NeuralGarSim: Geometry-Agnostic Garment Simulation with Neural Fields

Arihant Gaur[1,2] Navami Kairanda[1] Christian Theobalt[1] Vladislav Golyanik[1]

[1]MPI for Informatics, Saarland Informatics Campus, Germany [2]Saarland University, Saarland Informatics Campus, Germany

Abstract. Most existing mesh-based methods for garment simulation suffer from sensitivities tied to mesh discretisation and resolution. Recent approaches such as NeuralClothSim employ continuous neural fields as a promising alternative, but remain constrained by a 2D curvilinear parameterisation that models a single cloth panel, limiting their ability to represent garments with multiple panels stitched together. This paper introduces **NeuralGarSim**, a quasistatic garment simulator formulated directly in 3D Euclidean space. Our method accepts diverse input representations—including distance fields, meshes, point clouds, and Gaussians—as undeformed garment states, making it compatible with off-the-shelf reconstruction frameworks without additional preprocessing. We then represent the garment deformation as a neural field and define a nonlinear Kirchhoff–Love shell model directly in $\mathbb{R}^3$ by applying tangential differential calculus on the garment surface. This ensures that the neural garment simulation is parameterisation-space-free and naturally extends to arbitrary topologies. By minimising a potential energy functional, our method learns a 3D neural deformation field that predicts physically consistent deformations across garments with multiple panels, seams, and holes. It supports multiple material models and trains 2–4× faster than its predecessor, NeuralClothSim, while retaining the continuous, consistent, and memory-adaptive behaviour[1].

1 Introduction

Physics-based cloth simulation is central to modern computer graphics, underpinning applications ranging from visual effects and games to digital fashion design. Most existing systems rely on explicit and discrete representations, such as finite differences or finite elements on triangle meshes and particle systems [27, 45, 46, 50, 52, 83]. Despite their maturity, these discretisations impose fundamental limitations: the predicted cloth dynamics remain highly sensitive to mesh resolution and initial triangulation, often producing mesh-dependent wrinkle patterns [37, 90]. Because meshes are not inherently adaptive, simulators must either remesh—breaking the differentiability crucial for inverse problems [52]—or rerun the simulation on a denser mesh whenever a higher target resolution is desired, inflating both compute and memory budgets.

[1] Project page: `https://4dqv.mpi-inf.mpg.de/NeuralGarSim/`

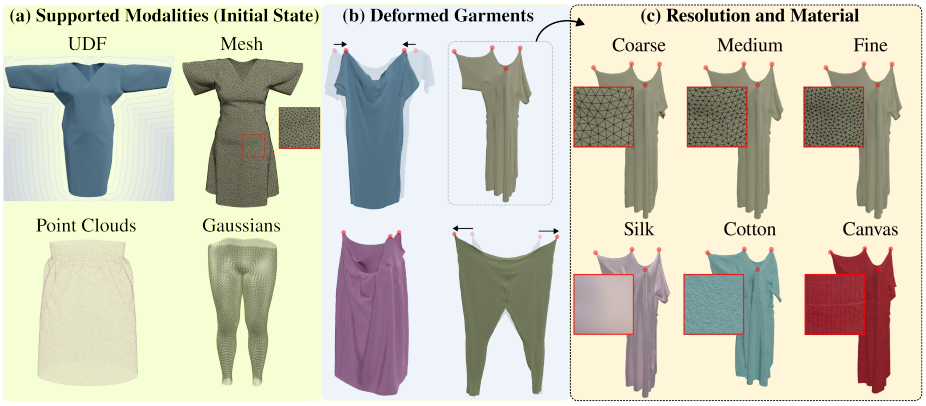


Fig. 1: NeuralGarSim is the first neural garment simulator to operate directly on the 3D garment surface without a 2D parameterisation, enabling continuous simulation of complex garments. (a): The reference garment surface can be provided as any one of UDF, triangle mesh, point cloud, or Gaussians—each supplying only the normal field and curvature tensor to the simulator. **(b):** NeuralGarSim produces physically consistent drape and fine-scale wrinkles across garments of varying topology **(c):** Wrinkle patterns remain consistent across different mesh resolutions (top); independently, swapping material parameters yields perceptually distinct drape characteristics (bottom).

Neural fields have emerged as a compelling paradigm for compact 3D scene representation [79, 88] and surface modelling [60, 85], typically realised as coordinate-based MLPs. Bridging these continuous representations with cloth simulation is therefore a highly promising direction. NeuralClothSim [37] is a notable step along this path. It trains a neural deformation field using the quasi-static[2] physics of a parametric Kirchhoff–Love shell [56]. The method minimises potential energy via analytic derivatives of the neural field, sidestepping the resolution dependence of traditional finite elements. However, it relies on classical parametric surface differential geometry [3, 87], *i.e.*, on the existence of a well-defined 2D parametric domain[3], which severely limits the complexity of garments it can handle.

The dependence on 2D parameterisation is particularly restrictive. Real garments comprise multiple panels, seams, and holes, making it difficult to construct a continuous parametric map whose derivatives remain analytic across boundaries. The classical finite element method (FEM) can cope with such discontinuities thanks to explicit meshes and local basis functions [64], but neural fields require global continuity and differentiability to evaluate energies via analytic derivatives [32, 40]. Hence, NeuralClothSim is limited to single-panel settings where suitable planar or periodic parameterisations are available, and extending neural cloth simulation to realistic garments demands a formulation that

[2] quasi-static denotes predicting the final equilibrium configuration under applied forces or boundary constraints.

[3] by parametric domain, we mean the flat cloth panel that serves as the 2D input to the neural deformation field in Kairanda et al. [37], not the network parameters.

dispenses with 2D parameter domains. Furthermore, prior neural and learning-based simulators [37, 68] are effectively tied to a single input representation, requiring a high-quality mesh as input. This prevents direct use with point clouds, 3D Gaussians [41] or other 3D representations from off-the-shelf reconstruction pipelines [63, 86].

We address these limitations with NeuralGarSim, a new method that reformulates cloth simulation entirely in 3D. The key enabler of our approach is *Tangential Differential Calculus (TDC)* [17, 72], a mathematical framework for defining differential operators directly on embedded surfaces and shapes without any local 2D parameterisations. By casting the nonlinear Kirchhoff–Love shell model into TDC, all strain quantities are computed directly in $\mathbb{R}^3$ from differential quantities of a neural deformation field, which has two immediate consequences. First, NeuralGarSim handles garments of arbitrary topology—multiple panels, seams and holes—because there is no need for a parametric map introducing discontinuities at boundaries. Second, the simulation only requires a surface normal field and curvature tensor at each sample point. Therefore, any 3D representation that can supply these quantities—distance fields, meshes, point clouds, or Gaussian splats—integrates directly without UV unwrapping or topology-specific preprocessing. This enables NeuralGarSim to accept garment assets from off-the-shelf reconstruction pipelines [41, 86] that prior neural simulators cannot utilise. To summarise, this paper advances the state of the art with the following **technical contributions:**

- We introduce NeuralGarSim, a parameterisation-free and continuous neural quasi-static cloth simulator operating directly on embedded surfaces, avoiding the need for a 2D parametric domain (Sec. 4).
- We cast Kirchhoff–Love thin shells into the TDC formalism, enabling seamless support for multiple geometric representations under a unified physical model (Sec. 4.1).

In experiments, NeuralGarSim produces discretisation-consistent wrinkle patterns across different mesh resolutions, supports multiple material models, and trains 2–4× faster than NeuralClothSim (Sec. 5). Fig. 1 highlights representative results. Our project page provides further visualisations, video and the code.

2 Related Work

Classical Cloth Simulation has matured over several decades [1, 2, 9, 24, 33, 47, 78, 82], converging on a common recipe: discretise the cloth with FEM [19, 64], timestep implicitly for stability [1, 45], and resolve contact and collisions robustly [46, 59, 66, 77]. Constitutive models capture fabric-dependent stress response [13, 84], while recent differentiable solvers [50, 52] expose gradients that aid tasks such as reconstruction [39, 49]. Despite steady progress, FEM methods [52] remain sensitive to mesh discretisation, require extremely fine meshes for fidelity, and are expensive to run or integrate with modern learning systems, motivating alternative formulations relying on neural fields [37].

