Precise Gradient Discontinuities in Neural Fields for Subspace Physics

MENGFEI LIU*, University of Toronto, Canada
YUE CHANG*, University of Toronto, Canada
ZHECHENG WANG, University of Toronto, Canada
PETER YICHEN CHEN, University of British Columbia, Canada
EITAN GRINSPUN, University of Toronto, Canada

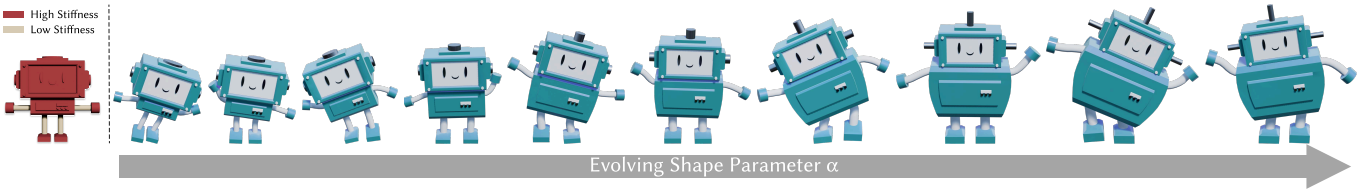


Fig. 1. *Dancing through life.* We introduce a neural field construction capable of representing discontinuities in spatial derivatives. Our approach allows both the domain and its internal interfaces to be parameterized over a shape space. This enables discretization-agnostic reduced-space simulation of heterogeneous materials over parametric shape families. In this animation, a stiff-bodied, soft-limbed robot dances its way from childhood to adulthood, each stage drawn from a parameteric shape family.

Discontinuities in spatial derivatives appear in a wide range of physical systems, from creased thin sheets to materials with sharp stiffness transitions. Accurately modeling these features is essential for simulation but remains challenging for traditional mesh-based methods, which require discontinuity-aligned remeshing—entangling geometry with simulation and hindering generalization across shape families.

Neural fields offer an appealing alternative by encoding basis functions as smooth, continuous functions over space, enabling simulation across varying shapes. However, their smoothness makes them poorly suited for representing gradient discontinuities. Prior work addresses discontinuities in function values, but capturing sharp changes in spatial derivatives while maintaining function continuity has received little attention.

We introduce a neural field construction that captures gradient discontinuities without baking their location into the network weights. By augmenting input coordinates with a smoothly clamped distance function in a lifting framework, we enable encoding of gradient jumps at evolving interfaces.

This design supports discretization-agnostic simulation of parametrized shape families with heterogeneous materials and evolving creases, enabling new reduced-order capabilities such as shape morphing, interactive crease editing, and simulation of soft-rigid hybrid structures. We further demonstrate that our method can be combined with previous lifting techniques to jointly capture both gradient and value discontinuities, supporting simultaneous cuts and creases within a unified model.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

*Indicates joint first authors.

Authors' addresses: Mengfei Liu, University of Toronto, Canada, mengfei.liu@mail.utoronto.ca; Yue Chang, University of Toronto, Canada, changyue.chang@mail.utoronto.ca; Zhecheng Wang, University of Toronto, Canada, zhecheng@cs.toronto.edu; Peter Yichen Chen, University of British Columbia, Canada, peter.chen@ubc.ca; Eitan Grinspun, University of Toronto, Canada, eitan@cs.toronto.edu.

SA Conference Papers '25, December 15–18, 2025, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-2137-3/2025/12
https://doi.org/10.1145/3757377.3763810

Additional Key Words and Phrases: Heterogeneous Elastodynamics, Discontinuity, Crease, Reduced-order modeling, Implicit neural representation

ACM Reference Format:
Mengfei Liu, Yue Chang, Zhecheng Wang, Peter Yichen Chen, and Eitan Grinspun. 2025. Precise Gradient Discontinuities in Neural Fields for Subspace Physics. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3757377.3763810

1 INTRODUCTION

Reduced-order modeling (ROM) [Barbič and James 2005] accelerates physical simulation by approximating high-dimensional dynamics with a low-dimensional modal subspace. However, it remains challenging for ROM to handle discontinuities in spatial derivatives, such as those arising from folds in creased materials or interfaces within heterogeneous solids. These sharp transitions introduce localized stiffness and high-frequency behavior that global basis often fail to capture effectively.

Mesh-based simulators capture such behavior by aligning discretizations with discontinuity interfaces, but this tightly couples geometry with simulation. As the interface evolves due to shape morphing, material variation, or creasing, the mesh must be rebuilt. This invalidates earlier ROM basis and limits the model's ability to generalize.

Several strategies attempt to decouple simulation from mesh topology. Extended finite element methods (XFEM) [Moës et al. 1999] enrich fixed meshes with discontinuous basis functions, allowing interfaces to move without remeshing. However, XFEM still requires a fixed background mesh and typically operates at full resolution, limiting its utility for reduced-order modeling or fast generalization across shape families.

Neural representations [Chang et al. 2023; Modi et al. 2024] offer an alternative: basis functions are modeled as continuous neural fields over spatial coordinates, making them agnostic to mesh discretization. This enables subspace simulation across parametric

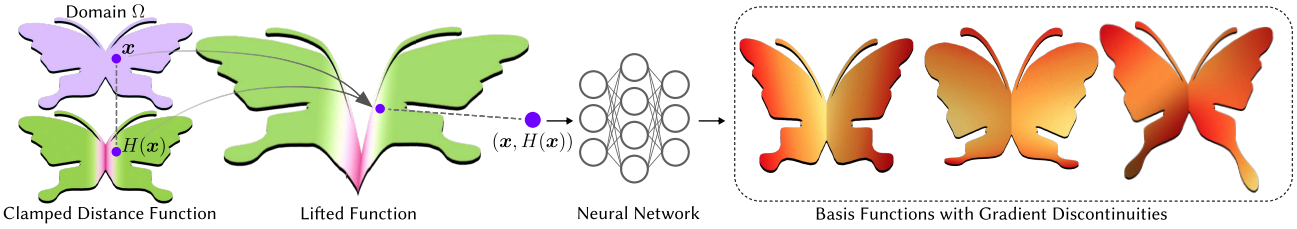


Fig. 2. Our method represents functions with discontinuous gradients by lifting the input domain into a higher-dimensional space. Starting from an input domain with internal interfaces, we construct a smoothly clamped distance field to augment the spatial coordinates. This defines a lifted domain where a neural network is trained to produce smooth basis functions. When restricted back to the original domain, the resulting basis captures sharp gradient transitions at the interface.

shape families [Chang et al. 2024]. However, their smoothness makes them ill-suited for discontinuities. While some approaches [Belhe et al. 2023; Liu et al. 2024] attempt to incorporate known discontinuity locations via feature alignment or differentiable fields, they embed geometric assumptions into network weights, reducing flexibility. As a result, even minor changes to the interface layout typically require retraining.

Recently, Chang et al. [2025] proposed augmenting input coordinates with a generalized winding number field to model discontinuities in function values, effectively lifting the domain. This enables neural fields to handle cutting phenomena in a reusable, generalizable way. We adopt a similar lifting construction, but target a different challenge: modeling *gradient discontinuities*—sharp changes in derivative, with continuity in the function itself. These arise naturally in simulations involving heterogeneous materials and creases and are not addressed by prior discontinuity handling techniques [Belhe et al. 2023; Chang et al. 2025; Liu et al. 2024].

The emergence of gradient discontinuities in solutions to weighted Laplace problems is well-established in the theory of elliptic PDEs: eigenfunctions of weighted Laplace operators can exhibit gradient discontinuities across interfaces where the weight function jumps [Gilbarg and Trudinger 2001]. However, this phenomenon remains underappreciated in the reduced-modeling and neural field literature. We highlight this connection and use it to motivate our construction, which explicitly encodes such discontinuities into the neural field representation.

