Engineering with Computers (2023) 39:943–958
https://doi.org/10.1007/s00366-022-01717-3

ORIGINAL ARTICLE

A CNN-based surrogate model of isogeometric analysis in nonlocal flexoelectric problems

Qimin Wang[2] · **Xiaoying Zhuang**[1,2]

Received: 3 August 2021 / Accepted: 30 June 2022 / Published online: 25 August 2022

Abstract
We proposed a convolutional neural network (CNN)-based surrogate model to predict the nonlocal response for flexoelectric structures with complex topologies. The input, i.e. the binary images, for the CNN is obtained by converting geometries into pixels, while the output comes from simulations of an isogeometric (IGA) flexoelectric model, which in turn exploits the higher-order continuity of the underlying non-uniform rational B-splines (NURBS) basis functions to fast computing of flexoelectric parameters, e.g., electric gradient, mechanical displacement, strain, and strain gradient. To generate the dataset of porous flexoelectric cantilevers, we developed a NURBS trimming technique based on the IGA model. As for CNN construction, the key factors were optimized based on the IGA dataset, including activation functions, dropout layers, and optimizers. Then the cross-validation was conducted to test the CNN's generalization ability. Last but not least, the potential of the CNN performance has been explored under different model output sizes and the corresponding possible optimal model layout is proposed. The results can be instructive for studies on deep learning of other nonlocal mech-physical simulations.

Keywords Convolutional neural network · Isogeometric analysis · NURBS trimming technique · Nonlocal flexoelectricity

1 Introduction

Unlike piezoelectricity coupling electric polarization directly with the strain, flexoelectricity is a nonlocal coupling of the polarization to the strain gradient, as shown in Fig. 1. While piezoelectricity exists only in non-centrosymmetric materials, flexoelectricity shows in nearly any material [27, 53]. Promising applications of flexoelectricity have been emerging like energy harvesters absorbing the mechanical vibrations [22, 32], actuators compatible for the semiconductor industries [8], self-powering microsensors [28, 31], as well as domain tailoring and polarization switching [17, 30]. In nano-scale systems, flexoelectricity is pronounced due to the arised large nonlocal strain gradients [2]. Simulation of flexoelectric materials is complicated due to the C^1 continuity requirement in the weak form. This numerical prerequisite complicates the implementation of the classical finite element methods based on Lagrange polynomials. Alternatively, the meshfree methods as presented in [3, 40] or isogeometric analysis (IGA) [21] can be used which intrinsically fulfill the C^1 requirement.

However, simulating simple bulk materials is not yet enough to excavate the potentials of the nonlocal flexoelectricity. Possible mechanisms of enhanced flexoelectricity in dielectrics mainly includes inner strain, polar nano-regions, non-crystalline polar phases, residual ferroelectricity, and surface piezoelectricity [12, 45]. Zhang et al. [55] found that the surface piezoelectricity can contribute about 70% of the enhanced flexoelectricity in $BaTiO_3$ ceramics. Moreover, the shape and configuration of materials have been reported affecting the piezoelectric and the flexoelectric effects [26, 35, 42]. To better develop flexoelectric devices based on the nonlocal electro-mechanical coupling effects, topology optimization considering the surface piezoelectricity has proved feasible for piezoelectric structures [34] and for flexoelectric structures [14, 36]. As meshfree methods tend to be computationally more expensive, IGA have advantages in representing complex boundary representations [15] when

✉ Xiaoying Zhuang
zhuang@iop.uni-hannover.de

[1] Department of Geotechnical Engineering, College of Civil Engineering, Tongji University, Shanghai 200092, China

[2] Chair of Computational Science and Simulation Technology, Institute of Photonics, Faculty of Mathematics and Physics, Leibniz University Hannover, 30167 Hannover, Niedersachsen, Germany

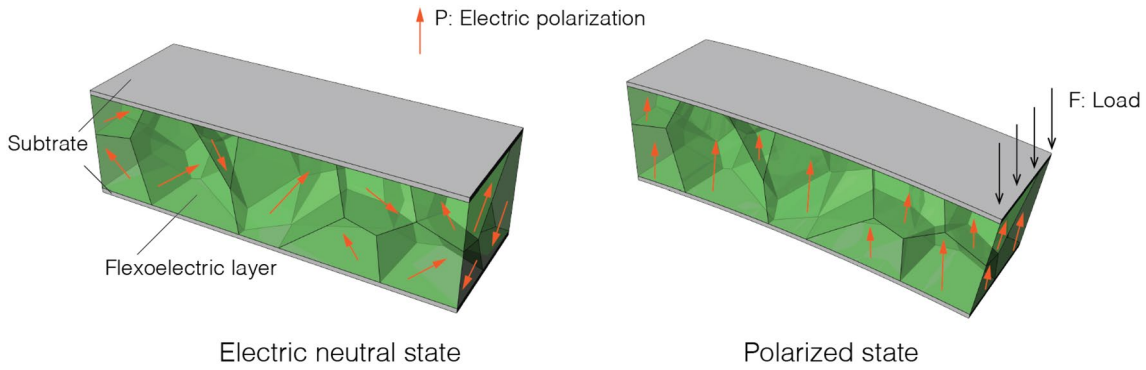


Fig. 1 Illustration of the flexoelectric effect in a cantilever energy harvester

adapting some improvements such as more sophisticated splines [37] or NURBS trimming technology [24].

Optimizing flexoelectricity with complex topologies required repetitive yet expensive simulations. Surrogate modeling serves right efficient for such problems, dramatically decreasing the computational cost by approximating the input-output relation of a system. Namely, given input parameters, e.g., initial/boundary/operational conditions, the quantities of interest (QoIs), such as electric potential, stress/strain, and their integrals/gradients can be obtained rapidly without conducting the simulations. The existing surrogate modeling approaches can be roughly categorized into two classes: projection-based reduced order models (ROMs) and data-fit models [7]. Common surrogate modeling techniques include: (i) Response Surface Models [33], (ii) Kriging Methods [11, 39], (iii) Radial Basis Functions [9, 51], and (v) Support Vector Regression [6, 46]. With the popularity of machine learning (ML), training surrogate models is also posed as a supervised learning problem [56], via Artificial Neural Networks (ANN) [5, 13, 52]. The Deep Neural Network (DNN), also called Deep learning, provides a better representation of non-linear input–output relationships compared to classical ANN [4]. Hamdia et al. [19] proved the accuracy of deep learning in solving flexoelectric problems. Goswami et al. [18] proposed a physics-informed DNN for modeling complex topologies like a fracture. Since first proposed in 1998 [29], the Convolutional Neural Network (CNN) has shown outstanding learning abilities in a wide range of applications including natural language [25, 54], speech recognition [1, 43], and computer vision [20, 41].

Although ML models have also shown promise in many disciplines outside the realm of computer science (e.g., aquatic sciences [23], atmospheric science [38], and biomedical science [48]), the capability of CNN in extracting geometry–physics linkages, at present, is still difficult to satisfy the engineering applications, especially in the nonlocal flexoelectricity. Inspired by the exciting predicting performance of CNNs in other fields, in this paper, we developed a CNN-based surrogate model for the efficient analysis of flexoelectric structures with complex topology. Our method further demonstrated the underlying potential of optimizing flexoelectricity.

The paper is arranged as follows: Sect. 2 summarizes the boundary value problem (BVP) and associated IGA formulation. Section 3 describes the CNN model before we present several benchmark problems in Sect. 4. The manuscript finishes with some concluding remarks in Sect. 5.

