun et al., 2023), thereby enabling real-time physics-based simulation.

3.2 Cut-Surface Lifting and Restriction

A finite cut permits a displacement discontinuity across the cut surface. Standard neural fields conditioned solely on spatial coordinates cannot represent this discontinuity. Inspired by the lifting-and-restriction view of WindLifter (Chang et al., 2025b) and DiscontGrad (Liu et al., 2025a), we obtain the cut-conditioned neural skinning fields by restricting a continuous neural field to the lifted domain defined by $\mathcal{L}_\Gamma$. We formulate this lifting-and-restriction operation as:

$$\mathcal{L}_\Gamma(\mathbf{X}) = (\mathbf{X}, \gamma_L(H_\Gamma(\mathbf{X}))), \qquad \boldsymbol{W}_\theta^\Gamma(\mathbf{X}) = \widetilde{\boldsymbol{W}}_\theta(\mathcal{L}_\Gamma(\mathbf{X})), \tag{3}$$

where $\mathcal{L}_\Gamma : \Omega \to \Omega \times \mathbb{R}^{2L}$ is the encoded lifting operator, $H_\Gamma : \Omega \to \mathbb{R}$ is a non-learned scalar lifting coordinate constructed explicitly from the cut surface, and $\gamma_L : \mathbb{R} \to \mathbb{R}^{2L}$ is its positional encoding. $\widetilde{\boldsymbol{W}}_\theta : \Omega \times \mathbb{R}^{2L} \to \mathbb{R}^m$ is a continuous neural field whose restriction through $\mathcal{L}_\Gamma$ yields the cut-conditioned neural skinning fields $\boldsymbol{W}_\theta^\Gamma : \Omega \to \mathbb{R}^m$. Unlike WindLifter (Chang et al., 2025b), which directly appends a scalar lifting coordinate, we condition the shared neural field on a multi-frequency positional encoding of $H_\Gamma(\mathbf{X})$:

$$\gamma_L(p) = \left(\sin(2^k p), \cos(2^k p)\right)_{k=0}^{L-1}, \tag{4}$$

where $p = H_\Gamma(\mathbf{X})$ denotes the scalar lifting coordinate and L is the number of frequency bands. This encoding helps the shared neural field adapt to varying cut configurations rather than converge to a cut-insensitive solution. By mapping opposite sides of the cut to distinct lifted inputs, $\mathcal{L}_\Gamma$ allows the skinning fields $\boldsymbol{W}_\theta^\Gamma$ to be discontinuous across Γ in the reference domain. Through Eq. (1), these skinning fields allow the deformation map to represent displacement discontinuities.

We construct H_Γ directly from the cut surface Γ and its boundary $\partial\Gamma$. For an input coordinate $\mathbf{X}$, we first find its closest point on Γ:

$$\mathbf{p}_\Gamma(\mathbf{X}) = \arg\min_{\mathbf{p}\in\Gamma} \|\mathbf{X} - \mathbf{p}\|_2. \tag{5}$$

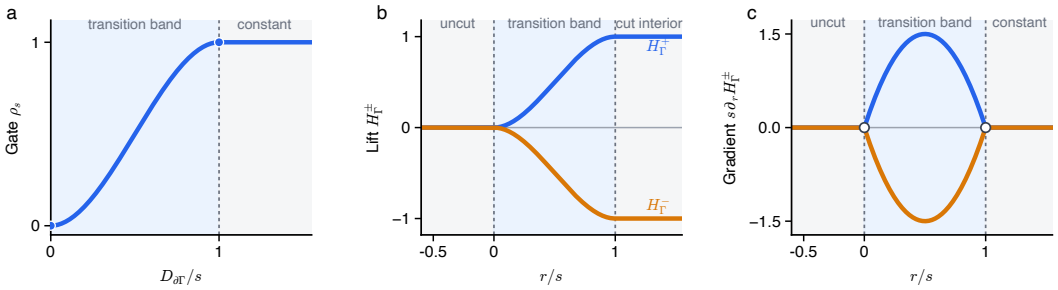

Figure 3: **Smooth cut-surface lifting profile.** (a) Smooth gate ρ_s defined in Eq. (8). (b) Lifting profiles $H_\Gamma^\pm(r) = \pm\rho_s(\max(r, 0))$ on opposite sides of the cut surface. Here, r is a signed coordinate across $\partial\Gamma$, with $r \le 0$ denoting the uncut region. (c) Normalized gradients $s\,\partial_r H_\Gamma^\pm$.

Using the oriented normal $\mathbf{n}_\Gamma$ at the closest point, we define a signed normal coordinate $\delta_\Gamma : \Omega \to \mathbb{R}$ and a binary side indicator σ_Γ:

$$\delta_\Gamma(\mathbf{X}) = \mathbf{n}_\Gamma(\mathbf{p}_\Gamma(\mathbf{X}))^\top (\mathbf{X} - \mathbf{p}_\Gamma(\mathbf{X})), \qquad \sigma_\Gamma(\mathbf{X}) = \operatorname{sgn}(\delta_\Gamma(\mathbf{X})). \tag{6}$$

The regions $\delta_\Gamma(\mathbf{X}) > 0$ and $\delta_\Gamma(\mathbf{X}) < 0$ define Ω_Γ^+ and Ω_Γ^-, respectively. Using σ_Γ as the lifting coordinate would therefore induce a constant separation across the cut surface. However, a finite cut surface is bounded by $\partial\Gamma$. Extending this separation beyond the cut-surface boundary would also separate points in the uncut region, whereas abruptly truncating it at $\partial\Gamma$ would introduce an artificial discontinuity in the lifting coordinate. To localize the separation to the cut surface, we measure the distance from the closest surface point to the cut-surface boundary:

$$D_{\partial\Gamma}(\mathbf{X}) = \min_{\mathbf{q}\in\partial\Gamma} \|\mathbf{p}_\Gamma(\mathbf{X}) - \mathbf{q}\|_2 . \tag{7}$$

For a transition width $s > 0$, we define a smooth gate ρ_s that attenuates the side indicator near the cut-surface boundary:

$$\rho_s(d) = \begin{cases} 1, & d \ge s, \\ 3\left(\frac{d}{s}\right)^2 - 2\left(\frac{d}{s}\right)^3, & 0 \le d < s, \end{cases} \qquad H_\Gamma(\mathbf{X}) = \sigma_\Gamma(\mathbf{X})\,\rho_s(D_{\partial\Gamma}(\mathbf{X})). \tag{8}$$

ρ_s increases smoothly from zero at the cut-surface boundary to one at the end of the transition band (Fig. 3(a)). For $D_{\partial\Gamma}(\mathbf{X}) \ge s$, $H_\Gamma(\mathbf{X}) = \sigma_\Gamma(\mathbf{X})$, preserving the separation between the two sides. Within the transition band, ρ_s smoothly attenuates both branches of H_Γ to zero at the cut-surface boundary (Fig. 3(b)). This construction yields a lifting profile with a continuous first derivative through the transition band and across its endpoints (Fig. 3(c)), avoiding additional gradient jumps at these junctions while preserving the intended discontinuity across the cut-surface interior.

3.3 Physics-Informed Self-Supervised Learning

Following the data-free learning formulation of Simplicits (Modi et al., 2024), we train cut-conditioned neural skinning fields by minimizing the expected elastic potential energy and encouraging orthogonality among the skinning weight functions. However, cut-conditioned learning introduces two additional challenges: First, uniform Monte Carlo sampling may provide insufficient coverage of the cut-surface neighborhood, where the prescribed cut most strongly influences the learned neural skinning fields. Second, although the lifted representation can represent discontinuities across the cut, the training objectives do not explicitly enforce them. To address these challenges, we adopt cut-aware importance sampling for potential-energy estimation and introduce a cross-side jump constraint to separate skinning weight vectors across the cut.

Physical Constraint. To ensure physical plausibility, we construct a low-energy subspace by minimizing the expected elastic potential energy over sampled subspace coordinates:

$$\mathcal{L}_{\text{pot}}(\theta) = \mathbb{E}_{\Gamma,\,\mathbf{z}\sim\mathcal{N}}\left[E_{\text{pot}}^{\theta,\Gamma}(\phi_\Gamma(\mathbf{X},\mathbf{z}))\right], \qquad E_{\text{pot}}^{\theta,\Gamma}(\phi_\Gamma(\mathbf{X},\mathbf{z})) = \int_\Omega \Psi\big(F_\Gamma^\theta(\mathbf{X},\mathbf{z})\big)\,\mathrm{d}\mathbf{X}, \tag{9}$$

where $\mathcal{N}$ denotes a Gaussian distribution over $\mathbb{R}^{12m}$ with zero mean and covariance $\mu^2 I_{12m}$, with $\mu = 0.2$, and $E_{\text{pot}}^{\theta,\Gamma}$ denotes the elastic potential energy induced by the cut-conditioned neural skinning fields. The expectation is taken over both the sampled subspace coordinates and the training cut configurations. In the spatial integral, Ψ denotes the hyperelastic energy density, and F_Γ^θ is the deformation gradient induced by $\boldsymbol{W}_\theta^\Gamma$. To improve sampling coverage near the cut surface, we adopt cut-aware importance sampling for the spatial integral. Let $d_\Gamma(\mathbf{X}) = \|\mathbf{X} - \mathbf{p}_\Gamma(\mathbf{X})\|_2$ denote the unsigned distance to the cut surface, and define the cut band and the remaining domain as:

$$\mathcal{B}_\Gamma^\varepsilon = \{\mathbf{X} \in \Omega : d_\Gamma(\mathbf{X}) < \varepsilon\}, \qquad \mathcal{R}_\Gamma^\varepsilon = \Omega \setminus \mathcal{B}_\Gamma^\varepsilon, \tag{10}$$

where $\varepsilon > 0$ is the cut-band width. For each Monte Carlo batch, we draw $N_{\mathrm{B}} = N_{\mathrm{R}} = N/2$ samples uniformly from the cut band $\mathcal{B}_\Gamma^\varepsilon$ and the remaining domain $\mathcal{R}_\Gamma^\varepsilon$, respectively. Let $\mathcal{S}_{\mathrm{B}}$ and $\mathcal{S}_{\mathrm{R}}$ denote the corresponding sample sets, and let $\mathcal{S} = \mathcal{S}_{\mathrm{B}} \cup \mathcal{S}_{\mathrm{R}}$ be the complete spatial sample batch. We evaluate the potential energy in Eq. (9) using the volume-corrected Monte Carlo estimator:

$$\widehat{E_{\mathrm{pot}}}^{\theta,\Gamma}(\phi_\Gamma(\mathbf{X},\mathbf{z})) = \frac{|\mathcal{B}_\Gamma^\varepsilon|}{N_{\mathrm{B}}} \sum_{\mathbf{X}_i \in \mathcal{S}_{\mathrm{B}}} \Psi\big(F_\Gamma^\theta(\mathbf{X}_i,\mathbf{z})\big) + \frac{|\mathcal{R}_\Gamma^\varepsilon|}{N_{\mathrm{R}}} \sum_{\mathbf{X}_i \in \mathcal{S}_{\mathrm{R}}} \Psi\big(F_\Gamma^\theta(\mathbf{X}_i,\mathbf{z})\big), \tag{11}$$

where $|\cdot|$ denotes volume. Weighting each region's sample mean by its volume yields an unbiased estimator of the domain integral, while allocating half of the sampling budget to the cut band.