Neural Cloth Simulation. Another line of work replaces traditional solvers with neural models [6, 7, 11, 21, 22, 53, 70]. PBNS [6] tackles static draping on virtual characters through self-supervised losses that enforce physics without simulation ground truth, and later variants extend the idea to garment dynamics [7, 70]. The garments in these methods are skinned to the human body, and deformations are driven primarily by body shape and pose. While effective for body-driven apparel, these networks cannot serve as general cloth simulators and inherit the discretisation limitations of classical mesh-based simulators. Pfaff *et al*. [68] instead learn general-purpose dynamics with graph neural networks but rely on synthetic training data [64]; HOOD [22] and ContourCraft [21] retain the same discrete representation while incorporating physics-based supervision. NeuralClothSim [36, 37] departs from meshes by representing deformation as a continuous neural field and evaluating energy analytically, thereby mitigating discretisation sensitivity. Yet it requires a 2D parametric domain: single panels are supported, but seams, cuts, and multi-panel garments remain out of reach. This limitation motivates our parameterisation-free formulation. Thin-Shell-SfT [36] builds on the same neural thin-shell formulation for monocular non-rigid surface tracking, whereas ∂^∞-Grid [38] accelerates neural differential-equation solving with differentiable feature grids. While these works address tracking and solver efficiency, we focus on parameterisation-free garment simulation across meshes, UDFs, point clouds, and 3D Gaussians.

3D Surface Representations for Garments. Neural implicit fields represent surfaces as zero-level sets of learned continuous functions [79, 88]. They have been adopted for geometry representation [12, 20, 55, 76], reconstruction [51, 85, 86], and geometric processing [60, 65, 89]. At the same time, explicit 3D representations such as 3D Gaussian splatting [41] and surface-aligned variants [25] are employed for dynamic garment modelling, for instance, MD-Splatting [18] and Thin-Shell-SfT [36]. GaussianGarments approach [69] further reconstructs simulation-ready clothing from multi-view video. In garments, neural and implicit fields have mainly supported representation, data-driven draping, and disentanglement: ULNeF [71] targets virtual try-on, Implicit Untangling [10] resolves garment-layer intersections, and DIG [48] and DrapeNet [15] model body-conditioned draping or garment generation. Compared to these methods, differentiable unsigned distance fields (UDFs) [20] improve open-surface regularity and make normals and curvature more reliable. Our contribution is complementary and multimodal: NeuralGarSim uses the differential quantities supplied by UDFs, meshes, point clouds, or 3D Gaussians as the simulation domain for a physics-based Kirchhoff–Love shell model.

Tangential Differential Calculus for Shells. TDC provides a coordinate-free framework for defining differential operators on embedded surfaces [16, 17], enabling shell and plate models [16, 17, 29–31, 73] to be formulated directly in $\mathbb{R}^3$ space. Prior work has applied TDC to Kirchhoff-Love [72] and related shell models, albeit with linear strain assumptions and non-neural classical FEM discretisations. *Our work is the first to couple TDC with neural fields, enabling*

fully nonlinear Kirchhoff-Love (KL) shell simulation directly in 3D space and avoiding the parametric domains required by earlier KL cloth simulators [13,37,80].

3 Preliminaries

We model garments as thin shells governed by the Kirchhoff–Love (KL) theory. We summarise the KL formulation (Sec. 3.1), review NeuralClothSim [37] as the closest prior work (Sec. 3.2), and outline Tangential Differential Calculus (TDC) [17, 72] background for the differential geometry of surfaces (Sec. 3.3).

3.1 The Kirchhoff-Love Shell Theory

A *thin shell* is a volumetric object whose thickness h is much smaller than its characteristic in-plane dimension L (*i.e.*, $h \ll L$). Under KL theory, 3D continuum mechanics is reduced to 2D kinematics posed on the *mid-surface* $\Gamma \subset \mathbb{R}^3$. Our formulation for shell deformation relies on the classical KL hypothesis: Straight lines that are initially normal to the mid-surface remain straight and normal to the mid-surface; they preserve their length during deformation, so the shell thickness stays constant.

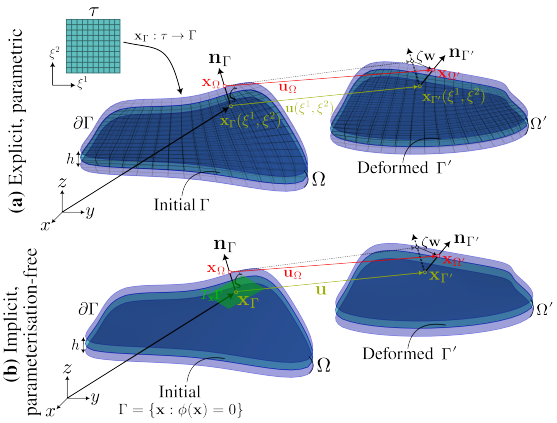


Fig. 2: Explicit vs implicit thin-shell modelling. (a): NeuralClothSim [37] parameterises the mid-surface via an explicit map $\mathbf{x}_\Gamma(\boldsymbol{\xi}) : \tau \subset \mathbb{R}^2 \to \Gamma$. **(b):** NeuralGarSim represents instead Γ as a zero level-set of an implicit field ϕ. Under the Kirchhoff-Love thin-shell assumption, a shell point $\mathbf{x}_\Omega$ is specified by its mid-surface anchor $\mathbf{x}_\Gamma$, thickness coordinate ζ, and normal $\mathbf{n}_\Gamma$, providing the kinematic basis for formulating the deformation field $\mathbf{u}_\Omega$.

Any point $\mathbf{x}_\Omega$ within the shell Ω is specified by its mid-surface projection $\mathbf{x}_\Gamma$ and a normal offset $\zeta \in [-h/2, h/2]$, as $\mathbf{x}_\Omega = \mathbf{x}_\Gamma + \zeta\,\mathbf{n}_\Gamma$ (see Fig. 2). Under the KL hypothesis, the 3D displacement field $\mathbf{u}_\Omega$ decomposes linearly through the thickness (thus eliminating any transverse shear) during shell deformation:

$$\mathbf{u}_\Omega(\mathbf{x}_\Gamma, \zeta) = \mathbf{u}(\mathbf{x}_\Gamma) + \zeta\,\mathbf{w}(\mathbf{x}_\Gamma), \tag{1}$$

where $\mathbf{u} := \mathbf{x}_{\Gamma'} - \mathbf{x}_\Gamma$ is the mid-surface displacement with the deformed midsurface Γ', and $\mathbf{w} := \mathbf{n}_{\Gamma'} - \mathbf{n}_\Gamma$ captures the change in the surface normal orientation.

3.2 NeuralClothSim [37]

NeuralClothSim parameterises the mid-surface via a 2D *curvilinear parametric domain* $\tau \subset \mathbb{R}^2$; see Fig. 2-(left). It then maps the coordinate samples $\boldsymbol{\xi} =$

$(\xi^1, \xi^2) \in \tau$ to 3D deformation $\mathbf{u}(\boldsymbol{\xi}; \Theta)$ using a neural field implemented as a coordinate-based multilayer perceptron (MLP) with weights Θ. The deformed midsurface $\mathbf{x}_{\Gamma'}(\boldsymbol{\xi})$ reads as

$$\mathbf{x}_{\Gamma'}(\boldsymbol{\xi}) = \mathbf{x}_{\Gamma}(\boldsymbol{\xi}) + \mathbf{u}(\boldsymbol{\xi}; \Theta), \tag{2}$$

where $\mathbf{x}_{\Gamma}(\boldsymbol{\xi})$ is an analytical surface defined on the parametric domain.

