

# Lifting the Winding Number: Precise Discontinuities in Neural Fields for Physics Simulation

YUE CHANG, University of Toronto, Canada  
MENGFEI LIU, University of Toronto, Canada  
ZHECHENG WANG, University of Toronto, Canada  
PETER YICHEN CHEN, MIT CSAIL, USA  
EITAN GRINSPUN, University of Toronto, Canada

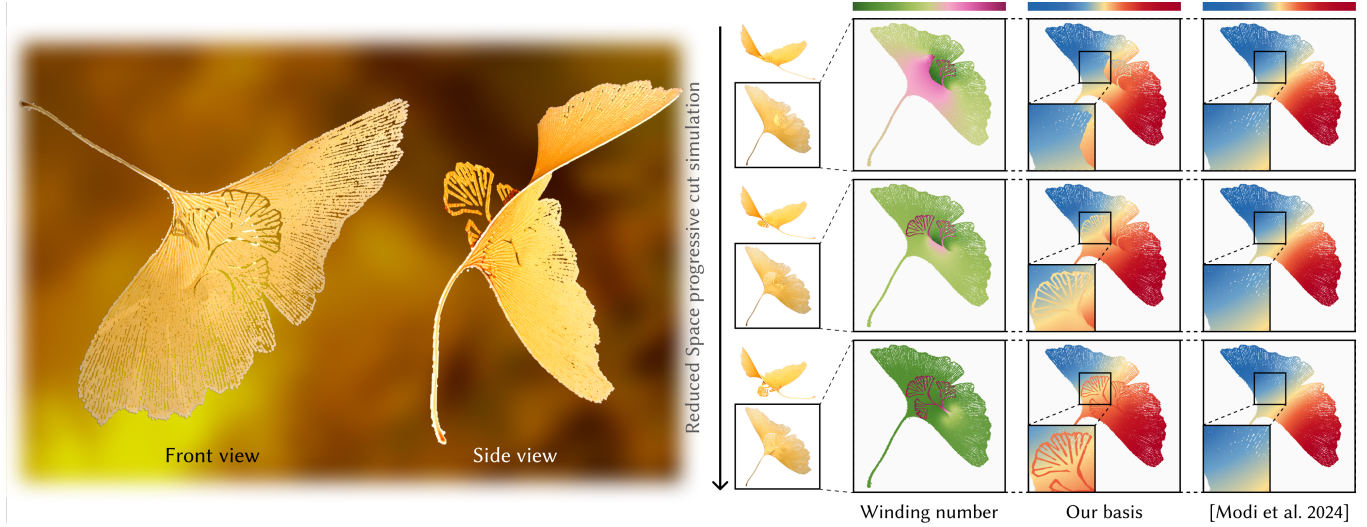


Fig. 1. Our method calculates a basis for reduced-space simulations that can represent shapes with a family of discontinuities. This basis is represented by neural fields. Unlike neural fields implemented by vanilla MLPs [Modi et al. 2024], which have built-in continuity, we represent discontinuities by construction using the winding number field. The basis can then be used for reduced-space simulation for progressive cutting.

Cutting thin-walled deformable structures is common in daily life, but poses significant challenges for simulation due to the introduced spatial discontinuities. Traditional methods rely on mesh-based domain representations, which require frequent remeshing and refinement to accurately capture evolving discontinuities. These challenges are further compounded in reduced-space simulations, where the basis functions are inherently geometry- and mesh-dependent, making it difficult or even impossible for the basis to represent the diverse family of discontinuities introduced by cuts.

Recent advances in representing basis functions with neural fields offer a promising alternative, leveraging their discretization-agnostic nature to

Authors' addresses: Yue Chang, University of Toronto, Canada, changyue.chang@mail.utoronto.ca; Mengfei Liu, University of Toronto, Canada, mengfei.liu@mail.utoronto.ca; Zhecheng Wang, University of Toronto, Canada, zhecheng@cs.toronto.edu; Peter Yichen Chen, MIT CSAIL, USA, pyc@csail.mit.edu; Eitan Grinspun, University of Toronto, Canada, eitan@cs.toronto.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1540-2/2025/08

<https://doi.org/10.1145/3721238.3730597>

represent deformations across varying geometries. However, the inherent continuity of neural fields is an obstruction to generalization, particularly if discontinuities are encoded in neural network weights.

We present *Wind Lifter*, a novel neural representation designed to accurately model complex cuts in thin-walled deformable structures. Our approach constructs neural fields that reproduce discontinuities precisely at specified locations, without “baking in” the position of the cut line. To achieve this, we augment the input coordinates of the neural field with the generalized winding number of any given cut line, effectively lifting the input from two to three dimensions. Lifting allows the network to focus on the easier problem of learning a 3D everywhere-*continuous* volumetric field, while a corresponding restriction operator enables the final output field to precisely resolve *strict* discontinuities. Crucially, our approach does not embed the discontinuity in the neural network’s weights, opening avenues to generalization of cut placement.

Our method achieves real-time simulation speeds and supports dynamic updates to cut line geometry during the simulation. Moreover, the explicit representation of discontinuities makes our neural field intuitive to control and edit, offering a significant advantage over traditional neural fields, where discontinuities are embedded within the network’s weights, and enabling new applications that rely on general cut placement.

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

Additional Key Words and Phrases: Cutting, Discontinuity, Reduced-order modeling, Implicit neural representation, Computational design

**ACM Reference Format:**

Yue Chang, Mengfei Liu, Zhecheng Wang, Peter Yichen Chen, and Eitan Grinspun. 2025. Lifting the Winding Number: Precise Discontinuities in Neural Fields for Physics Simulation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*, August 10–14, 2025, Vancouver, BC, Canada. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3721238.3730597>

**1 INTRODUCTION**

From laser-cut panels to handcrafting shapes with scissors, cuts pervade daily life. Simulating cuts in thin-walled structures such as leaves, paper, or fabric requires a kinematic representation that precisely resolves time-evolving spatial discontinuities. This is a requirement that is not met by current reduced order modeling (ROM) approaches [Barbič and James 2005].

ROM is an acceleration technique for physical simulations that is applicable for scenarios where the *deformation* complexity is low relative to the *geometric* complexity. In ROM, an offline precomputation identifies a truncated set of kinematic modes, which are leveraged in fast “reduced” simulation. By computing the dynamics of only a small number of modes, in lieu of a large number of geometric (e.g., mesh) degrees of freedom, ROMs offer speedups by orders of magnitude [Benner et al. 2015].

However, ROM’s benefits come at the cost of *precision* and *generalization*. ROMs are currently unable to treat cutting of thin-walled materials, particularly when cut placement may vary, cutting is progressive, or the cut boundary is complex. And yet, since detailed cuts introduce considerable geometric complexity (see Fig. 1), much could be gained if ROM could separate kinematics from geometry.

To effectively model cuts, ROM simulations would need a novel kinematic basis that spans the deformations induced by cutting and loading the deformable object. ROMs struggle with this, because cutting induces *strict* discontinuities in otherwise continuous displacement fields.

Current ROMs either lack representation or “bake in” the placement of discontinuities, therefore, they do not generalize over cut placement. For instance, mesh-based ROM mode precomputation ties the basis to the underlying discretization [Fulton et al. 2019; Shen et al. 2021; Sifakis and Barbič 2012]. Since progressing or altering a cut changes mesh connectivity, these ROMs cannot reuse the learned basis and must again precompute offline.

Recent work has leveraged continuous neural fields as kinematic basis representations [Chen et al. 2023b; Pan et al. 2022; Puri et al. 2024; Tao et al. 2024]. The advantage of these methods is that their precomputation is agnostic to the specific discretization of the domain geometry. However, current neural field techniques have shown limited capacity for representing high-frequency or discontinuous data [Belhe et al. 2023].

Indeed, efforts to represent discontinuous displacements in neural reduced simulations are nascent. Recently, Chang et al. [2023] trained on simulation snapshots of a small, manually-selected set of partial cuts, producing a displacement basis that encodes the discontinuities of partial cuts in the neural network’s weights. This limits their online reduced simulations to only the cut position seen during training, and limits the precision of the cut boundary due

to the neural network’s limited capacity to precisely represent a discontinuity along a curve.