Cross-Side Jump Constraint. Although the lifted representation can represent discontinuities across the cut, the learned skinning fields may remain continuous without an explicit cross-side constraint. To formulate this constraint, we define opposite-side sample sets by the sign of H_Γ:

$$\mathcal{S}_\Gamma^\pm = \{\mathbf{X} \in \mathcal{S} : \pm H_\Gamma(\mathbf{X}) > 0\}. \tag{12}$$

We independently draw random pairs $(\mathbf{X}^+, \mathbf{X}^-) \sim \mathcal{U}(\mathcal{S}_\Gamma^+) \times \mathcal{U}(\mathcal{S}_\Gamma^-)$. At each sampled point $\mathbf{X}$, we ℓ_2-normalize the skinning weight vector across handles: $\widehat{\boldsymbol{W}}_\theta^\Gamma(\mathbf{X}) = \boldsymbol{W}_\theta^\Gamma(\mathbf{X})/\|\boldsymbol{W}_\theta^\Gamma(\mathbf{X})\|_2$. We penalize the squared cosine similarity between the normalized weight vectors of each sampled pair:

$$\mathcal{L}_{\mathrm{jump}}(\theta) = \mathbb{E}_{\Gamma,\mathbf{X}^+,\mathbf{X}^-}\left[\left(\widehat{\boldsymbol{W}}_\theta^\Gamma(\mathbf{X}^+)^\top \widehat{\boldsymbol{W}}_\theta^\Gamma(\mathbf{X}^-)\right)^2\right]. \tag{13}$$

The jump loss promotes separation between paired skinning weight vectors across the cut, encouraging distinct deformation responses on its opposite sides.

Orthogonal Constraint. To enhance the expressiveness of the reduced subspace, we encourage different skinning weight functions to be mutually orthogonal (Modi et al., 2024). For a training batch $\mathcal{D} \sim \mathcal{S}$, let $\mathbf{X}_\mathcal{D}$ denote its point matrix and let $\boldsymbol{W}_\theta^\Gamma(\mathbf{X}_\mathcal{D}) \in \mathbb{R}^{|\mathcal{D}|\times m}$ denote the resulting skinning modes matrix. We define the orthogonality loss as:

$$\mathcal{L}_{\mathrm{orth}}(\theta) = \mathbb{E}_{\Gamma,\,\mathcal{D}\sim\mathcal{S}}\left[\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m}\left(\overline{\boldsymbol{W}}_{\theta,i}^\Gamma(\mathbf{X}_\mathcal{D})^\top \overline{\boldsymbol{W}}_{\theta,j}^\Gamma(\mathbf{X}_\mathcal{D})\right)^2\right], \tag{14}$$

where $\overline{\boldsymbol{W}}_{\theta,i}^\Gamma(\mathbf{X}_\mathcal{D})$ is obtained by ℓ_2-normalizing $\boldsymbol{W}_{\theta,i}^\Gamma(\mathbf{X}_\mathcal{D}) \in \mathbb{R}^{|\mathcal{D}|}$, the i-th column of the skinning modes matrix, across samples for training stability (Lei et al., 2026). The overall objective is:

$$\mathcal{L}(\theta) = \lambda_{\mathrm{pot}}\mathcal{L}_{\mathrm{pot}} + \lambda_{\mathrm{jump}}\mathcal{L}_{\mathrm{jump}} + \lambda_{\mathrm{orth}}\mathcal{L}_{\mathrm{orth}}, \tag{15}$$

where λ_{pot}, λ_{jump}, and λ_{orth} balance the corresponding constraints. However, these constraints often conflict in magnitude and direction, making naive joint optimization unstable. Following PhysSkin (Lei et al., 2026), we apply ConFIG (Liu et al., 2025b) to correct $\nabla_\theta\mathcal{L}_{\mathrm{pot}}$, $\nabla_\theta\mathcal{L}_{\mathrm{jump}}$, and $\nabla_\theta\mathcal{L}_{\mathrm{orth}}$ before each parameter update, to mitigate gradient conflicts during joint optimization. Finally, the learned fields define a compact, cut-conditioned subspace for implicit time integration in Eq. (2), enabling reduced-order simulation of 3D objects under varying cut configurations.

4 Experiments

4.1 Implementation Details

All experiments are conducted on a single NVIDIA RTX 4090 GPU. NeuralCut is implemented in PyTorch and trained using the Adam optimizer. For each 3D object, we recenter and normalize its geometry to the bounding volume $[-1, 1]^3$. Preprocessing details are provided in Appendix B. We use a 6-layer, 512-channel ResNet with ELU activations to parameterize the neural skinning field. The positional encoding in Eq. (4) uses $L = 8$ frequency bands. We set the cut-band width to $\varepsilon = 0.1$ and the transition width of the smooth gate ρ_s to $s = 0.05$. The objective weights are $\lambda_{\mathrm{pot}} = 1.0$, $\lambda_{\mathrm{jump}} = 10^6$, and $\lambda_{\mathrm{orth}} = 10^6$. NeuralCut requires only spatial samples from the reference domain Ω, without assuming a particular spatial discretization. We uniformly pre-sample a set of points from Ω for Monte Carlo integration (Modi et al., 2024; Lei et al., 2026). At each training iteration, we sample a cut configuration, partition the pre-sampled points according to the cut band and the remaining domain, and uniformly sample 4,096 points from each region. We use these samples to estimate the elastic potential energy with the volume-corrected Monte Carlo estimator in Eq. (11).

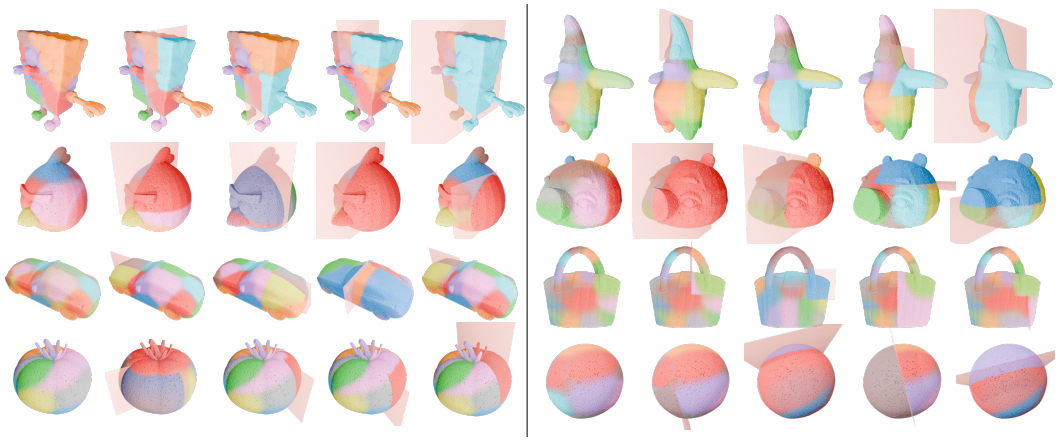

Figure 4: Cut-conditioned neural skinning modes under varying cut configurations. We visualize these modes by blending their assigned colors according to the skinning weights.

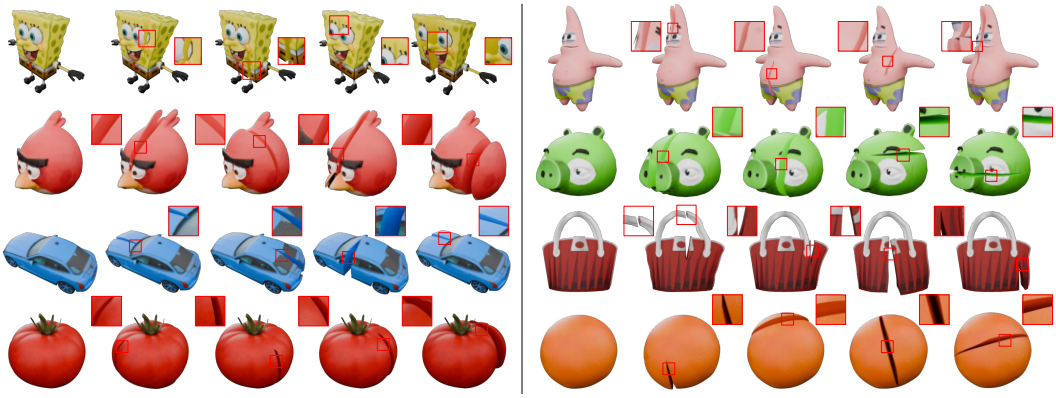

Figure 5: Qualitative results on reduced-order cut simulation under varying cut configurations. **Please refer to the supplementary video for dynamic results.**

4.2 Evaluation of Reduced-Order Cut Simulation

Baselines. Our first baseline group comprises data-free neural skinning-based reduced-order methods: Simplicits (Modi et al., 2024), FreeForm (Xiang et al., 2026), and PhysSkin (Lei et al., 2026). To our knowledge, NeuralCut is the first neural reduced-order simulation framework supporting finite cuts in deformable 3D objects. These baselines cannot simulate cuts; we compare skinning modes rather than cut simulation accuracy. Full-space baselines include FEM (Heiden et al., 2021) and MPM (Koh et al., 2026) solvers. We use high-accuracy FEM as the gold-standard reference for simulation accuracy. Appendix D details their cut representations and evolution.

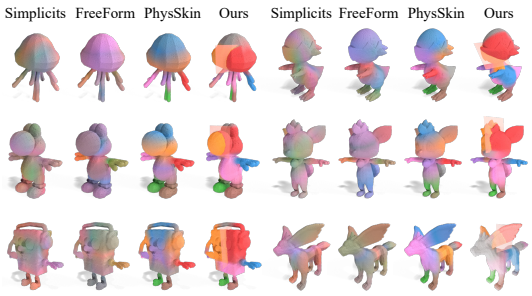

Figure 6: Comparison of skinning modes with data-free neural skinning baselines.

Skinning Comparison. Fig. 6 compares the learned skinning modes. Designed for uncut elastic objects, the baselines learn spatially continuous skinning fields without cut conditioning, so their weights remain continuous across the prescribed cut surface. In contrast, NeuralCut learns cut-conditioned skinning modes with distinct weight distributions on opposite sides of the cut.

Mesh-Based Cut Simulation. We evaluate NeuralCut on synthetic-object meshes (Xu et al., 2020; Chang et al., 2015; Sketchfab, 2026), tetrahedralized with fTetWild (Hu et al., 2020) for consistent solver inputs. FEM uses the resulting tetrahedra as simulation elements, whereas MPM and NeuralCut use the same vertices as simulation points. Appendix B details the processing of tetrahedra intersected by the cut surface. We fix a region on one side of the cut and apply a localized drag force to the opposite side. Fig. 7 compares deformations under the same cut configurations. NeuralCut produces deformation responses consistent with FEM, whereas MPM occasionally exhibits localized artifacts near the cut surface. Appendix Fig. 12 provides additional comparisons. Tab. 1 reports final-

state normalized mean squared error (NMSE) and relative deformation error (RDE). RDE quantifies the positional discrepancy from FEM relative to the magnitude of FEM deformation. It is computed as the ratio of the RMS position error to the RMS FEM displacement from rest, expressed as a percentage. Tab. 2 reports simulated vertex counts, basis-building time, and per-step runtimes on identical CPU and GPU hardware. These results demonstrate NeuralCut's excellent time efficiency. Fig. 4 visualizes learned skinning modes, and Fig. 5 shows their resulting deformations under varying cuts. Opposite cut sides deform distinctly, while the uncut region remains continuous. Appendix Figs. 13 and 14 show deformation sequences from real-time interaction in the Polyscope GUI (Sharp et al., 2019). These results show that the representation captures cut-induced displacement discontinuities in reduced-order simulation. We also couple NeuralCut with different robot configurations from RoboTwin 2.0 (Chen et al., 2025b) to simulate robotic cutting of deformable objects (Figs. 1 and 8), showing its potential for robot learning in cutting tasks. Appendix Figs. 15–17 show additional robot interactions, with simulation details in Appendix E.

Table 1: Cut simulation accuracy relative to FEM.

Method	NMSE ↓	RDE ↓
MPM	4.085×10^{-3}	5.939%
NeuralCut	$\mathbf{2.375 \times 10^{-4}}$	**1.064%**

Table 2: Cut simulation runtimes.