Parametric Energy Minimisation. The internal hyperelastic energy $\Psi(\boldsymbol{\epsilon}, \boldsymbol{\kappa}; \boldsymbol{\Phi})$ depends on membrane strain $\boldsymbol{\epsilon}$, bending strain $\boldsymbol{\kappa}$, and material parameters $\boldsymbol{\Phi}$. NeuralClothSim evaluates $\boldsymbol{\epsilon}(\boldsymbol{\xi}; \Theta)$ and $\boldsymbol{\kappa}(\boldsymbol{\xi}; \Theta)$ via classical parametric differential geometry [3, 74, 75], *i.e.*, first and second fundamental forms of both reference and deformed surfaces are constructed from the partial derivatives of $\mathbf{x}_{\Gamma}(\boldsymbol{\xi})$ and $\mathbf{u}(\boldsymbol{\xi}; \Theta)$ with respect to ξ^{α}. The equilibrium configuration is found by minimising the total potential energy over τ:

$$\min_{\Theta} \int_{\tau} \Psi\big(\boldsymbol{\epsilon}(\boldsymbol{\xi}; \Theta),\, \boldsymbol{\kappa}(\boldsymbol{\xi}; \Theta); \boldsymbol{\Phi}\big)\, d\boldsymbol{\xi} \;-\; \int_{\tau} \mathbf{f} \cdot \mathbf{u}(\boldsymbol{\xi}; \Theta)\, d\boldsymbol{\xi}. \tag{3}$$

Here, $\boldsymbol{\Phi}$ denotes fixed material parameters, while optimisation is performed only over the deformation-field weights Θ. We refer to Appendix C for the explicit expressions. The parametric formulation requires a continuous 2D domain, which is difficult to obtain for multi-panel garments with seams and holes without introducing chart boundaries or stitching artifacts; Sec. 5.6 empirically studies this limitation.

3.3 Tangential Differential Calculus (TDC)

TDC [17, 28, 31, 72] is a comprehensive mathematical framework for defining differential operators directly on embedded manifolds $\Gamma \subset \mathbb{R}^3$, circumventing the need for curvilinear coordinates, metric tensors, and Christoffel symbols required by classical parametric differential geometry [42, 57]. Below, we outline three operators that underpin the strain computation of our method.

Projection Operator. Suppose $\mathbf{P}$ is the orthogonal *projector* onto the tangent space $T_{\mathbf{x}}\Gamma$ at point $\mathbf{x}$, it reads as

$$\mathbf{P} = \mathbb{I} - \mathbf{n}_{\Gamma} \otimes \mathbf{n}_{\Gamma}, \tag{4}$$

where $\mathbf{n}_{\Gamma}$ is the surface normal, $\mathbb{I}$ is the 3×3 identity tensor, and "$\otimes$" denotes the outer vector product. Note that $\mathbf{P}$ projects any vector $\mathbf{v}$ onto $T_{\mathbf{x}}\Gamma$ so that $\mathbf{v}_t = \mathbf{P} \cdot \mathbf{v}$ is tangential (*i.e.*, $\mathbf{v}_t \cdot \mathbf{n}_{\Gamma} = 0$).

Tangential and Directional Gradients. For a scalar function $v : \Gamma \to \mathbb{R}$, its *tangential gradient* $\nabla_{\Gamma} v$ is given by the surface projection of its standard 3D gradient:

$$\nabla_{\Gamma} v(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \cdot \nabla \tilde{v}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{5}$$

where $\tilde{v}$ is any smooth extension of v into a neighbourhood of Γ. The tangential gradient, thus, lies in the tangent space: $\nabla_{\Gamma} v(\mathbf{x}) \in T_{\mathbf{x}}\Gamma$. For a vector field, the *directional gradient* is formed by applying tangential gradients component-wise;

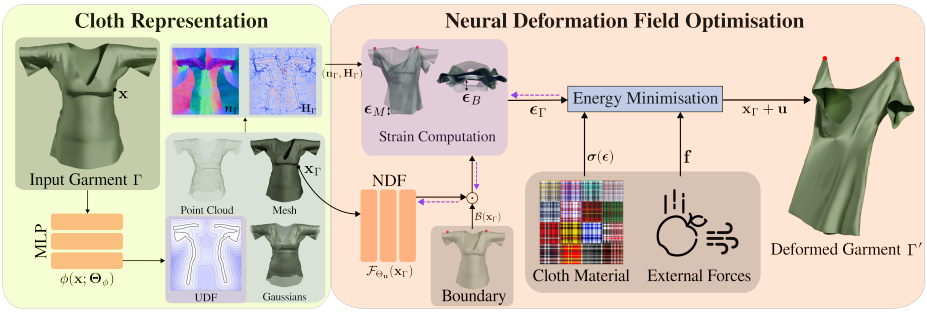


Fig. 3: Overview of NeuralGarSim. Left (Sec. 4.1): The input garment surface Γ is provided in one of four modalities—UDF, point cloud, mesh or 3D Gaussians. We initially fit UDF to an MLP $\phi(\mathbf{x};\Theta_\phi)$ with weights Θ_ϕ. The derivatives of the UDF supply geometric priors, i.e., normals $\mathbf{n}_\Gamma$ and the Weingarten map $\mathbf{H}$. For all other modalities, these priors are extracted directly from the input geometry. **Right (Sec. 4.2)**: The neural deformation field $\mathbf{u}(\mathbf{x}_\Gamma;\Theta_\mathbf{u})$ with weights $\Theta_\mathbf{u}$ applies boundary conditions $\mathcal{B}(\mathbf{x}_\Gamma)$ and combines with the geometric priors to compute membrane $\boldsymbol{\epsilon}_M$ and bending strains $\boldsymbol{\epsilon}_B$ (Eq. (14)) directly in $\mathbb{R}^3$. Minimising the total potential energy Π (Eq. (15)) optimises $\Theta_\mathbf{u}$, balancing elastic terms against external forces $\mathbf{f}$ to recover the deformed surface $\mathbf{x}_{\Gamma'} = \mathbf{x}_\Gamma + \mathbf{u}$.

tensor-valued extensions are given in Appendix A. In our strain formulation, the resulting directional membrane and bending strains are projected onto the tangent plane using $\mathbf{P}$.

Weingarten Map and Curvature. Surface curvature is captured by the *Weingarten map* $\mathbf{H}$, a symmetric in-plane tensor written as:

$$\mathbf{H} = \nabla_\Gamma^{\text{dir}} \mathbf{n}_\Gamma, \tag{6}$$

where $\nabla_\Gamma^{\text{dir}}$ denotes the directional surface gradient. $\mathbf{H}$ is required for bending strain computation in our method.

4 Our NeuralGarSim Approach

NeuralGarSim is a quasistatic neural cloth simulator that operates directly on reference geometries embedded in $\mathbb{R}^3$; see Fig. 3. Given an input garment $\mathcal{M}_\Gamma$ approximating the reference surface $\Gamma \subset \mathbb{R}^3$, we seek a neural deformation field (NDF) $\mathbf{u}(\cdot;\Theta_\mathbf{u})$ with weights $\Theta_\mathbf{u}$ that produces the simulated surface Γ' under updated boundary conditions $\mathcal{B}$ and external forces $\mathbf{f}$. The method first extracts geometric operators $(\mathbf{n}_\Gamma, \mathbf{P}, \mathbf{H})$ from the input representation (Sec. 4.1) and learns $\mathbf{u}(\cdot;\Theta_\mathbf{u})$ by minimising the total potential energy under the prescribed boundaries and forces (Sec. 4.2). TDC provides the coordinate-free bridge to Kirchhoff–Love shell mechanics: it does not require 2D parameterisation, but when a smooth parameterisation exists, the same geometric quantities are recovered.

4.1 Garment Representation

We take the undeformed garment surface Γ as input. The surface can be represented in different modalities $\mathcal{M}_\Gamma$, such as discrete meshes, point clouds, 3D Gaussians, or implicit level sets. We prefer level-set representations, as the TDC-based formulation is well-explored [31, 72] for implicits. Our simulation stage (see Sec. 4.2) requires *only* a normal field $\mathbf{n}_\Gamma$ and a Weingarten map $\mathbf{H}$ at each surface point. These are extracted differently for each input modality.

Garment as a Neural Level Set. Garments are *open* surfaces with complex seams and holes; we represent the undeformed garment surface Γ implicitly as the zero level of a differentiable UDF (DUDF) $\phi : \mathbb{R}^3 \to \mathbb{R}^+$ [20]:

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) = 0\}, \qquad \Omega = \left\{\mathbf{x} \in \mathbb{R}^3 : |\phi(\mathbf{x})| \leq \frac{h}{2}\right\}. \tag{7}$$

ϕ must be differentiable to compute higher-order differential geometric quantities. Therefore, we parametrise the distance field with an MLP $\phi(\mathbf{x}; \Theta_\phi)$ with weights Θ_ϕ and supervise it using the modified Eikonal problem of DUDF [20], which ensures that the distance field is differentiable with *hyperbolic scaling* (details in Appendix E). With the learned $\phi(\mathbf{x}; \Theta_\phi)$, the unit surface normal $\mathbf{n}_\Gamma$ follows from the field gradient, from which the TDC projection operator $\mathbf{P}$ and Weingarten map $\mathbf{H}$ follow via Eqs. (4) and (6).