2 IGA simulation of flexoelectricity

2.1 Governing equation for flexoelectricity

The weak form of flexoelectricity for a linear dielectric solid can be derived from the electric enthalpy density h which is a function of the linear strain tensor ε_{ij}, the electric field E_i $h\left(\varepsilon_{ij}, E_i\right)$ and the strain gradient $\varepsilon_{jk,l}$ [3]:

$$h\left(\varepsilon_{ij}, E_i, \varepsilon_{jk,l}\right) = \frac{1}{2}C_{ijkl}\varepsilon_{ij}\varepsilon_{kl} - e_{ikl}E_i\varepsilon_{kl} - \mu_{ijkl}E_i\varepsilon_{jk,l} - \frac{1}{2}\kappa_{ij}E_iE_j, \tag{1}$$

where e_{ikl}, κ_{ij} and C_{ijkl} denote the piezoelectricity tensor, the dielectric tensor and the elasticity tensor, respectively; $\mu_{ijkl} = d_{ijkl} - f_{ijkl}$ is the flexoelectric tensor which includes the direct d_{ijkl} and the reverse flexoelectricity tensor f_{ijkl}, respectively. Integrating h over the domain Ω, we obtain the total electric enthalpy

$$H = \frac{1}{2}\int_{\Omega}\left(\widehat{\sigma}_{ij}\varepsilon_{ij} + \widetilde{\sigma}_{ij}\varepsilon_{ij,k} - \widehat{D}_iE_i\right)\mathrm{d}\Omega, \tag{2}$$

where $\widehat{\sigma}_{ij}$, $\widetilde{\sigma}_{ij}$ and $\widehat{D}_i$ are defined as

$$\widehat{\sigma}_{ij} = \frac{\partial H}{\partial \varepsilon_{ij}}, \quad \widetilde{\sigma}_{ij} = \frac{\partial H}{\partial \varepsilon_{ij,k}}, \quad \widehat{D}_i = \frac{\partial H}{\partial E_i}. \tag{3}$$

The system kinetic energy K_E and the external work W_e can be written as

$$K_E = \frac{1}{2}\int_{\Omega} \rho \dot{u}_i \dot{u}_i \mathrm{d}\Omega, \quad W_e = \int_{\Gamma_t} \bar{t}_i u_i \mathrm{d}S - \int_{\Gamma_D} \omega\theta \mathrm{d}S, \tag{4}$$

where ρ is the density, θ and ω are the prescribed electric potential and surface charge density, and u_i and $\bar{t}_i$ are the prescribed mechanical displacements and traction. Using Hamilton principle and neglecting, the damping term leads to

$$\delta \int_{t_1}^{t_2} \left(K_E - H + W_{ext}\right)\mathrm{d}t = 0. \tag{5}$$

Substituting Eqs. (2)–(4) into Eq. (5), we have

$$\delta \int_{t_1}^{t_2} \left(\frac{1}{2}\int_{\Omega} \rho \dot{u}_i \dot{u}_i \mathrm{d}\Omega - \frac{1}{2}\int_{\Omega} \left(\hat{\sigma}_{ij}\varepsilon_{ij} + \tilde{\sigma}_{ijk}\varepsilon_{ij,k} - \hat{D}_i E_i\right)\mathrm{d}\Omega + \int_{\Gamma_t} \bar{t}_i u_i \mathrm{d}S - \int_{\Gamma_D} \omega\theta \mathrm{d}S \right)\mathrm{d}t = 0. \tag{6}$$

By reordering operations and ignoring the contribution of inertial forces to the static problems, the final weak form of the governing equation is

$$\int_{\Omega} \left(C_{ijkl}\delta\varepsilon_{ij}\varepsilon_{kl} - e_{kij}E_k\delta\varepsilon_{ij} - \mu_{ijkl}E_l\delta\varepsilon_{ij,k} - \kappa_{ij}\delta E_i E_j\right)\mathrm{d}\Omega - \int_{\Gamma_t} \bar{t}_i \delta u_i \mathrm{d}S + \int_{\Gamma_D} \omega\delta\theta \mathrm{d}S = 0. \tag{7}$$

2.2 IGA formulation

Our IGA is based on Non-Uniform rational B-splines (NURBS), which in turn originated from B-splines. The basis functions $N_{i,p}$ of p-order B-splines with $(n+1)$ control points are given as

$$\begin{cases} N_{i,p}(\xi) = \dfrac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \dfrac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \\ N_{i,0}(\xi) = \begin{cases} 1, & \xi_i \le \xi < \xi_{i+1}, \\ 0, & \text{others}. \end{cases} \end{cases} \tag{8}$$

where ξ is a knot vector in the parametric space. Introducing the weights ω_i to B-splines base functions, the mechanical displacement $\mathbf{u}$ and the electric potential ϕ on a p-order NURBS surface with $(m+1)(n+1)$ control points can be defined as

$$\mathbf{u}_{\xi,\eta} = \sum_N R_{i,j}(\xi,\eta)\mathbf{u}_i, \quad \phi_{\xi,\eta} = \sum_N R_{i,j}(\xi,\eta)\phi, \tag{9}$$

with

$$R_{i,j}(\xi,\eta) = \frac{N_{i,p}(\xi)N_{j,q}(\eta)\omega_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(\xi)N_{l,q}(\eta)\omega_{k,l}}. \tag{10}$$

The test functions have a similar structure. Substituting the trial functions and test functions into the weak form, Eq. (7) yields after some algebra to the final linear system of equations [15]:

$$\begin{pmatrix} \mathbf{K}_{\mathrm{uu}} & \mathbf{K}_{\mathrm{u}\phi} \\ \mathbf{K}_{\phi\mathbf{u}} & \mathbf{K}_{\phi\phi} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \phi \end{pmatrix} = \begin{pmatrix} \mathbf{f}_{\mathbf{u}} \\ \mathbf{f}_{\phi} \end{pmatrix}. \tag{11}$$

For details about IGA derivation, please refer to Appendix 1.

2.3 NURBS trimming technique

To facilitate the computing of flexoelectricity on complex topologies like inclusions and holes, we incorporated into our IGA formulation the NURBS trimming technique proposed by Kin et al. [24]. The flow of the trimming analysis is demonstrated in Fig. 2. First, the NURBS is meshed

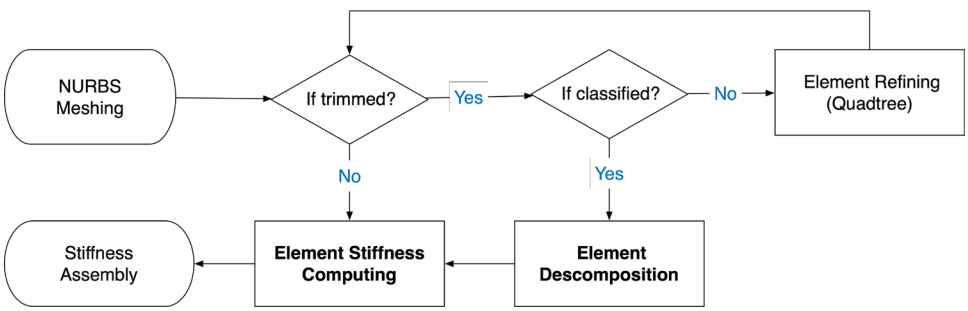


Fig. 2 The working flow of the NURBS trimming technique

in the parametric space, all the elements are identified and decomposed into triangular sub-elements. Then, for the numerical integration of the triangular sub-elements, the Jacobian matrix is calculated via the transformation of the sub-elements from the physical space to the Gaussian quadrature space. Finally, the element stiffness matrix and global stiffness matrix are assembled for solving the boundary value problem. Therefore, the key of the IGA on trimmed NURBS model lies in the identification of trimmed elements and the accuracy of the corresponding numerical integration.

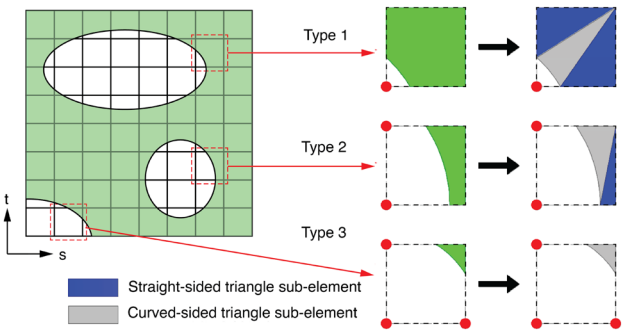


Fig. 4 Trimmed elements decomposition and classification

2.3.1 Trimmed elements identification

In the NURBS geometries, the distance of a point to a trimming curve can be obtained by finding the smallest distance between the point to its projections on the curve, e.g., d_2 is minimum projection distance, and thus is the distance of the point P to the curve $C(u)$, as shown in Fig. 3a.

Hence, the trimmed elements can be generally recognized by the distance relationship between the element's center, the element's vertices, and the curve, as shown in Fig. 3b. Possible conditions are as follows: (i) If $d_c < r_{\text{in}}$, the element is trimmed by the curve; (ii) If $d_c > r_{\text{out}}$, the element is normal; (iii) If $r_{\text{in}} \le d_c \le r_{\text{out}}$, the element needs refinement with the Quadtree algorithm [24].

2.3.2 Trimmed elements decomposition

In Jacobian computing, the trimmed elements can be divided into 3 types, based on the number of vertices outside of the physical domain, as shown in Fig. 4. NEFEM (NURBS integral finite element method) [44] is used to process straight-sided triangular sub-elements and curved-sided triangular sub-elements, as shown in Appendix 2.

