Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

aNeuSF A Factorized Neural Surrogate for Efficient Shape-Function Construction in Numerical Coarsening

Xuwei Wang
Zhejiang University
Wei Chen
Zhejiang University
Haroon Ijaz
Zhejiang University
Ming Li
liming@cad.zju.edu.cn

Zhejiang University

Research Article

Keywords:

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10068181/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

aNeuSF: A Factorized Neural Surrogate for Efficient Shape-Function Construction in Numerical Coarsening

Xuwei Wang[1], Wei Chen[1], Haroon Ijaz[1], Ming Li[1*]

[1]State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027, China.

*Corresponding author(s). E-mail(s): liming@cad.zju.edu.cn;

Abstract

Numerical coarsening simulates structures by embedding them into a coarse, regular background mesh while maintaining high simulation accuracy and avoiding tedious preprocessing and challenging volume mesh construction. Its practical applicability, however, is often limited by the high cost of constructing material-aware shape functions for individual coarse elements. We present **aNeuSF**, a factorized neural surrogate for efficient shape-function construction in numerical coarsening. Instead of learning the complete coarsening map directly, aNeuSF decomposes coarse-element shape functions into an analytical Bézier-order interpolator and a learned boundary-to-fine mapping predicted from coarse-element density fields. This analytical–learned factorization reduces the dimensionality of the learning target, improves trainability, and provides a natural mechanism for preserving key shape-function properties, including partition of unity and rigid-motion consistency. It further supports selectable Bézier orders, allowing different accuracy–efficiency trade-offs within a unified formulation. Experiments on linear-elastic static problems show that aNeuSF achieves competitive simulation accuracy while delivering substantial batched acceleration over conventional numerical coarsening baselines, with up to two orders of magnitude speedup in high-resolution and high-order settings.

Keywords: Factorized neural surrogate, Neural shape functions, Numerical coarsening, FEM, Bézier interpolation

1 Introduction

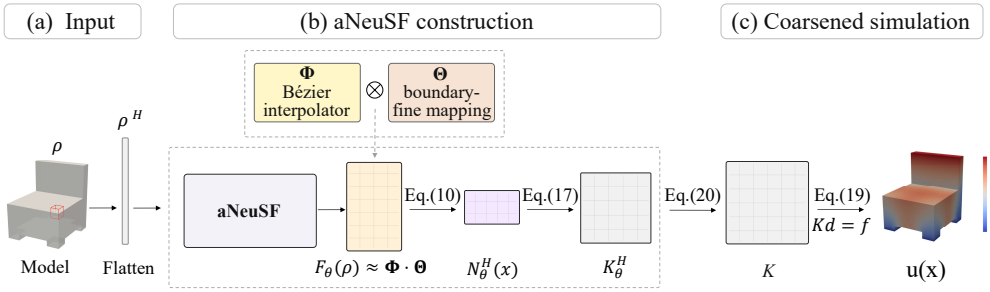


Fig. 1 Overview of the proposed aNeuSF-based numerical coarsening framework. For each coarse element, the voxelized density field is flattened and fed into aNeuSF to predict the surrogate mapping matrix $\mathbf{F}_\theta(\rho)$, replacing the costly conventional construction based on Bézier interpolation and boundary-to-fine mapping. The predicted mapping is used to construct the coarse-element shape-function matrix $\mathbf{N}_\theta^H(\mathbf{x})$, from which the coarse-element stiffness matrices are computed and assembled into the global stiffness matrix. The resulting linear system in Eq. (1) is then solved to obtain the displacement field

Simulation of the physical performance of structures is a fundamental problem, and is usually achieved via classical finite element methods (FEM) [1, 2]; however, these methods still incur challenges in conforming meshing. In visual computing, efficient physically based simulation has also been studied from different perspectives, including finite-element contact formulations for deformable objects [3], voxel-based topological updates for cutting simulation [4], and PDE-based real-time surface deformation with C^1 continuity [5]. Numerical coarsening [6–8], on the other hand, avoids the conformal meshing by embedding complex structures within a coarse, regular background mesh, by reconstructing material-aware shape functions that map coarse degrees of freedom (DOFs) to interior fine-mesh DOFs. Despite notable advances [8–11], the overall performance of numerical coarsening is hindered by its substantial computational cost in constructing the material-aware shape functions for each coarse element, which involves computing solutions to a small number of FEA problems.

Learning-based approaches have therefore emerged to predict numerical shape functions from material distributions, avoiding repeated fine FE solves. Related learning-based physical simulation studies have also explored neural surrogates for accelerating repeated numerical analysis, such as microstructure-property mapping [12] and multi-scale graph neural networks for physics-informed simulation [13]. Problem-Independent Machine Learning (PIML) frameworks [14–17] learn mappings from substructure material fields to multiscale shape functions, but supervised training with large output tensors can be data-hungry and may underuse physics in the loss, which limits generalization and scalability. Recent operator-learning variants address these issues by enforcing physics-motivated objectives and treating shape

functions as continuous fields. In parallel, neural shape functions [18] have been proposed for numerical coarsening in graphics, where embedding geometry constraints (e.g., rigid-motion invariance) as priors improves robustness compared with end-to-end regressors.

However, most existing approaches either prescribe a fixed interpolation behavior along coarse-element boundaries or rely on simplified assumptions about boundary deformation. Such treatments may restrict the attainable accuracy–efficiency balance and can produce overly stiff responses on coarse meshes. In addition, conventional numerical coarsening usually requires the repeated construction of material-aware shape functions for each coarse element, making shape-function generation a significant computational bottleneck at scale. To overcome this limitation, we propose **aNeuSF**, a factorized neural surrogate for efficient shape-function construction in numerical coarsening. Rather than learning coarse-element shape functions in a monolithic manner, aNeuSF separates their construction into an order-controllable analytical component and a learned material-aware component. This factorization is the central novelty of our approach: it reduces the complexity of the learning problem, improves the flexibility of the surrogate, and allows the interpolation order to be selected according to the desired balance between efficiency and accuracy. At the same time, the factorized formulation provides a natural way to retain essential shape-function properties, leading to efficient and reliable shape-function construction for coarse-scale simulation.

In addition, we introduce a Magnitude-Adaptive Smooth L_1 Loss (MASL1) for training aNeuSF, which reweights supervision according to target magnitude and helps the model better capture small-magnitude entries in the predicted boundary-to-fine mapping matrix. Extensive numerical experiments at both the element level and the simulation level show that aNeuSF achieves competitive accuracy together with substantial batched acceleration compared with conventional numerical coarsening baselines.

In summary, our main contributions are as follows:

1. We propose **aNeuSF**, a factorized neural surrogate for efficient shape-function construction in numerical coarsening. By learning the material-aware boundary-to-fine mapping from coarse-element density fields, aNeuSF avoids repeated per-element construction and substantially reduces the computational cost of shape-function generation.
2. We develop a factorized formulation that combines an analytical Bézier-order interpolator with a learned mapping matrix. This design is central to aNeuSF: it reduces the complexity of the learning target, helps preserve essential geometric and physical consistency, and enables selectable Bézier orders for balancing computational efficiency and approximation accuracy.
3. We introduce a magnitude-adaptive Smooth L_1 loss, termed MASL1, to train the proposed surrogate. The loss adaptively reweights supervision according to the target magnitude, improving the prediction of small-magnitude entries in the boundary-to-fine mapping matrix.

4. We conduct extensive element-level and simulation-level experiments to evaluate the proposed method. The results demonstrate that aNeuSF achieves competitive accuracy while providing substantial batched acceleration over conventional numerical coarsening baselines.

2 Related Work

2.1 Conventional shape functions

In the finite element method, shape functions are employed to interpolate displacements within elements and have been widely used in deformable simulation, contact modeling, voxel-based deformation, and physics-based surface manipulation [3–5]. Common formulations, such as bilinear and trilinear shape functions, satisfy essential geometric properties like the Partition of Unity and the Lagrange Property. However, the use of linear shape functions on coarse meshes can lead to significant overstiffness, resulting in inaccurate deformation predictions. Initial attempts [7, 19–21] at numerical coarsening predominantly focused on analytical solutions or optimization techniques. These methods are designed to simulate co-rotational linear materials and small-deformation hyperelastic materials. Kharevych et al. [19] proposed a method that calculates the effective elasticity tensor of coarse elements by solving harmonic displacement problems on fine meshes. Chen et al. [7] optimized non-conforming and matrix-valued shape functions, and later introduced a hierarchical construction of material-adapted refinable basis functions [22]. Subsequent works further improved coarsened simulation quality by developing C^1-continuous shape functions for better stress continuity [23] and EigenFit for approximating global eigenvalues on coarse meshes [24]. Nevertheless, many numerical coarsening methods still face limitations associated with scale-separation assumptions or costly element-wise shape-function construction. To address non-separated heterogeneous structures, Li et al. [8] proposed the CBN method, which combines Bézier interpolation [25] with boundary-interior transformations. Although effective, this formulation still requires per-element boundary-interior solves, and the cost increases further with higher interpolation orders.

2.2 Learning-based shape functions

In recent years, deep learning has provided new tools for mesh generation [26, 27], structural design [15, 16, 28–30], and simulation [13, 31–33]. In computational mechanics, learning-based methods have been explored for coarse-scale material substitution [34], nonlinear reduced subspaces [35–37], deformation correction [38], fluid synthesis [39], and data-driven detail enhancement [40, 41]. More recently, neural approaches have been introduced for constructing shape functions in numerical coarsening. Ni et al. [18] represented nonlinear shape functions using neural networks to improve generalization and accuracy. Huang et al. proposed a problem-independent machine learning (PIML) framework [15] that maps substructure material distributions to condensed stiffness matrices and numerical shape functions, and further

developed a mechanics-based data-free formulation [16] with DeepONet [42] to predict continuous multiscale shape functions without requiring pre-generated labeled data.

Despite these advances, existing learning-based approaches usually do not explicitly exploit the factorized structure of numerical coarsening shape-function construction. In particular, they often rely on fixed interpolation settings or do not separate the analytical boundary interpolation from the density-dependent boundary-to-fine construction, which motivates the factorized surrogate formulation developed in this work.