Contributions. We introduce a neural field construction for representing gradient discontinuities in a generalizable, simulation-ready form. Our key insight is to encode discontinuities via a lifting strategy that augments spatial coordinates with a non-trainable, smoothly clamped distance function. This approach enables sharp changes in function gradients without embedding interface geometry into network weights—allowing the same neural representation to be reused across families of shapes and materials.

We apply this construction to reduced-order simulation, enabling applications not previously demonstrated in this setting. Our method supports differentiable modeling of internal interfaces, allowing inverse design and real-time simulation under dynamic shape and material variation. It also enables interactive editing of creases and hybrid modeling of discontinuities in both function values and derivatives.

In summary, we present:

- a neural field architecture for encoding spatial gradient discontinuities via input lifting;
- a discretization-agnostic basis that generalizes across shape and material spaces;

and the first reduced-order simulation method to support

- combined cuts and folds,
- evolving creases, and
- heterogeneous materials with parameterized interfaces.

2 RELATED WORK

2.1 Discontinuity Representations

In physical simulation, gradient discontinuities are commonly handled by aligning the mesh discretization with underlying interfaces. This is a standard approach in modeling heterogeneous materials—where the mesh conforms to stiffness boundaries [Kim and Eberle 2022]—and in simulating creases, where folding patterns are embedded into the mesh topology [Choi et al. 2004; Narain et al. 2013; Zhu and Filipov 2021]. However, such mesh-dependent methods are typically limited to fixed discontinuity configurations on a single rest shape. When either the rest shape or the location of the discontinuities changes, remeshing becomes necessary. This process alters the structure of the system's Hessian, hampering maintainance of a consistent basis and in turn reduced-order simulation.

Extended finite element methods (XFEM) [Kaufmann et al. 2009; Moës et al. 1999; Ton-That et al. 2024] avoid remeshing by enriching a fixed mesh with basis functions that capture local discontinuities, including in gradients. XFEM is typically implemented at full resolution, making it less suitable for real-time or shape-varying reduced-order simulation. Like XFEM, we also target representation of gradient discontinuities, however, we do not rely on a fixed background mesh. By fully decoupling geometry from simulation degrees of freedom, we generalize to shape families with varying rest domains (see Fig. 15), and support reduced-order modeling.

At the core of our method is the use of neural fields to model gradient discontinuities. Prior neural approaches, such as Belhe et al. [2023], define feature fields over triangle meshes aligned to discontinuity interfaces, with subsequent work introducing differentiability [Liu et al. 2024]. Rather than learning mesh-aligned features, Chang

et al. [2025] proposed lifting input coordinates via a generalized winding number field to model discontinuities for progressive cut simulation. These efforts primarily focus on modeling discontinuities in the function values. In contrast, our work targets a different challenge: modeling functions that remain continuous but exhibit discontinuous gradients—an essential feature for simulating heterogeneous materials and sharp features such as creases.

2.2 Heterogeneous Elastodynamics

Heterogeneous elastodynamics refers to the simulation of elastic materials with spatially varying properties, such as stiffness. A common strategy is to perform full-space finite element simulations, which are often computationally intensive. To improve efficiency, previous work has explored spatial simplification techniques—such as mesh coarsening [Chen et al. 2017] and remeshing [Chen et al. 2015, 2018; Kharevych et al. 2009]—to reduce the number of degrees of freedom. Alternatively, Trusty et al. [2022] introduced a mixed discretization scheme that improves solver convergence without modifying the mesh. While these methods focus on optimizing full-space simulations of a single shape, our approach enables reduced simulation and generalizes across families of shapes.

Another line of work aims to accelerate simulations by reducing the dimensionality of the dynamical system itself. Reduced space methods [Barbič and James 2005; Benchekroun et al. 2023; Pentland and Williams 1989; Trusty et al. 2023] construct a linear subspace to eliminate redundant degrees of freedom while maintaining visually accurate motion. Our method can be seen as a generalization of reduced space simulation that extends across both shape and material spaces. Mukherjee et al. [2016] proposed an approach for updating the basis via incremental linear modal analysis. However, due to the matrix-based formulation, it is unclear how their method handles changes in mesh resolution or vertex count. In contrast, our model is designed to operate across varying discretizations, including different numbers of vertices.

2.3 Neural Physics Simulation

In recent years, neural networks have become powerful tools for accelerating physical simulation. They have been applied across a wide range of domains, including fluid modeling and reconstruction [Chu et al. 2022; Deng et al. 2023; Jain et al. 2024; Kim et al. 2019; Tao et al. 2024; Wang et al. 2024], collision handling [Cai et al. 2022; Romero et al. 2021; Yang et al. 2020], and cloth simulation [Bertiche et al. 2022; Kairanda et al. 2024; Zhang and Li 2024].

Our work is more closely related to approaches that apply neural networks to accelerate deformable simulation in reduced spaces. Holden et al. [2019], for example, learn nonlinear dynamics directly in a reduced coordinate system. In contrast, we preserve the physical formulation and focus on learning spatial variation. Fulton et al. [2019] introduce a learned nonlinear basis to better capture deformation behavior, with subsequent extensions improving expressivity and generalization [Lyu et al. 2024; Sharp et al. 2023; Shen et al. 2021]. While our approach uses a linear basis, it is designed to generalize across shapes and discretizations, going beyond the scope of earlier methods typically restricted to a single shape.

To support discretization-agnostic simulation, neural fields have gained popularity as spatial representations [Chang et al. 2024, 2023; Chen et al. 2023a; Modi et al. 2024]. Their continuous nature, however, makes it difficult to model discontinuities. The challenge of generalizing learned weights across different types of discontinuities remains largely unexplored. Chang et al. [2025] address this by proposing a method to represent a family of cuts using lifted coordinates in neural fields. Like that work, our method also tackles discontinuities within continuous neural fields using input lifting. However, while they focus on function-value discontinuities (e.g., for cut simulations), we address cases where the function is continuous but its gradients are not, such as in simulations involving heterogeneous materials or sharp creases.

3 REPRESENTING GRADIENT DISCONTINUITIES

Our goal is to construct neural fields capable of accurately capturing gradient discontinuities. Let $f : \Omega \to \mathbb{R}$ be a function defined over a domain $\Omega \subset \mathbb{R}^d$, where d is 2 or 3. We consider cases where the normal gradient of f is discontinuous across an internal interface Γ: for any point $\boldsymbol{x}_0 \in \Gamma$,

$$\lim_{\boldsymbol{x}\to\boldsymbol{x}_0^+} \frac{\partial f(\boldsymbol{x})}{\partial \mathbf{n}} \neq \lim_{\boldsymbol{x}\to\boldsymbol{x}_0^-} \frac{\partial f(\boldsymbol{x})}{\partial \mathbf{n}}\,, \quad \boldsymbol{x}_0 \in \Gamma\,, \quad \boldsymbol{x} \in \Omega\,, \tag{1}$$

with $\mathbf{n}$ denoting the normal to the interface.

Modeling the Gradient Discontinuity through Lifting. We adopt a similar lifting approach to Chang et al. [2025], who construct the *nonsmooth* field $f : \Omega \to \mathbb{R}$ by restricting an *everywhere smooth* field $\tilde{f} : \Omega \times \mathbb{R} \to \mathbb{R}$ to the graph of some suitably-chosen "height" function $H : \Omega \to \mathbb{R}$. We depart from the prior work in our choice of height function.

Chang et al. [2025] lift each point $\boldsymbol{x} \in \Omega$ to a point $\mathcal{L}(\boldsymbol{x}) \in \Omega \times \mathbb{R}$ via the graph of H,

$$\mathcal{L}(\boldsymbol{x}) = (\boldsymbol{x}, H(\boldsymbol{x}))\,, \tag{2}$$

then obtain f by restricting the smooth field $\tilde{f}$ back to the original domain through $\mathcal{L}$:

$$f(\boldsymbol{x}) = \tilde{f}(\mathcal{L}(\boldsymbol{x}))\,, \quad \boldsymbol{x} \in \Omega\,. \tag{3}$$

The domain $\Omega \times \mathbb{R}$ of $\tilde{f}$ is defined over the extrusion of the domain Ω along a new "lifting" dimension. $\tilde{f}(\boldsymbol{x})$ is then discretized with a neural field parameterized by weights θ, which we denote by $\tilde{f}_\theta$.