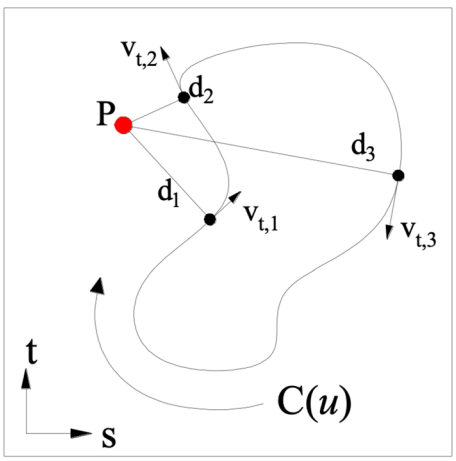
2.4 Numerical examples of the proposed IGA approach

2.4.1 Case I: an infinite plate with a circular hole

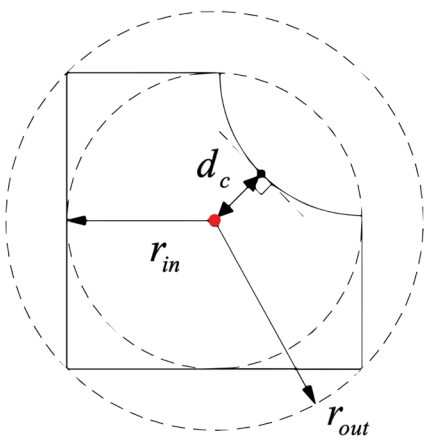
Our IGA method with NURBS trimming was implemented in MATLAB. To validate the proposed approach, we first investigated a infinite plate with a circular hole of radius $a = 1$ m (case I), which is subjected to a subjected to a far field traction ($\sigma = 1$ Pa) in the x direction, as shown in Fig. 5. A finite portion of the infinite plate is considered for analysis and, due to the symmetry of the problem in Fig. 5a, only a quarter of the portion requires modelling in Fig. 5b.

The plane stress conditions are assumed as our previous study [10], and, the elastic material properties used

Fig. 3 Positional relationship of points and elements with a trimming curve



(a) Projection for distance calculation.



(b) Trimming identification.

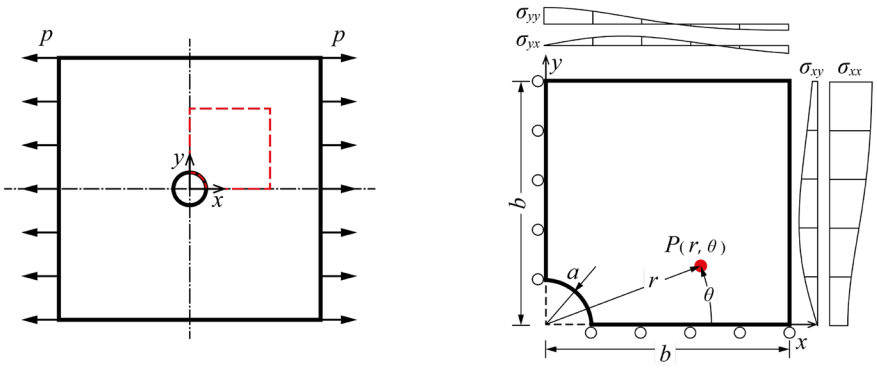


Fig. 5 An infinite plate with a circular hole subjected to a horizontal uniformed traction (case I)

(a) The small portion taken for analysis. (b) Boundary conditions applied on the quarter substructure.

are Young's modulus $E = 30$ MPa and Poisson's ratio $\mu = 0.3$. By using the polar coordinates shown in Fig. 5b, the stresses and displacements for this benchmark are given in an analytical solution in [49] as

$$\begin{aligned}
\sigma_{xx} &= 1 - \frac{a^2}{r^2}\left(\frac{3}{2}\cos(2\theta) + \cos(4\theta)\right) + \frac{3a^4}{2r^4}\cos(4\theta),\\
\sigma_{xy} &= -\frac{a^2}{r^2}\left(\frac{1}{2}\sin(2\theta) + \sin(4\theta)\right) + \frac{3a^4}{2r^4}\sin(4\theta),\\
\sigma_{yy} &= -\frac{a^2}{r^2}\left(\frac{1}{2}\cos(2\theta) - \cos(4\theta)\right) - \frac{3a^4}{2r^4}\cos(4\theta),
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
u &= \frac{a}{8G}\left(\frac{r}{a}(\kappa+1)\cos(\theta) + \frac{2a}{r}[(1+\kappa)\cos(\theta) + \cos 3(\theta)] - \frac{2a^3}{r^3}\cos 3(\theta)\right),\\
v &= \frac{a}{8G}\left(\frac{r}{a}(\kappa-3)\sin(\theta) + \frac{2a}{r}[(1-\kappa)\sin(\theta) + \sin 3(\theta)] - \frac{2a^3}{r^3}\sin 3(\theta)\right),
\end{aligned} \tag{13}$$

where G is the shear modulus and the Kolosov constant $\kappa = (3-\mu)/(1-\mu)$ for the plane strain assumption. Boundary conditions specified in [49] are applied on the right and upper edges as shown in Fig. 5b.

The performance of the proposed IGA formulation is studied using meshes with 36, 100, 324 and 1156 nodes as shown in Fig. 6b–d and the NURBS basis is used for all meshes. The proposed method shows good convergence characteristics in this problem involving stress concentration. Figure 6e shows σ_{xx} predicted along the left edge $x = 0$ showing comparable accuracy and smoothness, and close agreement with the analytical solution by Eq. (12), where $\sigma_{xx}(\theta = \pi/2, r = a) = 3$.

2.4.2 Case II: 2-D flexoelectric cantilever with multi-holes

Furthermore, the 2-D flexoelectric cantilever (case II) in Fig. 7 was investigated. The length L and the height h of the cantilever are 0.454 mm and 0.227 mm, respectively, while the thickness is set as 1 mm. To set up the boundary value problem (BVP), (i) the bottom edge is grounded; (ii) the left edge is fixed; and (iii) the free vertex of the top edge is applied with a point load of 100 mN. The NURBS model was meshed into 1024 elements with 1156 control points by the h-refinement [21] of 5, where h stands for the knot insertion times. Three holes are represented by the NURBS circles with a radius of $0.1h = 22.7$ μm.

As for case II, our MATLAB program solved the displacement $\mathbf{u}$ and the electric potential ϕ in Eq. (11) using the material properties in Table 1. Both scenarios whether the beam in Fig. 7 is with holes or not, are simulated. The impact of holes on electric potential is shown in Fig. 8. More detailed interactions of the defects with the structure response were revealed by the contours of displacement, the strain, and the strain gradient in Figs. 9 and 10.

3 CNN-based surrogate model

3.1 Dataset generation

Our CNN-based surrogate model is designed to predict the identical output from our IGA model, i.e., the basic unknowns including the displacement and the electric potential. The other physical fields like the strain/stress and the strain/stress gradient can be derived from the predicted basic unknowns. To simplify the generation of representative porous flexoelectric structures, the NURBS representation for circles is used to indicate the holes as shown in Fig. 11.

Similar to the boundary settings of Fig. 7, a cantilever is fixed at one end and applied with a point load of 100 mN