3 Problem formulation and Preliminaries

3.1 Problem formulation

Our primary objective is to compute the displacement field $\mathbf{u}(\mathbf{x})$ in a linear elasticity analysis of structure Ω, described as the solution to the following boundary value problem [43]:

$$\begin{cases} -\operatorname{div}\boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) = \mathbf{f}, & \text{in } \Omega, \\ \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}))\cdot\mathbf{n} = \boldsymbol{\tau}, & \text{on } \Gamma_N, \\ \mathbf{u}(\mathbf{x}) = \mathbf{u}_D, & \text{on } \Gamma_D, \end{cases} \tag{1}$$

where the Cauchy stress tensor $\boldsymbol{\sigma}(\mathbf{u})$ is defined via Hooke's law and the strain tensor is defined as a linearized approximation of the Green-Lagrange strain tensor, Γ_D is the displacement boundary with the prescribed displacement $\mathbf{u}_D$, Γ_N is the Neumann boundary with the external traction $\boldsymbol{\tau}$, and $\mathbf{f}$ represents the body force.

3.2 Preliminaries

3.2.1 Shape functions

To solve the boundary value problem (1), FEM is widely employed due to its ability to handle complex geometries and boundary conditions. The differential form in Eq. (1) can be stated as a weak form, the displacement $\mathbf{u}(\mathbf{x})$ satisfies:

$$a(\mathbf{u},\mathbf{v}) = l(\mathbf{v}), \tag{2}$$

where

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v})\mathrm{d}V = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u})^T\mathbf{D}(\mathbf{x})\boldsymbol{\varepsilon}(\mathbf{v})\mathrm{d}V, \tag{3}$$

and

$$l(\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}\mathrm{d}V + \int_{\Gamma_N} \boldsymbol{\tau}\cdot\mathbf{v}\,\mathrm{d}S, \tag{4}$$

where $\mathbf{v}$ is the test vector function.

FEM discretizes the continuous domain Ω into a finite set of simple subdomains (elements), and approximates the solution $\mathbf{u}(\mathbf{x})$ by interpolating nodal values through shape functions. Shape functions should satisfy the following fundamental properties: (i) **Partition of Unity**(**PU**) The sum of all shape functions is equal to one at any point within the element. (ii)**Lagrange Property** The shape function takes the value

one at its corresponding node and zero at all other nodes. (iii)**Nodal Interpolation Property**: the displacement field can be interpolated from nodal values $\{\mathbf{u}_i\}_{i=1}^{n}$ as

$$\mathbf{u}(\boldsymbol{\xi}) = \sum_{i=1}^{n} N_i(\boldsymbol{\xi})\, \mathbf{u}_i. \tag{5}$$

3.2.2 Numerical coarsening

Numerical coarsening is a coarse-scale simulation strategy that embeds complex structures into a coarse background mesh and encodes fine-scale geometric and material information through material-aware shape functions. These shape functions map a small number of coarse DOFs to the displacement field over the underlying fine-scale nodes, allowing the global problem to be solved on a reduced coarse system while avoiding conforming volume mesh construction.

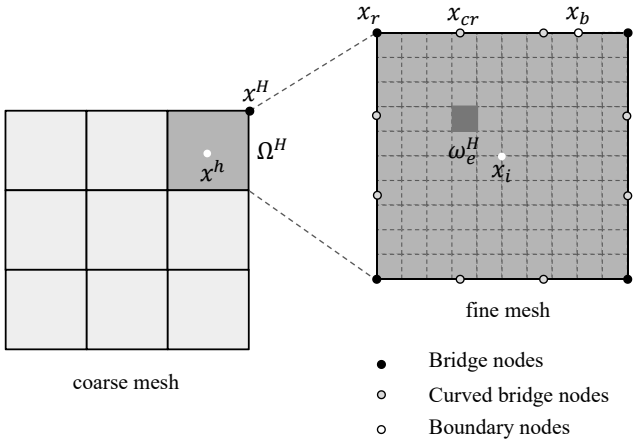


Fig. 2 Structure Ω, coarse mesh $M^H = \{\Omega^H, H = 1, 2, \ldots, n_H\}$, local fine mesh $\mathcal{M}^h = \{\omega_e^H \mid e = 1, 2, \ldots, n_e,\ H = 1, 2, \ldots, n_H\}$. Fine-grid nodes are classified into four categories: bridge nodes $\mathcal{X}_r$, curved bridge nodes (CBNs) $\mathcal{X}_{cr}$, boundary nodes $\mathcal{X}_b$, and interior nodes $\mathcal{X}_i$, with $\mathcal{X}_r \subseteq \mathcal{X}_b$. CBNs are additional nodes introduced between adjacent bridge nodes. In the proposed framework, both bridge nodes and curved bridge nodes are treated as master nodes and selected as DOFs

CBN shape functions [8] establish mappings from CBN nodes, defined along the boundaries of macro-elements, to their internal nodes, shown in Fig. 2. The shape function for each macro-element is derived as the matrix product of a Bézier interpolation matrix and a boundary-to-interior transformation matrix, which enables the shape functions to be expressed in an explicit matrix representation.

Bézier-order interpolator. Bézier interpolation is utilized to construct the mapping from coarse nodes to fine boundary nodes. For a cubic Bézier interpolation curve in 2D, the interpolation is defined as:

$$\begin{aligned} \mathbf{P}(t) &= \sum_{i=0}^{3} \varphi_i(t)\, \mathbf{P}_i, \\ \varphi_i(t) &= C_3^i t^i (1-t)^{3-i}, \end{aligned} \tag{6}$$

where $\mathbf{P}_i$ represents the control points, and $\varphi_i(t)$ are the Bernstein polynomials that define the interpolation. In the finite element framework, the corresponding Bézier interpolation matrix for a boundary node $\mathbf{x}_0$ is derived as:

$$\underset{(2b\times 6n)}{\mathbf{\Phi}} = \sum_{\text{assembly}} \underset{(2\times 6n)}{\boldsymbol{\varphi}_+}(\mathbf{x}_0), \quad \mathbf{x}_0 \in \mathcal{X}_b, \tag{7}$$

where $\boldsymbol{\varphi}_+(\mathbf{x}_0)$ is the matrix of Bézier interpolation basis functions, assembled from the boundary nodes.

$$\underset{(2\times 6n)}{\boldsymbol{\varphi}_+}(\mathbf{x}_0) = \sum_{\mathcal{X}_b} \boldsymbol{\varphi}(\mathbf{x}_0)\,, \quad \mathbf{x}_0 \in \mathcal{X}_b. \tag{8}$$

Let $\mathcal{X}_b$ denote the boundary nodes, where b and n represent the number of boundary nodes and bridge nodes, respectively. The dimension $6n$ is evident owing to the n line segments originating from the n bridge nodes, each of which is associated with $3n$ master nodes because of cubic Bézier interpolation. Consequently, the total dimension is $6n$, considering the x and y components. For all boundary nodes, the corresponding basis functions $\boldsymbol{\varphi}_+(\mathbf{x}_0)$ are arranged row-wise according to $\mathcal{X}_b$.

Boundary-interior mapping. To construct a transformation that maps boundary displacements to the interior DOFs, the local finite element equilibrium equation on the fine-scale mesh is first partitioned by separating the interior and boundary nodes. By enforcing static equilibrium at the interior nodes, the interior displacements can be expressed as a linear function of the boundary displacements. This relation defines the boundary-to-interior transformation matrix $\mathbf{\Theta}$, which extends the prescribed boundary displacements to all fine-scale DOFs within the local fine mesh. Here, b and i denote the numbers of boundary nodes and interior nodes, respectively.

Consequently, the matrix representation of the CBN shape functions is ultimately formulated as follows:

$$\underset{(2\times 6n)}{\mathbf{N}^H(\mathbf{x})} = \underset{2\times(2i+2b)}{\mathbf{N}^h(\mathbf{x})} \quad \underset{(2i+2b)\times 2b}{\mathbf{\Theta}} \quad \underset{(2b\times 6n)}{\mathbf{\Phi}}, \tag{9}$$

where $\mathbf{N}^h(\mathbf{x})$ is the matrix assembled from the shape functions on the microscopic mesh, $\mathbf{\Theta}$ is the boundary-to-interior transformation matrix, and $\mathbf{\Phi}$ is the Bézier interpolation matrix that maps coarse-element master-node DOFs to fine boundary nodes. In this formulation, $\mathbf{\Phi}$ first determines the fine-scale boundary displacement field from the coarse-element master nodes, $\mathbf{\Theta}$ extends the boundary displacements to the interior fine nodes, and $\mathbf{N}^h(\mathbf{x})$ interpolates the resulting fine-scale nodal displacement field within the local fine mesh.

While CBN-based methods [8] are effective, they rely on costly element-wise boundary-to-fine construction, which substantially increases computational complexity. This observation motivates our factorized neural surrogate, which replaces the repeated per-element construction with a learned mapping from the coarse-element density field, as described in the next section.

4 Methodology

Fig. 1 illustrates the proposed framework for efficient shape-function construction in numerical coarsening. Instead of repeatedly performing conventional per-element boundary-to-fine construction, aNeuSF learns a surrogate mapping from the coarse-element density field to the corresponding boundary-to-fine mapping matrix. Based on this predicted mapping, the coarse-element shape-function matrix is constructed in a factorized form. This design reduces the computational cost of shape-function construction while preserving the structured representation required for numerical coarsening.

4.1 Factorized aNeuSF shape-function construction

aNeuSF formulation. In contrast to the conventional CBN formulation, aNeuSF replaces the explicit per-element boundary-to-fine construction in Eq. (9) with a predicted mapping matrix obtained from the coarse-element density field. The resulting shape functions are then constructed through a factorized analytical-learned formulation. Specifically, in the 3D scenario, for a macro-element Ω^H with the associated micro-scale shape functions $\mathbf{N}^h(\mathbf{x})$, the aNeuSF shape-function matrix $\mathbf{N}_\theta^H(\mathbf{x})$ is formulated as:

$$\underset{(3\times 3n)}{\mathbf{N}_\theta^H(\mathbf{x})} = \underset{3\times 3(m+1)^3}{\mathbf{N}^h(\mathbf{x})} \quad \underset{3(m+1)^3\times 3n}{\mathbf{F}_\theta(\boldsymbol{\rho})}, \quad \mathbf{x}\in\Omega^H, \tag{10}$$

where $n = 8 + 12(\kappa-1) + 6(\kappa-1)^2$ is the total number of master nodes(bridge nodes and curved bridge nodes) in a coarse element, κ is the Bézier order. $\mathbf{F}_\theta(\boldsymbol{\rho})$ represents the boundary-to-fine mapping matrix obtained from the output of the trained aNeuSF model, which maps the macro-element density field $\boldsymbol{\rho}$ of resolution $m\times m\times m$ to the coefficients defining the coarse-element shape functions on the micro mesh.