Object	# Simulated Vertices (K)	Basis Building Time (ms)	Time Per Step		
			Ours	FEM	MPM
Sponge	72.5	22.6	**5.4 ms**	2.17 s	2.03 s
Starfish	96.0	27.1	**6.9 ms**	2.28 s	2.06 s
Bird	66.2	17.8	**7.0 ms**	2.25 s	2.07 s
Pig	29.3	17.2	**7.0 ms**	1.64 s	2.01 s
Car	207.8	30.2	**8.8 ms**	3.97 s	2.29 s
Bag	181.2	25.4	**8.6 ms**	4.10 s	2.11 s
Tomato	79.4	24.3	**6.8 ms**	2.53 s	2.07 s
Orange	46.0	19.1	**4.3 ms**	1.86 s	1.98 s

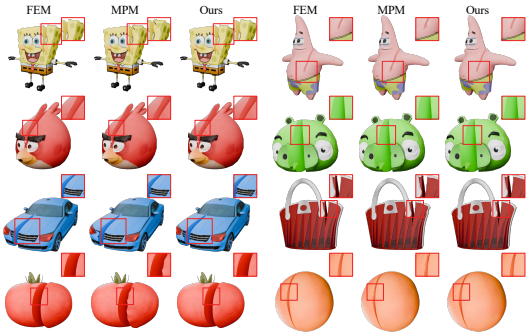

Figure 7: Qualitative comparison with full-space FEM and MPM solvers.

Simulation with 3DGS Objects. To demonstrate the discretization-agnostic formulation of NeuralCut, we apply NeuralCut to objects represented by 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023), binding the cut-conditioned neural skinning fields directly to individual Gaussians. Standard 3DGS reconstruction may retain redundant internal Gaussians and produce highly anisotropic or excessively large kernels. We therefore adopt Internal-Free Uniform Gaussian Regularization (Yang et al., 2026) to prune these redundant primitives and constrain kernel anisotropy and scale. During simulation, we apply EVSplitting (Feng et al., 2024a) to Gaussians intersected by each cut surface to reduce Gaussian overlap across the cut while preserving their visual contribution. We then evaluate the neural skinning fields at the undeformed Gaussian centers and use the resulting weights to update their centers and covariances from the reduced-order coordinates, while retaining opacity and spherical harmonic coefficients. Fig. 9 shows the resulting deformations under varying cut configurations, with detailed object appearance preserved. Appendix C details the algorithms for Gaussian binding, Gaussian regularization, and cut-aware splitting.

Cross-Object Generalization. PhysSkin (Lei et al., 2026) uses a transformer-based architecture to learn neural skinning fields with cross-object generalization. We augment its skinning decoder with $\gamma_L(H_\Gamma(\mathbf{X}))$ after its final ResNet block and train on the RigNet dataset (Xu et al., 2020) using the data-free objective in Sec. 3.3. Appendix F details the architecture. Fig. 10 and Appendix Fig. 11 show these skinning modes across objects and cut configurations.

4.3 Ablation Studies

Cut-Boundary Continuity. To evaluate the boundary attenuation in Sec. 3.2, we replace the smooth gate $\rho_s(d)$ with the hard gate $\mathbf{1}_{d>0}$, retaining zero lifting at the boundary and in the uncut region. We construct a plane Π_Γ orthogonal to Γ at the cut boundary $\partial\Gamma$ and select cut-side points $\mathcal{Q}_\Gamma$ within 10^{-3} of this plane. We orthogonally project each $\mathbf{X} \in \mathcal{Q}_\Gamma$ onto Π_Γ to obtain $\bar{\mathbf{X}}$, where $H_\Gamma(\bar{\mathbf{X}}) = 0$, and define the Uncut Continuity Discrepancy (UCD):

Table 3: Ablation study of smooth lifting for cut-boundary continuity.

Setting	UCD ($\times 10^{-3}$) ↓
Full Model	**0.194**
w/o Smooth Lifting	93.676

$$\mathrm{UCD} = \frac{1}{|\mathcal{Q}_\Gamma|} \sum_{\mathbf{X} \in \mathcal{Q}_\Gamma} \left\| \boldsymbol{W}_\theta^\Gamma(\mathbf{X}) - \boldsymbol{W}_\theta^\Gamma(\bar{\mathbf{X}}) \right\|_2 . \tag{16}$$

Tab. 3 shows lower UCD with smooth lifting, supporting continuity at the cut boundary.

Cross-Cut Separation. We evaluate cross-cut separation using two complementary metrics. Cross-Side Similarity (CSS) measures the mean squared cosine similarity of skinning weight vectors over all pairs of positive- and negative-lift samples. Opposite-Lift Similarity (OLS) measures the squared

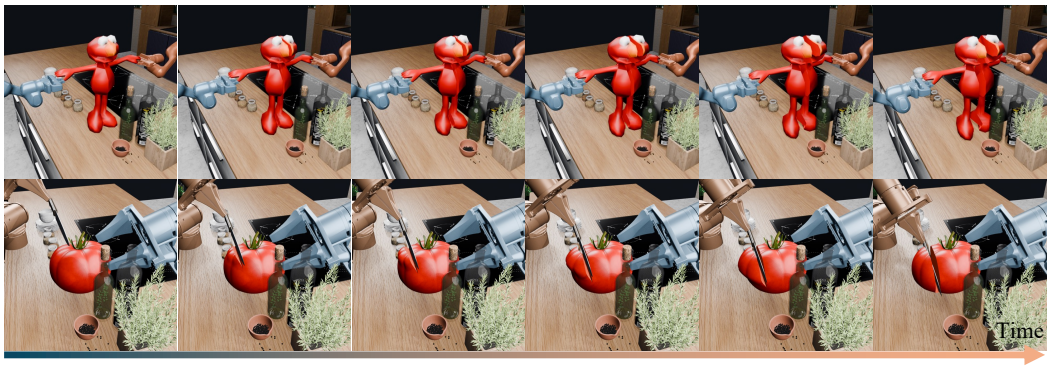

Figure 8: Qualitative results on reduced-order cut simulation coupled with robot interaction. **Please refer to the supplementary video for dynamic results.**



Figure 9: Qualitative results of reduced-order cut simulation for 3DGS objects.



Figure 10: Qualitative results on cross-object generalization using a transformer-based model. We visualize cut-conditioned neural skinning modes across diverse objects and cut configurations.

cosine similarity of skinning weight vectors at the same spatial coordinate $\mathbf{X}$ with lifting coordinate values $+1$ and -1. Lower CSS indicates greater cross-cut separation, while lower OLS indicates distinct skinning weights for opposite lifting coordinate values at the same spatial location. Tab. 4 shows that the full model achieves the lowest CSS and OLS, supporting each component's effectiveness.

Table 4: Ablation study of design choices for cross-cut separation.

Setting	CSS ($\times 10^{-3}$) ↓	OLS ↓
Full Model	**0.435**	**0.041**
w/o Pos. Enc.	6.227	0.476
w/o Smooth Lifting	7.133	0.753
w/o Cut-Aware Samp.	0.798	0.236
w/o Jump Constr.	35.362	0.723

5 Conclusion

In this paper, we present NeuralCut, a data-free learning framework that adopts a neural skinning-based reduced-order formulation for real-time cut simulation of deformable 3D objects. NeuralCut employs a geometry-derived lifting-and-restriction formulation to construct cut-conditioned neural skinning fields, which are learned through physics-informed self-supervision using cut-aware importance sampling and a cross-side jump constraint. Experiments demonstrate real-time reduced-order cut simulation across varying 3D objects and cut configurations, with objects represented as meshes or Gaussian splats. We further demonstrate the application of our method to robot interaction simulation and cross-object generalization of cut-conditioned skinning using a transformer-based model. As a limitation, NeuralCut currently assigns representative material colors to newly exposed mesh interiors; generative inpainting could provide richer interior textures.

AI USE STATEMENT

AI tools were used to improve the clarity and readability of the manuscript and assist with literature search. The authors take full responsibility for the final work.

ETHICS STATEMENT

Our study focuses on physics-based simulation of deformable-object cutting. All experimental evaluations are conducted using synthetic, publicly available, or self-captured datasets, curated to avoid sensitive or private content. We assert that this research has been carried out in accordance with the code of ethics.

REPRODUCIBILITY STATEMENT

In order to support reproducibility and verification, we present implementation and evaluation details in our paper, and will release the associated source code upon acceptance.

REFERENCES

P. Areias and Timon Rabczuk. Steiner-Point Free Edge Cutting of Tetrahedral Meshes with Applications in Fracture. *Finite Elements in Analysis and Design*, 132:27–41, 2017. doi: 10.1016/j.finel.2017.05.001.

Jernej Barbič and Doug L James. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics*, 24(3):982–990, 2005.

Cristian C. Beltran-Hernandez, Nicolas Erbetti, and Masashi Hamaya. SliceIt! – A Dual Simulator Framework for Learning Robot Food Slicing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4296–4302. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611596.

Otman Benchekroun, Jiayi Eris Zhang, Siddhartha Chaudhuri, Eitan Grinspun, Yi Zhou, and Alec Jacobson. Fast Complementary Dynamics via Skinning Eigenmodes. *ACM Transactions on Graphics*, 2023.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural Cloth Simulation. *ACM Transactions on Graphics*, 41(6), 2022. doi: 10.1145/3550454.3555491.

Daniel Bielser and Markus H. Gross. Interactive Simulation of Surgical Cuts. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, 2000.

Daniel Bielser, Pascal Glardon, Matthias Teschner, and Markus Gross. A State Machine for Real-Time Cutting of Tetrahedral Meshes. In *11th Pacific Conference on Computer Graphics and Applications*, pp. 377–386, 2003.

Oleksiy Busaryev, Tamal K. Dey, and Huamin Wang. Adaptive Fracture Simulation of Multi-Layered Thin Plates. *ACM Transactions on Graphics*, 32(4):52:1–52:6, 2013. doi: 10.1145/2461912.2461920.

Xinhao Cai, Eulalie Coevoet, Alec Jacobson, and Paul Kry. Active Learning Neural C-Space Signed Distance Fields for Reduced Deformable Self-Collision. In *Proceedings of Graphics Interface 2022*, pp. 92–100, 2022. doi: 10.20380/GI2022.11.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Yue Chang, Peter Yichen Chen, Zhecheng Wang, Maurizio M Chiaramonte, Kevin Carlberg, and Eitan Grinspun. LiCROM: Linear-subspace continuous reduced order modeling with neural fields. In *ACM SIGGRAPH Asia Conference Proceedings*, pp. 1–12, 2023.

Yue Chang, Otman Benchekroun, Maurizio M Chiaramonte, Peter Yichen Chen, and Eitan Grinspun. Shape Space Spectra. *ACM Transactions on Graphics*, 2025a.

Yue Chang, Mengfei Liu, Zhecheng Wang, Peter Yichen Chen, and Eitan Grinspun. Lifting the Winding Number: Precise Discontinuities in Neural Fields for Physics Simulation. In *ACM SIGGRAPH Conference Proceedings*, 2025b. doi: 10.1145/3721238.3730597.

Peter Yichen Chen, Maurizio M. Chiaramonte, Eitan Grinspun, and Kevin Carlberg. Model Reduction for the Material Point Method via an Implicit Neural Representation of the Deformation Map. *Journal of Computational Physics*, 478:111908, 2023a. doi: 10.1016/j.jcp.2023.111908.

Peter Yichen Chen, Jinxu Xiang, Dong Heon Cho, Yue Chang, GA Pershing, Henrique Teles Maia, Maurizio M Chiaramonte, Kevin Carlberg, and Eitan Grinspun. CROM: Continuous reduced-order modeling of PDEs using implicit neural representations. *International Conference on Learning Representations*, 2023b.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and Benchmarking for 3D Shape Variational Auto-Encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16251–16261, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

Zhili Chen, Miaojun Yao, Renguo Feng, and Huamin Wang. Physics-Inspired Adaptive Fracture Refinement. *ACM Transactions on Graphics*, 33(4):113:1–113:7, 2014. doi: 10.1145/2601097.2601115.

Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Physics Informed Neural Fields for Smoke Reconstruction with Sparse Data. *ACM Transactions on Graphics*, 41(4):119:1–119:14, 2022.