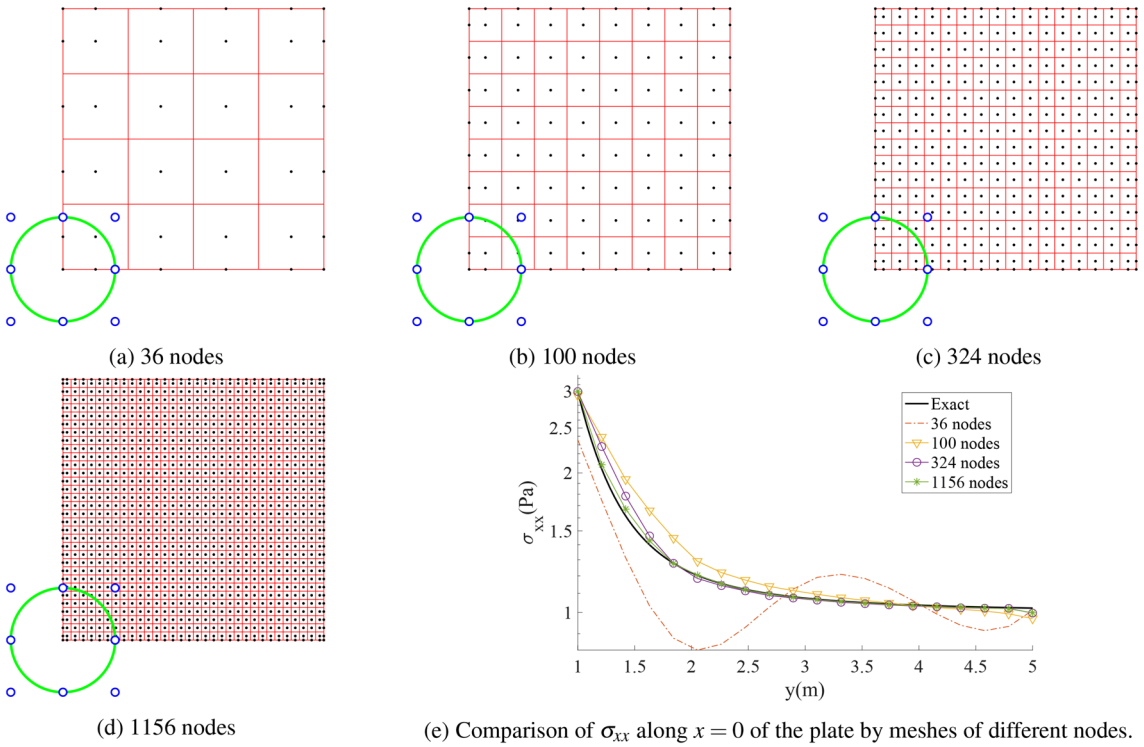


Fig. 6 NURBS meshes used for case I by the proposed approach

at the top vertex of the other end, while the bottom edge is grounded. Then the MATLAB's built-in uniform distribution algorithm was utilized to generate the circles with arbitrary radius r in a defined range $\{r \mid 0.05h \leq r \leq 0.2h\}$ and to guarantee an equal chance inside the defined limit boundary in Fig. 11. The corresponding material properties used are shown in Table 1. The physical meanings of these properties can be referred to in Eq. (1).

The dataset structure is shown in Fig. 12. For each sample, the CNN-based model input is a binary image recording the geometry information and the model output is an array of the nodal basic unknowns, i.e., the mechanical displacement components (u_1, u_2) and the electric potential (ϕ).

For the following study, we considered the number of holes (nb) ranging from 1 to 5. For each case, 200 samples were produced randomly. And three cases of model output numbers (with different h-refinements) were considered. In total, the dataset contains 3000 samples, as shown in Fig. 12. For most cases, the simulation is done in 2 min. But for samples with the high h-refinement, the calculation could take over 5 min due to the NURBS trimming analysis.

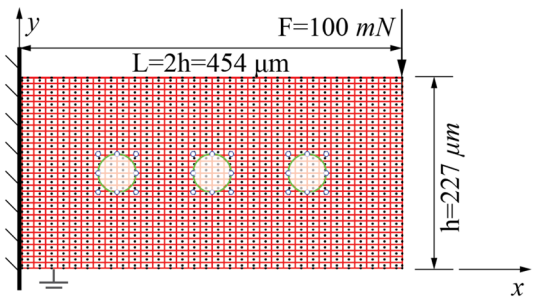
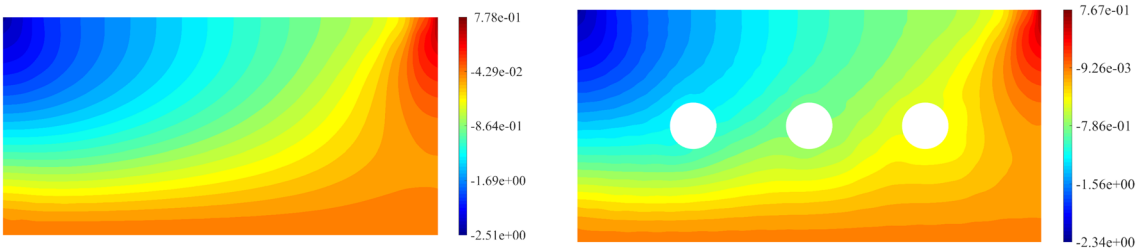


Fig. 7 NURBS meshing and boundary settings of the flexoelectric BVP (case II)

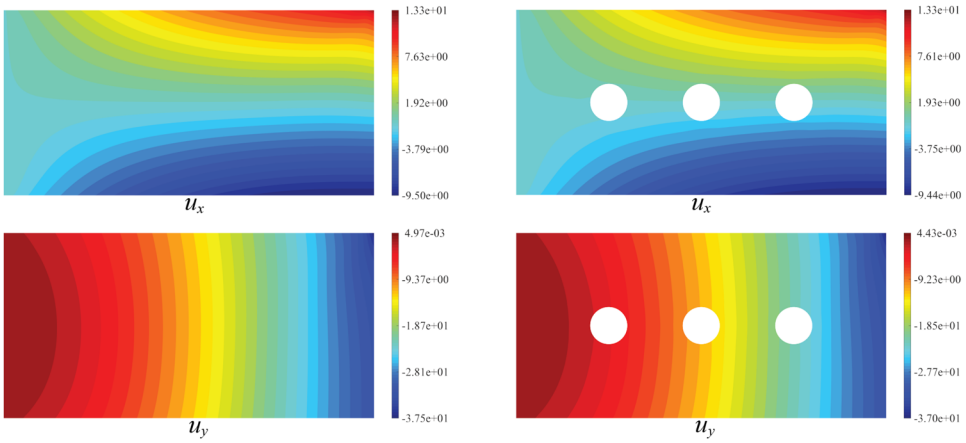
Table 1 Material properties of single crystal $BaTiO_3$ in case II

Properties	E	ν	e_{13}	μ_{12}	κ_{11}	κ_{13}	χ_{33}
Value	18.30	0.37	−4.4	1	11	12.48	1408
Unit	GPa	–	C/m^2	μC/m	nC/(V · m)	nC/(V · m)	–

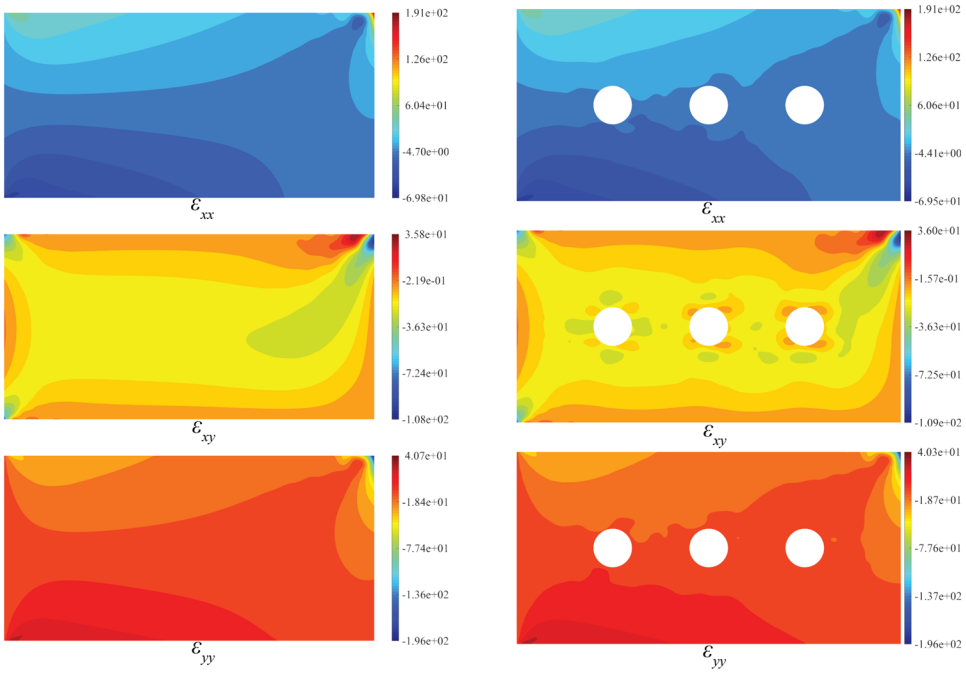


(a) Potential without holes (ϕ/mV). (b) Potential with holes (ϕ/mV).