Garment as a Discrete Representation. If the garment is provided as a *triangle mesh*, we compute $\mathbf{n}_\Gamma$ as the area-weighted vertex normal, and $\mathbf{H}$ is derived via the cotangent-weighted Laplacian [61]. For a *point cloud*, $\mathbf{n}_\Gamma$ is estimated by principal component analysis (PCA) on a local neighbourhood [35]. $\mathbf{n}_\Gamma$ is set as the axis of smallest variance extracted from the covariance matrix of the 3D Gaussian. For sufficiently dense meshes, point clouds, and 3D Gaussians, $\mathbf{H}$ can be set to zero without affecting the thin-shell energy approximation.

4.2 Garment Deformation

We model garment deformation with an NDF $\mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u}) : \Gamma \to \mathbb{R}^3$ that maps each reference point $\mathbf{x}_\Gamma$ to its displaced location $\mathbf{x}_{\Gamma'}$ in 3D. The field is parametrised by a coordinate-based MLP $\mathcal{F}_{\Theta_\mathbf{u}}$ with Siren activations [76], whose smooth periodic activations support the higher-order derivatives required by the bending strain. Unlike NeuralClothSim, which operates on parametric coordinates (Sec. 3.2), our NDF $\mathbf{u}$ takes the 3D coordinates on Γ as input, eliminating any dependency on 2D parameterisation. Dirichlet (pinning) conditions can be imposed as hard constraints by modulating the NDF output, as in Refs. [54, 58]. In particular, for a constrained boundary subset $\partial\Gamma \subset \Gamma$, we introduce a scalar distance function $\mathcal{B}(\mathbf{x})$ that is zero on $\partial\Gamma$, and one elsewhere:

$$\mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u}) = \mathcal{F}_{\Theta_\mathbf{u}}(\mathbf{x}_\Gamma) \cdot \mathcal{B}(\mathbf{x}_\Gamma). \tag{8}$$

Boundary conditions are, therefore, specified directly in $\mathbb{R}^3$, avoiding the 2D points required by parametric formulations [37]. Moreover, periodic boundary

conditions as in NeuralClothSim are not required, as the surface is represented implicitly. Once $\mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u})$ is trained, inference can be performed at any spatial resolution supported by the network capacity of the NDF. Next, we describe how to train the NDF to generate quasistatic cloth deformation. Detailed algorithms for NDF training and inference are provided in Algorithms 1 and 2 in Appendix K.

Strain Formulation. The core of our method is a nonlinear KL shell model formulated entirely in $\mathbb{R}^3$ using TDC, following classical shell mechanics and TDC-based shell formulations [31, 34, 72, 74]. Following the KL kinematic assumption (1), the 3D displacement vector $\mathbf{u}_\Omega$ at any point in the shell continuum is thus:

$$\mathbf{u}_\Omega(\mathbf{x}_\Gamma, \zeta; \Theta_\mathbf{u}) = \mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u}) + \zeta \mathbf{w}(\mathbf{x}_\Gamma; \Theta_\mathbf{u}), \tag{9}$$

where $\mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u})$ is the midsurface deformation and $\mathbf{w}(\mathbf{x}_\Gamma; \Theta_\mathbf{u})$ is the nonlinear director difference vector encoding orientation changes. For brevity, we drop the explicit dependence on $\mathbf{x}_\Gamma, \zeta$, and $\Theta_\mathbf{u}$ for further quantities in what follows. Because the KL model eliminates transverse shear, the director difference can be written as:

$$\mathbf{w} = \mathbf{H} \cdot \mathbf{u} - \nabla_\Gamma(\mathbf{u} \cdot \mathbf{n}_\Gamma). \tag{10}$$

To capture the large deformations and rotations happening in cloth, we employ the full Green-Lagrange strain tensor $\boldsymbol{\epsilon}^{\text{dir}}$, defined from the 3D directional deformation gradient $\nabla^{\text{dir}}_\Gamma \mathbf{u}_\Omega$ [34]. Neglecting the $\mathcal{O}(\zeta^2)$ term yields the standard split into membrane and bending strains:

$$\boldsymbol{\epsilon}^{\text{dir}} \approx \boldsymbol{\epsilon}^{\text{dir}}_M + \zeta \boldsymbol{\epsilon}^{\text{dir}}_B, \tag{11}$$

where the *membrane strain tensor* capturing stretching reads

$$\boldsymbol{\epsilon}^{\text{dir}}_M = \frac{1}{2}\left(\mathbf{F}_u + \mathbf{F}_u^\mathsf{T} + \mathbf{F}_u^\mathsf{T}\mathbf{F}_u\right), \tag{12}$$

and the *bending strain tensor* capturing curvature changes reads

$$\boldsymbol{\epsilon}^{\text{dir}}_B = \frac{1}{2}\left(\mathbf{F}_w + \mathbf{F}_w^\mathsf{T} + \mathbf{F}_u^\mathsf{T}\mathbf{F}_w + \mathbf{F}_w^\mathsf{T}\mathbf{F}_u\right), \tag{13}$$

with $\mathbf{F}_u = \nabla^{\text{dir}}_\Gamma \mathbf{u}$ and $\mathbf{F}_w = \nabla^{\text{dir}}_\Gamma \mathbf{w}$ denoting the directional gradients of the midsurface displacement and orientation, respectively. The Green–Lagrange expansion leading to (11) is detailed in Appendix B. *Both tensors remain nonlinear due to the quadratic terms.* As they are defined in $\mathbb{R}^{3\times3}$, we project them onto the tangent plane using $\mathbf{P}$ to arrive at the in-plane strains:

$$\boldsymbol{\epsilon}_M = \mathbf{P} \cdot \boldsymbol{\epsilon}^{\text{dir}}_M \cdot \mathbf{P} \quad \text{and} \quad \boldsymbol{\epsilon}_B = \mathbf{P} \cdot \boldsymbol{\epsilon}^{\text{dir}}_B \cdot \mathbf{P}. \tag{14}$$

In the above formulation, the bending strain $\boldsymbol{\epsilon}_B$ relies on the second-order derivatives of $\mathbf{u}$. This formulation aligns with our network's continuity (C^∞), while remaining fully non-linear and parameterisation-free.

4.3 Cloth Material Models and Energy Minimisation

We find the equilibrium configuration of cloth encoded in $\Theta^*_{\mathbf{u}}$ by minimising the total potential energy $\Pi = \Pi_{\text{int}} + \Pi_{\text{ext}}$ directly on the 3D reference manifold Γ. Here, the internal elastic energy Π_{int} depends on the chosen material and the geometric strains $(\boldsymbol{\epsilon}_M, \boldsymbol{\epsilon}_B)$. The external potential Π_{ext} is given by loads such as gravity, *e.g.*, $\mathbf{f} = \rho\mathbf{g}$, where ρ is the surface mass density and $\mathbf{g}$ is the gravitational acceleration, leading to:

$$\Theta^*_{\mathbf{u}} = \arg\min_{\Theta_{\mathbf{u}}} \Pi[\mathbf{u}(\mathbf{x}_\Gamma; \Theta_{\mathbf{u}})], \quad \text{with} \quad \Pi_{\text{ext}} = -\int_\Gamma \mathbf{f} \cdot \mathbf{u}\, d\mathbf{x}_\Gamma. \tag{15}$$

Our formulation is orthogonal to the choice of constitutive material law as long as it can be written as a material energy density $\Psi(\boldsymbol{\epsilon}_M, \boldsymbol{\epsilon}_B; \boldsymbol{\Phi})$. We implement two material models for modelling Π_{int}: the isotropic model is used for controlled benchmark validation, while the orthotropic StVK model captures direction-dependent fabric stiffness for garment examples.