Yitong Deng, Hong-Xing Yu, Diyang Zhang, Jiajun Wu, and Bo Zhu. Fluid Simulation on Neural Flow Maps. *ACM Transactions on Graphics*, 42(6):244:1–244:21, 2023.

L. Fan, F. M. Chitalu, and T. Komura. A Hybrid Lagrangian–Eulerian Formulation of Thin-Shell Fracture. *Computer Graphics Forum*, pp. e15273, 2025.

Qi-Yuan Feng, Geng-Chen Cao, Hao-Xiang Chen, Qun-Ce Xu, Tai-Jiang Mu, Ralph Martin, and Shi-Min Hu. EVSplitting: An Efficient and Visually Consistent Splitting Algorithm for 3D Gaussian Splatting. In *ACM SIGGRAPH Asia Conference Proceedings*, pp. 1–11, 2024a. doi: 10.1145/3680528.3687592.

Xudong Feng, Huamin Wang, Yin Yang, and Weiwei Xu. Neural-Assisted Homogenization of Yarn-Level Cloth. In *ACM SIGGRAPH Conference Proceedings*, 2024b.

Lawson Fulton, Vismay Modi, David Duvenaud, David IW Levin, and Alec Jacobson. Latent-space dynamics for reduced deformable simulation. In *Computer Graphics Forum*, volume 38, pp. 379–391. Wiley Online Library, 2019.

Eric Heiden, Miles Macklin, Yashraj S. Narang, Dieter Fox, Animesh Garg, and Fabio Ramos. DiSECt: A Differentiable Simulation Engine for Autonomous Robotic Cutting. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.067.

Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. Subspace Neural Physics: Fast Data-Driven Interactive Simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2019. doi: 10.1145/3309486.3340245.

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics*, 2020.

Jingwei Huang, Hao Su, and Leonidas Guibas. Robust Watertight Manifold Surface Generation Method for ShapeNet Models. *arXiv preprint arXiv:1802.01698*, 2018.

Jingwei Huang, Yichao Zhou, and Leonidas Guibas. ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups. *arXiv preprint arXiv:2005.11621*, 2020.

Pranav Jain, Ziyin Qu, Peter Yichen Chen, and Oded Stein. Neural Monte Carlo Fluid Simulation. In *ACM SIGGRAPH Conference Proceedings*, 2024. doi: 10.1145/3641519.3657438.

Prajjwal Jamdagni and Yan-Bin Jia. Robotic Cutting of Solids Based on Fracture Mechanics and FEM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8246–8251, 2019. doi: 10.1109/IROS40897.2019.8967988.

Lenka Jeřábková and Torsten Kuhlen. Stable Cutting of Deformable Objects in Virtual Environments Using XFEM. *IEEE Computer Graphics and Applications*, 29(2):61–71, 2009. doi: 10.1109/MCG.2009.32.

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. ACM, 2016.

Navami Kairanda, Marc Habermann, Christian Theobalt, and Vladislav Golyanik. NeuralClothSim: Neural Deformation Fields Meet the Thin Shell Theory. In *Advances in Neural Information Processing Systems*, 2024.

Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross. Enrichment Textures for Detailed Cutting of Shells. *ACM Transactions on Graphics*, 28(3):50:1–50:10, 2009. doi: 10.1145/1531326.1531356.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Byungsoo Kim, Vinicius C. Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. In *Computer Graphics Forum*, volume 38, pp. 59–70, 2019.

Hyunseo Koh, Chang-Yong Song, Youngjae Choi, Misa Viveiros, David Hyde, and Heewon Kim. CulinaryCut: A Physics-aware Vision-Language-Action Benchmark for Food Cutting via Material Point Method. In *Proceedings of the European Conference on Computer Vision*, 2026.

Dan Koschier, Jan Bender, and Nils Thuerey. Robust eXtended Finite Elements for Complex Cutting of Deformables. *ACM Transactions on Graphics*, 36(4):55:1–55:13, 2017. doi: 10.1145/3072959.3073666.

Yuanhang Lei, Tao Cheng, Xingxuan Li, Boming Zhao, Siyuan Huang, Ruizhen Hu, Peter Yichen Chen, Hujun Bao, and Zhaopeng Cui. PhysSkin: Real-Time and Generalizable Physics-Based Animation via Self-Supervised Neural Skinning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Minchen Li, Chenfanfu Jiang, Zhaofeng Luo, Wenxin Du, Chang Yu, Žiga Kovačič, and Tianyi Xie. *Physics-Based Simulation*. March 2026. doi: 10.5281/zenodo.20597655. URL https://doi.org/10.5281/zenodo.20597655. Open-source online book. Live version available at https://phys-sim-book.github.io/.

Li Liao, Pengfei Shen, and Yifan Peng. Boundary-aware Neural Model Reduction for PDEs. In *ACM SIGGRAPH Conference Proceedings*, pp. 1–11, 2026.

Mengfei Liu, Yue Chang, Zhecheng Wang, Peter Yichen Chen, and Eitan Grinspun. Precise Gradient Discontinuities in Neural Fields for Subspace Physics. In *ACM SIGGRAPH Asia Conference Proceedings*, 2025a. doi: 10.1145/3757377.3763810.

Qiang Liu, Mengyu Chu, and Nils Thuerey. ConFIG: Towards conflict-free training of physics informed neural networks. *International Conference on Learning Representations*, 2025b.

Aoran Lyu, Shixian Zhao, Chuhua Xian, Zhihao Cen, Hongmin Cai, and Guoxin Fang. Accelerate Neural Subspace-Based Reduced-Order Solver of Deformable Simulation by Lipschitz Optimization. *ACM Transactions on Graphics*, 43(6):1–10, 2024.

Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David IW Levin. Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Transactions on Graphics*, 43(4):1–11, 2024.

Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3D Gaussian Ray Tracing: Fast Tracing of Particle Scenes. *ACM Transactions on Graphics*, 2024.

Nicolas Moës, John Dolbow, and Ted Belytschko. A Finite Element Method for Crack Growth without Remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

Neil Molino, Zhaosheng Bao, and Ronald Fedkiw. A Virtual Node Algorithm for Changing Mesh Topology During Simulation. *ACM Transactions on Graphics*, 23(3):385–392, 2004. doi: 10.1145/1015706.1015734.

Andrew B. Mor and Takeo Kanade. Modifying Soft Tissue Models: Progressive Cutting with Minimal New Element Creation. In *Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 598–607, 2000.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Christoph J. Paulus, Lionel Untereiner, Hadrien Courtecuisse, Stéphane Cotin, and David Cazier. Virtual Cutting of Deformable Objects Based on Efficient Topological Operations. *The Visual Computer*, 31(6–8):831–841, 2015. doi: 10.1007/s00371-015-1123-x.

Cristian Romero, Dan Casas, Jesús Pérez, and Miguel Otaduy. Learning Contact Corrections for Handle-Based Subspace Dynamics. *ACM Transactions on Graphics*, 40(4), 2021. doi: 10.1145/3450626.3459875.

Nicholas Sharp, Cristian Romero, Alec Jacobson, Etienne Vouga, Paul Kry, David IW Levin, and Justin Solomon. Data-free learning of reduced-order kinematics. In *ACM SIGGRAPH Conference Proceedings*, pp. 1–9, 2023.

Nicholas Sharp et al. Polyscope, 2019. https://www.polyscope.run.

Siyuan Shen, Yin Yang, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. High-order differentiable autoencoder for nonlinear model reduction. *ACM Transactions on Graphics*, 2021.

Eftychios Sifakis and Jernej Barbic. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, pp. 1–50. 2012.

Eftychios Sifakis, Kevin G. Der, and Ronald Fedkiw. Arbitrary Cutting of Deformable Tetrahedralized Objects. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 73–80, 2007. doi: 10.2312/SCA/SCA07/073-080.

Sketchfab. Sketchfab: The Leading Platform for 3D & AR on the Web. Available at sketchfab.com, 2026.

Haozhe Su, Tao Xue, Chengguizi Han, and Mridul Aanjaneya. A-ULMPM: An Adaptively Updated Lagrangian Material Point Method for Efficient Physics Simulation without Numerical Fracture. *Computer Graphics Forum*, 41(2):325–341, 2022.

Yuanyuan Tao, Ivan Puhachov, Derek Nowrouzezahrai, and Paul Kry. Neural Implicit Reduced Fluid Simulation. In *ACM SIGGRAPH Asia Conference Proceedings*, pp. 1–11, 2024.

Quoc-Minh Ton-That, Paul G. Kry, and Sheldon Andrews. Generalized eXtended Finite Element Method for Deformable Cutting via Boolean Operations. *Computer Graphics Forum*, 43(8), 2024.

Jiahong Wang, Yinwei Du, Stelian Coros, and Bernhard Thomaszewski. Neural Modes: Self-supervised Learning of Nonlinear Modal Subspaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23158–23167, 2024a.

Yiming Wang, Siyu Tang, and Mengyu Chu. Physics-Informed Learning of Characteristic Trajectories for Smoke Reconstruction. In *ACM SIGGRAPH Conference Proceedings*, 2024b. doi: 10.1145/3641519.3657483.

Jun Wu, Rüdiger Westermann, and Christian Dick. A Survey of Physically Based Simulation of Cuts in Deformable Bodies. *Computer Graphics Forum*, 34(6):161–187, 2015. doi: 10.1111/cgf.12528.

Donglai Xiang, Vismay Modi, Rishit Dagli, Ty Trusty, Gilles Daviet, Anka He Chen, Nicholas Sharp, and David I.W. Levin. FreeForm: Reduced-Order Deformable Simulation from Particle-Based Skinning Eigenmodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32475–32484, June 2026.

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. RigNet: Neural rigging for articulated characters. *ACM Transactions on Graphics*, 2020.

Shuqi Yang, Xingzhe He, and Bo Zhu. Learning Physical Constraints with Neural Projections. In *Advances in Neural Information Processing Systems*, 2020.

Zesong Yang, Yuanhang Lei, Liyuan Cui, Yihang Chen, Jiaer Huang, Boming Zhao, Peter Yichen Chen, Hujun Bao, and Zhaopeng Cui. MeGAS: Thermomechanical Dynamic Gaussian Splatting for Thermophysical Scene Editing. In *Proceedings of the European Conference on Computer Vision*, 2026.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3DShape2VecSet: A 3D Shape Representation for Neural Fields and Generative Diffusion Models. *ACM Transactions on Graphics*, 42(4):1–16, 2023.

Meng Zhang and Jun Li. Neural Garment Dynamic Super-Resolution. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024. doi: 10.1145/3680528.3687610.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3D Shape Generation Based on Shape-Image-Text Aligned Latent Representation. *Advances in Neural Information Processing Systems*, 36: 73969–73982, 2023.

Zeshun Zong, Xuan Li, Minchen Li, Maurizio M Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. In *ACM SIGGRAPH Asia Conference Proceedings*, pp. 1–11, 2023.

APPENDIX

A BACKGROUND

A.1 CONTINUUM MECHANICS

NeuralCut follows the standard continuum-mechanics formulation for deformable 3D objects (Sifakis & Barbic, 2012). For the deformation map $\boldsymbol{x} = \phi_\Gamma(\mathbf{X}, \mathbf{z})$ introduced in Eq. (1), the deformation gradient and its Jacobian determinant are:

$$F_\Gamma(\mathbf{X}, \mathbf{z}) = \frac{\partial \phi_\Gamma(\mathbf{X}, \mathbf{z})}{\partial \mathbf{X}}, \qquad J_\Gamma(\mathbf{X}, \mathbf{z}) = \det F_\Gamma(\mathbf{X}, \mathbf{z}). \tag{17}$$

A hyperelastic constitutive model assigns the strain-energy density $\Psi(F_\Gamma)$ to each material point. The corresponding elastic potential energy is obtained by integrating this density over the reference domain Ω, as used in Eqs. (2) and (9).