**Contributions.** We propose a novel neural field construction and ROM approach specifically designed to *precisely* capture *strict* displacement discontinuities for a family of progressively-cut shapes over 2D domains. Our method enables reduced simulations of progressive cutting, for cut placements not seen during training. It embraces both training on existing deformation *data*, or training *data-free* by identifying natural vibration modes. The resulting reduced simulation maintains an explicit representation of cut positions as polylines that may be freely modified as the simulation proceeds, without remeshing or similar data structure updates.

Our mathematical approach is to transform the learning of a discontinuous field—a known challenge case for neural fields—into the learning of an *everywhere continuous* function—a fundamentally easier learning task. In a nutshell, we augment the input coordinates of the neural field with the generalized winding number of any given cut line [Jacobson et al. 2013], effectively lifting the input from two to three dimensions. Lifting creates 3D distance between corresponding sides of the cut, allowing the network to focus on the easy learning of a *continuous* volumetric field. A corresponding restriction operator enables final output *precisely* resolving *strict* discontinuities, even for cuts unseen during training.

Highlighting the key ingredients of *winding* and *lifting*, we call this *Wind Lifter*. After training a volumetric continuous neural field once, we can (re)compute the winding number at runtime, without any training, as a cut line is lengthened, moved, or reshaped. By combining the continuous neural field and the winding graph, we generate a field with precise, strict discontinuities, while maintaining the speed of ROM.

With Wind Lifter, the discontinuity placement is decoupled from the neural network’s weights, and editing the cut geometry is straightforward. Such editing is not possible with traditional neural fields, where the discontinuities are embedded within the network’s weights. As a result, our method enables novel applications, such as user-interactive design of cut shapes.

In summary, we:

- introduce Wind Lifter, a generalizable, precise, and easily editable representation of discontinuities in 2D neural fields;
- integrate this new capability with a discretization-agnostic ROM method to enable reduced simulation of cutting;
- demonstrate simulation results with complex cuts and cut shape generalization unparalleled by prior reduced modeling methods, including an interactive design prototype and a real-world comparison.

**2 RELATED WORK****2.1 Simulating Cutting of Deformable Bodies**

A common approach towards simulating cutting is refining and adapting the mesh to capture discontinuities. This approach needs to balance various considerations, such as the number of newly generated elements [Bielser and Gross 2000; Mor and Kanade 2000], the quality of the cut details [Busaryev et al. 2013], and material information [Chen et al. 2014]. While this approach is intuitive to

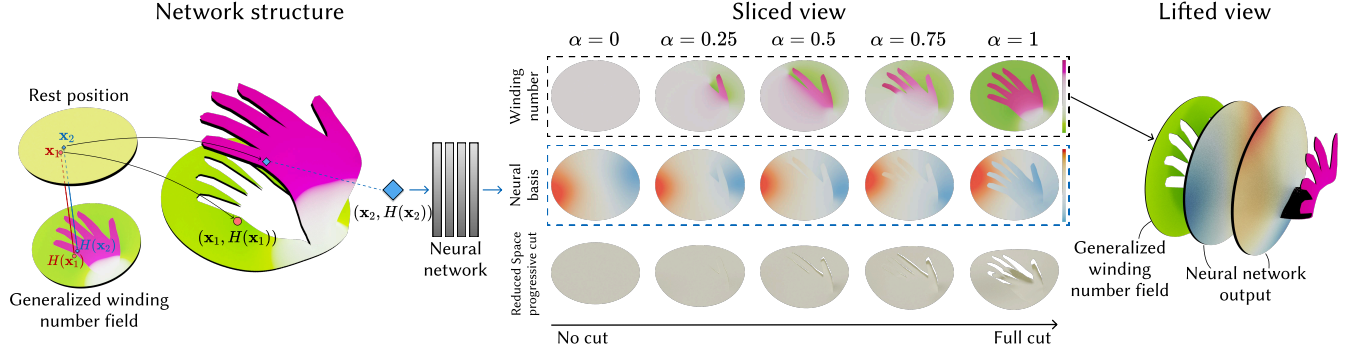


Fig. 2. We augment the input to our neural field with a generalized winding number, lifting the domain from 2D into 3D. As the cut progresses, the generalized winding number changes accordingly throughout the domain, capturing the family of discontinuities. The neural network takes the lifted input and outputs the basis function. This allows the neural network to learn a continuous function. The discontinuity is captured by the restriction of a continuous function over a 3D domain.

model, practical implementation is challenging. Remeshing can be challenging to implement [Zhang et al. 2019] and computationally expensive [Obiols-Sales et al. 2023]. By contrast, our approach takes a mesh-free approach through neural fields and completely bypasses the need to remesh.

Hybrid Lagrangian-Eulerian methods like the material point method [Jiang et al. 2016] handle cutting without remeshing by employing a combination of grid and particle representations. However, the particle-based representation lacks precise surface definition, resulting in artifacts such as numerical fractures and imprecise cutting [Fan et al. 2025; Su et al. 2022].

Extended finite element methods (XFEM) [Kaufmann et al. 2009; Koschier et al. 2017; Moës et al. 1999; Ton-That et al. 2024] maintain a fixed mesh and resolve discontinuities using discontinuous “enrichment functions.” Our method also resolves discontinuities using a discontinuous basis. However, unlike XFEM, our approach does not depend on a background mesh, separating the kinematic and geometric representations. This separation enables the simulation of shapes with intricate thin details (see Figure 1) and multiple cuts intersecting localized region corresponding to one mesh triangle or grid cell, situations that can challenge XFEM. Moreover, our separation allows for fast, reduced-order modeling, whereas XFEM solves partial differential equations (PDEs) in maximal degrees of freedom.

## 2.2 Neural Methods for Simulation

Neural networks have demonstrated efficiency in various areas of physics-based simulation, including deformable simulation [Chen et al. 2023a; Feng et al. 2024; Lyu et al. 2024], fluid simulation [Deng et al. 2023; Jain et al. 2024; Kim et al. 2019; Tao et al. 2024], and collision modeling and handling [Cai et al. 2022; Romero et al. 2021; Yang et al. 2020], among others.

Among all neural methods for physics-based simulation, our approach aligns most closely with reduced-space simulation techniques, which accelerate simulations by finding a reduced basis. These bases can either be mesh-dependent [Fulton et al. 2019; Shen et al. 2021] or mesh-independent [Chang et al. 2023], and can be data-based (derived from simulation sequences) [Chen et al. 2023b;

Zong et al. 2023] or data-free (derived from domain geometry) [Modi et al. 2024; Sharp et al. 2023].

However, *none of the data-free methods accommodate cutting during simulation*. Of the data-based methods, Chang et al. [2023] is able to reproduce only the cut seen during training, that cut being encoded in the neural network weights. By contrast, our method supports procedural cuts without requiring simulation snapshots for training.

For instance, when the reduced basis is computed without simulation data [Modi et al. 2024], the training process depends solely on sampling the domain. This approach fails to distinguish between an undamaged shape and one with a zero-volume cut, rendering it incapable of representing discontinuities in data-free training scenarios. To address these limitations, our method expressly represents lines of discontinuity and introduces a function representation capable of capturing evolving cut geometries even for zero-volume cuts.

In the context of discontinuity modeling with neural fields [Liu et al. 2024], Belhe et al. [2023] align feature fields from a triangle mesh to discontinuities, which is effective for compressing 2D physics simulation data. However, their approach learns a feature field defined on a mesh, requiring (re)meshing of the domain if discontinuities are placed, extended, or moved. This dependency makes their method unsuitable for modeling a family of discontinuities necessary for progressive cutting, or allowing cut placement and geometry to be edited interactively. Our approach allows for this broader functionality by separating the representation of the continuous field and the discontinuity.