Isotropic Linear Elasticity. For simple simulation and validation (Belytschko benchmarks; see Sec. 5.3), we employ an isotropic model governed by Young's modulus E and Poisson's ratio ν. Mapping the projected strains to in-plane stress $\boldsymbol{\sigma}$ and pre-integrating through the thickness h analytically, yields the membrane resultant $\tilde{\mathbf{n}}(\boldsymbol{\epsilon}_M)$ and bending resultant $\mathbf{m}(\boldsymbol{\epsilon}_B)$. After the thickness integration, the internal energy on the mid-surface can be evaluated as:

$$\Pi_{\text{int}} = \int_\Gamma (\tilde{\mathbf{n}} : \boldsymbol{\epsilon}_M + \mathbf{m} : \boldsymbol{\epsilon}_B)\, d\mathbf{x}_\Gamma. \tag{16}$$

Here, ":" denotes double contraction between second-order tensors.

Orthotropic StVK Material Model. In realistic garment simulation, fabrics exhibit direction-dependent stiffness. We implement the orthotropic St.Ṽenant–Kirchhoff (StVK) model defined by stiffness constants $\boldsymbol{\Phi} = \{a_{11}, a_{22}, a_{12}, G_{12}\}$ [14] applied to strain components evaluated in a local material tangent frame. As this energy density Ψ_{StVK} is non-linear across the shell thickness, we evaluate Π_{int} by integrating explicitly through the thickness via Simpson quadrature:

$$\Pi_{\text{int}} = \int_\Gamma \int_{-h/2}^{h/2} \Psi_{\text{StVK}}(\boldsymbol{\epsilon}_M + \zeta\boldsymbol{\epsilon}_B; \boldsymbol{\Phi})\, d\zeta\, d\mathbf{x}_\Gamma. \tag{17}$$

Optimisation. We approximate the continuous surface integral in (15) via Monte Carlo quadrature. At each iteration, we uniformly sample points $\{\mathbf{x}_i\}_i$ on $\mathcal{M}_\Gamma$: from mesh faces in the case of mesh inputs, from the zero level set of ϕ for neural UDFs, and from mean positions for point clouds and 3D Gaussians. We then query the fixed geometric priors from the input representation (Sec. 4.1), and evaluate the displacements $\mathbf{u}(\mathbf{x}_i; \Theta_{\mathbf{u}})$. Depending on the chosen material, we compute the corresponding internal energy (Eq. (16) or Eq. (17)) alongside the external forces, and train the NDF end-to-end to minimise Π. For visualisation, we extract a mesh $\mathcal{M}_{\Gamma'}$ by evaluating $\mathbf{u}$ at the vertices of the reference $\mathcal{M}_\Gamma$ and displacing them accordingly (Fig. 3-(right)).

Table 1: Performance comparison with NeuralClothSim [37]. Training times correspond to results in Tab. 2 (top) and cloth experiments (bottom). Consistent speedups of up to 4× are observed.

Dataset	NeuralClothSim	NeuralGarSim
Square Plate	13m 1s	**6m 26s**
Scordelis-Lo Roof	4h 49m	**2h 21m**
Pinched Cylinder	56m 4s	**27m 17s**
Napkin Moving Handles	6m 9s	**1m 35s**
Skirt Twist	21m 45s	**7m 14s**
Sleeve Buckle	11m 3s	**5m 55s**
Average	1h 6m 10s	**31m 35s**

Table 2: Maximum displacement at the target point for each Belytschko test. "*": *Guo et al.* use different materials matching corresponding analytical values.

Method	Square Plate	Scordelis-Lo	Pinched Cyl.
Analytical [5, 81]	0.487	0.3024	1.825e-5
Guo et al. [26]	2.566*	(N/A)	(N/A)
Bastek et al. [4]	(N/A)	0.297	(N/A)
NeuralClothSim [37]	**0.487**	0.2965	**1.825e-5**
NeuralGarSim (Ours)	**0.487**	**0.2987**	1.820e-5

5 Experiments

5.1 Implementation Details

Garment Deformation. The deformation field $\mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u})$ maps each reference point to the deformed surface point $\mathbf{x}_{\Gamma'} = \mathbf{x}_\Gamma + \mathbf{u}(\mathbf{x}_\Gamma; \Theta_\mathbf{u})$. We use five hidden layers of 256 units. We use the Adam optimiser [43] for training. We start from a learning rate of 10^{-4} with an exponential decay schedule [37]. To evaluate the energy in Eq. (15), we uniformly sample 1000 points on the cloth mesh $\mathcal{M}_\Gamma$. For linear isotropic material, we set the parameters to representative textile values, $\boldsymbol{\Phi} = (E, \nu, \rho, h) = (5000\,\mathrm{Pa}, 0.25, 0.144\,\mathrm{kg\,m^{-2}}, 1.2\,\mathrm{mm})$. For the orthotropic StVK material model, we use cotton and silk stiffness constants following fabric tables from Refs. [13, 62]. Each simulation is trained for the number of optimisation steps in the range $[2000 - 5000]$, depending on garment complexity.

NeuralGarSim is implemented in PyTorch [67]. For DUDF inputs, neural field fitting and NDF optimisation each complete in one to 15 minutes on a single NVIDIA H100, for a total of three to 17 minutes. For mesh, point cloud, and 3D Gaussian inputs, the end-to-end pipeline is shorter because the DUDF fitting stage is skipped. Consequently, only the shared NDF optimisation stage is run, which completes in 1 to 8 minutes depending on garment complexity.

5.2 Baselines and Datasets

We compare NeuralGarSim with NeuralClothSim [37], DiffCloth [50], DiffARCSim [52,64], and MeshGraphNets [68]. MeshGraphNets is a learned autoregressive dynamics surrogate rather than a quasistatic energy minimiser, so we report it in Appendix G.1. Our experiments cover classical Belytschko benchmarks [5, 81] and multi-panel garments, including T-shirts, dresses, and jackets [8, 44].

5.3 Classical Benchmarks

Validation. We first validate the physical accuracy of our method for estimating quasistatic deformation. To this end, we reproduce the Belytschko benchmark

experiments [5, 23], which test different modes of shell deformation. We represent the benchmark reference geometries as triangle meshes and apply the prescribed forces and boundary conditions for the square plate (membrane-dominated), Scordelis-Lo roof (combined membrane/bending), and pinched cylinder (bending-dominated) tests, measuring the maximum displacement at the target point. As shown in Tab. 2, our method closely matches the analytical solutions, confirming that the TDC-based strain formulation captures both membrane and bending modes. Details are provided in Appendix F.

Speed. We compare training time against NeuralClothSim in Tab. 1. Thanks to the fully 3D strain formulation and the avoidance of parametrisation space, NeuralGarSim consistently trains faster, with speed-ups of roughly 2–4×. The gain comes from evaluating all differential quantities directly on embedded samples. We therefore avoid constructing and differentiating through Christoffel-symbol terms on a 2D parametric domain as in [37], while still optimising the same quasi-static energy objective with comparable Monte Carlo sampling.

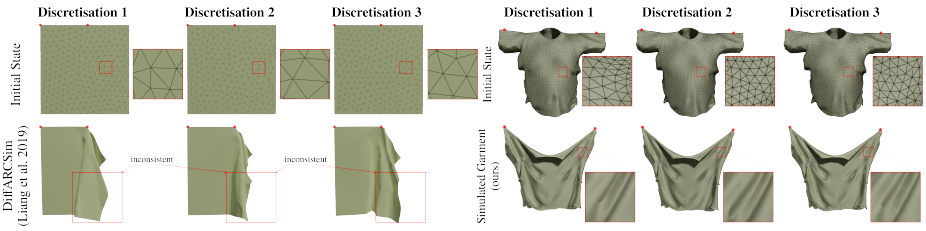


Fig. 4: Given different input discretisations (top row) with an identical simulation scenario, our method produces consistent folds and wrinkles due to continuous NDF. On the other hand, the mesh-based simulator, DiffARCSim [52], is sensitive to initial discretisation, despite enabling adaptive remeshing.