Linear Elasticity. For small deformations, the linear strain tensor and strain-energy density are:

$$\begin{aligned} \boldsymbol{\varepsilon}(F) &= \frac{1}{2}(F^\top + F) - \mathbf{I}_3, \\ \Psi_{\text{lin}}(F) &= \mu_{\text{mat}}\, \boldsymbol{\varepsilon}(F) : \boldsymbol{\varepsilon}(F) + \frac{\lambda_{\text{mat}}}{2} \operatorname{tr}(\boldsymbol{\varepsilon}(F))^2 . \end{aligned} \tag{18}$$

Here, : denotes double contraction. The Lamé coefficients are determined by Young's modulus Y and Poisson's ratio ν as:

$$\mu_{\text{mat}} = \frac{Y}{2(1+\nu)}, \qquad \lambda_{\text{mat}} = \frac{Y\nu}{(1+\nu)(1-2\nu)}. \tag{19}$$

Neo-Hookean Elasticity. To model large deformations, we use the compressible Neo-Hookean energy:

$$\Psi_{\text{NH}}(F) = \frac{\mu_{\text{mat}}}{2}\left(\operatorname{tr}(F^\top F) - 3\right) - \mu_{\text{mat}} \log J + \frac{\lambda_{\text{mat}}}{2}(\log J)^2, \qquad J = \det F, \tag{20}$$

The corresponding first Piola–Kirchhoff stress is:

$$\mathbf{P}(F) = \frac{\partial \Psi_{\text{NH}}}{\partial F} = \mu_{\text{mat}}(F - F^{-\top}) + \lambda_{\text{mat}} \log(J) F^{-\top}. \tag{21}$$

Progressive Constitutive Interpolation. Following Simplicits (Modi et al., 2024), we stabilize the early stages of physics-informed training by progressively interpolating between the two constitutive models:

$$\Psi_\tau(F) = (1 - \alpha_\tau)\Psi_{\text{lin}}(F) + \alpha_\tau \Psi_{\text{NH}}(F), \qquad \alpha_\tau \in [0, 1], \tag{22}$$

where τ denotes the training iteration and α_τ increases smoothly from zero to one. This continuation begins with the well-conditioned small-deformation model and gradually introduces the nonlinear material response used for the final neural skinning fields.

A.2 FULL-SPACE FEM IMPLICIT INTEGRATION

We next summarize the full-space implicit integrator underlying the reduced-order formulation in Sec. 3.1. Let $\mathbf{x}_t, \mathbf{v}_t \in \mathbb{R}^{3n}$ denote the nodal positions and velocities of a finite-element discretization at time step t. Given the full-space mass matrix $\mathbf{M}_{\text{full}}$, time-step size h, internal elastic force $\mathbf{f}_{\text{int}}$, and external force $\mathbf{f}_{\text{ext}}$, backward Euler updates the state according to:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\mathbf{v}_{t+1}, \qquad \mathbf{M}_{\text{full}}(\mathbf{v}_{t+1} - \mathbf{v}_t) = h\left(\mathbf{f}_{\text{int}}(\mathbf{x}_{t+1}) + \mathbf{f}_{\text{ext}}\right). \tag{23}$$

We define the inertial prediction and internal elastic force as:

$$\mathbf{y}_t = \mathbf{x}_t + h\mathbf{v}_t + h^2 \mathbf{M}_{\text{full}}^{-1} \mathbf{f}_{\text{ext}}, \qquad \mathbf{f}_{\text{int}}(\mathbf{x}) = -\nabla E_{\text{pot}}(\mathbf{x}). \tag{24}$$

Substitution into the backward Euler update yields the nonlinear equilibrium condition:

$$\frac{1}{h^2}\mathbf{M}_{\text{full}}(\mathbf{x}_{t+1} - \mathbf{y}_t) + \nabla E_{\text{pot}}(\mathbf{x}_{t+1}) = \mathbf{0}. \tag{25}$$

Incremental Potential. Eq. (25) is the stationarity condition of the incremental potential:

$$\mathcal{I}_t(\mathbf{x}) = \frac{1}{2h^2}\|\mathbf{x}-\mathbf{y}_t\|^2_{\mathbf{M}_{\text{full}}} + E_{\text{pot}}(\mathbf{x}), \qquad \mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \mathcal{I}_t(\mathbf{x}). \tag{26}$$

For finite elements $e \in \mathcal{T}$ with reference volumes V_e, the elastic potential is assembled as:

$$E_{\text{pot}}(\mathbf{x}) = \sum_{e\in\mathcal{T}} V_e \Psi(F_e(\mathbf{x}))\,. \tag{27}$$

Newton Solution. At Newton iteration k, we linearize the stationarity condition and solve:

$$\mathbf{H}^k \Delta\mathbf{x}^k = -\nabla\mathcal{I}_t(\mathbf{x}^k), \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k, \qquad \mathbf{H}^k = \frac{1}{h^2}\mathbf{M}_{\text{full}} + \nabla^2 E_{\text{pot}}(\mathbf{x}^k). \tag{28}$$

The effective stiffness matrix $\mathbf{H}^k$ can be regularized to maintain positive definiteness, while its finite-element sparsity permits efficient solution of the linearized system. Nevertheless, the optimization dimension still scales with the full set of nodal coordinates. NeuralCut instead restricts the deformation to the neural skinning subspace in Eq. (1) and advances only the reduced-order coordinates $\mathbf{z}$, yielding the reduced implicit update in Eq. (2).

B Data Preprocessing

Watertight Mesh Conversion. For mesh-based training and evaluation, we first convert each input triangle mesh into a watertight surface. Surface holes, non-manifold structures, and inconsistent face orientations can make inside–outside classification ambiguous and consequently compromise volumetric point sampling. We therefore adopt the open-source robust watertight manifold generation methods of Huang et al. (2018; 2020) to repair and close the input geometry. The resulting consistently oriented closed manifold defines a well-formed interior domain for subsequent geometric queries and stochastic volumetric cubature.

Surface Point Sampling. For experiments that require a surface point cloud, we employ Sharp Edge Sampling (SES) (Chen et al., 2025a). In addition to sampling the surface area, SES allocates samples to geometrically salient edges, improving the coverage of thin structures and high-curvature features that may be sparsely represented by area-only sampling. We follow the sampling procedure of Dora (Chen et al., 2025a) and retain the corresponding surface normals for the cross-object neural skinning model described in Appendix F.

Volumetric Point Sampling. The watertight surface permits reliable ray-based classification of uniformly sampled candidate points (Moenne-Loccoz et al., 2024). We draw candidates uniformly from the normalized bounding volume $[-1,1]^3$, cast rays against the oriented surface mesh, and retain the points classified inside the enclosed domain. This rejection-sampling procedure produces uniformly distributed interior points without constructing a tetrahedral mesh. The resulting points form the pre-sampled stochastic cubature set used by the cut-aware Monte Carlo procedure described in Sec. 3.3.

Interior Cut-Surface Extraction and Texturing. To represent newly exposed interiors, we intersect the prescribed cut surface Γ with an auxiliary tetrahedral mesh of the rest shape used only for surface extraction. For every crossed tetrahedral edge (i,j), the rest-space intersection vertex is determined by the edge parameter $u_{ij} \in [0,1]$, while its two copies are updated by the corresponding one-sided deformation maps:

$$\widetilde{\mathbf{X}}_{ij} = (1-u_{ij})\mathbf{X}_i + u_{ij}\mathbf{X}_j, \qquad \widetilde{\mathbf{x}}^{\pm}_{ij}(t) = \phi^{\pm}_{\Gamma}\left(\widetilde{\mathbf{X}}_{ij}, \mathbf{z}_t\right). \tag{29}$$

Here, $\phi^{\pm}_{\Gamma}$ denotes evaluation on the two one-sided branches of the cut-conditioned deformation map. The intersections within each tetrahedron form a triangle or quadrilateral, which we triangulate and duplicate with opposite winding to obtain two independently indexed cut sides. The resulting faces are appended to the surface mesh as $\mathcal{F}_{\text{int}}$, while the original boundary triangles form $\mathcal{F}_{\text{ext}}$. The added vertices therefore follow the reduced-order deformation without introducing additional degrees of freedom. For texturing, we register the source textured surface to the target rest surface and transfer exterior appearance through surface projection and barycentric texture sampling. Since the interior faces have no source-surface correspondence, we assign them a representative material color $\mathbf{c}_{\text{cut}}$ estimated from the source texture. All texture coordinates and material assignments are fixed in the rest configuration and reused throughout the simulation.

C SIMULATION WITH 3D GAUSSIAN REPRESENTATIONS

This section provides the technical details of coupling NeuralCut with 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023). We first regularize the reconstructed Gaussian representation, split the Gaussians intersected by a prescribed cut, and then evaluate the cut-conditioned neural skinning fields at the resulting Gaussian centers. The resulting weights bind the Gaussian representation to the reduced physical state used for simulation.

C.1 CONTINUOUS GAUSSIAN BINDING

A 3DGS representation consists of anisotropic Gaussian primitives $\mathcal{G} = \{\mathcal{G}_g\}_{g=1}^{N_G}$. In the rest configuration, each primitive is parameterized as:

$$\mathcal{G}_g^0 = \left\{\boldsymbol{\mu}_g^0, \mathbf{R}_g^0, \boldsymbol{s}_g^0, \alpha_g, \boldsymbol{c}_g\right\}, \qquad \boldsymbol{\Sigma}_g^0 = \mathbf{R}_g^0 \mathbf{S}_g^0 (\mathbf{S}_g^0)^\top (\mathbf{R}_g^0)^\top, \quad \mathbf{S}_g^0 = \mathrm{diag}(\boldsymbol{s}_g^0), \tag{30}$$

where $\boldsymbol{\mu}_g^0 \in \mathbb{R}^3$ is the Gaussian center, $\mathbf{R}_g^0 \in \mathrm{SO}(3)$ and $\boldsymbol{s}_g^0 \in \mathbb{R}_+^3$ define its covariance, α_g is its opacity, and $\boldsymbol{c}_g$ contains its spherical harmonic appearance coefficients.

After the cut-aware splitting described in Appendix C.3, we evaluate the neural skinning weights $\boldsymbol{W}_i^\Gamma(\boldsymbol{\mu}_g^0)$ at the rest center of every resulting Gaussian. Following the deformation map in Eq. (1), the corresponding local affine transformation is:

$$\mathbf{T}_g(\mathbf{z}) = [\mathbf{A}_g(\mathbf{z})\ \boldsymbol{t}_g(\mathbf{z})] = [\mathbf{I}_3\ \mathbf{0}] + \sum_{i=1}^{m} \boldsymbol{W}_i^\Gamma(\boldsymbol{\mu}_g^0)\mathbf{Z}_i, \tag{31}$$

where $\mathbf{A}_g \in \mathbb{R}^{3\times3}$ and $\boldsymbol{t}_g \in \mathbb{R}^3$ are the linear and translational components, respectively. At each simulation step, the current reduced-order coordinates update the Gaussian center and covariance as:

$$\boldsymbol{\mu}_g(\mathbf{z}) = \mathbf{A}_g(\mathbf{z})\boldsymbol{\mu}_g^0 + \boldsymbol{t}_g(\mathbf{z}), \qquad \boldsymbol{\Sigma}_g(\mathbf{z}) = \mathbf{A}_g(\mathbf{z})\boldsymbol{\Sigma}_g^0\mathbf{A}_g(\mathbf{z})^\top. \tag{32}$$

The opacity and spherical harmonic coefficients retain their rest values. The skinning weights are evaluated once after the cut-dependent splitting, whereas the local affine transformations are updated from the current reduced-order coordinates at every simulation step.

C.2 INTERNAL-FREE UNIFORM GAUSSIAN REGULARIZATION

Standard 3DGS reconstruction is supervised by external views and can retain redundant internal Gaussians. These primitives may become visible as floaters after large deformation. Moreover, highly anisotropic or excessively large Gaussian kernels can produce needle-like artifacts. We therefore adopt the Internal-Free Uniform Gaussian Regularization introduced by MeGAS (Yang et al., 2026) before physical simulation.