In doing so, Chang et al. [2025] sidestep the challenge of directly learning discontinuities by representing the target function as smooth in the lifted domain, aligning with the inherent smoothness prior of neural representations. Our work departs in the design of the height function $H(\boldsymbol{x})$. We seek a C^0 function, continuous in value but discontinuous in gradient across the interface Γ. Importantly, H is given in closed form and does not involve any learned parameters.

Choice of $H(\boldsymbol{x})$. We introduce a novel lifting strategy that departs from previous work on function-value discontinuities [Chang et al. 2025]. Our goal is to represent a continuous function with discontinuous gradients, a property critical for capturing material interfaces and creases. We define a new lifting function $H(\boldsymbol{x})$ that is given in closed form and requires no learned parameters. This function is

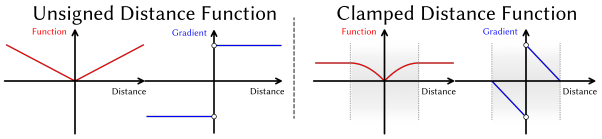


Fig. 3. We visualize the smoothly clamped distance function and its gradient. The function flattens beyond a threshold distance s, while preserving gradient discontinuities at the interface (where the distance is zero).

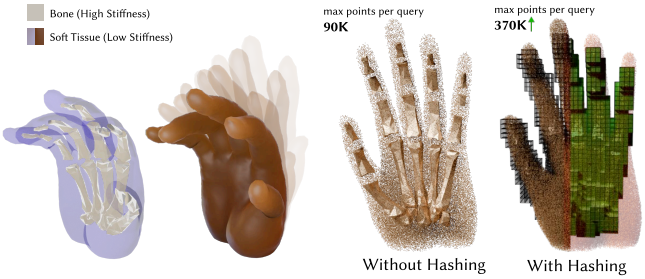


Fig. 4. We test our model on a complex scene of a hand with soft flesh and a stiff skeleton. This example demonstrates both reduced memory usage and speedup enabled by our spatial hash. The clamped distance function localizes queries by ignoring point pairs farther than s, making it well-suited for spatial hashing. The hash structure supports 4.1× more queries due to improved memory efficiency, and achieves a 3.6× speedup when tested with 90k query points (same as without hashing).

explicitly constructed from the interface geometry and designed to produce gradient discontinuities aligned with the interface normal. We begin by computing the unsigned distance from the input coordinates to the interface:

$$D(\boldsymbol{x}) = \min_{\boldsymbol{p}\in\Gamma} \|\boldsymbol{x} - \boldsymbol{p}\|_2,$$

where $\boldsymbol{p} \in \mathbb{R}^d$ is a point on the interface. After discretization, the interface Γ is represented by an explicit mesh $\mathcal{M} = \{V, E\}$. For a given point, we compute the closest distance to the nearest point on this mesh. We denote the distance function to the discrete interface $\mathcal{M}$ as $D_{\mathcal{M}}$.

Unsigned distance fields exhibit gradient discontinuities across the mesh due to their absolute-value-like behavior. While this may seem sufficient at first glance, the formulation has significant limitations. First, computing the unsigned distance field is inherently global: even for query points $\boldsymbol{x}$ located far from the interface, the computation may still involve traversing many primitives resulting in unnecessary overhead. Second, unsigned distance fields are known to produce singularities near the medial axis. Although prior work [Madan and Levin 2022] addresses this by applying a softmin across primitives, the resulting formulation remains global.

Since the key property we seek from the distance field is its absolute-value-like behavior near the interface, where gradient discontinuities occur, we do not need to compute exact distance values far from the interface; it is sufficient for the field to remain smooth in those regions. To this end, we adopt a smoothly clipped distance field that preserves the discontinuity near the interface while localizing computation within a specified threshold. This formulation also helps alleviate singularities caused by the medial axis by nullifying the gradient beyond the threshold region.

As shown in Figure 3, the final height function is constructed by smoothly clamping the distance function:

$$H(\boldsymbol{x}) = \|D_{\mathcal{M}}(\boldsymbol{x})\|_{\mathrm{SC}}$$

where $\|\cdot\|_{\mathrm{SC}}$ is the smooth clamping function [Chen et al. 2023b]:

$$\|D(\boldsymbol{x})\|_{\mathrm{SC}} = \begin{cases} \|D(\boldsymbol{x})\|_2 - \frac{1}{2s}\|D(\boldsymbol{x})\|_2^2, & \text{if } D(\boldsymbol{x}) < s, \\ \frac{1}{2}s, & \text{if } D(\boldsymbol{x}) \geq s. \end{cases} \tag{4}$$

This formulation has the advantage that only distances within a fixed threshold s need to be computed, making it well suited for acceleration structures such as spatial hashing. As shown in Figure 4, it requires 4.1 times less memory during queries and achieves a 3.6 × speedup compared to querying the same number of points without hashing. We computed the gradient norm error under different values of s (Figure 7) and observed no significant variation across settings.

4 ADAPTATION TO REDUCED-SPACE SIMULATION

4.1 Reduced-space Simulation

We consider an elastic body on a reference domain $\Omega \subset \mathbb{R}^d$, with deformation described by a displacement field $\boldsymbol{u}(\boldsymbol{x})$. In reduced-space simulation with skinning eigenmode subspace [Benchekroun et al. 2023; Trusty et al. 2023], this field is approximated using low-dimensional coordinates $\boldsymbol{z} \in \mathbb{R}^k$ and basis functions $\boldsymbol{\phi}_j$:

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{j=1}^{k} \boldsymbol{z}_j \boldsymbol{\phi}_j(\boldsymbol{x}) \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix},$$

where $\boldsymbol{z}_j \in \mathbb{R}^{d\times(d+1)}$ and $\boldsymbol{\phi}_j : \Omega \to \mathbb{R}$. Following recent work on neural reduced models [Chang et al. 2023; Modi et al. 2024], we represent basis functions using a neural field $\boldsymbol{\phi}(\boldsymbol{x}) = f(\boldsymbol{x})$. To support varying geometry or material properties, we condition on some parameter α (i.e., $\boldsymbol{\phi}^\alpha$), which corresponds to some (hand-built or learned) parameterization of the geometry (e.g., shape, material interfaces) or scenario (e.g., material stiffness, boundary conditions).

At each time step t, the reduced coordinate $\boldsymbol{z}^{t+1}$ is updated by minimizing

$$\boldsymbol{z}^{t+1} = \arg\min_{\boldsymbol{z}} \frac{1}{2}\left\|\boldsymbol{z} - 2\boldsymbol{z}^t + \boldsymbol{z}^{t-1}\right\|^2 + h^2 \int_\Omega \Psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z})\right) \, d\boldsymbol{x}, \tag{5}$$

where h is the time step size. The first term models inertia; the second is the elastic energy. We opted to use a St. Venant–Kirchhoff model to evaluate Ψ, although in principle any hyperelastic model could be substituted.

The energy Ψ depends on the spatial gradient $\partial\boldsymbol{\phi}^\alpha/\partial\boldsymbol{x}$, computed via automatic differentiation. The deformation gradient $\mathbf{F}$ is formed as a linear combination of basis gradients and reduced coordinates. We approximate the integral using stochastic cubature with uniform sampling and optimize using gradient descent.