## 2.3 Inside/Outside Descriptors

Modelling discontinuities inherently requires querying whether a point is on one side or the other. Signed distance fields (SDFs) and occupancy functions [Mescheder et al. 2019] both come to mind. Both rely on a level set to represent a boundary. However, these approaches represent insideness only for closed domains, whereas a cut may be partial or incomplete, i.e., a curve of discontinuity need not be a closed curve. Indeed, an open curve of discontinuity may evolve in time to close up, further begging for a richer representation.

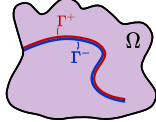
The winding number is a fundamental concept in mathematics, particularly in complex analysis, geometry, and topology. It measures how many times a closed curve winds around a given point in the plane, by integrating the “angle of projection” along the curve with relation to the query point. As a field over all points on the plane it is piecewise constant with jump conditions across the curve [Reinhart 1960; Shimrat 1962], making it a natural candidate for “strict” insideness tests. Jacobson et al. [2013] proposed its generalization (GWN) to open curves, where it is harmonic with jump conditions across the curve, thereby serving as a “soft” insideness test. The GWN has been implemented for point clouds [Barill et al. 2018] and curved surfaces [Feng et al. 2023], and applied to garment modeling [Chi and Song 2021; Hu et al. 2018], geometry processing [Xu et al. 2023; Zhou et al. 2016], and computer vision [Müller et al. 2021]. Our work leverages the GWN to represent discontinuities.

### 3 HOW TO REPRESENT A DISCONTINUOUS FIELD USING A CONTINUOUS NEURAL NETWORK

We aim to accurately represent 2D fields with lines of discontinuity. The field value differs depending on the side from which the discontinuity is approached. Let  $f : \Omega \rightarrow \mathbb{R}$  be a real-valued function over  $\Omega \subset \mathbb{R}^2$  discontinuous across a curve  $\Gamma$ ,

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_0^+} f(\mathbf{x}) \neq \lim_{\mathbf{x} \rightarrow \mathbf{x}_0^-} f(\mathbf{x}), \quad \mathbf{x}_0 \in \Gamma, \quad \mathbf{x} \in \Omega, \quad (1)$$

where  $\mathbf{x}_0^+$  and  $\mathbf{x}_0^-$  indicate approaching  $\mathbf{x}_0$  from opposing sides  $\Gamma^+$  and  $\Gamma^-$ , respectively (see incident figure). With a discontinuity, arbitrarily small changes to the evaluation point may produce a finite jump in the field value. We would like to leverage available tools for training neural fields to learn such a function. Unfortunately, typical neural field architectures are poorly suited for this representation task.



**Challenges.** Neural fields are typically continuous and differentiable because they are represented by neural networks, such as fully connected feedforward networks, which use smooth activation functions (e.g., sigmoid, or sine). Rahaman et al. [2019] study the spectral bias of such networks, observing that smooth activation functions naturally favor smoother outputs, and common architectures have a natural tendency to learn low-frequency components of a target function first. These properties pose a challenge to precisely representing sharp changes or discontinuities.

Some techniques may ameliorate, but do not inherently resolve this challenge. For instance, input encoding methods, such as Fourier feature mappings or sinusoidal positional embeddings [Tancik et al. 2020], and improved activation functions, such as SIREN [Sitzmann et al. 2020], improve the ability to approximate high-frequency details, but they do not inherently and explicitly model strict discontinuities.

**Proposed architecture.** We do *not* seek to improve what appears to be a fundamental inability of neural fields to represent strict discontinuities. We avoid this limitation and embrace the low-frequency bias of neural fields. We will use feature augmentation to make

the learning task easier, simplifying our task to learning only a *continuous* field.

Let  $\tilde{f}_\theta : \Omega \times \mathbb{R} \rightarrow \mathbb{R} : (x, y, z) \mapsto \tilde{f}_\theta(x, y, z)$  be a neural field over a *volumetric* domain  $\Omega \times \mathbb{R} \subset \mathbb{R}^3$  parameterized by neural network weights  $\theta$ . Our learning task will seek a *smooth* neural field  $\tilde{f}_\theta$ , without discontinuities, by appropriately choosing and leveraging the augmented feature  $z$ .

Let  $z = H(\mathbf{x})$ , where  $\mathbf{x} = (x, y)$ . Here,  $H : \Omega \rightarrow \mathbb{R}$  is a function that is discontinuous over  $\Gamma$ . We require that  $H$  have an analytical representation, that is,  $H$  need not be trained using a neural network.

**Lifting.** Our augmentation has a simple geometric interpretation. As depicted in the incident figure, our approach lifts the domain into a graph of  $H(\mathbf{x})$  over  $\Omega$ :



$$\mathcal{L}(\mathbf{x}) = (x, H(\mathbf{x})) = (x, y, H(x, y)), \quad \mathbf{x} \in \Omega. \quad (2)$$

Since  $H$  is discontinuous over  $\Gamma$ , the graph  $\mathcal{L}$  has separated boundaries  $\Gamma^+$  and  $\Gamma^-$ .

**Central idea: Discontinuity by restriction.** Our central idea is to define the 2D field  $f$  as the restriction of the volumetric neural field  $\tilde{f}_\theta$  to the graph  $\mathcal{L}$ :

$$f = \tilde{f}_\theta \circ \mathcal{L}, \quad \text{or} \quad f(\mathbf{x}) = \tilde{f}_\theta(\mathcal{L}(\mathbf{x})), \quad \mathbf{x} \in \Omega. \quad (3)$$

**Why it works.** The benefit of lifting is that it decouples spatial discontinuity from the trainable continuous function  $\tilde{f}_\theta$ . Without lifting, the jumps across  $\Gamma$  must be captured by the weight of the neural field, which complicates training. However, our method alleviates this problem. By lifting, two positions  $\mathbf{x}_0^- \rightarrow \Gamma^-$  and  $\mathbf{x}_0^+ \rightarrow \Gamma^+$ , which are close in  $\Omega$ , are mapped to distinct coordinates in 3D, enabling  $\tilde{f}_\theta$  to represent a smooth function, while the discontinuity emerges from its restriction to the discontinuous graph.

As a result, optimizing network weights becomes easier, as no sharp jumps need to be fit, as shown in the very right part in Figure 2. Another advantage of this construction is that it simplifies editing of the field  $f$ , as the output can be directly controlled by modifying the height function  $H(\mathbf{x})$ , as depicted in Figure 5.

**Winding graph.** While any function discontinuous over  $\Gamma$  may serve as the height  $H$ , we choose the generalized winding number field [Jacobson et al. 2013], a harmonic function with jump boundary condition across  $\Gamma$  amenable to fast evaluation [Barill et al. 2018]

$$H(\mathbf{x}) = \int_0^1 \frac{\Gamma'(s) \cdot (\Gamma(s) - \mathbf{x})^\perp}{|\Gamma(s) - \mathbf{x}|^2} ds, \quad (4)$$

where  $\Gamma : (0, 1) \rightarrow \Omega : s \mapsto \Gamma(s)$ , and  $(\cdot)^\perp$  rotates a vector by  $\pi/2$ . We refer to the graph of  $H(\mathbf{x})$  as the *winding graph*.