For each Gaussian, we record its maximum alpha-blending contribution $w_g^{\max}$ over all pixels in the training views and remove primitives whose contribution is below a threshold τ_{prune}:

$$\mathcal{G}' = \left\{\mathcal{G}_g \in \mathcal{G} \;\middle|\; w_g^{\max} > \tau_{\text{prune}}\right\}. \tag{33}$$

Let $s_g^1 \geq s_g^2 \geq s_g^3$ denote the ordered principal scales of $\mathcal{G}_g$. To suppress needle-like kernels, we penalize an excessive ratio between the two largest principal scales with the following loss:

$$\mathcal{L}_{\text{aniso}} = \frac{1}{|\mathcal{G}'|} \sum_{\mathcal{G}_g \in \mathcal{G}'} \max\left(\frac{s_g^1}{s_g^2} - a,\ 0\right), \tag{34}$$

where a is the anisotropy threshold. We additionally clamp the principal scales component-wise as $\boldsymbol{s}_g \leftarrow \min(\boldsymbol{s}_g, \tau_{\text{scale}})$ to prevent excessively large kernels. After pruning, the remaining Gaussian primitives are fine-tuned with the original 3DGS reconstruction objective.

C.3 EFFICIENT AND VISUALLY CONSISTENT 3D GAUSSIAN SPLITTING

An anisotropic Gaussian intersected by a cut surface can extend across both sides of the cut and blur the resulting boundary. For every cut introduced during simulation, we therefore identify the

intersected Gaussians from the cut position and process them with EVSplitting (Feng et al., 2024a) before evaluating their neural skinning weights. Gaussians outside the finite extent of the cut remain unchanged.

Locally, let the cut surface be represented by the plane $\mathcal{P}_\Gamma = \{\mathbf{X} \in \mathbb{R}^3 \mid \boldsymbol{n}_\Gamma^\top \mathbf{X} + b_\Gamma = 0\}$, where $\boldsymbol{n}_\Gamma$ is a unit normal. For a Gaussian with center $\boldsymbol{\mu}_g^0$ and covariance $\boldsymbol{\Sigma}_g^0$, we define:

$$d_g = \boldsymbol{n}_\Gamma^\top \boldsymbol{\mu}_g^0 + b_\Gamma, \qquad \tau_g = \sqrt{\boldsymbol{n}_\Gamma^\top \boldsymbol{\Sigma}_g^0 \boldsymbol{n}_\Gamma}, \qquad \boldsymbol{l}_g = \boldsymbol{\Sigma}_g^0 \boldsymbol{n}_\Gamma. \tag{35}$$

Only Gaussians whose effective support intersects the finite cut surface are split. Following EVSplitting, we define the mass fractions and the normalized boundary density as:

$$C_g^\pm = \frac{1}{2}\left(1 \pm \operatorname{erf}\left(\frac{d_g}{\sqrt{2}\tau_g}\right)\right), \qquad D_g = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{d_g^2}{2\tau_g^2}\right). \tag{36}$$

The parent Gaussian is replaced by two primitives on the negative and positive sides of the cut, with opacities and centers:

$$\alpha_g^\pm = \alpha_g C_g^\pm, \qquad \boldsymbol{\mu}_g^- = \boldsymbol{\mu}_g^0 - \frac{\boldsymbol{l}_g D_g}{\tau_g C_g^-}, \qquad \boldsymbol{\mu}_g^+ = \boldsymbol{\mu}_g^0 + \frac{\boldsymbol{l}_g D_g}{\tau_g C_g^+}, \tag{37}$$

and covariances:

$$\boldsymbol{\Sigma}_g^- = \boldsymbol{\Sigma}_g^0 + \frac{\boldsymbol{l}_g \boldsymbol{l}_g^\top}{\tau_g^2}\left(\frac{d_g D_g}{\tau_g C_g^-} - \frac{D_g^2}{(C_g^-)^2}\right), \tag{38}$$

$$\boldsymbol{\Sigma}_g^+ = \boldsymbol{\Sigma}_g^0 - \frac{\boldsymbol{l}_g \boldsymbol{l}_g^\top}{\tau_g^2}\left(\frac{d_g D_g}{\tau_g C_g^+} + \frac{D_g^2}{(C_g^+)^2}\right). \tag{39}$$

This closed-form construction preserves the zero-, first-, and second-order moments of the weighted Gaussian distribution while reducing Gaussian overflow across the cut. The spherical harmonic coefficients of the parent are copied to both resulting primitives. Finally, we separately evaluate $\boldsymbol{W}_\theta^\Gamma(\boldsymbol{\mu}_g^-)$ and $\boldsymbol{W}_\theta^\Gamma(\boldsymbol{\mu}_g^+)$ at the two split centers. The resulting cut-conditioned weights are then used in Eq. (31) to drive the two Gaussian primitives during reduced-order simulation.

D Details on Baseline Methods

FEM. DiSECt (Heiden et al., 2021) prescribes the complete cut surface before simulation and duplicates every tetrahedron intersected by this surface. Virtual nodes are inserted at the intersections between the cut surface and tetrahedral edges, producing independent element copies on the two sides of the cut. Pairs of virtual nodes originating from the same edge are connected by cutting springs. The knife is represented by a signed-distance field, and penalty-based contact forces are evaluated between the knife and the intersected edge segments. As the knife loads the interface, the corresponding spring stiffness is reduced in proportion to the contact-force magnitude. Progressive weakening of these springs provides a continuous damage model, and material separation occurs as their stiffness approaches zero.

MPM. The CulinaryCut simulator (Koh et al., 2026) uses the moving-least-squares material point method (MLS-MPM). The deformable body is represented by Lagrangian material points coupled through a background Eulerian grid, with each point carrying its deformation gradient and a scalar damage state. Particle mass and momentum are transferred to the grid, where corotated elasticity, J_2 plasticity, and contact are evaluated before the updated state is transferred back to the particles. The knife is represented as a signed-distance collider, and its momentum exchange with the material determines the contact impulse. The resulting local damage evolution reduces the material resistance along the knife trajectory and allows the particle representation to separate without conforming remeshing, thereby producing the cut topology through the particle–grid dynamics.

E Robot Interaction Simulation

To couple NeuralCut with articulated robot manipulation, we use gripper-equipped robot configurations from RoboTwin 2.0 (Chen et al., 2025b). Each robot is loaded from its URDF as a fixed-base

articulated model, retaining the collision geometry of each link. We specify a sequence of end-effector key poses for approach, contact, and cutting or dragging, and interpolate the target pose over time. In bimanual scenes, the two manipulators are controlled independently, allowing one gripper to stabilize the object while the other executes the prescribed interaction.

At each frame, a Jacobian-based inverse-kinematics controller maps the six-dimensional end-effector pose error $\boldsymbol{\xi}_t$ to the target joint velocities:

$$\dot{\boldsymbol{q}}_t^{\mathrm{tar}} = \mathbf{J}_t^{\dagger}\boldsymbol{\xi}_t + \left(\mathbf{I} - \mathbf{J}_t^{\dagger}\mathbf{J}_t\right) k_{\mathrm{ns}}\left(\boldsymbol{q}_{\mathrm{ref}} - \boldsymbol{q}_t\right), \tag{40}$$

where $\mathbf{J}_t$ is the end-effector spatial Jacobian, $\mathbf{J}_t^{\dagger}$ is its pseudoinverse, and the null-space term regularizes the arm toward a reference posture $\boldsymbol{q}_{\mathrm{ref}}$. The finger joints are driven directly by the prescribed gripper state. We first advance the robot articulation and then evaluate contacts between the updated link geometries and the deformable object. The resulting interaction forces are incorporated into the reduced-order implicit solve, coupling the robot motion with the cut-conditioned deformation simulated by NeuralCut.

F Cross-Object Generalization

3D Shape VAE. A pretrained 3D shape VAE represents object geometry in a shared latent space. 3DShape2VecSet represents a 3D neural field with a compact set of latent tokens (Zhang et al., 2023), and Michelangelo adopts this transformer-based design in its shape VAE (Zhao et al., 2023). Given the preprocessed surface samples described in Appendix B, farthest point sampling selects a sparse set $\mathbf{P}_s$ from the dense point cloud $\mathbf{P}_d$. These sparse points aggregate the dense surface geometry through cross-attention, after which K self-attention blocks produce the latent set $\mathbf{F}_s \in \mathbb{R}^{M\times d}$:

$$\mathbf{C}_s = \mathrm{CrossAttn}(\gamma(\mathbf{P}_s), \gamma(\mathbf{P}_d)), \qquad \mathbf{F}_s = \mathrm{SelfAttn}^{(1:K)}(\mathbf{C}_s), \tag{41}$$

where γ denotes point-wise positional encoding. The geometry decoder refines $\mathbf{F}_s$ and conditions each spatial query $\mathbf{X} \in \mathbb{R}^3$ on the resulting tokens to predict its occupancy:

$$\widehat{o}(\mathbf{X}) = f_{\mathrm{geo}}(\mathrm{CrossAttn}(\gamma(\mathbf{X}), \mathrm{SelfAttn}_{\mathrm{dec}}(\mathbf{F}_s))). \tag{42}$$

The encoder and geometry decoder are jointly pretrained through occupancy reconstruction and variational latent regularization. We use the pretrained Michelangelo configuration with 4,096 surface samples, $M = 256$ latent tokens of dimension $d = 768$, and $K = 8$ self-attention blocks.

PhysSkin-Based Cut-Conditioned Skinning Model. PhysSkin (Lei et al., 2026) builds cross-object generalization on the pretrained 3D shape VAE by discarding its geometry decoder, freezing the Michelangelo encoder, and using $\mathbf{F}_s$ as the object-specific geometry condition. It then decodes $\mathbf{F}_s$ into spatially continuous neural skinning fields. First, m learnable handle queries $\mathbf{Q}_h \in \mathbb{R}^{m\times d}$ extract the handle latents $\mathbf{F}_h \in \mathbb{R}^{m\times d}$. A second cross-attention block then produces a point-wise feature $\mathbf{F}_p(\mathbf{X}) \in \mathbb{R}^d$ for each $\mathbf{X} \in \Omega$:

$$\mathbf{F}_h = \mathrm{CrossAttn}(\mathbf{Q}_h, \mathrm{SelfAttn}(\mathbf{F}_s)), \qquad \mathbf{F}_p(\mathbf{X}) = \mathrm{CrossAttn}(\gamma(\mathbf{X}), \mathbf{F}_h). \tag{43}$$

This two-stage decoder transfers the object-level shape representation to spatially varying skinning features through a compact set of learned handles.

To support varying cut configurations while preserving the pretrained shape representation, we retain PhysSkin's ResNet-based skinning decoder and introduce the cut condition only after its final ResNet block. Let $\mathbf{f}_\theta(\mathbf{X};\mathbf{P})$ denote its output feature. The cut-conditioned neural skinning weights are computed as:

$$\mathbf{f}_\theta(\mathbf{X};\mathbf{P}) = \mathrm{ResNet}_\theta(\mathbf{F}_p(\mathbf{X}) \oplus \gamma(\mathbf{X})), \tag{44}$$

$$\boldsymbol{W}_\theta^{\Gamma}(\mathbf{X};\mathbf{P}) = f_{\mathrm{out},\theta}(\mathbf{f}_\theta(\mathbf{X};\mathbf{P}) \oplus \gamma_L(H_\Gamma(\mathbf{X}))), \tag{45}$$

where $\oplus$ denotes concatenation and $f_{\mathrm{out},\theta}$ is the output layer. The notation $\boldsymbol{W}_\theta^{\Gamma}(\mathbf{X};\mathbf{P}) \in \mathbb{R}^m$ explicitly indicates its dependence on the input object. The pretrained shape encoder and the attention-based feature decoder remain unchanged. We train the model on RigNet using the same data-free objective as in Sec. 3.3.

G Additional Qualitative Results

Cross-Object Generalization. Fig. 11 presents additional cut-conditioned skinning fields from the Transformer-based extension across diverse 3D shapes and cut configurations.

Comparison with Full-Space Solvers. Fig. 12 presents additional qualitative comparisons of NeuralCut, full-space FEM, and MPM under different cut configurations.