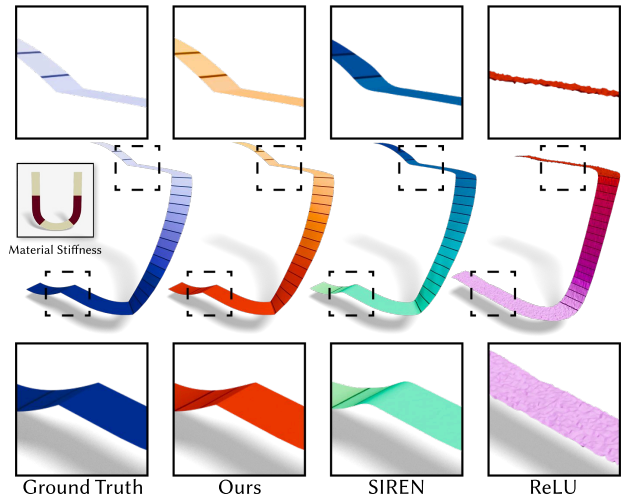


Fig. 5. We compared our method against basis functions produced by other neural field architectures on a heterogeneous 2D "U"-shaped domain. For visualization, the 2D shape is lifted along the Y-axis to represent the scalar basis function, with additional color coding to indicate its value. The SIREN MLP [Sitzmann et al. 2020], used in prior works [Chang et al. 2024; Modi et al. 2024], fails to capture sharp variations at the material interface. A ReLU-based neural field, which permits C^0 continuity, does not converge to the correct solution. In contrast, our method successfully captures the sharp gradient transitions across the interface.

4.2 Training on Varying Shapes and Material Stiffness

As mentioned before, we train a reduced model that adapts to a family of shapes and varying material properties. To capture these variations, we introduce a parameter α that encodes shape and stiffness information.

When α represents different shapes with changing interfaces, the domain Ω^α, the interface Γ^α and its explicit representation $\mathcal{M}^\alpha$ all depend on α. In our implementation, we treat $\mathcal{M}^\alpha$ as a nonlinear, user-defined function that aligns with the discontinuity interface. Accordingly, the height field becomes a function of α:

$$H^\alpha(\boldsymbol{x}) = \|D_{\mathcal{M}^\alpha}(\boldsymbol{x})\|_{\mathrm{SC}}$$

As α controls the shape and material variations, the network weights are also conditioned on α to adapt to the resulting changes in the domain. Thus, the parameter-conditioned neural field is defined as

$$f^\alpha(\boldsymbol{x}) = \tilde{f}^\alpha_\theta(\mathcal{L}^\alpha(\boldsymbol{x})) \quad \text{and} \quad \mathcal{L}^\alpha(\boldsymbol{x}) = (\boldsymbol{x}, H^\alpha(\boldsymbol{x})).$$

We parameterize the basis functions for reduced-space simulation using a neural field, denoted as $\boldsymbol{\phi}^\alpha(\boldsymbol{x}) = f^\alpha(\boldsymbol{x})$. To train the basis, we uniformly sample a parameter α in each epoch, representing a domain Ω^α, and then draw spatial samples $\boldsymbol{x} \in \Omega^\alpha$ uniformly. The neural network weights θ are optimized by minimizing a loss function evaluated on these samples:

$$L = E[\boldsymbol{\phi}^\alpha_\theta].$$

The exact formulation of $E[\boldsymbol{\phi}]$ depends on the application and will be discussed in detail in Section 4.3 and Section 4.4.

4.3 Heterogeneous Elastodynamics

Our method can be applied to compute basis functions for heterogeneous materials in a discretization-agnostic manner.

To construct the basis functions $\boldsymbol{\phi}$ for the skinning eigenmode subspace, we solve a generalized eigenvalue problem associated with the elastic energy Laplacian. In the case of co-rotational elasticity, this corresponds to a Laplacian operator weighted by the spatially varying material stiffness [Benchekroun et al. 2023].

We extend the method of Chang et al. [2024] to heterogeneous materials by introducing a spatially varying weight function $w(\boldsymbol{x})$. Following their variational formulation, the eigenproblem is expressed as minimizing the weighted Dirichlet energy:

$$E_D[\boldsymbol{\phi}_i] = \frac{1}{2}\int_\Omega w(\boldsymbol{x})|\nabla\boldsymbol{\phi}_i|^2 d\Omega, \tag{6}$$

subject to the unit norm constraint $\boldsymbol{\phi}_i \in \mathcal{U}$, where $\mathcal{U} = \{f \in L^2(\Omega) \mid |f|_2 = 1\}$, and the orthogonality condition $\boldsymbol{\phi}_i \in \mathrm{span}\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{i-1}\}^\perp$. The orthogonality condition is enforced via Gram–Schmidt orthogonalization. Given a candidate function $\overline{\phi}_i$, we project out its component in the span of the previous basis $\{\phi_1, \ldots, \phi_{i-1}\}$, and then normalize the result to obtain ϕ_i. Since their method handles only homogeneous settings, it fails to capture sharp material transitions due to the smoothness of the neural field. Our formulation overcomes this limitation.

Gradient discontinuities in heterogeneous materials. In heterogeneous materials, the varying stiffness induces gradient discontinuities across interfaces Γ, resulting in a *jump condition* on the normal derivative [Gilbarg and Trudinger 2001],

$$w_1\frac{\partial u}{\partial \mathbf{n}} = w_2\frac{\partial u}{\partial \mathbf{n}}, \tag{7}$$

where $\frac{\partial u}{\partial \mathbf{n}}$ denotes the normal derivative, and $\mathbf{n}$ is the unit normal to Γ. Refer to the supplemental material for a derivation.

While this behavior is familiar in mesh-based PDE contexts, it remains underutilized in neural simulation frameworks. Our method is the first to represent these discontinuities in a generalizable neural field architecture, enabling basis functions that remain valid across parametric shape and material spaces.

As illustrated in Figure 5 and Figure 9, directly optimizing Equation 6 using a standard MLP leads to overly smoothed results that fail to capture the sharp discontinuities at material interfaces. In contrast, our formulation preserves these transitions accurately. Figure 8 visualizes the gradient and its error. Our method achieves lower gradient error than SIREN-based neural fields, particularly at interfaces where discontinuities occur. Unlike traditional mesh-based FEM approaches, where basis computation is tightly coupled to a specific discretization, our method is discretization-agnostic. This enables the computation of basis functions across a family of shapes (Figure 15), offering improved generalization.

4.4 Creasing

Our method can also compute basis functions for subspace simulations of creasing, where the displacement field has discontinuous derivatives due to the crease.

Unlike the previous section, which used a weighted Dirichlet energy, we compute the basis here by minimizing the Hessian energy [Stein et al. 2018], subject to unit-norm and orthogonality constraints:

$$E_H[\boldsymbol{\phi}_i] = \int_\Omega \|\nabla^2 \boldsymbol{\phi}_i\|_F^2 d\Omega \,, \tag{8}$$

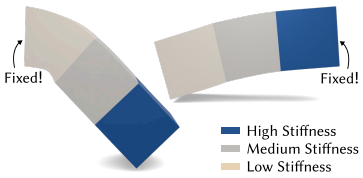
where $\|\cdot\|_F$ is the Frobenius norm and $\nabla^2\boldsymbol{\phi} \in \mathbb{R}^{2\times 2}$ denotes the second-order partial derivatives. The unit-norm and orthogonality constraints are implemented in the same fashion as in the previous section. Note that this formulation does not impose explicit boundary conditions at the crease. The gradient change arises naturally from the neural field, which allows for discontinuities at the crease. Figure 16 shows the mode produced by our formulation. In contrast, traditional neural fields fail to capture such discontinuities due to their inherently smooth nature.

5 RESULTS

All experiments are trained and tested on an NVIDIA RTX 4090 GPU. Our method is implemented in PyTorch and optimized using the Adam optimizer. The network architecture is a 5-layer, 128-channel SIREN MLP with positional encoding up to a maximum frequency of 2^5. The clamping factor is set to $s = 1/8$ for all 2D examples, $s = 1/25$ for the hand-with-skeleton example, and $s = 1/16$ for all other 3D cases.