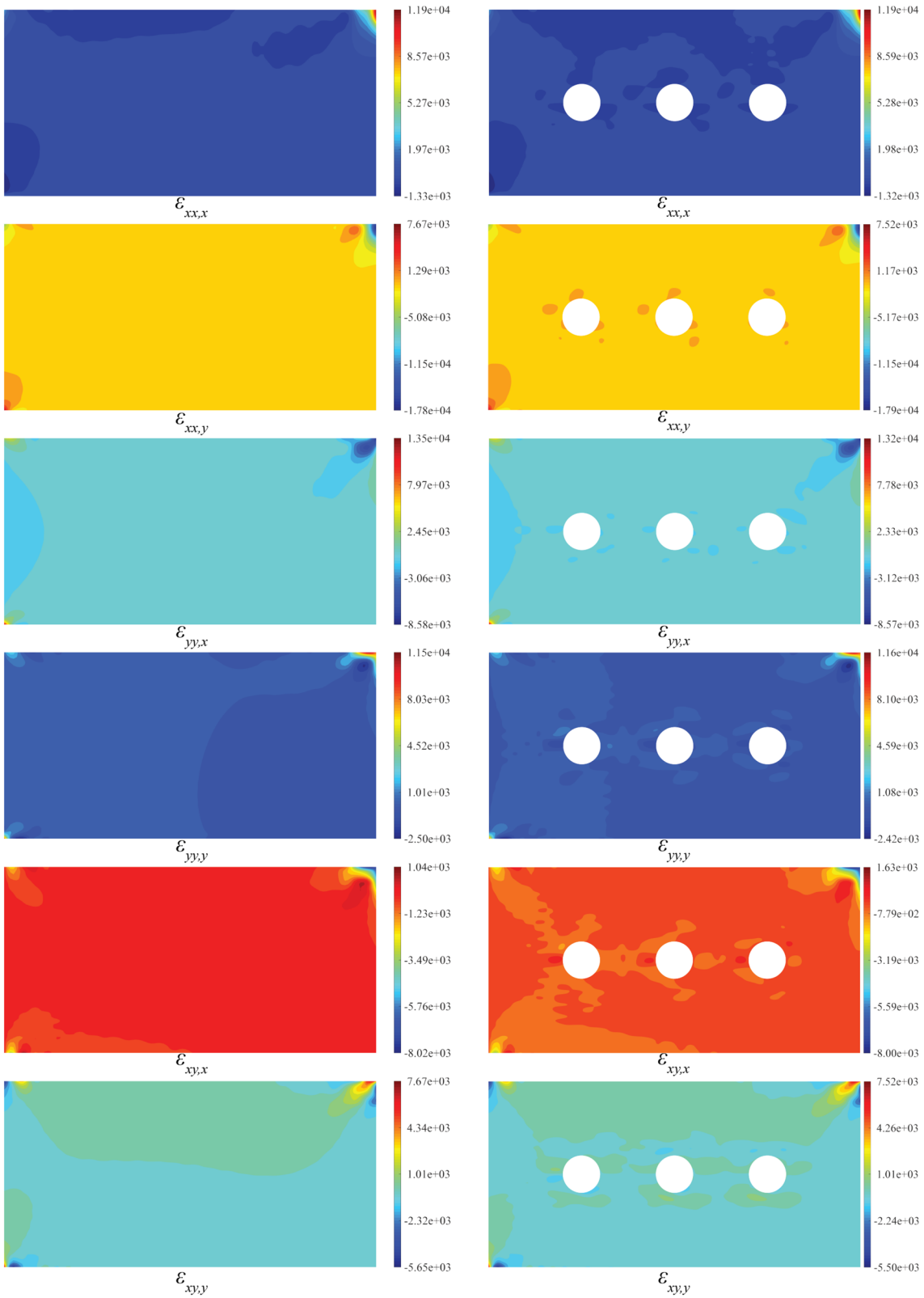
Fig. 8 Electric potential contours of case II using our IGA formulation



(a) Displacement without holes (*mm*). (b) Displacement with holes (*mm*).



(c) Strain without holes. (d) Strain with holes.



(a) Strain gradient without holes (mm^{-1}.

(b) Strain gradient with holes (mm^{-1}).

Fig. 10 Strain gradient contours of case II using our IGA formulation

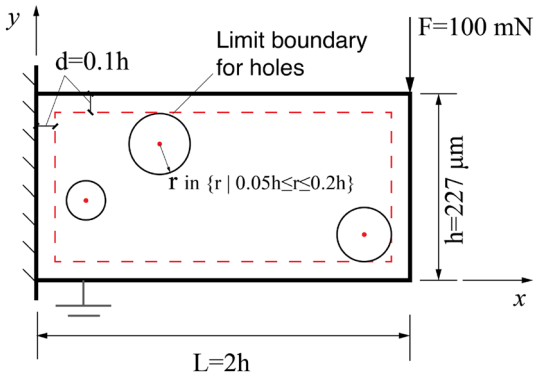


Fig. 11 Setup of the boundary value problem for data generation

3.2 Data preprocessing

The dataset is randomly split into the training set (80% data) and the testing dataset (20% data) in the model training. For computational efficiency, the pixels of the binary images are reduced to 64×32. The Min–Max normalization is applied to the physical fields data, which on the dataset $E = \{e_i\}$ is written as

$$\text{Normalized}(e_i) = \frac{e_i - E_{\min}}{E_{\max} - E_{\min}}. \tag{14}$$

After the Min–Max normalization, a good balance between the model bias and the data variance is achieved, judging from the loss difference between test and train datasets in

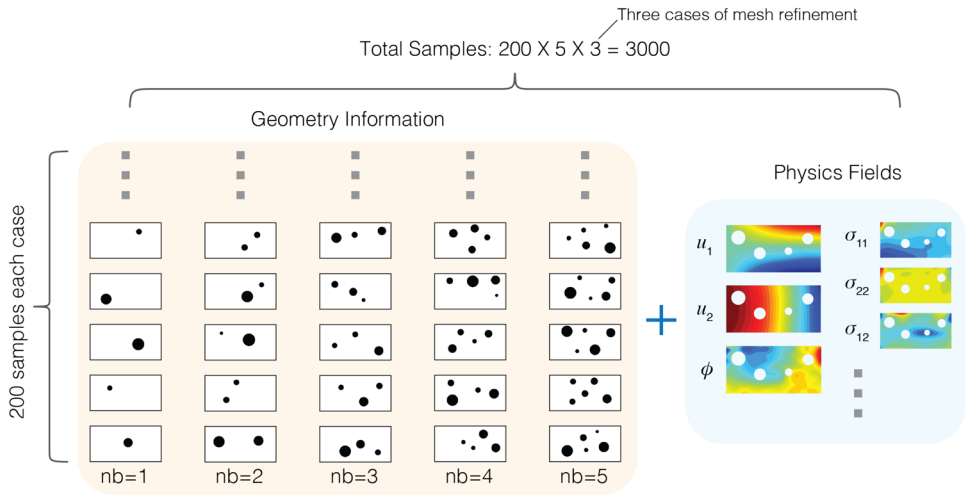
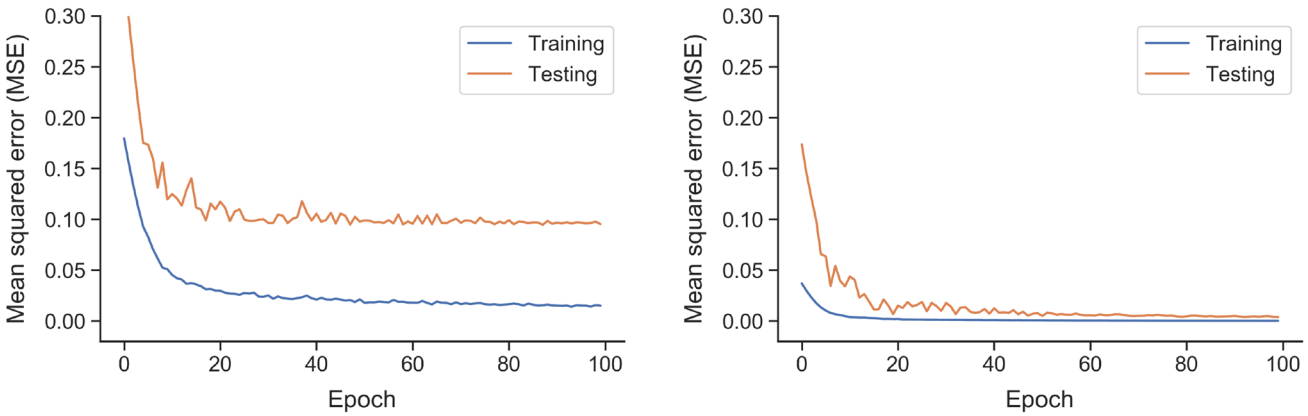


Fig. 12 The IGA dataset components of the CNN model



(a) The loss without normalization.

(b) The loss with normalization.