This factorized formulation provides three practical advantages. First, it reduces the dimensionality of the learning target. Instead of directly learning a dense coarse-to-fine coarsening map associated with all fine boundary DOFs, aNeuSF learns the combined boundary-to-fine mapping with respect to a smaller set of Bézier master-node DOFs. Second, the analytical interpolation component helps retain the structured properties of conventional shape functions, including partition of unity and rigid-motion consistency. Third, since the number of master nodes n depends on the Bézier order κ, the same formulation naturally supports different interpolation orders for accuracy-efficiency trade-offs.

Surrogate target and shape-function consistency. The predicted matrix $\mathbf{F}_\theta(\boldsymbol{\rho})$ is used as a learned surrogate for the conventional density-dependent mapping encoded by the boundary-to-interior transformation and the Bézier interpolation matrix in

Eq. (9). In other words, aNeuSF does not explicitly construct $\boldsymbol{\Theta} \cdot \boldsymbol{\Phi}$ during inference; instead, it learns the combined mapping effect from the coarse-element density field and directly predicts $\mathbf{F}_\theta(\boldsymbol{\rho})$. This design keeps the analytical micro-grid shape-function matrix $\mathbf{N}^h(\mathbf{x})$ in the final representation, while replacing the costly element-wise construction of the density-dependent mapping with batched neural prediction.

The factorized form in Eq. (10) also provides a structured way to inherit the desirable behavior of conventional numerical coarsening shape functions. Since $\mathbf{N}^h(\mathbf{x})$ is the analytical micro-scale shape-function matrix and $\mathbf{F}_\theta(\boldsymbol{\rho})$ is trained to approximate the target boundary-to-fine mapping, the resulting coarse-element shape functions are encouraged to preserve key consistency properties, including partition of unity and rigid-motion consistency.

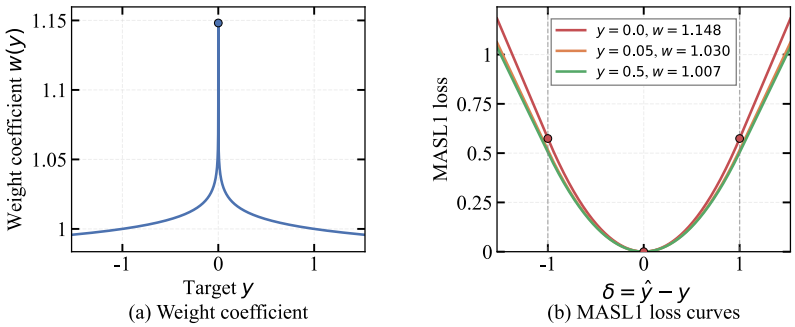


Fig. 3 Illustration of the proposed MASL1 loss. (a) The magnitude-adaptive weight coefficient $w(y) = 1/(|y| + \varepsilon)^\gamma$, which assigns larger weights to small-magnitude entries of the target boundary-to-fine mapping matrix. (b) MASL1 loss curves for representative target values $y = 0.0$, 0.05, and 0.5, defined as $w(y)\mathcal{L}_{\text{smooth}}(\delta;\beta)$, where $\delta = \hat{y} - y$ is the prediction residual and $\beta = 1.0$

4.2 Learning objective for mapping-matrix prediction

The Smooth L_1 loss is defined as:

$$\mathcal{L}_{\text{smooth}}(\delta;\beta) = \begin{cases} \dfrac{\delta^2}{2\beta}, & |\delta| < \beta, \\ |\delta| - \dfrac{\beta}{2}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\delta = \hat{y} - y$ is the prediction residual, $\hat{y}$ denotes the predicted value, y is the ground-truth target, and $\beta > 0$ is the smoothing parameter controlling the transition between the quadratic and linear regimes.

Magnitude-**A**daptive **S**mooth L_1 Loss (**MASL1**) extends the Smooth L_1 loss by incorporating a magnitude-based weighting mechanism. It enhances the standard Smooth L_1 loss by assigning larger weights to small-magnitude target entries in the

predicted boundary-to-fine mapping matrix. For the b-th training sample in a mini-batch, the predicted and ground-truth mapping matrices are flattened as $\hat{\mathbf{y}}^{(b)}, \mathbf{y}^{(b)} \in \mathbb{R}^D$, where $D = 3(m+1)^3 \cdot 3n$ is the number of scalar entries in each boundary-to-fine mapping matrix. The weight for the j-th scalar target entry is computed as

$$w_{bj} = \frac{1}{(|y_{bj}| + \varepsilon)^{\gamma}}, \tag{12}$$

where y_{bj} denotes the j-th ground-truth scalar entry of the b-th training sample, ε is a small constant (e.g., $\varepsilon = 10^{-6}$) introduced to avoid division-by-zero, and γ is the magnitude sensitivity coefficient controlling the suppression strength for large-magnitude targets.

The MASL1 loss is then obtained by applying the magnitude-adaptive weight w_{bj} to the Smooth L_1 loss over all scalar entries in the current mini-batch:

$$\mathcal{L}_{\mathrm{MASL1}} = \frac{1}{BD} \sum_{b=1}^{B} \sum_{j=1}^{D} w_{bj}\, \mathcal{L}_{\mathrm{smooth}}\left(\delta_{bj}; \beta\right), \tag{13}$$

where B is the mini-batch size, D is the number of scalar entries in each flattened boundary-to-fine mapping matrix, and $\delta_{bj} = \hat{y}_{bj} - y_{bj}$ is the prediction residual of the j-th scalar entry in the b-th training sample.

In Fig. 3, the left plot illustrates the weighting coefficient $w(t)$ for $\gamma = 0.01$. The magnitude-adaptive weighting explicitly increases the loss contribution of small-magnitude target values, a behavior not captured by the standard Smooth L_1 formulation.

4.3 Coarsened simulation based on aNeuSF

After constructing the aNeuSF shape-function matrix $\mathbf{N}_\theta^H(\mathbf{x})$ for each macro-element, the displacement solution for the linear elasticity problem (1) over the structure Ω^H can be computed within the conventional finite element analysis framework.

Let $\mathbf{d}_n$ denote the discrete nodal displacement vector to be determined on Ω^H. The displacement field $\mathbf{u}(\mathbf{x})$ throughout the macro-element is then obtained through interpolation using the aNeuSF shape-function matrix $\mathbf{N}_\theta^H(\mathbf{x})$:

$$\mathbf{u}^H(\mathbf{x}) = \mathbf{N}_\theta^H(\mathbf{x})\mathbf{d}_n. \tag{14}$$

Based on this, the strain can be obtained as:

$$\boldsymbol{\varepsilon}^H(\mathbf{x}) = \mathbf{B}_\theta^H(\mathbf{x})\, \mathbf{d}_n, \tag{15}$$

where $\mathbf{B}_\theta^H(\mathbf{x})$ denotes the derivative operator of $\mathbf{N}_\theta^H(\mathbf{x})$ with respect to $\mathbf{x}$, as follows:

$$\mathbf{B}_\theta^H(\mathbf{x}) = \frac{\partial \mathbf{N}_\theta^H(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathbf{N}^h(\mathbf{x})}{\partial \mathbf{x}}\, \mathbf{F}_\theta(\boldsymbol{\rho}). \tag{16}$$

The stiffness matrix of the macro-element $\mathbf{K}_\theta^H$ can then be obtained as follows:

$$\begin{aligned}\mathbf{K}_\theta^H &= \int_{\Omega^H} \left(\mathbf{B}_\theta^H(\mathbf{x})\right)^{\mathrm{T}} \mathbf{D}(\mathbf{x})\, \mathbf{B}_\theta^H(\mathbf{x})\, \mathrm{d}\Omega^H \\ &= \sum_{e=1}^{n_e} \mathbf{F}_\theta^{\mathrm{T}} \left(\int_{\omega_e} \left(\frac{\partial \mathbf{N}^h(\mathbf{x})}{\partial \mathbf{x}} \right)^{\mathrm{T}} \mathbf{D}_e(\mathbf{x}) \frac{\partial \mathbf{N}^h(\mathbf{x})}{\partial \mathbf{x}}\, \mathrm{d}\omega_e \right) \mathbf{F}_\theta \\ &= \sum_{e=1}^{n_e} \mathbf{K}_{\theta,e}^H.\end{aligned} \tag{17}$$

Here n_e means the total number of micro-elements in Ω^H, and $\mathbf{K}_{\theta,e}^H$ can be calculated by:

$$\mathbf{K}_{\theta,e}^H = \mathbf{F}_\theta^{\mathrm{T}} \left(\int_{\omega_e} \left(\frac{\partial \mathbf{N}^h(\mathbf{x})}{\partial \mathbf{x}} \right)^{\mathrm{T}} \mathbf{D}_e(\mathbf{x}) \frac{\partial \mathbf{N}^h(\mathbf{x})}{\partial \mathbf{x}}\, \mathrm{d}\omega_e \right) \mathbf{F}_\theta. \tag{18}$$

Let Ω be a solid structure, and let $\mathcal{M}^H$ and $\mathcal{M}^h$ denote its coarse and fine meshes, respectively. The discrete displacement vector $\mathbf{d}$ of the linear elasticity problem in Eq. (1) can then be computed as follows:

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \tag{19}$$

where

$$\mathbf{K} = \sum_{H=1}^{n_H} \mathbf{K}_\theta^H = \sum_{H=1}^{n_H} \sum_{e=1}^{n_e} \mathbf{K}_{\theta,e}^H. \tag{20}$$

Algorithm 1: Numerical coarsening using the trained aNeuSF model for linear elasticity

Input: A structure Ω, its coarse mesh $\mathcal{M}^H = \left\{\Omega^H,\ H = 1, 2, \ldots, n_H\right\}$, fine mesh $\mathcal{M}^h = \left\{\omega_e^H \mid e = 1, 2, \ldots, n_e,\ H = 1, 2, \ldots, n_H\right\}$

Output: The coarse-element shape functions $\mathbf{N}_\theta^H(\mathbf{x})$ constructed by aNeuSF, and the approximated displacement solution $\mathbf{d}$ to Eq. (1)

for $H = 1, 2, \ldots, n_H$ **do**
- Predict the boundary-to-fine mapping matrix $\mathbf{F}_\theta^H(\boldsymbol{\rho})$ using the trained aNeuSF model;
- Construct the coarse-element shape functions $\mathbf{N}_\theta^H(\mathbf{x})$ for Ω^H using Eq. (10);
- Compute the coarse-element stiffness matrix $\mathbf{K}_\theta^H$ from the fine-scale stiffness contributions $\mathbf{K}_{\theta,e}^H$ for Ω^H using Eq. (17);

end

Assemble the global stiffness matrix $\mathbf{K}$;

Solve the discretized system in Eq. (19) to obtain the approximate displacement solution $\mathbf{d}$;

4.4 Framework based on aNeuSF

Algorithm 1 shows how to solve linear elasticity problems through numerical coarsening, leveraging the aNeuSF model. The algorithm takes as input a structure Ω, defined over a coarse mesh and a corresponding fine mesh. The outputs are the coarse-element shape functions $\mathbf{N}_\theta^H(\mathbf{x})$ constructed by aNeuSF and the approximate displacement solution $\mathbf{d}$ obtained from the discretized governing equation in Eq. (1).