5.1 Heterogenous Materials

Capturing Gradient Discontinuities. We demonstrate the ability of our basis to capture gradient discontinuities for a U shape (Figure 5) and a 2D snail (Figure 9). Our method is the only one to capture sharp gradient discontinuities at material stiffness changes, with results comparable to FEM, while being discretization agnostic—a point elaborated in the next paragraph. In contrast, a traditional neural field implemented using SIREN fails to capture the gradient discontinuity, exhibiting smoothing artifacts at the interface. We also implemented a neural field with a C^0 continuous activation function (ReLU). The hope was that such an activation function would allow the neural field to represent a C^0 continuous basis function. However, since the gradient discontinuity of ReLU cannot be aligned with the gradient discontinuity in the object, it fails to capture the gradient discontinuity at the stiffness interface.



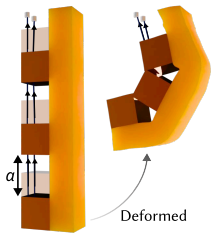
We can further expand our model to capture complex gradient discontinuities, as shown in Figure 4, which depicts a hand with soft flesh and a stiff skeleton simulated to curl up naturally. Our method is also capable of capturing heterogeneous materials with multiple stiffness regions. As shown in the inset illustration, two cantilever simulations of the same cuboid bar with three distinct stiffness regions exhibit markedly different behaviour when pinned at the low-stiffness end compared to the high-stiffness end. In addition, our approach accommodates materials with varying elastic properties. Figure 10 shows a block with stiff ends and a soft center stretched under different Poisson ratios: as the Poisson's ratio ν increases, the deformation becomes more volume-preserving, with the block approaching incompressibility at $\nu = 0.49$.

Discretization-Agnostic Representation. Figure 15 demonstrates the discretization-agnostic nature of our method. We train a neural field over a shape space of 2D robots with heterogeneous material distributions, where the arms and legs are softer than the torso. While traditional FEM can capture gradient discontinuities, it requires explicit meshing of each domain. For shape families, maintaining consistent discretizations across all instances is often impractical or impossible, making it difficult to construct a shared basis across the space. As shown in the second row of Figure 15, the number of vertices and faces varies significantly across shapes, preventing the reuse of a common reduced model.

In contrast, our method is independent of mesh discretization. A single trained neural field generalizes across the entire shape space, enabling basis inference on arbitrary geometries without remeshing, while maintaining accuracy comparable to FEM. This flexibility supports dynamic shape morphing during simulation without recomputation, as shown in the supplementary video.

In Figure 1, we show a continuously evolving family of robots performing a dancing motion. These robots are sampled from a 3D shape space consisting of soft limbs and a stiff body. We train a single model to represent basis functions across the entire space. As a result, we can simulate the dancing motion with smoothly changing robot shapes, without remeshing or restarting the simulation. Figure 6 shows another example: a family of cartoon animals with rigid skulls and soft tissue, parameterized by a variable α. The skull is 100 times stiffer than the surrounding head. As the animal nods and shakes its head, it morphs continuously from a fox to a bear. The neural basis adapts seamlessly: low-stiffness regions like the ears and nose exhibit large deformations, while the stiff skull remains relatively rigid.

Differentiable Subspace Physics Simulation. Our representation is differentiable with respect to the shape parameter α, making it suitable for shape optimization tasks. Figure 14 and the inset illustrates an example involving objects resembling robotic fingers used for grasping. Each finger consists of a single long, soft cuboid bar, along with three smaller, stiffer blocks attached along one side. The grasping motion is controlled by adjusting the gap between the tops of the stiff blocks, as shown in the inset. Starting from an initial configuration, a more strongly bent finger shape is obtained by optimizing the shape code α.



We perform forward simulation for 200 time steps, applying spring forces between the endpoints of each stiff segment to mimic strings stretching across different parts of the robot finger. We compute the mean final displacement along the y-axis, denoted as $\boldsymbol{u}_\alpha$, and determine the optimal shape parameter α^* by maximizing the

	# basis	# simulated vertices	# vertices (interface)	# elements (interface)	Basis building time (ms)	Time per step (ms)
Snail (Figure 9)	21	5.9k	21	18	79.06	23.52
2D Robot, $\alpha = 0$ (Figure 15)	7	19.0k	98	93	37.42	19.71
2D Robot, $\alpha = 0.33$ (Figure 15)	7	25.5k	98	93	41.71	23.03
2D Robot, $\alpha = 0.67$ (Figure 15)	7	48.7k	98	93	68.60	37.91
2D Robot, $\alpha = 1$ (Figure 15)	7	102.4k	98	93	144.60	76.51
Bar (Figure 11)	4	40.0k	44	40	86.28	2.78
Robot finger (Figure 14)	4	8.6k	105	160	24.81	4.19
Shoe (Figure 12)	4	30.0k	1924	2598	226.52	2.17
3D Robot (Figure 1)	11	23.7k	174	326	370.94	5.16
Crease (Figure 16)	4	150.0k	20	19	87.42	5.34
Crease and Cut (Figure 18)	2	50.0k	104	101	7.75	1.87

Table 1. We collected simulation timing data for the examples presented. Our method exhibits high performance, frequently achieving near real-time simulation speeds, even for cases involving a large number of simulated vertices. The reported basis building time reflects the duration required to perform network inference and construct the basis. While this step may introduce some overhead, it is only performed at the beginning of the simulation or when the design parameter α changes, and is therefore invoked infrequently.

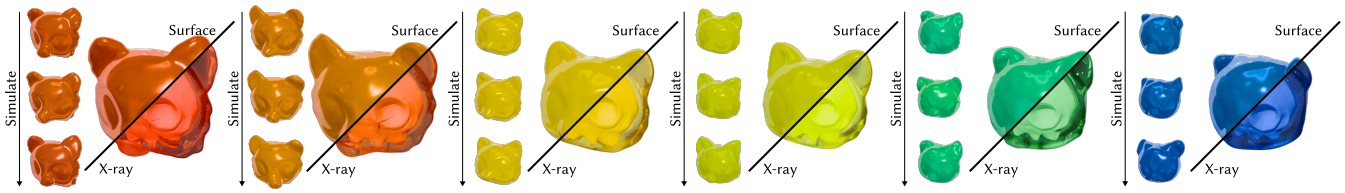


Fig. 6. Simulation of a parameterized shape family morphing from a fox to a bear, performing nodding and shaking motions. Each shape includes a skull that is 100× stiffer than the surrounding soft tissue, shown in the X-ray view. Our neural basis adapts across the family: soft regions (ears, nose) undergo large deformations, while the stiff skull remains rigid.

displacement norm:

$$\alpha^* = \arg\max_{\alpha} \|\boldsymbol{u}_\alpha\|^2.$$

As shown in Figure 14, the optimized shape (right) bends significantly more than the initial guess (left), demonstrating the effectiveness of our method for shape optimization in reduced space.

We present another example where α controls the stiffness parameter (Figure 13). We perform forward simulation for 120 time steps and optimize α by minimizing the L_2 difference between the final and target displacement fields. After optimization, the resulting deformation more closely matches the target.

Generalization. Our framework generalizes not only across shape families but also over spatially varying material layouts, including parametric changes in stiffness. In Figure 12, we vary the stiffness ratio between the upper part of the shoe (foot-facing) and the sole (ground-facing), distinguished visually by color. A material parameter α controls this ratio: larger α values increase the stiffness of the upper relative to the sole. We compress the shoe from both above and below and observe that as α increases, the sole becomes increasingly deformable, leading to greater compression in the lower region.

To further demonstrate the ability of our method to generalize across unseen material configurations, we evaluate it on a second example involving a bar with a cuboid rest state (Figure 11). Here, the basis for simulation was trained using only five different stiffness distributions, each parameterized by a material variable α. We then performed twist simulations on bars with both seen and unseen α values within the training range. Our method not only delivers excellent simulation performance on examples from the training set, but also generalizes effectively to bars with stiffness distributions not encountered during training. It successfully creates sharp transitions at the boundaries between different materials while maintaining the structural integrity of the stiffer middle section.