5.4 Discretisation Independence

Setup. The key advantage of NeuralGarSim inherited from NeuralClothSim [37] is that the deformation is represented continuously (as a coordinate-based MLP; see Sec. 4.2) and, therefore, the simulation quality is independent of the input mesh discretisation $\mathcal{M}_\Gamma$. To demonstrate this, for each garment in the experiment, we generate three alternative discretisations via isotropic remeshing; see Fig. 4-(top row). Each mesh preserves the overall geometry and pinned boundary set. We then run the simulations under identical forces and the same pinned garment regions for ours, DiffARCSim [52] and DiffCloth [50]. Note that DiffARCSim is adaptive (*i.e.*, it supports adaptive remeshing), whereas DiffCloth is not and runs at the fixed initial resolution.

Quantitative Metric. We report full-surface Chamfer distance d_C and *wrinkle-region Chamfer distance* $d_{\mathcal{W}}$, which quantifies the consistency of high-curvature regions across discretisations. Since the NDF provides correspondences, we extract

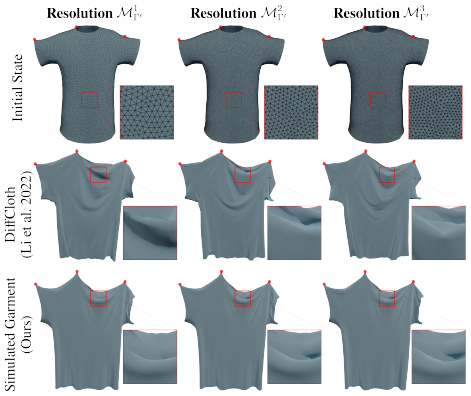


Fig. 5: Discretisation independence. Qualitative results at three mesh resolutions. NeuralGarSim produces consistent wrinkle patterns; DiffCloth shows sensitivity to mesh connectivity.

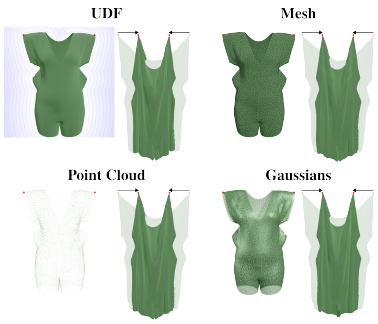


Fig. 6: Multi-modal consistency evaluation. Qualitative results for four input representations. Small cross-modal distances confirm that simulation is not significantly affected by the choice of input.

Table 3: Wrinkle Chamfer distances $d_{\mathcal{W}}$ and full Chamfer distances d_C across three discretisations. NeuralGarSim achieves consistently lower values than DiffCloth.

	$d_{\mathcal{W}}(\cdot,\cdot)^{\tau=75}$		$d_C(\cdot,\cdot)$	
Method	$(\mathcal{M}^1_{\Gamma'}, \mathcal{M}^2_{\Gamma'})$	$(\mathcal{M}^1_{\Gamma'}, \mathcal{M}^3_{\Gamma'})$	$(\mathcal{M}^1_{\Gamma'}, \mathcal{M}^2_{\Gamma'})$	$(\mathcal{M}^1_{\Gamma'}, \mathcal{M}^3_{\Gamma'})$
DiffCloth	0.035	0.034	0.020	0.021
Ours	**0.011**	**0.013**	**0.009**	**0.009**

Table 4: Cross-modal Chamfer distances relative to the mesh reference.

Modality	$d_{\mathcal{W}}(\mathcal{M}^{\text{ref}}_{\Gamma'}, \mathcal{M}^i_{\Gamma'})^{\tau=75}$	$d_C(\mathcal{M}^{\text{ref}}_{\Gamma'}, \mathcal{M}^i_{\Gamma'})$
Mesh	ref.	ref.
UDF	0.017	0.013
Point cloud	0.	0.
Gaussians	0.017	0.014

$\mathcal{M}_{\Gamma'}$ using the initial mesh topology $\mathcal{M}_\Gamma$, whereas the baselines directly output meshes. The wrinkle skeleton of a mesh $\mathcal{M}$ is the vertex subset defined as

$$\mathcal{W}(\mathcal{M}) = \{v \in \mathcal{M} : |H(v)| \geq H^{(\tau)}(\mathcal{M})\}, \tag{18}$$

where $H(v)$ is vertex mean curvature and $H^{(\tau)}$ denotes the 75th percentile over vertices of $\mathcal{M}$. $d_{\mathcal{W}}(\mathcal{M}^i_{\Gamma'}, \mathcal{M}^j_{\Gamma'})$ is the bidirectional Chamfer distance between simulation wrinkles $\mathcal{W}(\mathcal{M}^i_{\Gamma'})$ and $\mathcal{W}(\mathcal{M}^j_{\Gamma'})$ from two initial discretisations $\mathcal{M}^i_\Gamma$ and $\mathcal{M}^j_\Gamma$. A lower value indicates more consistent wrinkles.

Results. Fig. 4 and Fig. 5 show results for T-shirts under three discretisations, while Tab. 3 reports the corresponding full-surface and wrinkle-region Chamfer distances. NeuralGarSim produces visually consistent fold patterns and achieves the lowest $d_{\mathcal{W}}$ across all methods. Both DiffARCSim (despite adaptive remeshing) and DiffCloth exhibit wrinkle patterns that shift substantially with input mesh connectivity, as expected from discretisation-dependent element formulations.

5.5 Multi-Modal Input Representations

To the best of our knowledge, NeuralGarSim is the only simulator in our comparisons that natively supports multiple input representations, while all baselines



Fig. 7: Multi-material simulation. We simulate the same garment with isotropic linear, cotton StVK, and silk StVK parameters. Anisotropic stiffness changes fold orientation and wrinkle density.

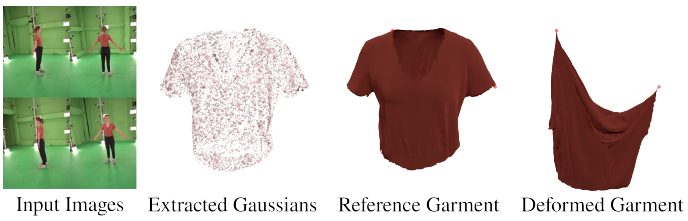


Fig. 8: Simulation on reconstructed 3D Gaussians. GaussianGarments [69] provides surface-registered 3D Gaussians from multi-view images. NeuralGarSim uses the extracted Gaussians directly as the simulation input. For clear visualisation, we show the same undeformed Gaussian asset as a reference garment and show the predicted quasistatic drape under pinned shoulders.

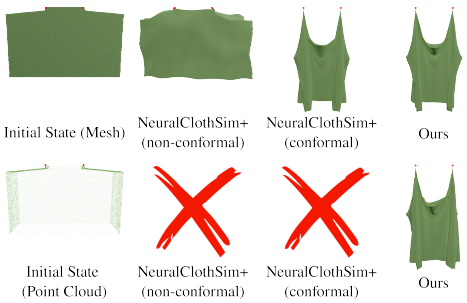
require triangle meshes. Fig. 6 shows NeuralGarSim applied to four input modalities: a differentiable UDF (primary representation), a triangle mesh, a point cloud, and 3D Gaussians encoding the same reference garment. The resulting deformations are perceptually consistent across modalities: large-scale fold structure and drape direction are preserved, and fine-scale wrinkles appear in the same surface regions. Cross-modal Chamfer distances reported in Tab. 4 are small, confirming that input representation does not significantly affect simulation fidelity.

Gaussians from Captured Data. We simulate a garment reconstructed from multi-view video using GaussianGarments [69]. We use its registered surface Gaussians as input and ignore appearance optimisation and learned dynamics, since NeuralGarSim requires only splat orientations for normals and is independent of colour or opacity. As shown in Fig. 8, NeuralGarSim predicts the quasistatic equilibrium under gravity and pinned shoulders without constructing a 2D parametric domain.

Multi-Material Simulation. Fig. 7 shows a garment simulated under isotropic linear elasticity and orthotropic StVK with cotton and silk parameters. The linear model produces uniform drape with equal resistance in all directions. The cotton StVK configuration, with higher warp stiffness ($a_{11}>a_{22}$), produces stiffer folds aligned with the warp direction, resulting in a more structured upright drape. The silk StVK configuration, with lower stiffness and higher shear compliance (G_{12}), drapes more loosely with softer folds. These distinct results confirm that

the orthotropic StVK model captures material anisotropy without changing the simulation architecture.