**Progressive cutting.** Compared to a permanent discontinuity, progressive cutting involves the notion that the curve of discontinuity lengthens over time. We introduce the parameter  $\alpha \in [0, 1]$  to mark the fraction of  $\Gamma$  that has been cut thus far. We use  $\Gamma^\alpha$  to denote the portion of  $\Gamma$  corresponding to the first  $\alpha$ -fraction of its length, and we account for  $\alpha$  by modifying the integration bounds:

$$H^\alpha(\mathbf{x}) = \int_0^\alpha \frac{\Gamma'(s) \cdot (\Gamma(s) - \mathbf{x})^\perp}{|\Gamma(s) - \mathbf{x}|^2} ds. \quad (5)$$



Similarly, since the cut progression generally affects the optimal kinematic basis, we also condition the volumetric neural field on  $\alpha$ :

$$\tilde{f}_\theta^\alpha : [0, 1] \times \Omega \times \mathbb{R} \rightarrow \mathbb{R} : (\alpha, x, y, z) \xrightarrow{\text{inference}} \tilde{f}_\theta^\alpha(x, y, z). \quad (6)$$

The lifting and restriction operators are unaffected, but for completeness we indicate their dependency on  $\alpha$ :

$$\mathcal{L}^\alpha(\mathbf{x}) = (\mathbf{x}, H^\alpha(\mathbf{x})) \quad \text{and} \quad f^\alpha(\mathbf{x}) = \tilde{f}_\theta^\alpha(\mathcal{L}^\alpha(\mathbf{x})). \quad (7)$$

*Outlook.* Equations 5–7 summarize the construction of a function  $f^\alpha : \Omega \rightarrow \mathbb{R}$  discontinuous across part of the curve  $\Gamma$  depending on the cutting extent  $\alpha$ . As we have seen,  $f^\alpha$  is constructed by restricting a learned volumetric neural field ( $\tilde{f}_\theta$ ) to the *winding graph*  $H(\mathbf{x})$  using the *lifting operator*  $\mathcal{L}$ .

We are ready to construct ROMs that precisely resolve discontinuities, applying the same constructions and notation to kinematics-specific fields.

## 4 REDUCED ORDER MODELING

*Kinematics.* Consider a thin-walled elastic body parameterized by the domain  $\Omega \subset \mathbb{R}^2$ . The deformed position of the body in three-space is given by the displacement field  $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$ . Reduced order modeling (ROM) seeks to represent this displacement field using a small number  $k$  of coordinates, that is, via a reduced configuration  $\mathbf{z} \in \mathbb{R}^k$  [An et al. 2008; Barbič and James 2005; Kim and James 2009]. In particular, linear ROM, which we will consider here, requires that  $\mathbf{u}$  be *linear* in  $\mathbf{z}$ , that is,  $\mathbf{u} = \mathbf{z}^T \Phi$ , where  $\Phi^T = [\phi_1, \dots, \phi_k]$ , and  $\phi_i : \Omega \rightarrow \mathbb{R}^3$  is a displacement *basis*.

In our setting, the displacement basis must represent a discontinuity over  $\Gamma$  depending on the cutting extent  $\alpha$ . Therefore,  $\Phi$  is dependent on  $\alpha$ . We will be reminded of this dependency with the notation  $\Phi^\alpha$ , and omit the decorations where it is clear from context.

### 4.1 Precomputation: Training

*Training overview.* Recalling Equation 3, we construct a displacement basis field  $\phi^\alpha$  discontinuous across  $\Gamma$  by training a volumetric neural field  $\tilde{\phi}_\theta^\alpha$  continuous over  $\Omega \times \mathbb{R}$ . While  $\tilde{\phi}_\theta^\alpha$  and  $\phi^\alpha$  are vector-valued, nothing changes in the lifting construction, which remains simply  $\phi^\alpha = \tilde{\phi}_\theta^\alpha \circ \mathcal{L}^\alpha$ .

Our main contribution (Section 3) is orthogonal to the choice of training scheme. In fact, our neural discontinuity representation is compatible with both data-driven and data-free approaches. Below, we will develop how to incorporate the proposed winding-number-lifting approach in either of the training setups. Regardless of the approach taken, the goal is to train the weights  $\theta$ .

*Data-driven basis learning.* In the data-driven setting, we begin by collecting training snapshots, recording each displacement field  $\mathbf{u}_j$  and cut progression  $\alpha_j$  at time increment  $j$ . The neural field  $\tilde{\phi}_\theta^\alpha$  is then trained by minimizing the reconstruction loss over all the simulation snapshots:

$$\mathcal{L}_{\text{data-driven}} = \sum_j \left\| \mathbf{z}_j^T \tilde{\phi}_\theta \circ \mathcal{L} - \mathbf{u}_j \right\|_2^2, \quad (8)$$

where  $\mathbf{z}_j \in \mathbb{R}^k$  is the reduced coordinate corresponding to the optimal reconstruction at time increment  $j$ . Note that  $\|f\|_2^2 = \int_\Omega f(\mathbf{x})^2 d\mathbf{x}$

is the  $L_2$  norm on  $\Omega$ , which is estimated via uniform stochastic cubatures [An et al. 2008; Carlberg 2011]. For more details on the training process, please refer to [Chang et al. 2023]. The key distinction in our approach is the incorporation of cutting, achieved by restricting the neural field to the lifting function. Furthermore, we ensure that both  $\Phi$  and  $\mathcal{L}$  explicitly depend on  $\alpha_j$ , the cut progression parameter.

*Data-free basis learning.* In addition to the previous data-driven training setting, our neural field can also be trained in a data-free fashion. Following Modi et al. [2024], we trained the neural network by minimizing the elastic energy

$$\mathcal{L}_{\text{data-free}} = E_{\text{elas}} = \int_\Omega \Psi(\mathbf{u}(\mathbf{x})) d\mathbf{x} \quad (9)$$

, where  $\Psi$  is the elastic energy density, e.g., St. Venant-Kirchhoff (StVK) material [Barbič and James 2005]. For more details on the data-free training, please refer to the supplementary material and [Modi et al. 2024]. We emphasize that the loss function only involves an analytically defined elastic energy and does not involve any training data (e.g., from full-order simulations).

### 4.2 Dynamic Subspace Simulation

Following Chang et al. [2023], the reduced configuration  $\mathbf{z}$  is updated at each time step via the optimization

$$\mathbf{z}_{j+1} = \operatorname{argmin}_{\mathbf{z}} \frac{1}{2} \|\mathbf{z} - \mathbf{z}_{j+1}^{\text{pred}}\|^2 + h^2 \int_\Omega \Psi(\mathbf{u}(\alpha, \mathbf{x}, \mathbf{z})) d\mathbf{x}, \quad (10)$$

where  $h$  is the time step size, and  $\mathbf{z}_{j+1}^{\text{pred}} = 2\mathbf{z}_j - \mathbf{z}_{j-1}$ . Evaluating the elastic energy  $\Psi$  involves computing the deformation gradient  $\mathbf{F} = \partial \mathbf{u}(\alpha, \mathbf{x}, \mathbf{z}) / \partial \mathbf{x}$ . This gradient can be directly obtained from the reduced-space coordinate  $\mathbf{z}$  and the spatial gradient of the neural field,  $\partial \phi_i^\alpha / \partial \mathbf{x}$ , which is efficiently calculated via automatic differentiation of the neural network. We evaluate the domain integral using uniform stochastic cubature. Our examples optimize (10) using gradient descent; we also implemented Newton’s method and observed similar performance.

### 4.3 Implementation

*Progressive cutting and cut placement editing.* Our progressive cutting simulations increase the cutting extent  $\alpha$  over time. Our interactive design application modifies  $\Gamma$  during simulation. Changing either  $\alpha$  or  $\Gamma$  immediately affects the winding graph, which is computed as-needed analytically; crucially, it does not require retraining the volumetric neural field.

Our implementation represents  $\Gamma$  as a collection of polylines (allowing for more than one cut). The evaluation of the generalized winding number locates the appropriate bounds of integration by walking along the polyline until the fractional length  $\alpha$  has been walked (refer to supplemental material).

*Strain singularity at crack tip.* The elastic energy  $\Psi$  involves the spatial gradient of displacements  $\nabla_{\mathbf{x}} \mathbf{u}$ , and in turn  $\nabla_{\mathbf{x}} H$ . While the winding number itself is bounded, its gradient diverges approaching the endpoints of  $\Gamma$ . In mechanics this is known as the crack tip strain singularity, a recognized challenge to force calculations [Mousavi et al. 2011]. Following typical treatments, we smooth the  $\epsilon$ -neighborhood around the endpoints of  $\Gamma^\alpha$  by multiplying the

generalized winding number by a cubic spline kernel function (refer to supplemental material).