5 Implementation details on aNeuSF

5.1 Dataset generation

We build a new dataset named **aNeuSF-N** by randomly generating 3D voxel cubes of size $(N \times N \times N)$ with volume fractions ranging from 0 to 1. Some voxel cubes with different volume fractions and shapes when N is three are shown in Fig. 4. The dataset is generated randomly, ensuring that each cube consists of a single connected component. The final dataset contains voxel cubes with densities spanning from sparse to high-density configurations. The generated dataset is shuffled to form a set of 100,000 samples, and split into training, validation, and test sets in a 7:2:1 ratio. Employing the method of [8], we perform computational analysis on the aNeuSF-N dataset to generate the data labels required for model training. The parameters are configured as follows: fine-mesh resolution $N = 3, 4, 5$, with Young's modulus, Poisson's ratio, and density set to 1×10^5, 0.3, and 1×10^3, respectively.

5.2 Architecture of aNeuSF

The proposed aNeuSF model adopts an MLP architecture, as shown in Fig. 5. The input is a coarse-element density tensor of size $m \times m \times m$, which is flattened into an m^3-dimensional feature vector. The network output is reshaped into a matrix of size $3(m+1)^3 \times 3n$, representing the predicted boundary-to-fine mapping matrix. Since the output dimension depends on the Bézier order through the number of master nodes, we implement a lightweight aNeuSF model for each κ in practice, rather than using a single network sized for the highest order. This design preserves the efficiency benefits of lower-order interpolation while supporting the selection of Bézier orders for different accuracy-efficiency trade-offs. Accordingly, selectable Bézier orders are realized through lightweight order-specific models with different output dimensions.

5.3 Settings

We train the aNeuSF model using Pytorch 2.4.1+cu124 and implemented the simulation using C++ on a standard workstation equipped with an NVIDIA GeForce GTX 4090, an Intel Core i7-12700 2.1 GHz CPU. For the training procedure, we implement Adam as the optimizer of aNeuSF, whose learning rate is set to 0.005, the batch size to 256, and epochs to 500.

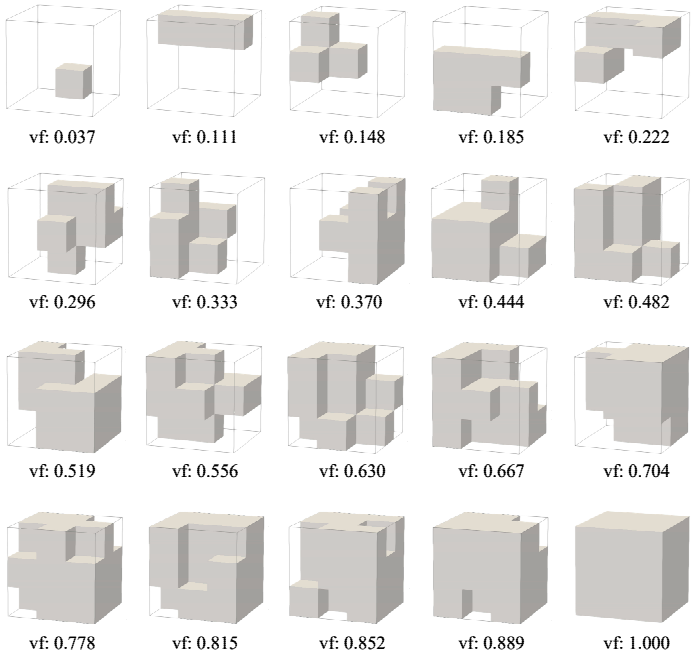


Fig. 4 Representative voxelized substructures with different volume fractions and shapes in dataset **aNeuSF-3**

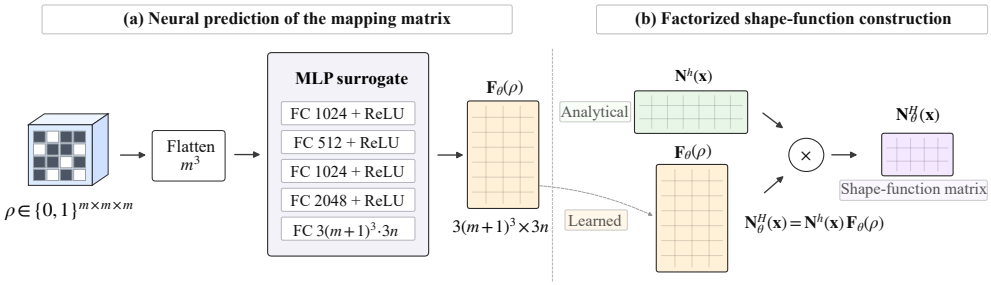


Fig. 5 Architecture and factorized construction of the proposed aNeuSF model. (a) The voxelized coarse-element density tensor $\rho \in \{0,1\}^{m\times m\times m}$ is flattened and passed through an MLP to predict the boundary-to-fine mapping matrix $\mathbf{F}_\theta(\rho) \in \mathbb{R}^{3(m+1)^3\times 3n}$. (b) The coarse-element shape-function matrix is constructed in a factorized form as $\mathbf{N}^H_\theta(\mathbf{x}) = \mathbf{N}^h(\mathbf{x})\mathbf{F}_\theta(\rho)$

6 Experiments and results

6.1 The performance of trained aNeuSF model

In this section, we take Bézier order κ=3 as an example and analyze the training process and error distributions of the trained aNeuSF on the test set of aNeuSF-3. We

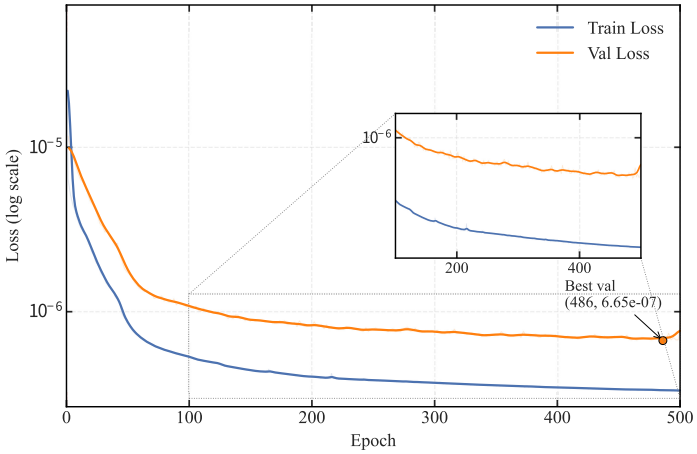


Fig. 6 Trends of loss and prediction error in the training process of aNeuSF trained on dataset aNeuSF-3

evaluate the performance of predictive models on the test set through the following error metrics: MSE, RMSE, MAE, Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). These metrics are widely used to assess model accuracy, with lower values indicating better predictive performance. To specifically evaluate the model's fundamental property, we compute the Row Sum Error (RSE) per sample, defined as the MAE between the predicted and true row-wise sums.

6.1.1 Training time cost

The training process took approximately 24.3h with 500 epochs on the training set of aNeuSF-3.

6.1.2 Convergence and error distribution

Fig. 6 shows that both the training loss and the prediction MSE of aNeuSF-3 decrease rapidly in early epochs and then plateau, reaching near-minimum values at around 400 epochs.

After training, we evaluated the trained aNeuSF model on the aNeuSF-3 test set. Fig. 7 reports the entire test-set error distributions using Empirical Cumulative Distribution Functions (ECDFs) of MAE (left) and Row Sum Error (right). To further examine sample-wise behavior, we additionally present the worst-case sample and analyze its error distribution at the single-sample level in Fig. 8.

MAE distribution. The left red curve in Fig. 7 indicates the overall performance of the aNeuSF model on the whole test set concerning MAE. With 99% of samples exhibiting errors below 0.00053, most of the predictions demonstrate high precision. From the left figure in Fig. 8, the majority of the predicted data items for the worst

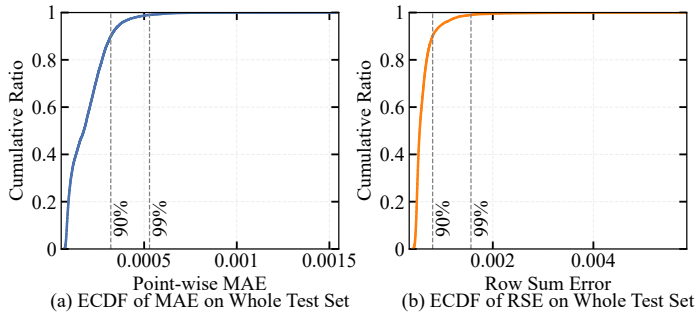


Fig. 7 Empirical cumulative distribution functions (ECDFs) of prediction errors on the aNeuSF-3 test set. (a) Distribution of the point-wise mean absolute error (MAE). (b) Distribution of the row sum error (RSE). The vertical dashed lines indicate the 90% and 99% quantiles

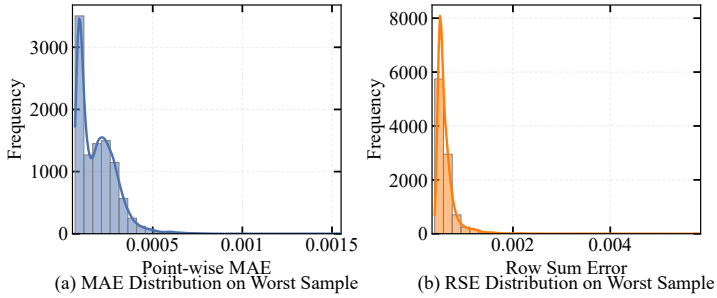


Fig. 8 Error distributions for the worst-case test sample from the aNeuSF-3 test set. (a) Distribution of the point-wise mean absolute error (MAE). (b) Distribution of the row sum error (RSE). The histograms and kernel density estimates show the single-sample error characteristics

sample exhibit very small errors, representing the highest proportion and indicating extremely low point-wise MAE errors for most data items among the worst sample.