5.2 Creasing

Capturing Gradient Discontinuities. Our method captures the gradient discontinuity introduced by the crease. We train a 2D neural field using the loss defined in Equation 8 and visualize the resulting basis functions by lifting the scalar-valued output along the y-axis. As shown in Figure 16, our approach (first row) successfully recovers the sharp edge along the crease, whereas a standard SIREN MLP (second row) fails to produce crease-aware basis functions due to its inherent continuity. Notably, the loss function in this example contains no explicit information about the discontinuity, as all spatial samples share the same stiffness. Thus, the gradient discontinuity is captured entirely by the network architecture, without relying on additional cues from the loss.

Interactive Designs of Crease Shapes. Since our boundary representation is explicit, the crease shape can be interactively edited during simulation. Using the basis functions from the previous example, we perform reduced simulations where the crease geometry is modified in real time. This allows the simulation to adapt immediately to user edits without requiring a restart, offering flexibility in editing the crease shape during runtime.

Modeling both Discontinuities in Function and Gradient. Since both our method and [Chang et al. 2025] adopt a lifting-based approach, we extend our framework to capture both function and gradient discontinuities, enabling simultaneous modeling of cuts and creases within a single model. This is achieved by incorporating their generalized winding number field as an additional dimension in our lifting function.

Specifically, we augment Equation 2 by adding an extra dimension for a second lifting function that captures function discontinuities, defined by the generalized winding number field from [Chang et al. 2025]:

$$\mathcal{L}(\boldsymbol{x}) = (\boldsymbol{x}, H_d(\boldsymbol{x}), H_{gwn}(\boldsymbol{x})),$$

where $H_d(\boldsymbol{x})$ is the smoothly clamped distance function from our method, and $H_{gwn}(\boldsymbol{x})$ is the generalized winding number field.

As shown in Figure 18, the paper deforms smoothly at the beginning of the simulation. As the crease develops, sharp turns emerge along it and, as the shape of the deer is progressively cut, the deformation becomes increasingly concentrated near the crease. This extension allows us to construct basis functions that are discontinuous in value, evident in the cuts of the deformed shape, while also exhibiting gradient discontinuities, such as those along the creases. This example demonstrates the flexibility and extensibility of our framework.

6 DISCUSSIONS AND FUTURE WORK

In this work, we proposed a novel neural field representation for capturing spatial gradient discontinuities and adapted it to reduced-space simulation of heterogeneous materials and creases. This enables several new applications, including shape optimization for reduced-order models of heterogeneous materials and simulations where cuts and creases evolve at runtime.

As with all methods based on neural networks, our approach has limitations in generalization when the test data distribution deviates significantly from the training set. In particular, when the test shape differs substantially from those seen during training, our method can still capture the visual appearance of the displacement field, including gradient discontinuities, but the resulting basis functions may be less physically meaningful. As shown in Figure 19, when trained on a single crease shape and then simulated on a crease far from the training distribution, the model still captures the gradient discontinuity at the new crease location, but the paper no longer bends naturally along the crease. In addition, for the twisting cuboid shown in Figure 11, our model can accurately handle unseen configurations within the range of α values used during training, but it fails to generalize once α moves beyond this range, resulting in noticeably less sharp boundaries. One possible way to alleviate this issue is to include more diverse training data by sampling additional values of α. To demonstrate this, in Figure 19 right, when trained on both crease shapes, the model successfully reproduces natural bending behavior in both cases.

A key property of our method is that the interface is explicitly represented and defined by the user. An interesting direction for future work is to automate this process or make it physics-driven, eliminating the need for manual alignment. This could potentially be achieved using differentiable techniques similar to those proposed in [Liu et al. 2024]. In addition, it would be valuable to investigate how the method extends to more complex material settings, such as tiled structures and porous media, where explicit interface control could aid design and optimization.

ACKNOWLEDGMENTS

We would like to thank our lab system administrator, John Hancock, and our financial officer, Xuan Dam, for their invaluable administrative support in making this research possible. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) grant RGPIN-2021-03733.

REFERENCES

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

Yash Belhe, Michaël Gharbi, Matthew Fisher, Iliyan Georgiev, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Discontinuity-aware 2D neural fields. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 41, 6 (2023). https://doi.org/10.1145/3550454.3555484

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

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2022. Neural Cloth Simulation. *ACM Trans. Graph.* 41, 6, Article 220 (nov 2022), 14 pages. https://doi.org/10.1145/3550454.3555491

Xinhao Cai, Eulalie Coevoet, Alec Jacobson, and Paul Kry. 2022. Active Learning Neural C-space Signed Distance Fields for Reduced Deformable Self-Collision. In *Proceedings of Graphics Interface 2022* (Montréal, Quebec) *(GI 2022)*. Canadian Information Processing Society, 92 – 100. https://doi.org/10.20380/GI2022.11

Yue Chang, Otman Benchekroun, Maurizio M. Chiaramonte, Peter Yichen Chen, and Eitan Grinspun. 2024. Shape Space Spectra. arXiv:2408.10099 [cs.GR] https://arxiv.org/abs/2408.10099

Yue Chang, Peter Yichen Chen, Zhecheng Wang, Maurizio M. Chiaramonte, Kevin Carlberg, and Eitan Grinspun. 2023. LiCROM: Linear-Subspace Continuous Reduced Order Modeling with Neural Fields. In *SIGGRAPH Asia 2023 Conference Papers* (, Sydney, NSW, Australia,) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 111, 12 pages. https://doi.org/10.1145/3610548.3618158

Yue Chang, Mengfei Liu, Zhecheng Wang, Peter Yichen Chen, and Eitan Grinspun. 2025. Lifting the Winding Number: Precise Representation of Complex Cuts in Subspace Physics Simulations. arXiv:2502.00626 [cs.GR] https://arxiv.org/abs/2502.00626

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-aware numerical coarsening for fabrication design. *ACM Trans. Graph.* 36, 4, Article 84 (July 2017), 15 pages. https://doi.org/10.1145/3072959.3073669

Desai Chen, David I. W. Levin, Shinjiro Sueda, and Wojciech Matusik. 2015. Data-driven finite elements for geometry and material design. *ACM Trans. Graph.* 34, 4, Article 74 (July 2015), 10 pages. https://doi.org/10.1145/2766889

Honglin Chen, Changxi Zheng, and Kevin Wampler. 2023b. Local Deformation for Interactive Shape Editing. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages. https://doi.org/10.1145/3588432.3591501

Jiong Chen, Hujun Bao, Tianyu Wang, Mathieu Desbrun, and Jin Huang. 2018. Numerical coarsening using discontinuous shape functions. *ACM Trans. Graph.* 37, 4, Article 120 (July 2018), 12 pages. https://doi.org/10.1145/3197517.3201386

Peter Yichen Chen, Jinxu Xiang, Dong Heon Cho, Yue Chang, G A Pershing, Henrique Teles Maia, Maurizio M Chiaramonte, Kevin Thomas Carlberg, and Eitan Grinspun. 2023a. CROM: Continuous Reduced-Order Modeling of PDEs Using Implicit Neural Representations. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=FUORz1tG8Og

Min-Hyung Choi, Min Hong, and Samuel Welch. 2004. Modeling and simulation of sharp creases. In *ACM SIGGRAPH 2004 Sketches* (Los Angeles, California) *(SIGGRAPH '04)*. Association for Computing Machinery, New York, NY, USA, 95. https://doi.org/10.1145/1186223.1186342

Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. 2022. Physics Informed Neural Fields for Smoke Reconstruction with Sparse Data. *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 41, 4 (aug 2022), 119:1–119:15.

Yitong Deng, Hong-Xing Yu, Diyang Zhang, Jiajun Wu, and Bo Zhu. 2023. Fluid Simulation on Neural Flow Maps. *ACM Trans. Graph.* 42, 6 (2023).

Lawson Fulton, Vismay Modi, David Duvenaud, David I. W. Levin, and Alec Jacobson. 2019. Latent-space Dynamics for Reduced Deformable Simulation. *Computer Graphics Forum* (2019).