*Joint training.* Since  $\Phi$  is a  $k$ -dimensional basis, we must repeat the lifting construction  $k$  times. Conceptually, the simplest approach is to train  $k$  volumetric networks, however, this is not strictly necessary, since a volumetric neural field is not limited to a 1-dimensional (real-valued) output. In our implementation, we parameterize all  $k$  displacement bases using the same neural network weights  $\theta$  by making  $\hat{\phi}_\theta^\alpha$  a  $k$ -valued field, i.e.,  $\phi_i^\alpha = \hat{\phi}_{\theta,i}^\alpha \circ \mathcal{L}^\alpha$ .

## 5 RESULTS

Table 1 summarizes the timing statistics for the examples in our paper. Time per step increases with cut changes due to the need to re-evaluate the neural network and update the basis for the new cut shape. Without cut changes, our method matches traditional reduced space simulations. Our method runs at 28.5 ~ 42.4 fps with cut changes and 62 ~ 166 fps without. For reference, full-space simulation takes around 1.1s per frame without cut changes during simulation. This gives our method a  $30\times \sim 183\times$  speedup over full-space simulations while enabling cut changes during simulation. All results are reported using an NVIDIA RTX 4090 GPU and an AMD 7950X CPU. Unless otherwise noted, we use a 5-layer, 128-channel SIREN MLP with positional encoding up to maximum frequency of  $2^3$ . Winding numbers are scaled by a factor of 32 to ensure sufficient separation of lifted and 2D coordinates.

*Discretization-agnostic domain representation.* Our method does not assume a specific type of domain discretization, instead relying only on the ability to sample the domain. All examples below use stochastic cubature drawn from a uniform distribution over the domain [Chang et al. 2023; Modi et al. 2024]. Figures 10 and 14 are rendered with a mesh bound to the shape, while other figures are visualized as point clouds with colors sampled from a texture.

### 5.1 Evaluation on Data-Free Settings

*Comparison to prior work.* To the best of our knowledge, our method is the first data-free model reduction method to support displacement field discontinuities, progressive cutting, or generalization of cut placement. As depicted by the butterfly in Figure 3 and the small ginkgo leaves in Figure 1, our method enables a ROM with highly detailed cut structures. All of our data-free examples use the 18-dimensional basis presented above.

*Generalization.* We explore generalization to *external loads* and *cut geometry* unseen during training. Since data-free ROMs do not explicitly consider external loads during training, any external force may demonstrate generalization. We applied a sequence of external forces at different stages of a progressive helical cut, inducing varying deformations responsive to the changing geometry of the evolving discontinuity (see Figure 9 and accompanying video). We cut a helix from real paper and observed that its final sagged state agrees with the simulated result (see Figure 6).

To evaluate generalization over cut geometry, we trained the network weights  $\theta$  on three “training” cuts, then froze the weights and simulated three significantly different “test” cuts, observing plausible results (see Figure 7 and accompanying video).

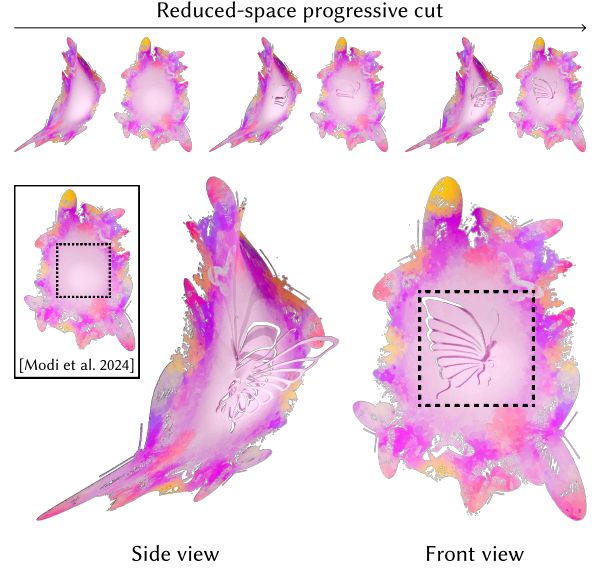


Fig. 3. Our method is able to capture the complex cut of this butterfly shape, this is not achievable by previous data-free method [Modi et al. 2024] that uses traditional MLP due to the built-in continuity.

*Interactive design.* We prototyped an interactive cut editor (see Figure 5). The designer draws and edits the cut geometry, applies external forces, while observing the corresponding deformation in real time. The cut position and shape may be edited without restarting the simulation, with the deformation updating in real time (refer to accompanying video). To the best of our knowledge, real-time physical preview as a cut geometry is altered has not been reported in the literature. Implementing interactive cut editing with a pure neural field representation [Chang et al. 2023] is not possible since the discontinuity is “baked in” to the network weights; on the other hand, if a mesh-based discontinuity representation is introduced [Belhe et al. 2023], interactive editing would necessitate ongoing remeshing to align mesh edges with  $\Gamma$ , which has not been demonstrated, and has the potential to be costly or to exhibit basis-projection artifacts.

### 5.2 Evaluation on Data-Driven Settings

We also evaluate our method in data-driven settings. The training data and ground-truth full-space simulations for all examples in this section are generated using the data generation code publicly released by Chang et al. [2023], which implements a FEM solver for stable Neo-Hookean energy [Smith et al. 2018]. We train  $k = 20$  displacement modes for all examples in this section.

*Interactive deformation editing.* We trained a ROM on simulations of tugging on square sheet with various etched angular cuts (see Figures 10 and 4). By connecting many instances of this ROM end to end, we obtain a reduced simulation of a “kirigami tower.”

We demonstrate generalizability with an interactive interface that enables users to edit the cut geometry while the simulation is active (see Figures 11 and 4). Starting from a straight cut, users can

|                                  | Number of Vertices (k) | Training Time (s) | Time per Step<br>With Cut (ms) | Time per Step<br>Without Cut (ms) |
|----------------------------------|------------------------|-------------------|--------------------------------|-----------------------------------|
| <b>hand</b> (Figure 2)           | 50                     | 53                | 23.56                          | 8.57                              |
| <b>leaf</b> (Figure 1)           | 50                     | 78                | 25.40                          | 9.96                              |
| <b>butterfly</b> (Figure 3)      | 90                     | 73                | 34.86                          | 14.89                             |
| <b>helical</b> (Figure 9)        | 100                    | 161               | 35.32                          | 14.77                             |
| <b>generalization</b> (Figure 7) | 100                    | 81                | 35.11                          | 15.15                             |
| <b>kirigami</b> (Figure 10)      | 137                    | 2760              | 30.95                          | 16.01                             |
| <b>slit</b> (Figure 14)          | 20                     | 1740              | 25.33                          | 6.02                              |

Table 1. We provide the number of vertices and timing details for all examples. Thanks to the compact neural representation, our approach significantly reduces training time compared to many neural methods that require hours or even days. Our training time ranges from just under a minute to slightly less than an hour. During inference, the time cost is higher when cut changes are involved, as these require both neural network inference and basis updates.

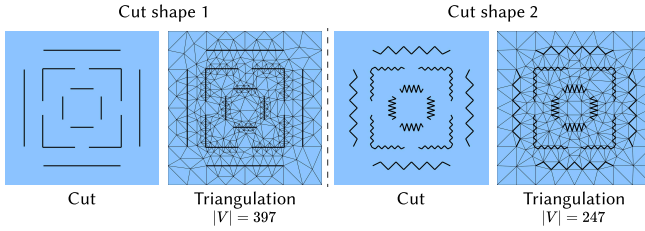


Fig. 4. The cut change can lead to a big difference in discretization using the mesh-based method. We applied constrained Delaunay triangulation with the same setting for these two cut shapes, and the resulting number of vertices is significantly different.

drag and adjust the control points of the kirigami cuts. After editing, the resulting zig-zag cut shape significantly differs from the initial straight cut, which was not included in the training data (Figure 4).