Interactive Simulation. Figs. 13–14 present interactive GUI simulations, while Figs. 15–17 show robot interaction simulations.

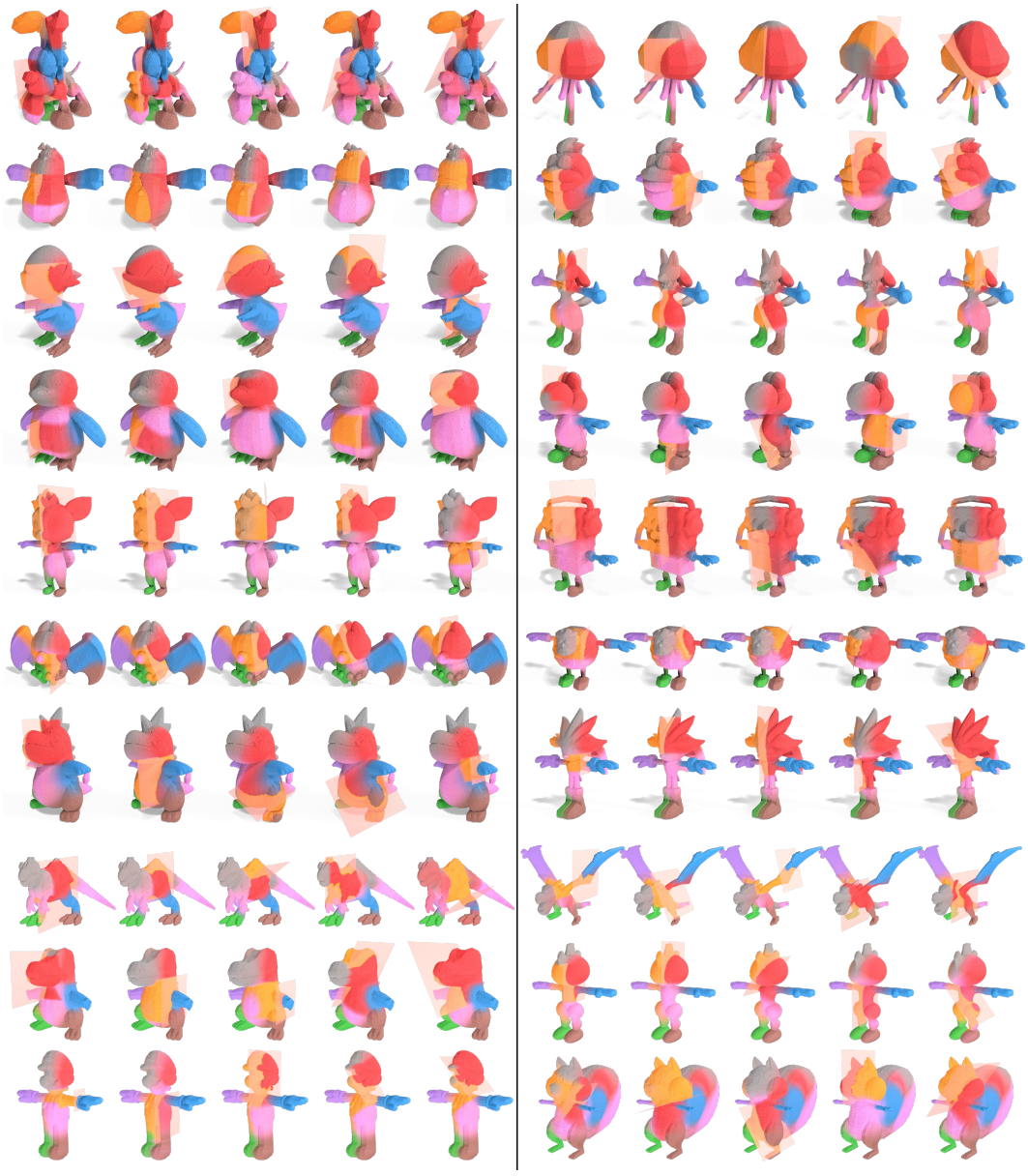

Figure 11: Additional qualitative results on Transformer-based cross-object generalization across diverse shapes and cut configurations.

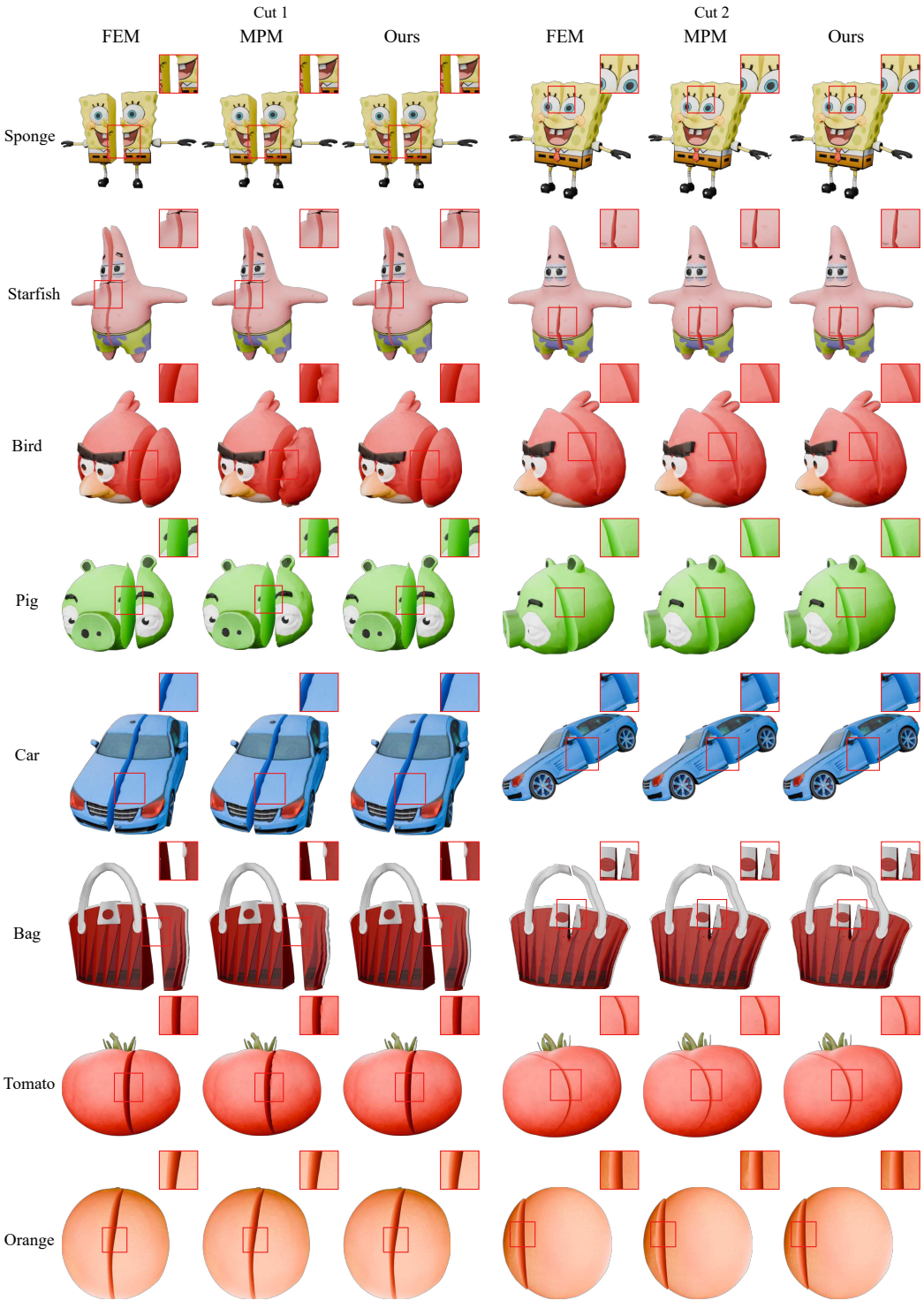

Figure 12: Additional qualitative comparisons with full-space FEM (Heiden et al., 2021) and MPM (Koh et al., 2026) under different cut configurations.

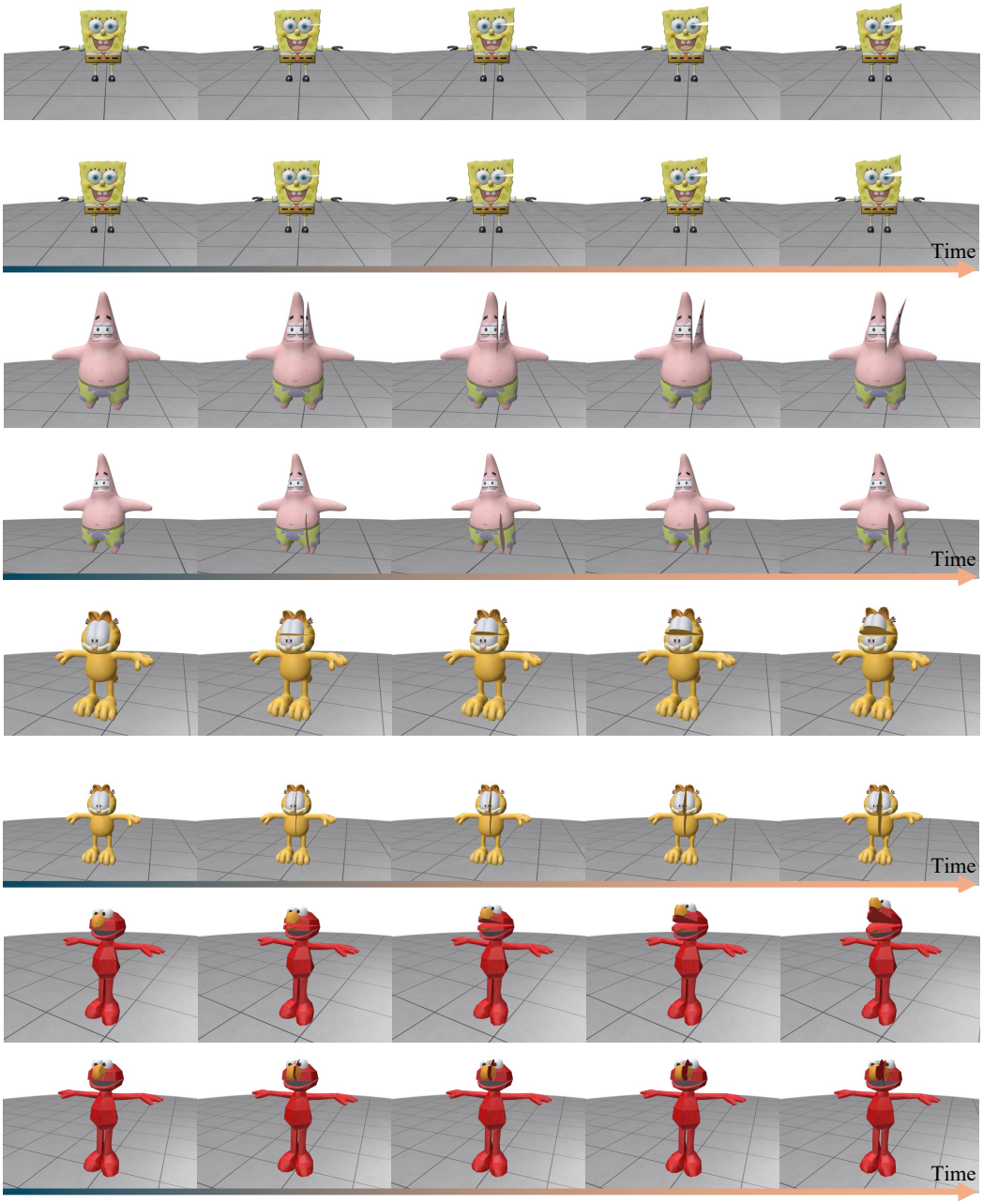

Figure 13: Interactive GUI results for reduced-order cut simulation. **Please refer to the supplementary video for dynamic results.**