Row sum error. Based on the physical properties of shape functions, the sum of each row in aNeuSF's predicted values should ideally equal one. This property is closely related to the partition-of-unity requirement of shape functions. To evaluate this property of aNeuSF, we plotted the cumulative curve of row-sum errors of the test set. The right curve in Fig. 7 reveals that the majority of samples exhibit very low errors. Notably, 99% of the samples have errors less than 0.00157. This distribution underscores the overall accuracy of the test set, with most samples concentrated at very low error levels. Additionally, from the right figure in Fig. 8, a large proportion of the predicted data items for the worst sample show minimal row sum errors, highlighting the dominance of low row sum errors for the majority of data points in the worst sample.

Table 1 Comparison of different neural networks in terms of test-set performance metrics

Model	Size	MSE↓ ($\times 10^{-6}$)	RMSE↓ ($\times 10^{-4}$)	MAE↓ ($\times 10^{-4}$)	MAPE↓ (%)	SMAPE↓ (%)
MLP-3	68.7M	0.806	8.98	1.94	4.67	**2.31**
aNeuSF	69.3M	**0.669**	**8.18**	**1.90**	**4.30**	2.36
MLP-5	71.4M	0.730	8.55	1.98	4.50	2.46
3D U-Net	69.1M	1.37	11.7	3.09	5.97	4.09
ResNet	49.7M	9.14	30.2	5.99	14.97	5.76

Table 2 Comparison of different loss functions in terms of test-set performance metrics

Loss	MSE↓ ($\times 10^{-6}$)	RMSE↓ ($\times 10^{-4}$)	MAE↓ ($\times 10^{-4}$)	MAPE↓ (%)	SMAPE↓ (%)
MSE loss	0.90	9.48	1.96	4.92	2.27
Smooth L_1	0.69	8.31	**1.80**	4.36	**2.14**
MASL1	**0.67**	**8.18**	1.90	**4.30**	2.36
MAMSE	1.02	10.1	2.12	5.16	2.50
MAE loss	11.2	33.4	8.24	16.1	9.67

6.2 Ablation studies

This section compares different architectures of neural networks and demonstrates the effectiveness of the MASL1. It also evaluates the impact of magnitude sensitivity coefficients (gamma), highlighting the improvements achieved with optimal gamma values.

6.2.1 Architecture of network

We compare aNeuSF with several neural network architectures: 3-layer MLP, 5-layer MLP, 3D U-Net [44], and ResNet [45]. As shown in Table 1, lower values indicate better performance across all metrics. aNeuSF outperforms most architectures, including the deeper MLP-5, which, despite having more parameters, performs worse on the test set. This highlights aNeuSF's ability to balance complexity with robust error minimization. Additionally, aNeuSF surpasses both 3D U-Net and ResNet in this task.

6.2.2 Loss function

To assess the effectiveness of the proposed MASL1 loss function, we compared it against four baselines (MSE Loss, Smooth L_1, MAMSE, and MAE Loss) using multiple regression metrics on the test set. As shown in Table 2, MASL1 achieves the lowest MSE, RMSE, and MAPE among the compared losses, while remaining competitive on MAE and SMAPE. These results suggest that the magnitude-adaptive

weighting strategy helps improve the modeling of small-magnitude entries in the predicted mapping matrix. In contrast, MSE tends to amplify large residuals because of its quadratic penalty, whereas Smooth L_1 alleviates this issue but does not explicitly account for target magnitude.

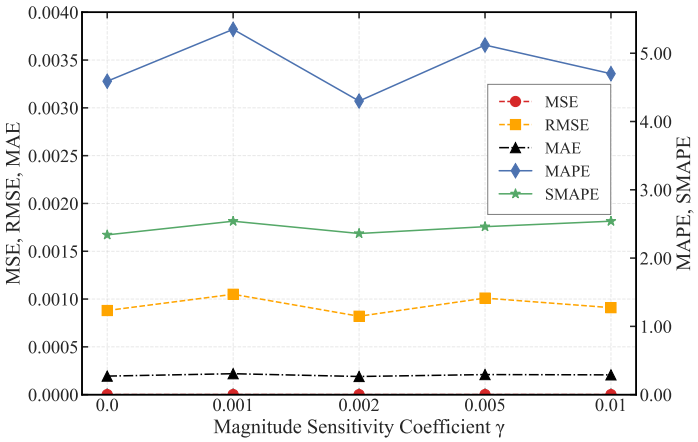


Fig. 9 Comparison of different thresholds of the magnitude sensitivity coefficient (γ) on the performance of the aNeuSF model, evaluated through five error metrics

6.2.3 Magnitude sensitivity coefficient (γ)

This section explores the impact of the Magnitude Sensitivity Coefficient (γ) with values of 0, 0.001, 0.002, 0.005, and 0.01. As shown in Fig. 9, the optimal γ value is 0.002, where most error metrics are minimized. A small, non-zero γ improves the model's ability to handle magnitude variations without sacrificing accuracy, while larger values lead to increased errors.

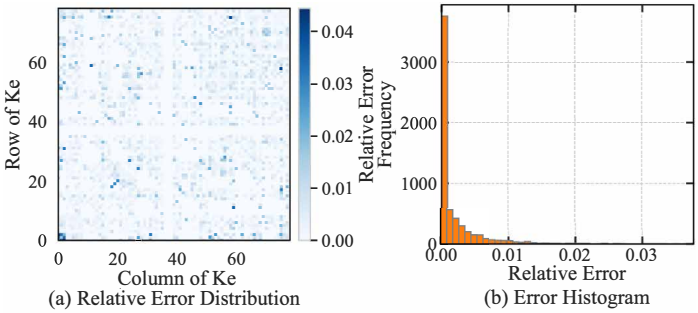


Fig. 10 Relative error analysis of the 78×78 element stiffness matrix K_e for a median aNeuSF-3 sample at $\kappa = 2$ (MAE at the 50th percentile). (a) Spatial distribution of relative errors in the K_e matrix; (b) Histogram of relative error frequencies

Table 3 Mean SMAPE of element stiffness matrices over different volume-fraction intervals on the aNeuSF-3 test set for $\kappa = 2$ and $\kappa = 3$

Volume-fraction interval	SMAPE (%)	
	$\kappa = 2$	$\kappa = 3$
$[0.9, 1]$	0.044	0.057
$[0.8, 0.9)$	0.056	0.063
$[0.7, 0.8)$	0.080	0.076
$[0.6, 0.7)$	0.114	0.095
$[0.5, 0.6)$	0.154	0.117
$[0.4, 0.5)$	0.277	0.200
$[0.3, 0.4)$	0.359	0.261
$[0.2, 0.3)$	0.512	0.380
$[0.1, 0.2)$	0.971	0.833
$[0, 0.1)$	5.422	8.558

6.3 Numerical experiments

This section presents a numerical evaluation of aNeuSF, beginning with the assessment of its accuracy and computational efficiency at the element level. The performance of aNeuSF is then compared to the conventional CBNs method in terms of both accuracy and efficiency. Following that, simulation experiments on various 3D models are presented, where the simulation accuracy is evaluated across different methods. Finally, the impact of varying Bézier orders on performance is discussed.

6.3.1 Performance comparison with conventional method

We compare our method with the conventional CBNs approach [8] in terms of computational accuracy and efficiency, since both methods construct shape functions in a similar manner and are therefore directly comparable.

Accuracy. Since the element stiffness matrix is assembled from the underlying shape functions, its accuracy directly reflects the quality of the predicted shape functions. We therefore further evaluate the prediction error of the element stiffness matrix. Table 3 and Fig. 11 show the SMAPE of the element stiffness matrix across different volume-fraction (vf) intervals on the aNeuSF-3 test set for $\kappa = 2$ and $\kappa = 3$. Overall, the error increases as vf decreases, indicating that low-density elements are more challenging to predict. In common engineering regimes with medium-to-high vf ($\text{vf} \geq 0.5$), the errors remain no more than 0.154% for $\kappa = 2$ and 0.117% for $\kappa = 3$, demonstrating stable and accurate performance. Overall, our method predicts the element stiffness matrix accurately and robustly in the medium and high vf regimes. Further, Fig. 10 visualizes the relative error distribution and histogram of the 78×78 element stiffness matrix on the worst sample at $\kappa = 2$, which shows that most relative errors remain below a low threshold, confirming that our method delivers high precision and is well-suited for practical applications in computational mechanics.