Fig. 13 Loss curves with/without the Min–Max normalization

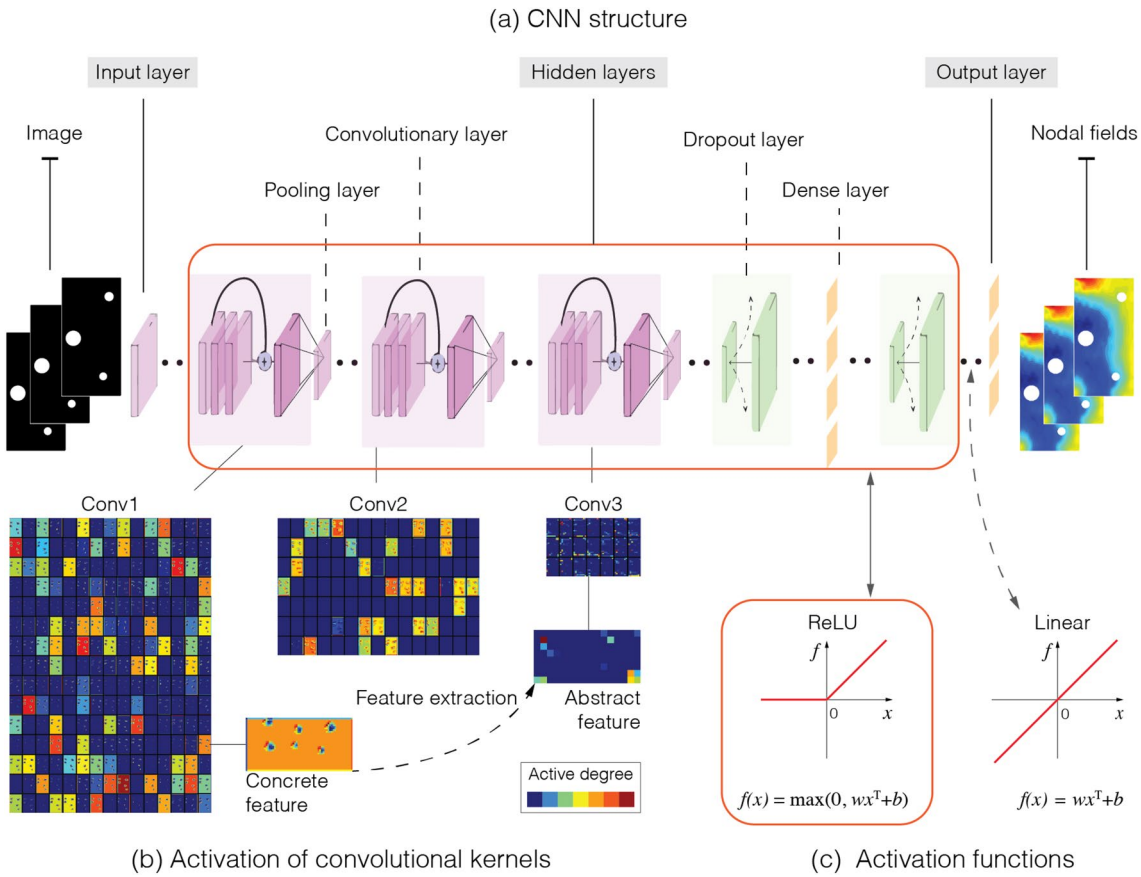


Fig. 14 The CNN framework of our surrogate model

pre-experiments, as shown in Fig. 13. The metric for model loss adopted here is the mean square error (MSE), which is the sum of squared distances between targets e_i and predictions $\hat{e}_i$.

$$\mathrm{MSE}(E) = \frac{1}{n}\sum_{i=1}^{n}\left(e_i - \hat{e}_i\right)^2. \tag{15}$$

3.3 Model building

The proposed CNN-based surrogate model is shown in Fig. 14a. Realized in Python with PyTorch library, the model input is binary images of flexoelectric porous beams, while the output is the same as that from IGA simulations. When the training data are input in our model, three 2-D convolutional layers (Con2D) and three Max-Pooling2D layers are arranged alternately, to gradually extract the underline data features at different abstract extend as the network grows deeper, as shown in Fig. 14b. Then to avoid the common overfitting issue in CNN, the dropout layer is followed. In the end, the fully connected Dense layer is added according to make the learned features match the dimension of the output.

Unlike classic pixel-to-pixel predictions in CNNs, our CNN model predicts nodal fields as the output array instead, which ameliorates the cost of reducing image resolution, so that the training efficiency and prediction accuracy is improved.

3.3.1 Activation function

ReLU (Rectified Linear Unit) activation functions exhibit certain biological meaning [16] and are widely used in deep neural network applications including image

Table 2 Comparison of the model loss using common optimizers

Optimizer	Validation loss (MSE)	Training time (s)
Adam	0.0068	62
AdaDelta	0.0077	65
SGD	0.0113	91

recognition. The ReLU is used for all the layers in our models, except that the final hidden layer must use linear activation for prediction's purpose, as shown in Fig. 14c.

3.3.2 Optimizer

Under the same training settings (200 samples with 80% as training dataset, epochs = 50, batch size = 20), the Adaptive Moment Estimation (Adam) outperforms other common adaptive techniques, as shown in Table 2.

3.3.3 Dropout

Dropout is a regularization technique for neural network models to avoid overfitting [47]. Through dropout layers, random neurons are ignored during one training epoch, so their contribution to the downstream neurons and weight updates on the backward pass is temporally removed. Dropout layers are normally used after all convolutional layers and pooling layers as indicated in our CNN-based surrogate model (Fig. 15a). An alternative way is to drop out entire feature maps from the convolutional layer which are then not used during pooling. This is called spatial dropout [50], as illustrated in Fig. 15b.

A low dropout has a minimal effect while a high value may result in under-learning. In our model, the dropout value is set to 50%, which seems close to the optimal for a wide range of networks [47]. Under the same training settings (200 samples with 80% as training dataset, epochs = 50, batch size = 20), the validation loss of (a) is 0.0068, which is slightly better than that of (b), which is 0.0088. So our model adopted the normal dropout layers.

4 Results and analysis

Our CNN model could predict the identical physical unknowns as our IGA model at a stable accuracy by the model selection and hypermeter tuning. Details of training settings and results will be discussed in the following. We first tested the accuracy of our model by cross-validation, and then analyzed the potential of our model performance and possible optimal model layout.

4.1 Cross-validation

That *k*-fold cross-validation (CV) is a resampling procedure to estimate the generalization ability of the model on new data. Here *k* refers to the number of groups that the samples are to be split into. To validate our model, 10-fold cross-validation was conducted, as shown in Fig. 16. Under the same training settings (200 samples with 80% as training