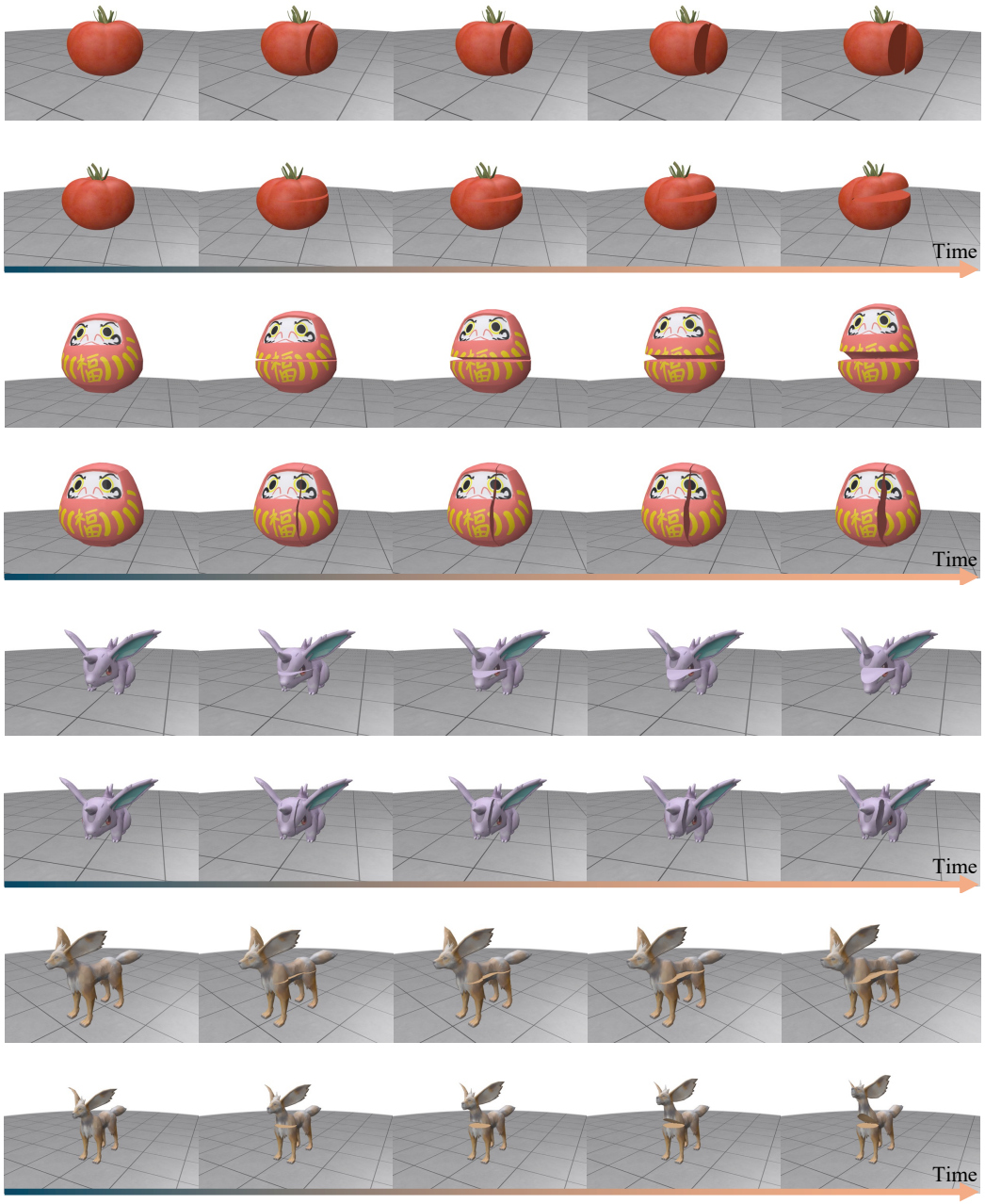

Figure 14: Interactive GUI results for reduced-order cut simulation. **Please refer to the supplementary video for dynamic results.**



Figure 15: Additional results on reduced-order cut simulation coupled with robot interaction. **Please refer to the supplementary video for dynamic results.**

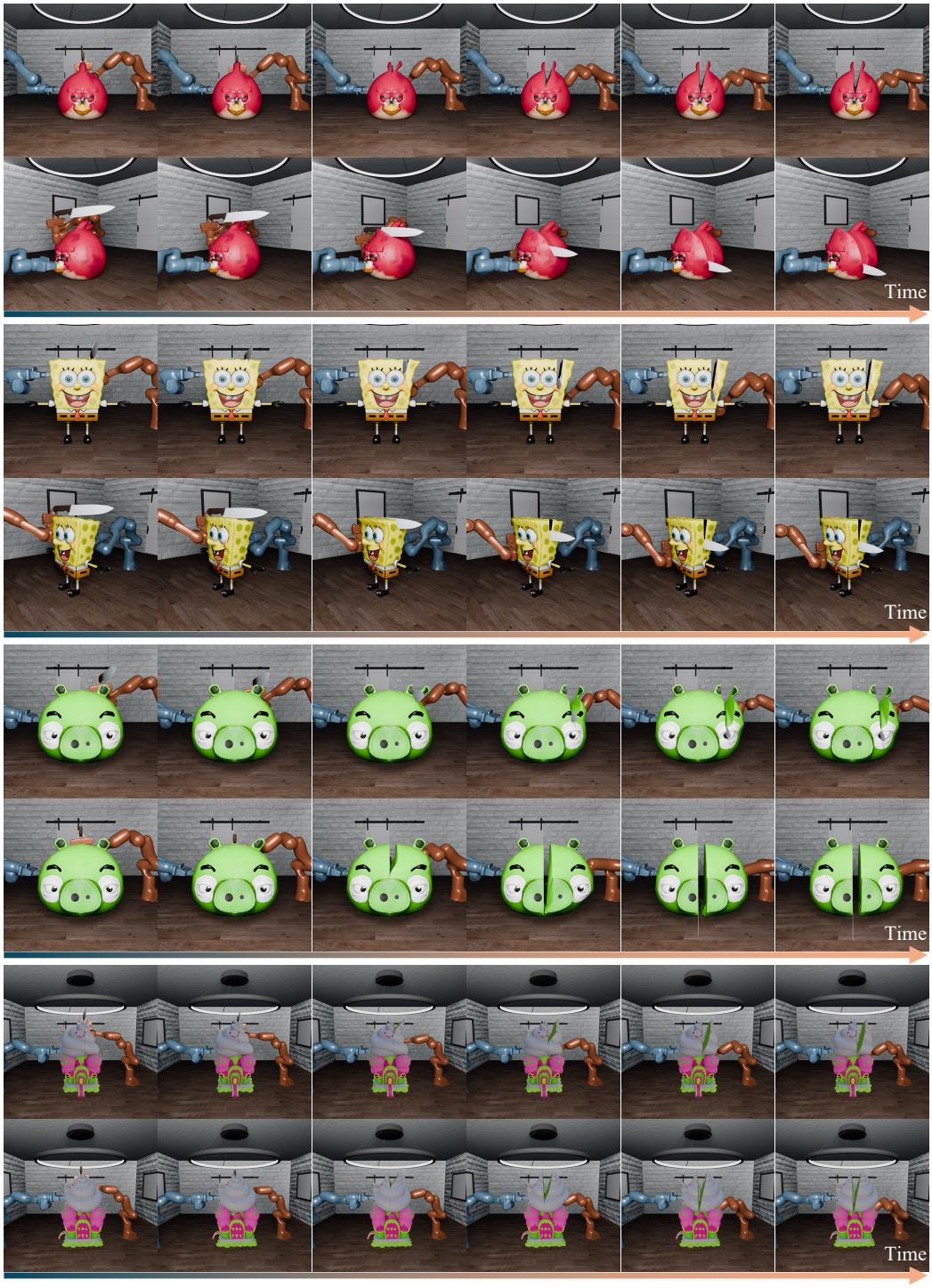

Figure 16: Additional results on reduced-order cut simulation coupled with robot interaction. **Please refer to the supplementary video for dynamic results.**

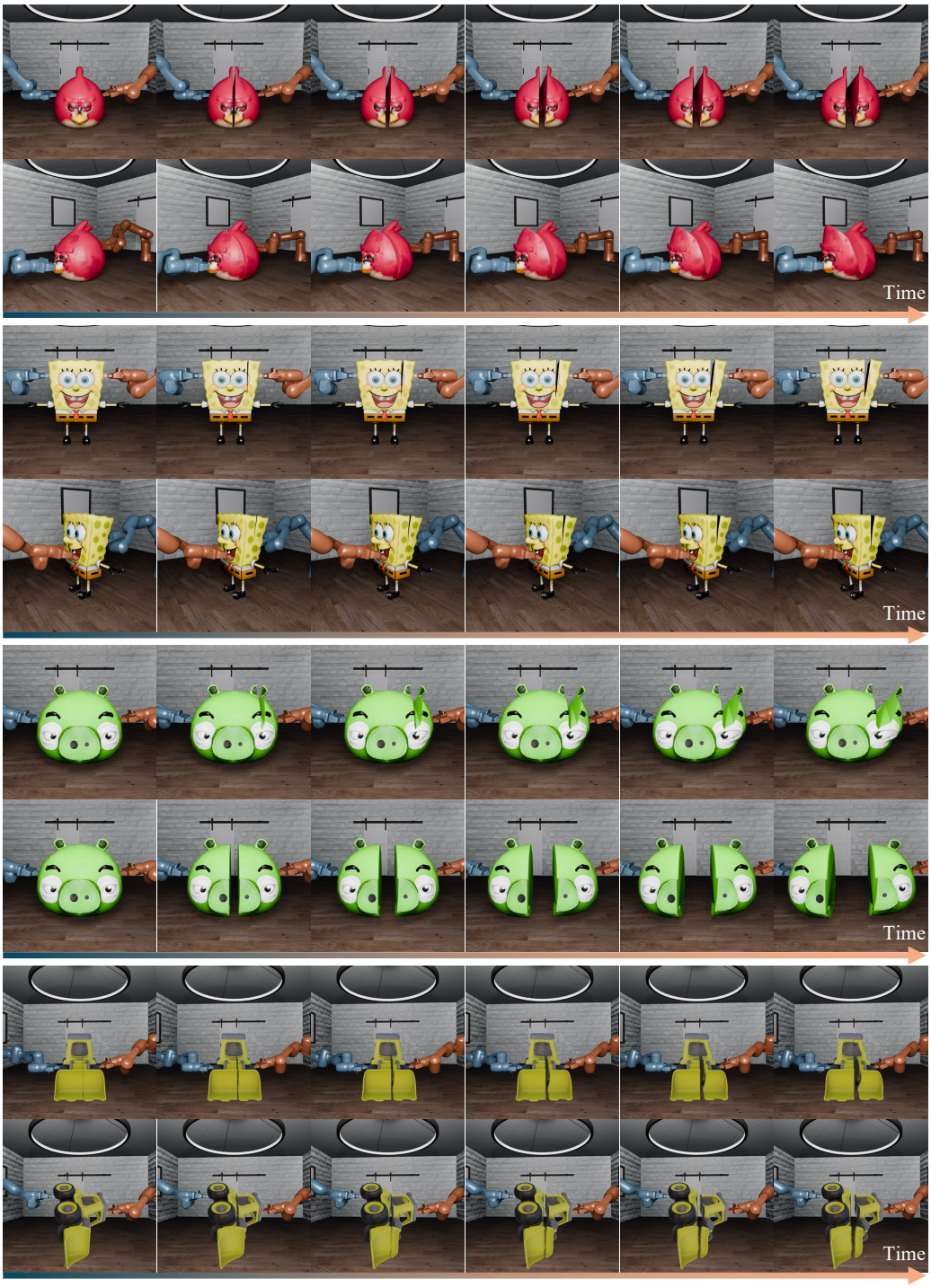

Figure 17: Additional results on reduced-order cut simulation coupled with robot interaction. **Please refer to the supplementary video for dynamic results.**