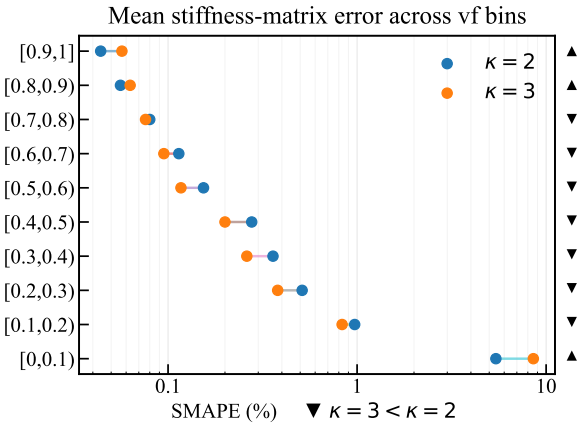


Fig. 11 SMAPE across volume-fraction (vf) bins comparing ($\kappa = 2$) and ($\kappa = 3$). Lines connect the two orders per bin; ▼/▲ indicate ($\kappa = 3 < \kappa = 2$) / ($\kappa = 3 > \kappa = 2$)

Table 4 Runtime performance of aNeuSF compared with the numerical CBNs method on the aNeuSF-5 test set with $\kappa = 3$. Total runtime, average per-sample runtime, and speedup are reported for different aNeuSF batch sizes

Method	Batch size	Total time (s)↓	Avg. time (ms)↓	Speedup (×)↑
CBNs [8]	–	71.87	7.187	1.0
aNeuSF	1	10.54	1.054	6.8
	8	1.90	0.190	37.8
	16	1.21	0.121	59.2
	32	0.88	0.088	81.5
	64	0.72	0.072	100.1
	128	0.671	0.0671	107.2
	256	0.665	0.0665	108.1
	512	0.64	0.064	111.6
	1024	**0.53**	**0.053**	**134.9**

Computational efficiency. We evaluate the computational efficiency of aNeuSF against the CBNs across various batch sizes on the test set of aNeuSF-5, which contains 10,000 samples. Table 4 shows that aNeuSF achieves a 134.9× speedup at batch size 1024, reducing runtime from 71.87s (CBNs) to 0.53s.

Moreover, Table 5 indicates that CBNs runtime grows sharply with resolution and Bézier order (e.g., from 3.147 s at level 3, $\kappa = 1$ to 71.87 s at level 5, $\kappa = 3$), whereas aNeuSF remains nearly resolution-invariant (less than 1 s). Fig. 12 shows aNeuSF speedup over the numerical CBNs method at batch size 256 across Bézier orders and resolutions. As a result, the speedup increases with problem difficulty, reaching up to 108.1× at the highest resolution.

Table 5 Runtime comparison between aNeuSF and the numerical CBNs method across different resolutions and Bézier orders. The aNeuSF results are measured using batched inference with a batch size of 256, whereas the numerical CBNs baseline is computed element by element

Method	κ	Total time (s)↓		
		Resolution 3	Resolution 4	Resolution 5
CBNs	1	3.147	8.652	18.31
	2	4.996	14.47	29.29
	3	10.94	31.10	71.87
aNeuSF	1	0.599	0.590	0.595
	2	0.600	0.602	0.613
	3	0.609	0.634	0.665

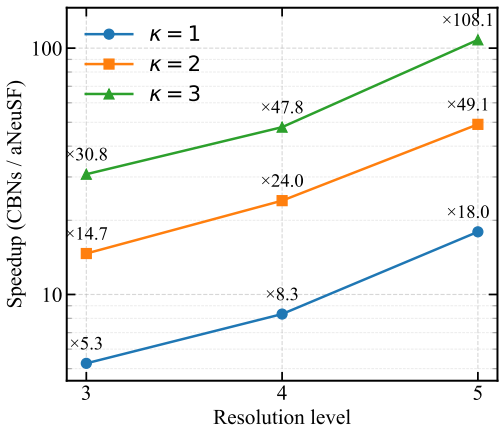


Fig. 12 aNeuSF speedup over the numerical CBNs method at batch size 256 across Bézier orders κ (1,2,3) and resolution levels (3,4,5), where each cell reports the speedup ratio

6.3.2 Simulation experiments on 3D models

To conduct a quantitative comparison between aNeuSF, the numerical CBNs method [8], and MsFEM [46] in terms of simulation accuracy metrics, we use six representative geometric models from the BrepGen furniture dataset [47] for simulation experiments. Additionally, the results are compared against fine-mesh reference solutions to assess the accuracy of each method.

Settings. The displacement fields of the resulting solutions are evaluated under the following conditions: all furniture models are fixed at the base, and a vertical uniformly distributed load of $p = 1$ is applied to the designated force-bearing surface regions. Our simulation experimental data are reported in SI (International System of Units) units. We use $\kappa = 3$ as the default Bézier order and set the fine-mesh resolution within

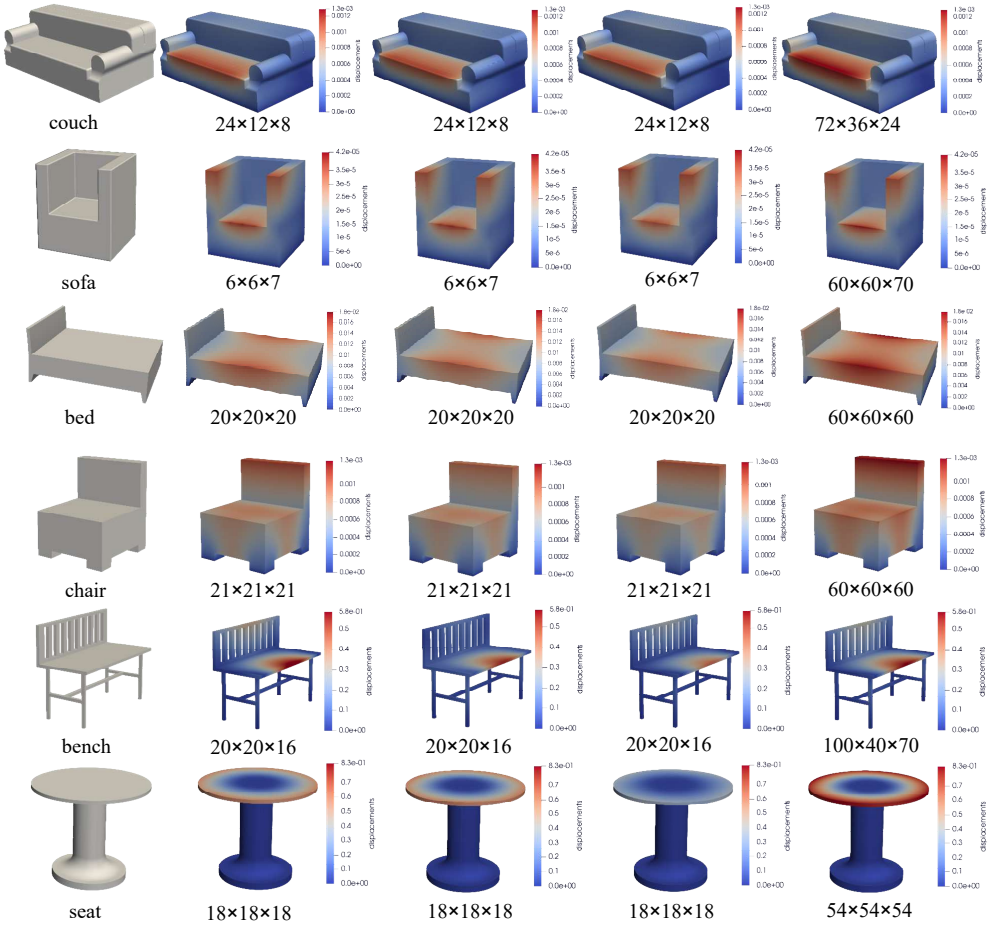
Table 6 Comparison of full-field displacement errors and top-5% displacement/stress errors for six furniture models on coarsened meshes. All errors are computed relative to the fine-mesh FEA reference. Full-field displacement errors are reported as normalized errors (%), whereas top-5% errors are reported as SMAPE (%)

Model	Method	Full disp. error (%)	Top-5% error (%)	
			Disp.	Stress
couch	CBNs	4.67	9.05	49.4
	aNeuSF	4.79	9.13	49.3
	MsFEM	4.21	9.41	50.5
sofa	CBNs	0.66	0.45	10.7
	aNeuSF	0.78	0.46	10.9
	MsFEM	0.96	0.96	20.3
bed	CBNs	9.98	15.8	38.3
	aNeuSF	10.0	17.3	57.3
	MsFEM	12.6	22.6	35.8
chair	CBNs	5.73	9.04	33.9
	aNeuSF	7.76	16.9	38.6
	MsFEM	9.08	13.2	37.3
bench	CBNs	3.20	7.81	24.7
	aNeuSF	6.41	6.58	30.3
	MsFEM	7.48	25.6	37.3
seat	CBNs	8.05	33.7	29.3
	aNeuSF	7.52	31.0	31.0
	MsFEM	16.7	81.7	62.4

each coarse element (Fig. 13) to $m = 3$. Each 3D model is partitioned into coarse elements during the FEA preprocessing step.

Simulation accuracy. The left column in Fig. 13 shows six furniture models from top to bottom. The displacement cloud maps provide a qualitative comparison against the fine-mesh FEA reference. Across the six models, aNeuSF generally reproduces the global deformation patterns more faithfully than MsFEM and, in several cases, remains visually close to the numerical CBNs results computed on the same coarsened meshes.

Table 6 further reports the full-field displacement errors together with the top-5% displacement and stress errors, where the top-5% region emphasizes the most critical parts of the solution field. Overall, aNeuSF achieves displacement accuracy that is generally competitive with CBNs and is more reliable than MsFEM on most models. In particular, it remains close to CBNs on couch and sofa, outperforms both CBNs and MsFEM on bench and seat in the top-5% displacement metric, and still performs substantially better than MsFEM on the most difficult case, seat. For stress, the trend is more case-dependent: aNeuSF stays close to CBNs on couch and sofa and improves upon MsFEM on sofa, bench, and seat, while larger discrepancies remain on bed.



(a) Models (b) Numerical CBNs (c) aNeuSF (d) MsFEM (e) Fine mesh

Fig. 13 Column (a) shows the six furniture models (from top to bottom): couch, sofa, bed, chair, bench, and seat. (b) presents the displacement fields computed on the coarsened meshes generated by the numerical CBNs method [8]. (c) and (d) show the results produced by our method and by MsFEM (Multiscale FEM) [46], respectively, using the same coarsened meshes. (e) reports the reference displacement fields obtained from fine-mesh FEA. Overall, our method produces displacement contours that closely match those of the CBNs approach and exhibit higher fidelity to the fine-mesh reference than MsFEM across all models

6.4 Discussion of Bézier orders

In this section, we discuss the impact of different Bézier orders in terms of error distributions, computational efficiency, and simulation accuracy.