5.6 Ablation: Necessity of TDC



A natural question is whether a fully 3D TDC formulation is necessary, or whether the existing 2D parametrisation of NeuralClothSim can be extended to handle multi-panel garments by stitching UV charts at seams. To test this, we construct **NeuralClothSim+** by: (a) extending the single-panel UV domain of NeuralClothSim [37] to a multi-panel garment with a per-panel parameterisation and (b) replacing the 2D NDF input (*i.e.*, parametric coordinates $\boldsymbol{\xi}$) with 3D positions $\mathbf{x}_\Gamma$. Since NeuralClothSim requires the strain expression to be defined in 2D parametric coordinates, we recover parametric derivatives from NDF through the chain rule, $\partial \mathbf{u}/\partial \boldsymbol{\xi} = (\partial \mathbf{u}/\partial \mathbf{x}_\Gamma) J_{\text{rest}}$, where $J_{\text{rest}} = \partial \mathbf{x}_\Gamma/\partial \boldsymbol{\xi}$ is the per-face Jacobian of the *rest* configuration. Although simulations seem plausible at first, NeuralClothSim+ suffers from the following fundamental limitations that TDC eliminates by construction.

Limitations of the Parametric Extension. With a conformal parameterisation, NeuralClothSim+ produces a plausible drape, but with a non-conformal parameterisation of the same garment, it fails to converge since distorted per-face Jacobians transfer inconsistent parametric gradients to the strain computation. Moreover, it cannot be applied to input modalities where a 2D parameterisation is unavailable or ill-defined, such as point clouds. In contrast, NeuralGarSim requires only surface normals and produces physically consistent simulations in all cases. Additional results are provided in Appendix C.1.

6 Discussion and Conclusion

NeuralGarSim enables garment simulation directly in 3D without an explicit 2D parameterisation, while preserving simulation fidelity and supporting different mesh extraction resolutions post-training. It matches NeuralClothSim on Belytschko benchmarks, trains 2–4× faster, and reduces wrinkle-region Chamfer distance compared to DiffCloth in discretisation experiments. It supports multi-panel garments across distance fields, meshes, point clouds, and 3D Gaussians, with consistent behaviour across geometric modalities and material-dependent drape. Current limitations are the quasistatic setting and the absence of inertia, damping, and collisions. Extending the method to dynamics, robust collision handling, faster optimisation, and inverse problems remains future work.

Acknowledgements

We thank Rhaleb Zayer and Briac Toussaint for insightful discussions on tangential differential calculus and shell theory.

References

1. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Conference on Computer Graphics and Interactive Techniques. SIGGRAPH (1998)
2. Barr, A.H.: Global and local deformations of solid primitives. In: Conference on Computer Graphics and Interactive Techniques. SIGGRAPH (1984)
3. Basar, Y., Krätzig, W.B.: Mechanik der Flächentragwerke: theorie, berechnungsmethoden, anwendungsbeispiele. Springer-Verlag (2013)
4. Bastek, J.H., Kochmann, D.M.: Physics-informed neural networks for shell structures. European Journal of Mechanics-A/Solids (2023)
5. Belytschko, T., Stolarski, H., Liu, W.K., Carpenter, N., Ong, J.S.: Stress projection for membrane and shear locking in shell finite elements. Computer Methods in Applied Mechanics and Engineering (1985)
6. Bertiche, H., Madadi, M., Escalera, S.: Pbns: physically based neural simulation for unsupervised garment pose space deformation. ACM Transactions on Graphics (2021)
7. Bertiche, H., Madadi, M., Escalera, S.: Neural cloth simulation. ACM Transactions on Graphics (2022)
8. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-garment net: Learning to dress 3d people from images. In: International Conference on Computer Vision (ICCV) (2019)
9. Bridson, R., Fedkiw, R., Anderson, J.: Robust treatment of collisions, contact and friction for cloth animation. In: ACM Transactions on Graphics (2002)
10. Buffet, T., Rohmer, D., Barthe, L., Boissieux, L., Cani, M.P.: Implicit untangling: A robust solution for modeling layered clothing. ACM Transactions on Graphics (2019)
11. Chen, R., Chen, L., Parashar, S.: Gaps: Geometry-aware, physics-based, self-supervised neural garment draping. In: International Conference on 3D Vision (3DV) (2024)
12. Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. In: Neural Information Processing Systems (NeurIPS) (2020)
13. Clyde, D., Teran, J., Tamstorf, R.: Modeling and data-driven parameter estimation for woven fabrics. In: ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA) (2017)
14. Clyde, D.C.: Numerical Subdivision Surfaces for Simulation and Data Driven Modeling of Woven Cloth. University of California, Los Angeles (2017)
15. De Luigi, L., Li, R., Guillard, B., Salzmann, M., Fua, P.: DrapeNet: Garment Generation and Self-Supervised Draping. In: Computer Vision and Pattern Recognition (CVPR) (2023)
16. Delfour, M., Zolésio, J.: Tangential differential equations for dynamical thin/shallow shells. Journal of Differential Equations (1996)
17. Delfour, M.C., Zolesio, J.P.: Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization. SIAM (2011)
18. Duisterhof, B.P., Mandi, Z., Yao, Y., Liu, J.W., Song, S., Shou, M.Z., Ichnowski, J.: Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes. arXiv preprint arXiv:2312.00583 (2023)
19. Etzmuß, O., Keckeisen, M., Straßer, W.: A fast finite element solution for cloth modelling. In: Proc. of The Pacific Conference on Computer Graphics and Applications (2003)

20. Fainstein, M., Siless, V., Iarussi, E.: Dudf: Differentiable unsigned distance fields with hyperbolic scaling. In: Computer Vision and Pattern Recognition (CVPR) (2024)
21. Grigorev, A., Becherini, G., Black, M., Hilliges, O., Thomaszewski, B.: ContourCraft: Learning to resolve intersections in neural multi-garment simulations. In: ACM SIGGRAPH Conference Papers (2024)
22. Grigorev, A., Black, M.J., Hilliges, O.: Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In: Computer Vision and Pattern Recognition (CVPR) (2023)
23. Grinspun, E., Gingold, Y., Reisman, J., Zorin, D.: Computing discrete shape operators on general meshes. In: Computer Graphics Forum (2006)
24. Grinspun, E., Hirani, A.N., Desbrun, M., Schröder, P.: Discrete shells. In: ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA) (2003)
25. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Computer Vision and Pattern Recognition (CVPR) (2024)
26. Guo, H., Zhuang, X., Rabczuk, T.: A deep collocation method for the bending analysis of kirchhoff plate. Computers, Materials & Continua (2019)
27. Guo, Q., Han, X., Fu, C., Gast, T., Tamstorf, R., Teran, J.: A material point method for thin shells with frictional contact. ACM Transactions on Graphics (2018)
28. Gurtin, M.E., Ian Murdoch, A.: A continuum theory of elastic material surfaces. Archive for rational mechanics and analysis (1975)
29. Hansbo, P., Larson, M.G.: Finite element modeling of a linear membrane shell problem using tangential differential calculus. Computer Methods in Applied Mechanics and Engineering (2014)
30. Hansbo, P., Larson, M.G.: Continuous/discontinuous finite element modelling of kirchhoff plate structures in r 3 using tangential differential calculus. Computational Mechanics (2017)
31. Hansbo, P., Larson, M.G., Larsson, F.: Tangential differential calculus and the finite element modeling of a large deformation elastic membrane problem. Computational Mechanics (2015)
32. Hao, Z., Liu, S., Zhang, Y., Ying, C., Feng, Y., Su, H., Zhu, J.: Physics-informed machine learning: A survey on problems, methods and applications. arXiv preprint arXiv:2211.08064 (2022)
33. Harmon, D., Vouga, E., Tamstorf, R., Grinspun, E.: Robust treatment of simultaneous collisions. ACM Transactions on Graphics (2008)
34. Hjelmstad, K.D.: Fundamentals of structural mechanics. Springer (2005)
35. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques (1992)
36. Kairanda, N., Habermann, M., Naik, S., Theobalt, C., Golyanik, V.: Thin-Shell-SfT: Fine-grained monocular non-rigid 3d surface tracking with neural deformation fields. In: Computer Vision and Pattern Recognition (CVPR) (2025)
37. Kairanda, N., Habermann, M., Theobalt, C., Golyanik, V.: NeuralClothSim: Neural deformation fields meet the thin shell theory. In: Neural Information Processing Systems (NeurIPS) (2024)
38. Kairanda, N., Naik, S., Habermann, M., Sharma, A., Theobalt, C., Golyanik, V.: ∂^∞-Grid: A neural differential equation solver with differentiable feature grids. In: International Conference on Learning Representations (ICLR) (2026)