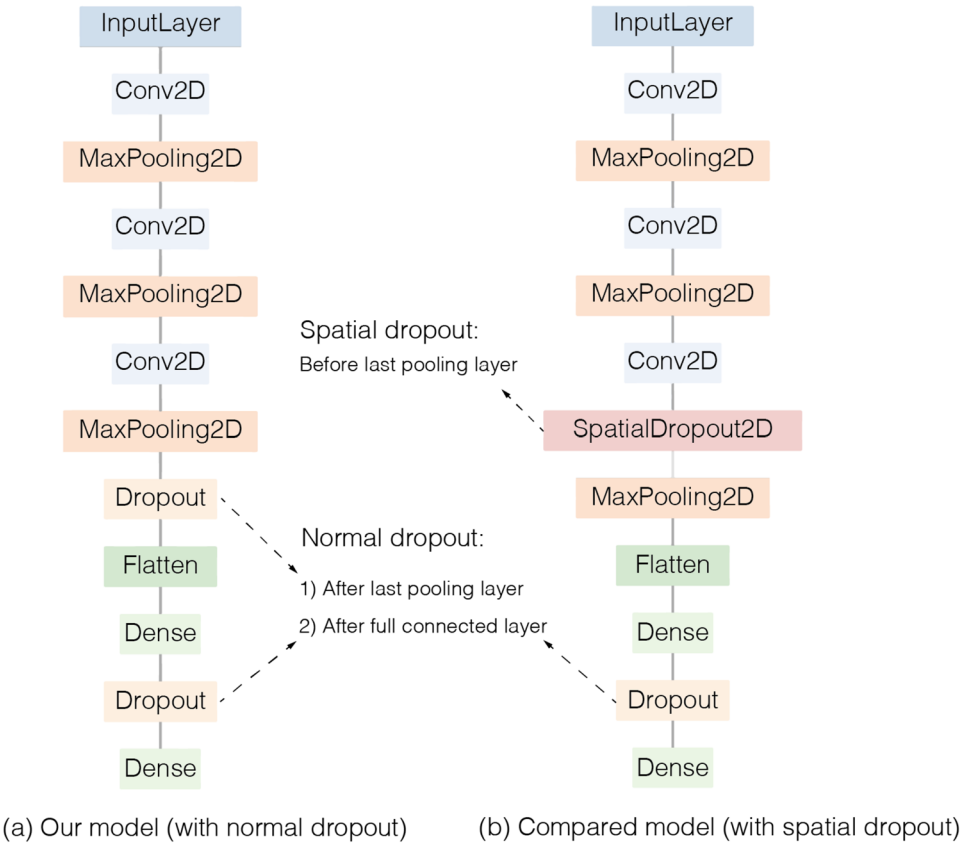


Fig. 15 Comparison of difference dropout settings in CNN structures

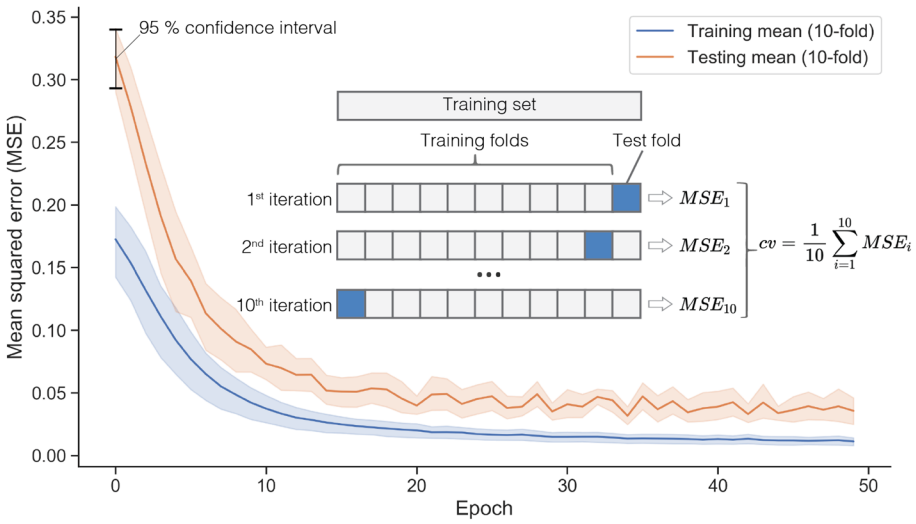


Fig. 16 10-fold cross-validation of the proposed CNN model

Table 3 The layers summary of the current CNN model

Layer (type)	Output shape	Params.
input_1 (InputLayer)	(None, 64, 32, 1)	0
conv2d_1 (Conv2D)	(None, 64, 32, 256)	2560
max_pooling2d_1 (MaxPooling2D)	(None, 32, 16, 256)	0
conv2d_2 (Conv2D)	(None, 32, 16, 128)	295,040
max_pooling2d_2 (MaxPooling2D)	(None, 16, 8, 128)	0
conv2d_3 (Conv2D)	(None, 16, 8, 64)	73,792
max_pooling2d_3 (MaxPooling2D)	(None, 8, 4, 64)	0
dropout_1 (Dropout)	(None, 8, 4, 64)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 32)	65,568
dropout_2 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, x^1)	–

x is the output size. $x = \{100, 324, 1156\}$ for h-refinement $= \{3, 4, 5\}$

dataset, epochs = 50, batch size = 20), we get the average of the MSE (cv) over 10 folds when the training process is repeated 10 times. The final steady state of cv indicates that our model converges as the training time increases.

4.2 Model performance analysis

In building our CNN-based surrogate model, the prediction errors exist due to the discretization from the analytic NURBS into pixels, due to the reduction of the pixels in data preprocessing, and finally the CNN approximation. Meanwhile, with the specified CNN layers, our model may not be right the best for the given problem.

It would therefore be interesting to see how to minimize the total errors by adjusting the size of the model output and convolutional layers. The layers of our current model are summarized in Table 3. The size of the model output is determined by the aforementioned h-refinement. For a scalar physical field like electric potential, the output size ranges in {100, 324, 1156} while h is in {3, 4, 5}. The influence of

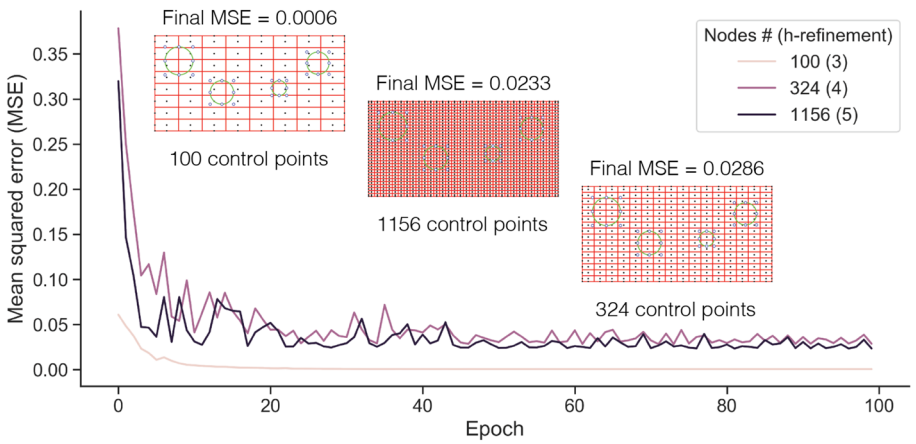


Fig. 17 Impact of model output size on the proposed CNN model

Table 4 The MSE variation under different model scales and output size

h (model output size)	Model scale				
	1/16	1/4	1	4	16
3 (100)	0.0027	0.0027	0.0015	0.0040	0.0029
4 (324)	0.0102	0.0102	0.0275	0.0204	0.0093
5 (1156)	0.0220	0.0406	0.0123	0.0121	0.0297

the model output size on the model performance is shown in Fig. 17.

In the analysis, 1000 samples were randomly selected, where 80% samples were set as the training dataset, while the rest 20% as the testing dataset. For all cases, the MSE turned to converge after the first 100 epochs. The training time is comparable, around 200 seconds each. When h is 3, the model output size is 100, and the corresponding MSE is the smallest, 6×10^{-4}. As the model output size increased, MSE jumped around 40 times bigger, which means the current model might have no potential for predicting under the larger output size. In fact, when h is 3 with 100 control points, the mesh is very coarse. Thus, the error from NURBS trimming of our IGA model is considerable compared to the finer cases.

To improve the performance of our basic CNN model in Table 3, the size of convolutional layers is our following concern. We denote the size of convolutional layers as scale = 1. There are about 4.4×10^5 hyperparameters for the basic model, and most are in the convolutional layers. In the following analysis, the neurons in all the three Conv2D layers are changed by the same scale. Considering the practical training time, the explored network scale is limited in $\{2^{-4}, 2^{-2}, 2^0, 2^2, 2^4\}$. For a given scale 2^i, the number of hyperparameters in the corresponding model is about $4.4 \cdot 2^i \times 10^5$. 200 samples were randomly selected, where 80% samples were set as the training dataset, while the rest 20% as the testing dataset.

The MSE is evaluated for the first 100 epochs. For varying output sizes, the potential optimal size of convolutional layers differs, as shown in Table 4. While the original model scale is most suitable for the case that output size is 100, the increased model scale did make our model predict better in larger output sizes of 324 and 1156. The number of hyperparameters reflects the network complexity. As neurons in convolutional layers increase, more subtle features can be captured, which might keep pace with the increasing model output size.

5 Conclusion

Flexoelectricity is a nonlocal electromechanical coupling phenomenon between strain gradient and electric polarization, which is normally computationally costly for simulating structures with complex topology and further optimizing. We developed an IGA model with the NURBS trimming technique to firstly realize the amiable simulation of porous flexoelectric structures. Then a CNN-based surrogate model is proposed to predict the identical output from our IGA model. Instead of conventional pixel-to-pixel prediction, predicting physical fields of the control points helps to eliminate loss in model accuracy, as well as to save training time of deep learning. Moreover, details of CNN model building are discussed, including the choice of activation functions, dropout layers, and optimizers based on our IGA dataset. Finally, the CNN model performances by changing model output size and convolutional layers size have been analyzed, which might be useful to build similar CNN frameworks predicting nonlocal mech-physical problems.