David Gilbarg and Neil S. Trudinger. 2001. *Elliptic Partial Differential Equations of Second Order* (2nd ed.). Springer. https://doi.org/10.1007/978-3-642-61798-0

Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. 2019. Subspace neural physics: fast data-driven interactive simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '19)*. Association for Computing Machinery, New York, NY, USA, Article 6, 12 pages. https://doi.org/10.1145/3309486.3340245

Pranav Jain, Ziyin Qu, Peter Yichen Chen, and Oded Stein. 2024. Neural Monte Carlo Fluid Simulation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 9, 11 pages. https://doi.org/10.1145/3641519.3657438

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

Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross. 2009. Enrichment textures for detailed cutting of shells. *ACM Trans. Graph.* 28, 3, Article 50 (jul 2009), 10 pages. https://doi.org/10.1145/1531326.1531356

Lily Kharevych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. 2009. Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. Graph.* 28, 3, Article 51 (July 2009), 8 pages. https://doi.org/10.1145/1531326.1531357

Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. 2019. Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 59–70.

Theodore Kim and David Eberle. 2022. Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses* (Vancouver, British Columbia, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 259 pages. https://doi.org/10.1145/3532720.3535628

Chenxi Liu, Siqi Wang, Matthew Fisher, Deepali Aneja, and Alec Jacobson. 2024. 2D Neural Fields with Learned Discontinuities. In *Computer Graphics Forum*. Wiley Online Library, e70023.

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

Abhishek Madan and David I.W. Levin. 2022. Fast Evaluation of Smooth Distance Constraints on Co-Dimensional Geometry. *ACM Trans. Graph.* 41, 4 (2022).

Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David I. W. Levin. 2024. Simplicits: Mesh-Free, Geometry-Agnostic Elastic Simulation. *ACM Trans. Graph.* 43, 4, Article 117 (jul 2024), 11 pages. https://doi.org/10.1145/3658184

Nicolas Moës, John Dolbow, and Ted Belytschko. 1999. A finite element method for crack growth without remeshing. *Internat. J. Numer. Methods Engrg.* 46, 1 (1999), 131–150.

R. Mukherjee, X. Wu, and H. Wang. 2016. Incremental Deformation Subspace Reconstruction. *Comput. Graph. Forum* 35, 7 (Oct. 2016), 169–178.

Rahul Narain, Tobias Pfaff, and James F. O'Brien. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph.* 32, 4, Article 51 (July 2013), 8 pages. https://doi.org/10.1145/2461912.2462010

A. Pentland and J. Williams. 1989. Good vibrations: modal dynamics for graphics and animation. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '89)*. Association for Computing Machinery, New York, NY, USA, 215–222. https://doi.org/10.1145/74333.74355

Cristian Romero, Dan Casas, Jesús Pérez, and Miguel Otaduy. 2021. Learning Contact Corrections for Handle-Based Subspace Dynamics. *ACM Transactions on Graphics (TOG)* 40, 4, Article 131 (jul 2021), 12 pages. https://doi.org/10.1145/3450626.3459875

Nicholas Sharp, Cristian Romero, Alec Jacobson, Etienne Vouga, Paul Kry, David I.W. Levin, and Justin Solomon. 2023. Data-Free Learning of Reduced-Order Kinematics. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 40, 9 pages. https://doi.org/10.1145/3588432.3591521

Siyuan Shen, Yang Yin, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. 2021. High-order differentiable autoencoder for nonlinear model reduction. *arXiv preprint arXiv:2102.11026* (2021).

Jonathan Richard Shewchuk. 2005. Triangle: A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator. https://www.cs.cmu.edu/~quake/triangle.html.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, Vol. 33. 7462–7473.

Oded Stein, Eitan Grinspun, Max Wardetzky, and Alec Jacobson. 2018. Natural Boundary Conditions for Smoothing in Geometry Processing. *ACM Trans. Graph.* 37, 2, Article 23 (May 2018), 13 pages. https://doi.org/10.1145/3186564

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

Quoc-Minh Ton-That, Paul G. Kry, and Sheldon Andrews. 2024. Generalized eXtended Finite Element Method for Deformable Cutting via Boolean Operations. *Computer Graphics Forum* 43, 8 (2024), 13 pages.

Ty Trusty, Otman Benchekroun, Eitan Grinspun, Danny M. Kaufman, and David I.W. Levin. 2023. Subspace Mixed Finite Elements for Real-Time Heterogeneous Elastodynamics. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 112, 10 pages. https://doi.org/10.1145/3610548.3618220

Ty Trusty, Danny Kaufman, and David I.W. Levin. 2022. Mixed Variational Finite Elements for Implicit Simulation of Deformables. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 40, 8 pages. https://doi.org/10.1145/3550469.3555418

Yiming Wang, Siyu Tang, and Mengyu Chu. 2024. Physics-Informed Learning of Characteristic Trajectories for Smoke Reconstruction. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Article 53, 11 pages. https://doi.org/10.1145/3641519.3657483

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

Meng Zhang and Jun Li. 2024. Neural Garment Dynamic Super-Resolution. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 122, 11 pages. https://doi.org/10.1145/3680528.3687610

Yi Zhu and Evgueni T. Filipov. 2021. Rapid multi-physics simulation for electro-thermal origami systems. *International Journal of Mechanical Sciences* 202-203 (2021), 106537. https://doi.org/10.1016/j.ijmecsci.2021.106537

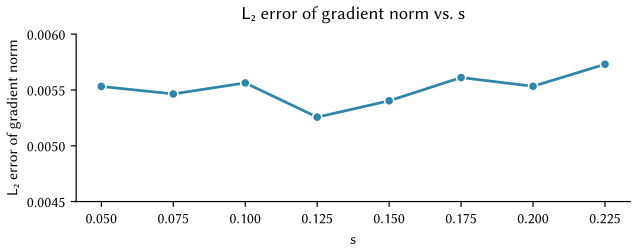


Fig. 7. We compared the L_2 error of the gradient norm for different values of s in Equation 4 and found no clear dependence on s. The ground truth was computed using FEM on the shape shown in Figure 5.

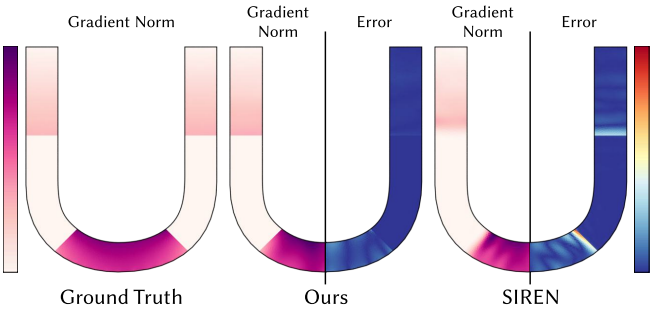


Fig. 8. We visualized both the gradient norm and its error. Our method yields smaller gradient errors than SIREN-based neural fields, particularly at interfaces where discontinuities occur.

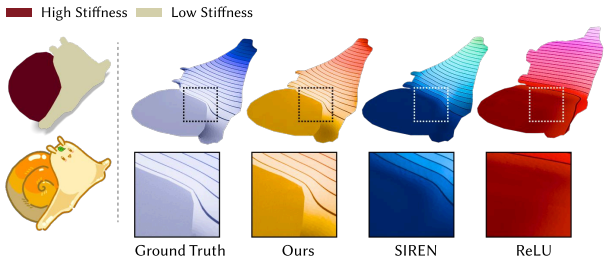


Fig. 9. We did another comparison on a snail shape where the shell is 100× stiffer than the body. SIREN and ReLU-based neural fields fail to capture the sharp gradient changes, while our method successfully recovers the gradient discontinuities.

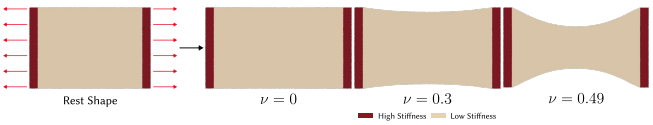


Fig. 10. Our method accommodates materials with varying properties. Here, a block with stiff ends and a soft center is stretched under different Poisson's ratios. As the Poisson's ratio ν increases, the deformation becomes more volume-preserving, with the block approaching incompressibility at $\nu = 0.49$.