This type of editing is particularly challenging for prior mesh-based methods, including neural fields that rely on meshes [Belhe et al. 2023]. As shown in Figure 4, we applied constrained Delaunay triangulation [Shewchuk 2005] to two different cut shapes. The resulting number of vertices and triangle arrangement for the two cut patterns differs significantly, highlighting that remeshing is a global operation and that alignment of edges and points to revised cut lines would be a non-trivial (unexplored) alternative.

**One-shot generalization.** To evaluate quantitatively the generalizability of our method, we compare it with previous approaches that also use neural fields to construct the basis for reduced-space simulation, including LiCROM [Chang et al. 2023] and Simplicits [Modi et al. 2024]. As a third baseline, the DANN network proposed by Belhe et al. [2023], which is designed to better capture discontinuities, replaces the standard MLP in LiCROM, keeping the remainder of LiCROM intact.

We first run full-space simulations on cuts at different positions and train all methods using simulation snapshots from one cut position. For Simplicits, which does not require simulation trajectories and only samples the rest shape, we trained solely on the undeformed shape samples found in the same training set.

As shown in the top row of Figure 14, when evaluated on the training set, the reconstruction errors for all data-driven methods are

below 0.14%. Our approach achieves the most accurate reconstruction of deformation details and exhibits the lowest reconstruction error among them.

To further highlight our method’s advantage in modeling discontinuities, we compare the reconstruction error of our method against LiCROM [Chang et al. 2023] across various network scales and activation functions. As shown in Figure 13, our method reduces the mean squared error by an order of magnitude under the same settings. Furthermore, our method demonstrates faster convergence compared to LiCROM.

Since Simplicits [Modi et al. 2024] does not require training data, it is expected to have a larger reconstruction error. Simplicits does not consider cuts or discontinuities in its vibration mode analysis, and it also cannot leverage training data that exemplifies the displacements induced by discontinuities. For these reasons, a Simplicits model does not “know” that a cut exists. Since the neural field used in Simplicits has built-in continuity, the reduced simulations keep the two sides of the cut glued together. This result reflects the state of the art in geometry-agnostic data-free model reduction until now. By contrast, our approach is explicitly discontinuity-aware, and therefore accurately captures cutting.

When tested on cut shapes not included in training data, our method demonstrates greater generalizability. We have shown the one-shot generalizability of our approach. As shown in the bottom line of Figure 14, when tested on cut shapes with altered cut positions, all previous methods exhibit limitations in reconstructing visually reasonable results. In contrast, our method is flexible enough to adapt to new cut positions by simply updating the  $\Gamma$  to the new cut position *without* retraining the neural basis. This adaptation is nontrivial for previous methods because their basis functions are either fully coupled to the neural network weights [Chang et al. 2023] or require remeshing and retraining of the entire domain to handle new cuts [Belhe et al. 2023]. We plot the reconstruction error as a function of the normalized gap difference between the test shapes and those used during training. As shown in Figure 12, the error increases with the percentage difference between training and test cuts. No significant visual artifacts are observed when the difference is below 15%.

## 6 DISCUSSIONS AND FUTURE WORK

In this work, we proposed a novel neural representation technique for capturing precise displacement discontinuities in reduced-order models, with a focus on cuts in thin-walled deformable structures. Our method leverages a generalized winding number field to encode discontinuities, offering significant improvements over traditional mesh-based approaches and other neural field techniques.

**Collision.** One critical area for improvement is collision handling. While our method successfully represents displacement discontinuities, real-world applications often involve interacting objects where (self-)collision detection and response are critical [Zesch et al. 2023].

**Complex cuts.** The cut representation in this paper is limited to piecewise linear curves. Future work could explore extending this approach to incorporate higher-order parametric representations, such as Bézier curves [Spainhour et al. 2024].

**Out-of-Distribution Challenges.** Thanks to the discretization-agnostic approach, our method demonstrates robust generalization capabilities not seen before in subspace physics simulations. However, generalization remains a challenging aspect, particularly for significant, out-of-distribution scenarios. As shown in Figure 8, our method’s performance decreases when tested on winding number distributions significantly different from the training data. Future research could explore adaptive training strategies or augmentation techniques that ensure the winding number distribution covers a broader range of scenarios [Grangier et al. 2023]. Another avenue would be to exploit the volumetric nature of the neural field: it would be interesting to explore whether the field may be trained on multiple alternative cut positions, i.e., supervised by its restriction onto more than one winding graph.

## 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

- Steven S. An, Theodore Kim, and Doug L. James. 2008. Optimizing cubature for efficient integration of subspace deformations. *27*, 5 (2008). <https://doi.org/10.1145/1409060.1409118>
- 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.
- Gavin Barill, Nia Dickson, Ryan Schmidt, David I.W. Levin, and Alec Jacobson. 2018. Fast Winding Numbers for Soups and Clouds. *ACM Transactions on Graphics* (2018).
- 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>
- Peter Benner, Serkan Gugercin, and Karen Willcox. 2015. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review* 57, 4 (2015), 483–531.
- Daniel Bielser and Markus H. Gross. 2000. Interactive Simulation of Surgical Cuts. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications (PG'00)*. IEEE Computer Society, USA, 116.
- Oleksiy Busaryev, Tamal K. Dey, and Huamin Wang. 2013. Adaptive Fracture Simulation of Multi-Layered Thin Plates. *ACM Trans. Graph. (SIGGRAPH)* 32, 4, Article 52 (jul 2013), 6 pages. <https://doi.org/10.1145/2461912.2461920>
- 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>
- Kevin Thomas Carlberg. 2011. *Model reduction of nonlinear mechanical systems via optimal projection and tensor approximation*. Ph. D. Dissertation. Stanford University.
- 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>
- Peter Yichen Chen, Maurizio M. Chiaramonte, Eitan Grinspun, and Kevin Carlberg. 2023a. Model reduction for the material point method via an implicit neural representation of the deformation map. *J. Comput. Phys.* 478 (2023), 111908. <https://doi.org/10.1016/j.jcp.2023.111908>
- 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. 2023b. 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>
- Zhili Chen, Miaojun Yao, Renguo Feng, and Huamin Wang. 2014. Physics-inspired adaptive fracture refinement. *ACM Trans. Graph.* 33, 4, Article 113 (jul 2014), 7 pages. <https://doi.org/10.1145/2601097.2601115>
- Cheng Chi and Shuran Song. 2021. GarmentNets: Category-Level Pose Estimation for Garments via Canonical Space Shape Completion. In *The IEEE International Conference on Computer Vision (ICCV)*.
- 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).
- L. Fan, FM Chitalu, and T Komura. 2025. A Hybrid Lagrangian–Eulerian Formulation of Thin-Shell Fracture. In *Computer Graphics Forum*. Wiley Online Library, e15273.
- Nicole Feng, Mark Gillespie, and Keenan Crane. 2023. Winding Numbers on Discrete Surfaces. *ACM Trans. Graph.* 42, 4, Article 36 (jul 2023), 17 pages. <https://doi.org/10.1145/3592401>
- Xudong Feng, Huamin Wang, Yin Yang, and Weiwei Xu. 2024. Neural-Assisted Homogenization of Yarn-Level Cloth. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, New York, NY, USA.
- 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 Grangier, Pierre Ablin, and Awni Hannun. 2023. Adaptive training distributions with scalable online bilevel optimization. *arXiv preprint arXiv:2311.11973* (2023).
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral meshing in the wild. *ACM Trans. Graph.* 37, 4, Article 60 (jul 2018), 14 pages. <https://doi.org/10.1145/3197517.3201353>
- Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. 2013. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.* 32, 4, Article 33 (jul 2013), 12 pages. <https://doi.org/10.1145/2461912.2461916>
- 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>
- Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*. 1–52.
- 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>
- 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 Doug L. James. 2009. Skipping steps in deformable simulation with online model reduction. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/1661412.1618469>
- Dan Koschier, Jan Bender, and Nils Thuerey. 2017. Robust eXtended finite elements for complex cutting of deformables. *ACM Trans. Graph.* 36, 4, Article 55 (jul 2017), 13 pages. <https://doi.org/10.1145/3072959.3073666>
- Chenxi Liu, Siqi Wang, Matthew Fisher, Deepali Aneja, and Alec Jacobson. 2024. 2D Neural Fields with Learned Discontinuities. *arXiv preprint arXiv:2408.00771* (2024).
- 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.
- Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.



- Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David I. W. Levin. 2024. Simplicitis: Mesh-Free, Geometry-Agnostic Elastic Simulation. *ACM Trans. Graph.* 43, 4, Article 117 (jul 2024), 11 pages. <https://doi.org/10.1145/3658184>
- Andrew B. Mor and Takeo Kanade. 2000. 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 (MICCAI '00)*. Springer-Verlag, Berlin, Heidelberg, 598–607.
- S. E. Mousavi, E. Grinspun, and N. Sukumar. 2011. Higher-order extended finite elements with harmonic enrichment functions for complex crack problems. *Internat. J. Numer. Methods Engrg.* 86, 4–5 (2011), 560–574. <https://doi.org/10.1002/nme.3098> <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.3098>
- 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.
- Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. 2021. On Self-Contact and Human Pose. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Octavi Obiols-Sales, Abhinav Vishnu, Nicholas Malaya, and Aparna Chandramowlishwaran. 2023. ADARNet: Deep Learning Predicts Adaptive Mesh Refinement. In *Proceedings of the 52nd International Conference on Parallel Processing*. 524–534.
- Shaowu Pan, Steven L Brunton, and J Nathan Kutz. 2022. Neural Implicit Flow: a mesh-agnostic dimensionality reduction paradigm of spatio-temporal data. *arXiv preprint arXiv:2204.03216* (2022).
- Vedant Puri, Aviral Prakash, Levent Burak Kara, and Yongjie Jessica Zhang. 2024. SNF-ROM: Projection-based nonlinear reduced order modeling with smooth neural fields. *arXiv preprint arXiv:2405.14890* (2024).
- Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 5301–5310.
- Bruce L. Reinhart. 1960. The winding number on two manifolds. *Annales de l'Institut Fourier* 10 (1960), 271–283. <https://api.semanticscholar.org/CorpusID:53078482>
- 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>.
- M. Shimrat. 1962. Algorithm 112: Position of point relative to polygon. *Commun. ACM* 5, 8 (aug 1962), 434. <https://doi.org/10.1145/368637.368653>
- Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*. 1–50.
- 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.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages. <https://doi.org/10.1145/3180491>
- Jacob Spainhour, David Gunderman, and Kenneth Weiss. 2024. Robust containment queries over collections of rational parametric curves via generalized winding numbers. <https://doi.org/10.48550/arXiv.2403.17371> [arXiv:2403.17371 \[cs.CG\]](https://arxiv.org/abs/2403.17371)
- Haozhe Su, Tao Xue, Chengguizi Han, and Mridul Aanjaneya. 2022. A-ULMPM: An Adaptively Updated Lagrangian Material Point Method for Efficient Physics Simulation without Numerical Fracture. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 325–341.
- Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, Vol. 33. 7537–7547.
- Yuan Yuan 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.
- Rui Xu, Zhiyang Dou, Ningna Wang, Shiqing Xin, Shuangmin Chen, Mingyan Jiang, Xiaohu Guo, Wenping Wang, and Changhe Tu. 2023. Globally Consistent Normal Orientation for Point Clouds by Regularizing the Winding-Number Field. *ACM Trans. Graph.* 42, 4, Article 111 (jul 2023), 15 pages. <https://doi.org/10.1145/3592129>
- Shuqi Yang, Xingzhe He, and Bo Zhu. 2020. Learning Physical Constraints with Neural Projections. In *Advances in Neural Information Processing Systems*.
- Ryan S Zesch, Vismay Modi, Shinjiro Sueda, and David IW Levin. 2023. Neural Collision Fields for Triangle Primitives. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.
- Wei Qun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, et al. 2019. AMReX: a framework for block-structured adaptive mesh refinement. *The Journal of Open Source Software* 4, 37 (2019), 1370.
- Qingnan Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. Mesh Arrangements for Solid Geometry. *ACM Transactions on Graphics (TOG)* 35, 4 (2016).
- Zeshun Zong, Xuan Li, Minchen Li, Maurizio M Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. 2023. Neural Stress Fields for Reduced-order Elastoplasticity and Fracture. *arXiv preprint arXiv:2310.17790* (2023).

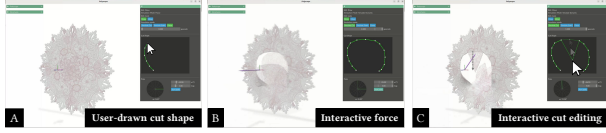


Fig. 5. Our method enables interactive drawing and editing of cut shapes with real-time previews. Users can drag polyline vertices to modify the cut shape during an ongoing simulation, in real-time. This is highly challenging for mesh-based methods due to the frequent need to constantly re-mesh the entire domain and has not been demonstrated.

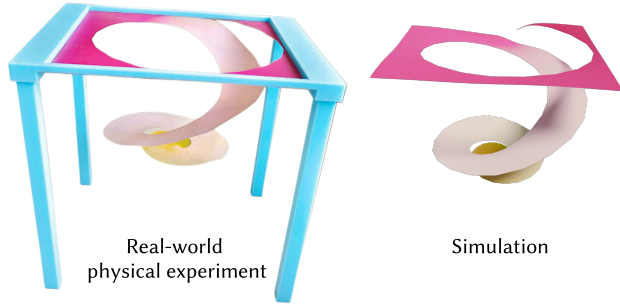


Fig. 6. Real-world Physical Experiment. We performed a qualitative comparison with a real-world experiment. A helical shape was cut and photographed as it sagged under gravity under the same boundary conditions. The simulated deformation qualitatively matches the real-world data.

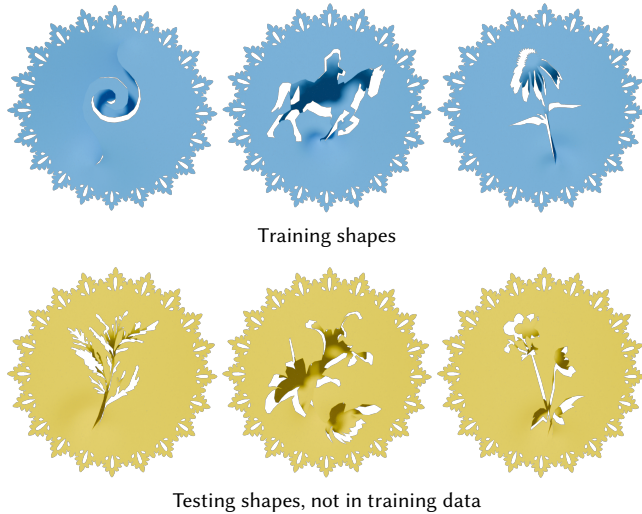


Fig. 7. We train our method with the three blue cut shapes (top), and test it on three significantly different cut shapes not included in training (bottom). We can still get reasonable deformation for the testing set.



Fig. 8. When we train our method on a clockwise helical cut and test with a counter-clockwise helical cut, the final deformation is different from the deformation tested on the training set. This is because the winding number distribution is significantly different from the training set.

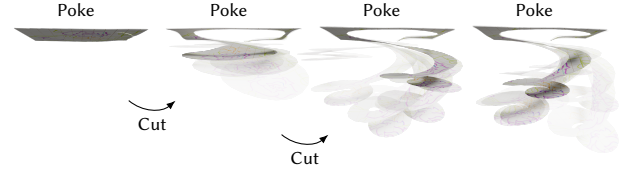


Fig. 9. Our method generalizes to loadings (forces) unseen during training. To demonstrate this, we applied a previously unseen force at various stages of the progressive cut, resulting in deformations that reflect the evolving geometry at each stage.