Table 7 Top-5% displacement relative errors of aNeuSF on six furniture models under different Bézier orders. The model is trained on aNeuSF-3. The Fine mesh columns report errors relative to the fine-mesh FEA reference, whereas the CBNs columns report errors relative to the CBNs solution

Model	$\kappa = 1$		$\kappa = 2$		$\kappa = 3$	
	Fine mesh	CBNs	Fine mesh	CBNs	Fine mesh	CBNs
couch	0.0016	1.47×10^{-5}	0.00154	7.69×10^{-6}	0.00153	1.26×10^{-5}
sofa	6.95×10^{-6}	7.98×10^{-6}	3.61×10^{-6}	6.62×10^{-8}	3.30×10^{-6}	5.81×10^{-8}
bed	0.00718	0.00305	0.0055	0.00213	0.00391	0.000472
chair	0.00906	0.00477	0.00585	0.00265	0.00597	0.00287
bench	0.0123	0.00976	0.0109	0.0093	0.000911	0.00165
seat	0.0556	0.0189	0.041	0.0247	0.0195	0.00149

6.4.1 Error distribution comparison

To further analyze the impact of Bézier order on prediction quality, we compare the error distributions under different κ values. From Table 3, it is evident that $\kappa = 3$ generally produces lower errors than $\kappa = 2$ across most vf ranges, with the exception of the particularly challenging low-vf interval. This suggests that $\kappa = 3$ performs better in the medium-to-high vf ranges, where the element stiffness matrix is more stable and predictable. However, in the low-density region with vf $\in [0, 0.1)$, $\kappa = 2$ yields slightly lower errors compared to $\kappa = 3$, which is expected due to the complex behavior and stiffness challenges in this extreme regime. This degradation is expected because extremely low-volume-fraction samples contain only sparse structural support, making the resulting stiffness operators more sensitive to small prediction errors in the learned mapping matrix. Nonetheless, across the broader range of vf values, our method demonstrates robust and accurate performance for both $\kappa = 2$ and $\kappa = 3$, with $\kappa = 3$ generally offering superior accuracy in most scenarios.

6.4.2 Computational efficiency comparison

When comparing different values of κ in Table 5, the CBNs method shows a significant increase in runtime with higher κ, due to the rising computational cost of numerical assembly as the interpolation order increases. For example, the runtime increases drastically from 3.147 s at level 3 with $\kappa = 1$ to 71.87 s at level 5 with $\kappa = 3$. On the other hand, aNeuSF demonstrates nearly constant runtime, with only a slight increase as κ increases. This leads to a significant speedup of aNeuSF over CBNs as the interpolation order grows, with the largest gains observed for higher-order cases, as shown in Fig. 12.

6.4.3 Simulation accuracy comparison

The results presented in Table 7 and Fig. 14 compare the Top-5% displacement relative errors of six furniture models computed using aNeuSF with different Bézier orders (κ

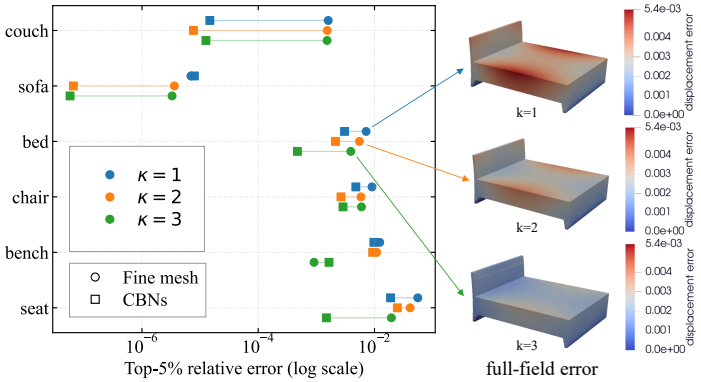


Fig. 14 Displacement errors of aNeuSF on six furniture models under different Bézier orders. Left: Top-5% displacement relative error, where circles denote errors relative to the fine-mesh FEA reference, squares denote errors relative to the CBNs baseline, and connecting segments indicate the difference between the two references (lower is better). Right: full-field displacement errors relative to the fine-mesh FEA reference

= 1, 2, 3). The errors are reported in two columns: the first compares aNeuSF to the fine-mesh reference solution, and the second compares aNeuSF to the CBNs method.

Across all models, aNeuSF's relative errors remain small when compared to the fine-mesh reference. As the Bézier order κ increases from one to three, we observe a slight improvement in accuracy for most models, suggesting that higher κ values help refine the approximation of displacement fields. For more complex models, such as bench and seat, the improvements are less pronounced, but the trend still holds: higher Bézier orders result in slightly lower errors compared to lower orders. This indicates that aNeuSF maintains accuracy even for more challenging cases, with the errors still within acceptable limits when compared to both the fine mesh and the CBNs method.

7 Conclusion and future work

In this work, we presented **aNeuSF**, a factorized neural surrogate for efficient shape-function construction in numerical coarsening. The proposed method decomposes the construction of coarse-element shape functions into an analytical Bézier-order interpolator and a learned boundary-to-fine mapping predicted from coarse-element density fields. This factorized analytical-learned formulation preserves essential shape-function properties while supporting selectable Bézier orders for different accuracy-efficiency trade-offs. In addition, we studied a magnitude-adaptive Smooth L_1 loss (MASL1) for training aNeuSF, which helps improve the modeling of small-magnitude entries in the predicted mapping matrix. Numerical experiments showed that aNeuSF achieves competitive accuracy together with substantial batched acceleration compared with conventional numerical coarsening baselines.

The main limitations of this work are as follows. First, the current framework is developed for linear-elastic static analysis and voxelized coarse-element density

fields at fixed input resolution. Although this setting is sufficient to validate the proposed factorized surrogate for shape-function construction, extending the framework to nonlinear materials and large-deformation regimes remains future work. Second, the proposed approach is data-driven and therefore requires labeled training data generated from numerical coarsening analysis. Third, as indicated by the results in Table 3, the current method is more reliable for medium-to-high volume fractions, whereas its accuracy decreases for structures with very low volume fractions.

A promising direction for future work is to extend the current learning pipeline beyond voxelized density fields to more general geometry representations, including implicit surface representations via signed distance functions (SDFs) [48], boundary representations (B-rep) commonly used in CAD [49], and point-cloud-based encodings [50]. Neural radiance-field techniques such as NeRF [51] may also be explored as a more speculative direction. Based on such representations, it would be valuable to construct neural mappings to multiscale shape functions or even directly to coarse-element stiffness operators. Another important direction is to improve generalization to geometries with smooth and freeform boundaries. In addition, it will be worthwhile to extend the proposed framework to nonlinear simulation settings, including hyperelastic constitutive behavior [52], nonlinear material responses such as elastoplasticity and fracture [53], and geometric nonlinearity induced by large deformations [54].

Declarations

Author contributions. **Xuwei Wang**: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Visualization, Writing-Original Draft, Writing-Review & Editing. **Wei Chen**: Methodology, Investigation, Data curation, Writing-Review & Editing. **Haroon Ijaz**: Writing-Review & Editing. **Ming Li**: Supervision, Conceptualization, Methodology, Validation, Formal Analysis, Project Administration, Funding Acquisition, Resources, Writing-Review & Editing.

Funding. This work was supported by the NSF of China (No. 62372401) and the "Pioneer" and "Leading Goose" R&D Program of Zhejiang Province (No. 2024C01103,2025C01086).

Competing interests. The authors declare no conflict of interest.

Data availability. Data will be available upon request.

References

[1] Panchal, J.H., Kalidindi, S.R., McDowell, D.L.: Key computational modeling issues in integrated computational materials engineering. Computer-Aided Design **45**(1), 4–25 (2013) https://doi.org/10.1016/j.cad.2012.06.006

[2] Alexandersen, J., Lazarov, B.S.: Topology optimisation of manufacturable microstructural details without length scale separation using a spectral coarse

basis preconditioner. Computer Methods in Applied Mechanics and Engineering **290**, 156–182 (2015) https://doi.org/10.1016/j.cma.2015.02.028

[3] Hirota, G., Fisher, S., State, A.: An improved finite-element contact model for anatomical simulations. The Visual Computer **19**, 291–309 (2003) https://doi.org/10.1007/s00371-002-0188-5

[4] Qi, D., Milef, N., De, S.: Divided Voxels: an efficient algorithm for interactive cutting of deformable objects. The Visual Computer **37**, 1113–1127 (2021) https://doi.org/10.1007/s00371-020-01856-y

[5] Wang, S., Xiang, N., Xia, Y., You, L., Zhang, J.: Real-time surface manipulation with C^1 continuity through simple and efficient physics-based deformations. The Visual Computer **37**(9–11), 2741–2753 (2021) https://doi.org/10.1007/s00371-021-02169-4

[6] Chen, D., Levin, D.I., Matusik, W., Kaufman, D.M.: Dynamics-aware numerical coarsening for fabrication design. ACM Transactions on Graphics (TOG) **36**(4), 1–15 (2017) https://doi.org/10.1145/3072959.3073669

[7] Chen, J., Bao, H., Wang, T., Desbrun, M., Huang, J.: Numerical coarsening using discontinuous shape functions. ACM Transactions on Graphics (TOG) **37**(4), 1–12 (2018) https://doi.org/10.1145/3197517.3201386

[8] Li, M., Hu, J.: Analysis of heterogeneous structures of non-separated scales using curved bridge nodes. Computer Methods in Applied Mechanics and Engineering **392**, 114582 (2022) https://doi.org/10.1016/j.cma.2022.114582

[9] Yvonnet, J.: Computational Homogenization of Heterogeneous Materials with Finite Elements vol. 258. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-18383-7

[10] Tan, V.B.C., Raju, K., Lee, H.P.: Direct fe2 for concurrent multilevel modelling of heterogeneous structures. Computer Methods in Applied Mechanics and Engineering **360**, 112694 (2020) https://doi.org/10.1016/j.cma.2019.112694

[11] Le, M.V., Yvonnet, J., Feld, N., Detrez, F.: The coarse mesh condensation multiscale method for parallel computation of heterogeneous linear structures without scale separation. Computer Methods in Applied Mechanics and Engineering **363**, 112877 (2020) https://doi.org/10.1016/j.cma.2020.112877

[12] Zhu, L., Wang, X., Zhong, W., Li, M.: Learning microstructure–property mapping via label-free 3d convolutional neural network. The Visual Computer **40**(10), 7025–7044 (2024) https://doi.org/10.1007/s00371-024-03411-5