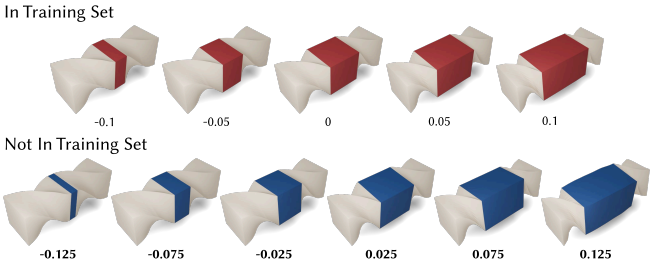


Fig. 11. Our method generalizes to unseen material configurations. In this cuboid bar example, the model was trained on only five distinct material layouts, each parameterized by α. Despite this limited supervision, the simulation accurately handles both seen configurations and unseen configurations with α values within the training range, producing sharp transitions and stiffness boundaries. However, when evaluated on α values outside the training range, the model exhibits generalization failure, with the boundaries appearing noticeably less sharp.

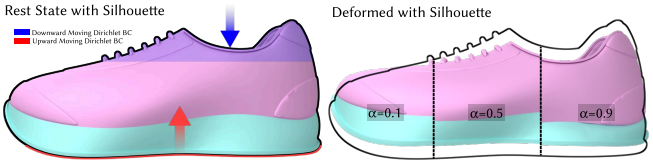


Fig. 12. We assign different stiffness values to the upper (pink) and sole (blue) regions of the shoe, controlled by a parameter α. Higher α values increase the stiffness of the upper relative to the sole. When compressed from both above and below, the sole exhibits greater deformation as α increases.

Fig. 13. We set the parameter α to control the stiffness at the bottom of the bridge. Using differentiable optimization, we optimize α to match the target displacement.

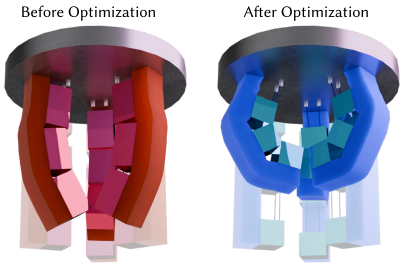


Fig. 14. Our model is differentiable with respect to the shape parameter α, enabling shape optimization in reduced space. We optimize the spacing between bars in the robot fingers. The optimized shape (right) exhibits greater deformation compared to the initial guess (left).

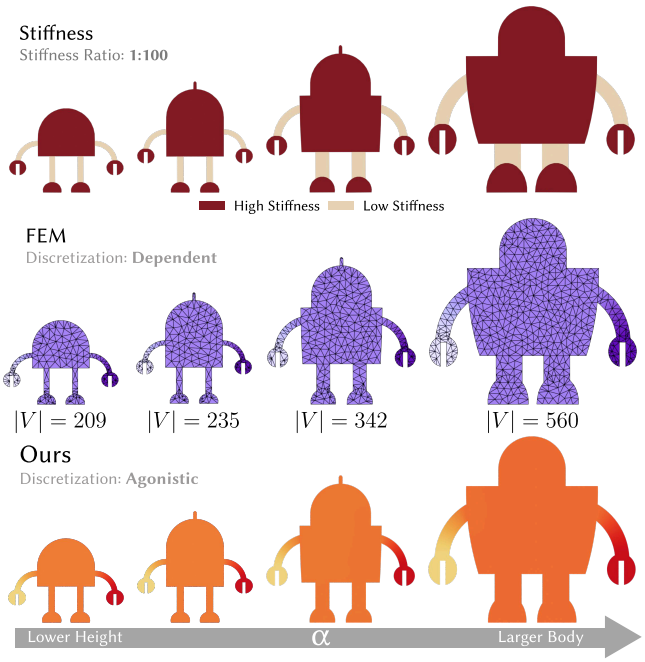


Fig. 15. Our model generalizes across heterogeneous domains, which is challenging for traditional methods based on eigenanalysis of discrete operators. We construct a robot shape space with varying material stiffness (first row) and apply constrained Delaunay triangulation [Shewchuk 2005], resulting in meshes with 209–560 vertices. This variation hinders mesh-based methods from representing a consistent basis (second row), while our model captures basis functions across all shapes with a single network (third row). Scalar basis functions are visualized using color in the second and third rows.

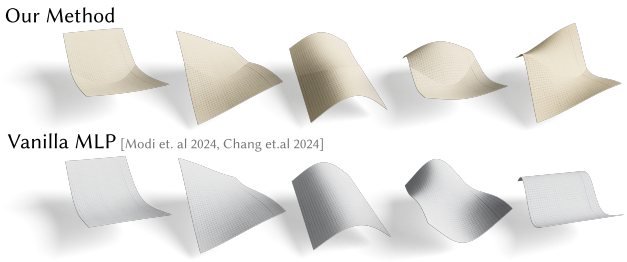


Fig. 16. Comparison of basis functions learned by our method (top row) and a standard SIREN MLP [Chang et al. 2024; Modi et al. 2024] (bottom row) on a 2D domain with a crease. The scalar-valued basis functions are visualized by lifting along the y-axis. Our method successfully captures the sharp gradient discontinuity introduced by the crease, while the vanilla MLP fails to represent this non-smooth behavior due to its inherent continuity.

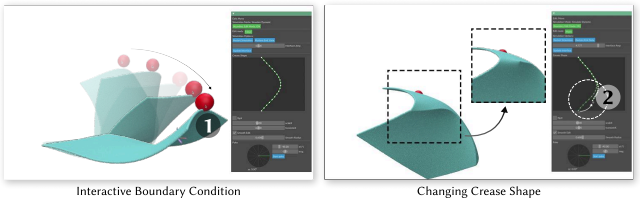


Fig. 17. Our method enables interactive editing of both crease geometry and boundary conditions. (1) highlights how dragging the red sphere changes the boundary condition, while (2) shows how the crease shape can be modified by editing the polyline directly in the interface.

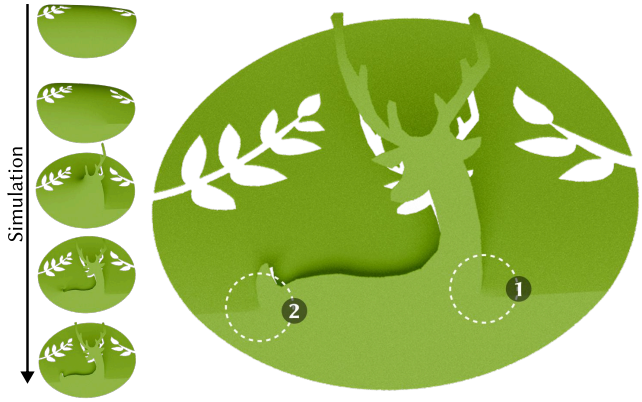


Fig. 18. Our method can be combined with [Chang et al. 2025], enabling simulation of a paper-like material undergoing both progressive cutting and creasing. The sequence on the left shows the deformation evolving from smooth bending to sharp folds as creases and cuts develop. Region (1) highlights a transition from crease to cut, while region (2) shows the reverse: a transition from cut to crease.

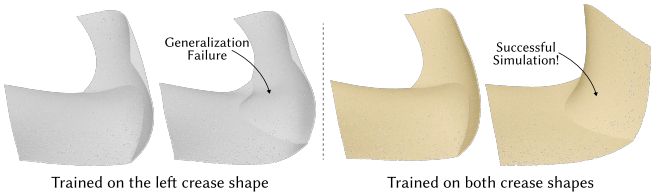


Fig. 19. On the left, we show simulation results from a model trained on a single crease shape. When tested on a significantly different crease shape, it captures the gradient discontinuity but produces behavior that is not physically meaningful. On the right, training on both crease shapes enables the model to successfully simulate both conditions.