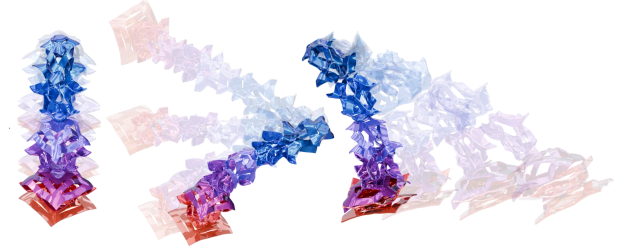


Fig. 10. We trained the basis on simulations of tugging on a *single* kirigami sheet. By connecting many instances of this kirigami sheet end to end, we obtain a reduced simulation of this “kirigami tower.”

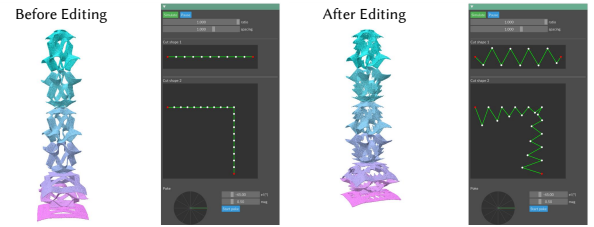


Fig. 11. We can interactively edit the cut shape for the kirigami. The cut lines of the kirigami change from a straight line to a ‘z’ shaped curve.

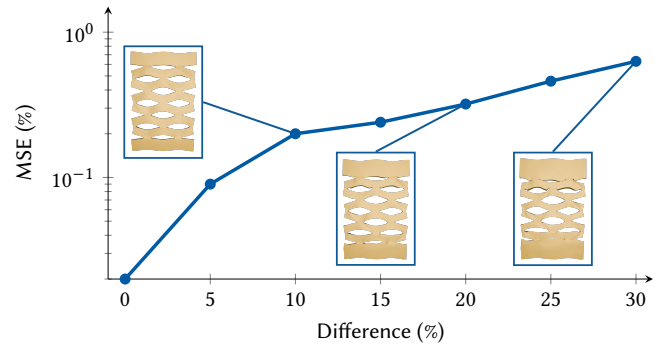


Fig. 12. We compute the percentage mean squared error on cut shapes that deviate progressively from the training set by scaling the cut gap. As the deviation increases, the reconstruction error also rises. When the difference is below 15%, no significant visual artifacts are observed. Beyond this threshold, while the discontinuity remains accurately captured, deformations begin to appear near the cut edges.

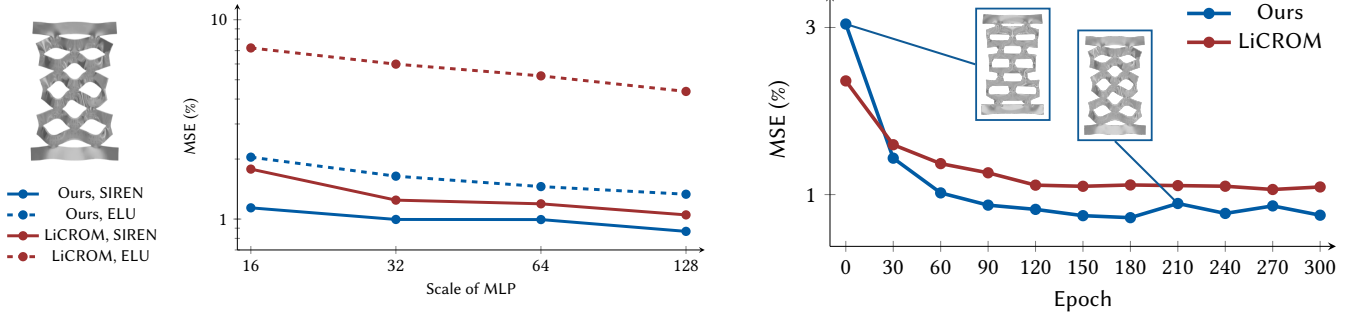


Fig. 13. We compared the percentage mean squared error (MSE) of our method with LiCROM [Chang et al. 2023] using different MLP scales and activation functions. Our method achieves significantly lower error across all settings, as shown on the left. Additionally, it demonstrates faster convergence during training, as shown on the right.

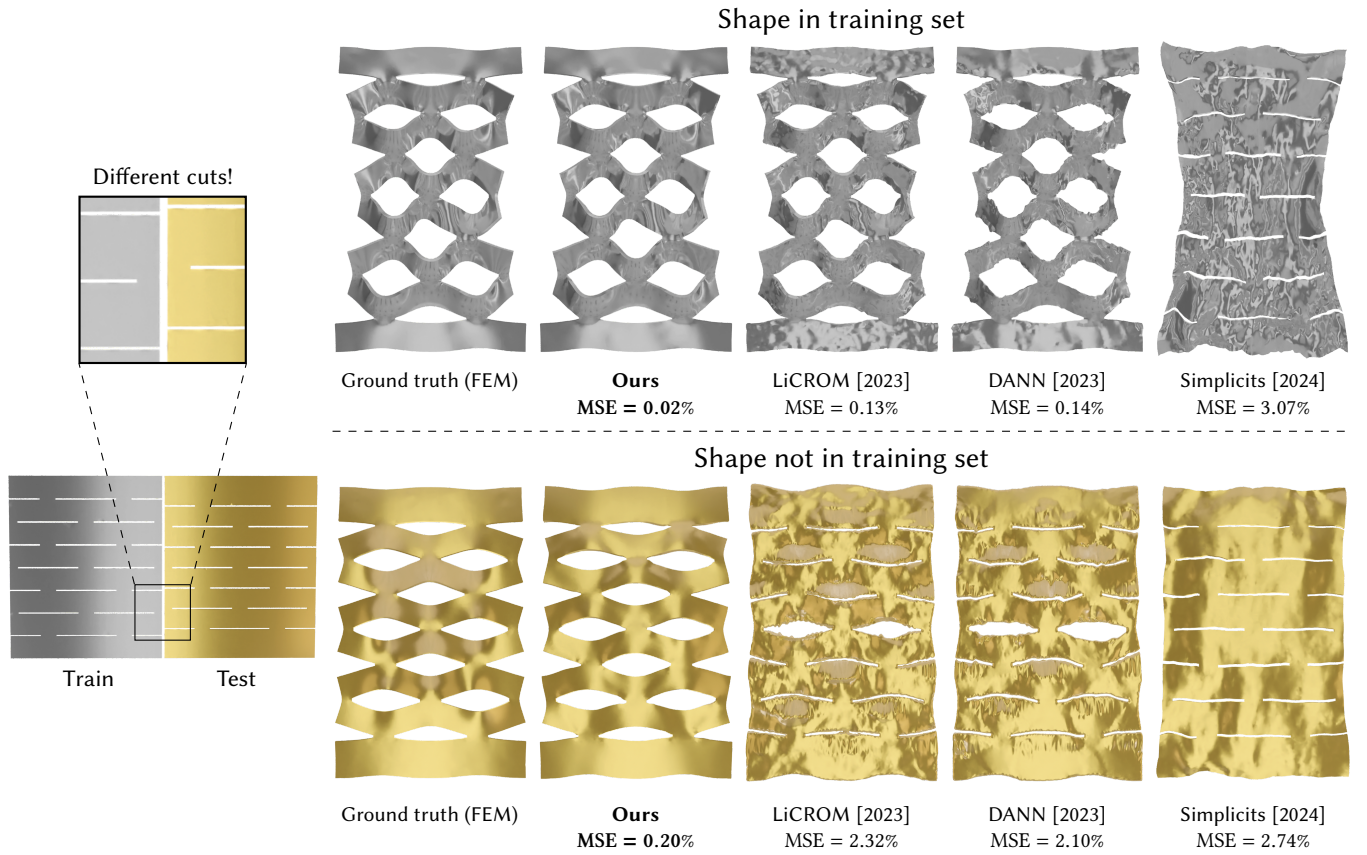


Fig. 14. We have compared our percentage reconstruction error with other methods. Our method has the lowest reconstruction error for both shapes in and not in the training set. Moreover, our method is the only one that can generate a visually reasonable result when tested on a cut that is different from the training set.