39. Kairanda, N., Tretschk, E., Elgharib, M., Theobalt, C., Golyanik, V.: ϕ-SfT: Shape-from-template with a physics-based deformation model. In: Computer Vision and Pattern Recognition (CVPR) (2022)
40. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nature Reviews Physics (2021)
41. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (2023)
42. Kiendl, J., Hsu, M.C., Wu, M.C., Reali, A.: Isogeometric kirchhoff–love shell formulations for general hyperelastic materials. Computer Methods in Applied Mechanics and Engineering (2015)
43. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
44. Korosteleva, M., Lee, S.H.: Generating datasets of 3d garments with sewing patterns. In: Neural Information Processing Systems (NeurIPS) (2021)
45. Li, C., Tang, M., Tong, R., Cai, M., Zhao, J., Manocha, D.: P-cloth: interactive complex cloth simulation on multi-gpu systems using dynamic matrix assembly and pipelined implicit integrators. ACM Transactions on Graphics (2020)
46. Li, J., Daviet, G., Narain, R., Bertails-Descoubes, F., Overby, M., Brown, G.E., Boissieux, L.: An implicit frictional contact solver for adaptive cloth simulation. ACM Transactions on Graphics (2018)
47. Li, M., Kaufman, D.M., Jiang, C.: Codimensional incremental potential contact. ACM Transactions on Graphics (2021)
48. Li, R., Guillard, B., Remelli, E., Fua, P.: Dig: Draping implicit garment over the human body. In: Asian Conference on Computer Vision (ACCV) (2022)
49. Li, Y., Chen, H.y., Larionov, E., Sarafianos, N., Matusik, W., Stuyck, T.: Diffavatar: Simulation-ready garment optimization with differentiable simulation. In: Computer Vision and Pattern Recognition (CVPR) (2024)
50. Li, Y., Du, T., Wu, K., Xu, J., Matusik, W.: Diffcloth: Differentiable cloth simulation with dry frictional contact. ACM Transactions on Graphics (2022)
51. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Computer Vision and Pattern Recognition (CVPR) (2023)
52. Liang, J., Lin, M., Koltun, V.: Differentiable cloth simulation for inverse problems. Neural Information Processing Systems (NeurIPS) (2019)
53. Liao, Z., Wang, S., Komura, T.: Senc: Handling self-collision in neural cloth simulation. In: European Conference on Computer Vision (ECCV) (2024)
54. Liu, S., Zhongkai, H., Ying, C., Su, H., Zhu, J., Cheng, Z.: A unified hard-constraint framework for solving geometrically complex pdes. In: Neural Information Processing Systems (NeurIPS) (2022)
55. Long, X., Lin, C., Liu, L., Liu, Y., Wang, P., Theobalt, C., Komura, T., Wang, W.: Neuraludf: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In: Computer Vision and Pattern Recognition (CVPR) (2023)
56. Love, A.E.H.: A treatise on the mathematical theory of elasticity. Cambridge university press (2013)
57. Lu, J., Zheng, C.: Dynamic cloth simulation by isogeometric analysis. Computer Methods in Applied Mechanics and Engineering (2014)
58. Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., Johnson, S.G.: Physics-informed neural networks with hard constraints for inverse design. SIAM Journal on Scientific Computing (2021)

59. Ly, M., Jouve, J., Boissieux, L., Bertails-Descoubes, F.: Projective dynamics with dry frictional contact. ACM Transactions on Graphics (2020)
60. Mehta, I., Chandraker, M., Ramamoorthi, R.: A level set theory for neural implicit evolution under explicit flows. In: European Conference on Computer Vision (ECCV) (2022)
61. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and mathematics III (2003)
62. Miguel, E., Miraut, D., Otaduy, M.A.: Modeling and estimation of energy-based hyperelastic objects. In: Computer Graphics Forum (2016)
63. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision (ECCV) (2020)
64. Narain, R., Samii, A., O'brien, J.F.: Adaptive anisotropic remeshing for cloth simulation. ACM Transactions on Graphics (2012)
65. Novello, T., Da Silva, V., Schardong, G., Schirmer, L., Lopes, H., Velho, L.: Neural implicit surface evolution. In: International Conference on Computer Vision (ICCV) (2023)
66. Otaduy, M.A., Tamstorf, R., Steinemann, D., Gross, M.: Implicit contact handling for deformable objects. In: Computer Graphics Forum (2009)
67. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Neural Information Processing Systems (NeurIPS) (2019)
68. Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P.: Learning mesh-based simulation with graph networks. In: International Conference on Learning Representations (ICLR) (2021)
69. Rong, B., Grigorev, A., Wang, W., Black, M.J., Thomaszewski, B., Tsalicoglou, C., Hilliges, O.: Gaussian Garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. In: International Conference on 3D Vision (3DV) (2025)
70. Santesteban, I., Otaduy, M.A., Casas, D.: SNUG: Self-Supervised Neural Dynamic Garments. In: Computer Vision and Pattern Recognition (CVPR) (2022)
71. Santesteban, I., Otaduy, M.A., Thuerey, N., Casas, D.: ULNeF: Untangled layered neural fields for mix-and-match virtual try-on. In: Neural Information Processing Systems (NeurIPS) (2022)
72. Schöllhammer, D., Fries, T.P.: Kirchhoff–love shell theory based on tangential differential calculus. Computational mechanics (2019)
73. Schöllhammer, D., Fries, T.P.: Reissner–mindlin shell theory based on tangential differential calculus. Computer Methods in Applied Mechanics and Engineering (2019)
74. Simo, J.C., Fox, D.D., Rifai, S.M.: On a stress resultant geometrically exact shell model. part ii: The linear theory; computational aspects. Computer Methods in Applied Mechanics and Engineering (1989)
75. Simo, J.C., Fox, D.D.: On a stress resultant geometrically exact shell model. part i: Formulation and optimal parametrization. Computer Methods in Applied Mechanics and Engineering (1989)
76. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Neural Information Processing Systems (NeurIPS) (2020)
77. Tang, M., Liu, Z., Tong, R., Manocha, D.: Pscc: Parallel self-collision culling with spatial hashing on gpus. Proc. ACM Comput. Graph. Interact. Tech. (2018)

78. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. SIGGRAPH Comput. Graph. (1987)
79. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. Computer Graphics Forum (Eurographics State of the Art Reports) (2022)
80. Thomaszewski, B., Wacker, M., Straßer, W.: A consistent bending model for cloth simulation with corotational subdivision finite elements. In: ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA) (2006)
81. Timoshenko, S., Woinowsky-Krieger, S.: Theory of plates and shells. McGraw-hill New York (1959)
82. Volino, P., Thalmann, N.M.: Implementing fast cloth simulation with collision response. In: Proceedings Computer Graphics International (2000)
83. Wang, H.: Gpu-based simulation of cloth wrinkles at submillimeter levels. ACM Transactions on Graphics (2021)
84. Wang, H., O'Brien, J.F., Ramamoorthi, R.: Data-driven elastic models for cloth: modeling and measurement. ACM Transactions on Graphics (2011)
85. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Neural Information Processing Systems (NeurIPS) (2021)
86. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: International Conference on Computer Vision (ICCV) (2023)
87. Wempner, G., Talaslidis, D.: Mechanics of solids and shells. CRC Press (2003)
88. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. In: Computer Graphics Forum (Eurographics State of the Art Reports) (2022)
89. Yang, G., Belongie, S., Hariharan, B., Koltun, V.: Geometry processing with neural fields. In: Neural Information Processing Systems (NeurIPS) (2021)
90. Zhang, J.E., Dumas, J., Fei, Y., Jacobson, A., James, D.L., Kaufman, D.M.: Progressive shell qasistatics for unstructured meshes. ACM Transactions on Graphics (2023)