[13] Wei, L., Freris, N.M.: Multi-scale graph neural network for physics-informed fluid simulation. The Visual Computer **41**, 1171–1181 (2025) https://doi.org/10.1007/

s00371-024-03402-6

[14] Huang, M., Du, Z., Liu, C., Zheng, Y., Cui, T., Mei, Y., Li, X., Zhang, X., Guo, X.: Problem-independent machine learning (piml)-based topology optimization—a universal approach. Extreme Mechanics Letters **56**, 101887 (2022) https://doi.org/10.1016/j.eml.2022.101887

[15] Huang, M., Cui, T., Liu, C., Du, Z., Zhang, J., He, C., Guo, X.: A problem-independent machine learning (piml) enhanced substructure-based approach for large-scale structural analysis and topology optimization of linear elastic structures. Extreme Mechanics Letters **63**, 102041 (2023) https://doi.org/10.1016/j.eml.2023.102041

[16] Huang, M., Liu, C., Guo, Y., Zhang, L., Du, Z., Guo, X.: A mechanics-based data-free problem independent machine learning (piml) model for large-scale structural analysis and design optimization. Journal of the Mechanics and Physics of Solids **193**, 105893 (2024) https://doi.org/10.1016/j.jmps.2024.105893

[17] Ma, X., Huang, M., Du, Z., Guo, Y., Liu, C., Mei, Y., Guo, X.: A high-performance parallel algorithm based on problem independent machine learning (piml) for large-scale topology optimization. Acta Mechanica Sinica **42**(3), 425942 (2026) https://doi.org/10.1007/s10409-025-25942-x

[18] Ni, N., Xu, Q., Li, Z., Fu, X.-M., Liu, L.: Numerical coarsening with neural shape functions. In: Computer Graphics Forum, vol. 42, p. 14736 (2023). https://doi.org/10.1111/cgf.14736 . Wiley Online Library

[19] Kharevych, L., Mullen, P., Owhadi, H., Desbrun, M.: Numerical coarsening of inhomogeneous elastic materials. ACM Transactions on graphics (TOG) **28**(3), 1–8 (2009) https://doi.org/10.1145/1531326.1531357

[20] Nesme, M., Kry, P.G., Jeřábková, L., Faure, F.: Preserving topology and elasticity for embedded deformable models. In: ACM SIGGRAPH 2009 Papers, pp. 1–9 (2009). https://doi.org/10.1145/1576246.1531358

[21] Torres, R., Rodríguez, A., Espadero, J.M., Otaduy, M.A.: High-resolution interaction with corotational coarsening models. ACM Transactions on Graphics (TOG) **35**(6), 1–11 (2016) https://doi.org/10.1145/2980179.2982414

[22] Chen, J., Budninskiy, M., Owhadi, H., Bao, H., Huang, J., Desbrun, M.: Material-adapted refinable basis functions for elasticity simulation. ACM Transactions on Graphics (TOG) **38**(6), 1–15 (2019) https://doi.org/10.1145/3355089.3356567

[23] Chen, W., Li, M.: Coarsened c1-continuous shape functions using b-spline patches for simulating nonlinear heterogeneous structures. Computer Methods in Applied Mechanics and Engineering **421**, 116808 (2024) https://doi.org/10.1016/j.cma.2024.116808

[24] Chen, Y.J., Levin, D.I., Kaufmann, D., Ascher, U., Pai, D.K.: Eigenfit for consistent elastodynamic simulation across mesh resolution. In: Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 1–13 (2019). https://doi.org/10.1145/3309486.3340248

[25] Farin, G.: Curves and Surfaces for CAGD: a Practical Guide, 5th edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)

[26] Zhang, Y., Shen, G., Wang, Q., Qian, Y., Wei, M., Qin, J.: Geobi-gnn: geometry-aware bi-domain mesh denoising via graph neural networks. Computer-Aided Design **144**, 103154 (2022) https://doi.org/10.1016/j.cad.2021.103154

[27] Narayanan, A., Pan, Y., Persson, P.-O.: Learning topological operations on meshes with application to block decomposition of polygons. Computer-Aided Design **175**, 103744 (2024) https://doi.org/10.1016/j.cad.2024.103744

[28] Yang, X., Li, M., Zhu, L., Zhong, W.: Evolutionary Discrete Multi-Material Topology Optimization Using CNN-Based Simulation Without Labeled Training Data. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 2: 41st Computers and Information in Engineering Conference (CIE), pp. 002–02027 (2021). https://doi.org/10.1115/DETC2021-68487

[29] Meng, Z., Lv, S., Gao, Y., Zhong, C., An, K.: Data-driven reliability-based topology optimization by using the extended multi scale finite element method and neural network approach. Computer Methods in Applied Mechanics and Engineering **438**, 117837 (2025) https://doi.org/10.1016/j.cma.2025.117837

[30] Ijaz, H., Wang, X., Chen, W., Lin, H., Li, M.: Physically reliable 3d styled shape generation via structure-aware topology optimization in unified latent space. Computer-Aided Design **183**, 103864 (2025) https://doi.org/10.1016/j.cad.2025.103864

[31] Romero, C., Casas, D., Chiaramonte, M., Otaduy, M.A.: Learning contact deformations with general collider descriptors. In: SIGGRAPH Asia 2023 Conference Papers, pp. 1–10 (2023). https://doi.org/10.1145/3610548.3618229

[32] Lei, Z., Liang, Y., Cheng, G., Yang, D., Chen, G.: Stress-related discrete variable topology optimization with handling non-physical stress concentrations. Computer Methods in Applied Mechanics and Engineering **431**, 117293 (2024) https://doi.org/10.1016/j.cma.2024.117293

[33] Zhu, L., Wang, X., Zhong, W., Li, M.: Learning microstructure–property mapping via label-free 3d convolutional neural network. The Visual Computer **40**(10), 7025–7044 (2024) https://doi.org/10.1007/s00371-024-03411-5

[34] Chen, D., Levin, D.I., Sueda, S., Matusik, W.: Data-driven finite elements for

geometry and material design. ACM Transactions on Graphics (TOG) **34**(4), 1–10 (2015) https://doi.org/10.1145/2766889

[35] Fulton, L., Modi, V., Duvenaud, D., Levin, D.I., Jacobson, A.: Latent-space dynamics for reduced deformable simulation. In: Computer Graphics Forum, vol. 38, pp. 379–391 (2019). https://doi.org/10.1111/cgf.13645 . Wiley Online Library

[36] Romero, C., Casas, D., Pérez, J., Otaduy, M.: Learning contact corrections for handle-based subspace dynamics. ACM Transactions on Graphics (ToG) **40**(4), 1–12 (2021) https://doi.org/10.1145/3450626.3459875

[37] Shen, S., Yin, Y., Shao, T., Wang, H., Jiang, C., Lan, L., Zhou, K.: High-order differentiable autoencoder for nonlinear model reduction. arXiv preprint arXiv:2102.11026 (2021) https://doi.org/10.48550/arXiv.2102.11026

[38] Luo, R., Shao, T., Wang, H., Xu, W., Chen, X., Zhou, K., Yang, Y.: Nnwarp: Neural network-based nonlinear deformation. IEEE transactions on visualization and computer graphics **26**(4), 1745–1759 (2018) https://doi.org/10.1109/TVCG.2018.2881451

[39] Kim, B., Azevedo, V.C., Thuerey, N., Kim, T., Gross, M., Solenthaler, B.: Deep fluids: A generative network for parameterized fluid simulations. In: Computer Graphics Forum, vol. 38, pp. 59–70 (2019). https://doi.org/10.1111/cgf.13619 . Wiley Online Library

[40] Chu, M., Thuerey, N.: Data-driven synthesis of smoke flows with cnn-based feature descriptors. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017) https://doi.org/10.1145/3072959.3073643

[41] Lahner, Z., Cremers, D., Tung, T.: Deepwrinkles: Accurate and realistic clothing modeling. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 667–684 (2018). https://doi.org/10.1007/978-3-030-01225-0_41

[42] Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nature machine intelligence **3**(3), 218–229 (2021) https://doi.org/10.1038/s42256-021-00302-5

[43] Hughes, T.J.R.: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications, Mineola, NY (2000)

[44] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28 . Springer

[45] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[46] Hou, T.Y., Wu, X.-H.: A multiscale finite element method for elliptic problems in composite materials and porous media. Journal of computational physics **134**(1), 169–189 (1997) https://doi.org/10.1006/jcph.1997.5682

[47] Xu, X., Lambourne, J., Jayaraman, P., Wang, Z., Willis, K., Furukawa, Y.: Brepgen: A b-rep generative diffusion model with structured latent geometry. ACM Transactions on Graphics (TOG) **43**(4), 1–14 (2024) https://doi.org/10.1145/3658129

[48] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 165–174 (2019). https://doi.org/10.1109/CVPR.2019.00025

[49] Lambourne, J.G., Willis, K.D., Jayaraman, P.K., Sanghi, A., Meltzer, P., Shayani, H.: Brepnet: A topological message passing system for solid models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12773–12782 (2021). https://doi.org/10.1109/CVPR46437.2021.01258

[50] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017). https://doi.org/10.1109/CVPR.2017.16

[51] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) https://doi.org/10.1145/3503250

[52] Franke, M., Klein, D.K., Weeger, O., Betsch, P.: Advanced discretization techniques for hyperelastic physics-augmented neural networks. Computer Methods in Applied Mechanics and Engineering **416**, 116333 (2023) https://doi.org/10.1016/j.cma.2023.116333

[53] Zong, Z., Li, X., Li, M., Chiaramonte, M.M., Matusik, W., Grinspun, E., Carlberg, K., Jiang, C., Chen, P.Y.: Neural stress fields for reduced-order elastoplasticity and fracture. In: SIGGRAPH Asia 2023 Conference Papers, pp. 1–11 (2023). https://doi.org/10.1145/3610548.3618207

[54] Luo, R., Shao, T., Wang, H., Xu, W., Chen, X., Zhou, K., Yang, Y.: Nnwarp: Neural network-based nonlinear deformation. IEEE Transactions on Visualization and Computer Graphics **26**, 1745–1759 (2020) https://doi.org/10.1109/TVCG.2018.2881451