Appendix 1: Further derivation of IGA formulation

The contribution of a Gaussian integral point to Eq. (11) is

$$\begin{cases} \mathbf{K}^i_{\mathrm{uu}} = \mathbf{B}^{\mathrm{T}}_{\mathrm{u}} C_{ijkl} \mathbf{B}_{\mathrm{u}} \left|\mathbf{J}^i\right| W^i \\ \mathbf{K}^i_{\mathrm{u}\phi} = \mathbf{B}^{\mathrm{T}}_{\mathrm{u}} e_{kj} \mathbf{B}_{\phi} \left|\mathbf{J}^i\right| W^i + \mathbf{H}^{\mathrm{T}}_{\mathrm{u}} \mu_{ijkl} \mathbf{B}_{\phi} \left|\mathbf{J}^i\right| W^i \\ \mathbf{K}^i_{\phi_{\mathbf{u}}} = \mathbf{B}^{\mathrm{T}}_{\phi} e^{\mathrm{T}}_{kjj} \mathbf{B}_{\mathbf{u}} \left|\mathbf{J}^i\right| W^i + \mathbf{B}^{\mathrm{T}}_{\phi} \boldsymbol{\mu}^{\mathrm{T}}_{ijkl} \mathbf{H}_{\mathbf{u}} \left|\mathbf{J}^i\right| W^i \\ \mathbf{K}^i_{\phi\phi} = \mathbf{B}^{\mathrm{T}}_{\phi} \kappa_{ij} \mathbf{B}_{\phi} \left|\mathbf{J}^i\right| W^i \end{cases} \tag{16}$$

where $\mathbf{J}^i$ and W^i, are the Jacobian and the weight of the integration point, respectively; $\mathbf{B}_{\mathbf{u}}$ and $\mathbf{B}_{\phi}$ are the corresponding gradient operators for $\mathbf{u}$ and ϕ; $\mathbf{H}_{\mathbf{u}}$ is Hessian operator for $\mathbf{u}$.

According to the gradient operation of NURBS, $\mathbf{B}_{\mathbf{u}}$, $\mathbf{B}_{\phi}$, and $\mathbf{H}_{\mathbf{u}}$ can be written as

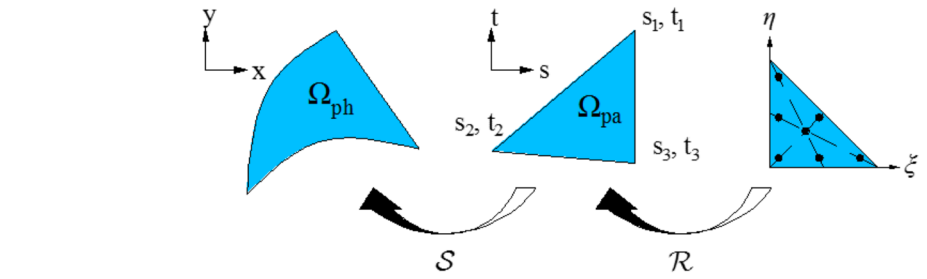
$$\begin{aligned} \mathbf{B}^R_{\mathrm{u}} &= \begin{pmatrix} \frac{\partial R}{\partial x} & 0 & \frac{\partial R}{\partial y} \\ 0 & \frac{\partial R}{\partial y} & \frac{\partial R}{\partial x} \end{pmatrix}^{\mathrm{T}}, \quad \mathbf{B}^R_{\phi} = \begin{pmatrix} \frac{\partial R}{\partial x} \\ \frac{\partial R}{\partial y} \end{pmatrix}, \\ \mathbf{H}^R_{\mathrm{u}} &= \begin{pmatrix} \frac{\partial^2 R}{\partial x^2} & 0 & \frac{\partial^2 R}{\partial y \partial x} & \frac{\partial^2 R}{\partial x \partial y} & 0 & \frac{\partial^2 R}{\partial y^2} \\ 0 & \frac{\partial^2 R}{\partial y \partial x} & \frac{\partial^2 R}{\partial x^2} & 0 & \frac{\partial^2 R}{\partial y^2} & \frac{\partial^2 R}{\partial x \partial y} \end{pmatrix}^{\mathrm{T}}, \end{aligned} \tag{17}$$

which can all be derived from the explicit expressions of NURBS base functions in Eqs. (8)–(10).

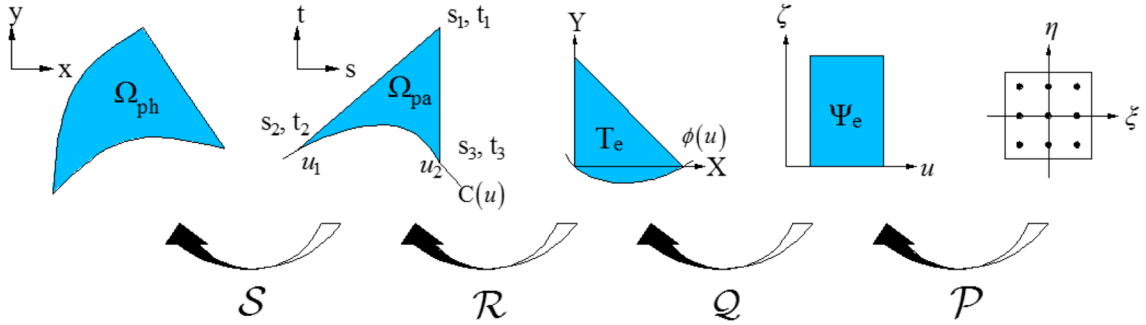
Appendix 2: Jacobian computation of triangle sub-elements

Straight-sided triangular elements

The transformations of the straight-sided triangle element from the physical space to the Gaussian quadrature space is shown in Fig. 18a.



(a) The straight-sided triangular sub-elements.



(b) The curved-sided triangular sub-elements.

Fig. 18 Spatial transformation of triangular sub-elements

The Jacobian of a straight-sided triangular element is determined by transformation R and S, i.e.,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial s}{\partial \xi} & \frac{\partial t}{\partial \xi} \\ \frac{\partial s}{\partial \eta} & \frac{\partial t}{\partial \eta} \end{pmatrix} \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{pmatrix} = \mathbf{J}_{\mathrm{R}} \cdot \mathbf{J}_{\mathrm{S}}. \tag{18}$$

Transformation $\mathrm{S} : \{s,t\} \rightarrow \{x,y\}$ is from the parametric space Ω_{pa} to the physical space Ω_{ph}. We have

$$\mathrm{S} : \begin{Bmatrix} x \\ y \end{Bmatrix} = \sum_{i=0}^{n} R_{i,p}(s,t) \begin{Bmatrix} x_i \\ y_i \end{Bmatrix}, \tag{19}$$

and

$$\mathbf{J}_{\mathrm{s}} : \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{pmatrix} = \begin{pmatrix} \sum R_{,s} x & \sum R_{,s} y \\ \sum R_{,t} x & \sum R_{,t} y \end{pmatrix}. \tag{20}$$

Transformation $\mathrm{R} : \{\xi,\eta\} \rightarrow \{s,t\}$ transforms the Gaussian quadrature for triangle elements into the parametric space Ω_{pa}, written in natural coordinates as

$$\mathrm{R} : \begin{cases} s = (\eta)s_1 + (1-\xi-\eta)s_2 + (\xi)s_3 \\ t = (\eta)t_1 + (1-\xi-\eta)t_2 + (\xi)t_3 \end{cases}, \tag{21}$$

and

$$\mathbf{J}_{\mathrm{R}} : \begin{pmatrix} \frac{\partial s}{\partial \xi} & \frac{\partial t}{\partial \xi} \\ \frac{\partial s}{\partial \eta} & \frac{\partial t}{\partial \eta} \end{pmatrix} = \begin{pmatrix} s_3 - s_2 & t_3 - t_2 \\ s_1 - s_2 & t_1 - t_2 \end{pmatrix}. \tag{22}$$

Curved-sided triangular elements

The transformations of the curved-sided triangle element from the physical space to the Gaussian quadrature space is shown in Fig. 18b.

The Jacobian of a straight-sided triangular element is determined by transformation P, Q, R and S, i.e.,

$$\begin{aligned} \mathbf{J} &= \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial \zeta}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial \zeta}{\partial \eta} \end{pmatrix} \begin{pmatrix} \frac{\partial X}{\partial \psi} & \frac{\partial Y}{\partial \psi} \\ \frac{\partial Y}{\partial \zeta} & \frac{\partial Y}{\partial \zeta} \end{pmatrix} \begin{pmatrix} \frac{\partial s}{\partial X} & \frac{\partial t}{\partial X} \\ \frac{\partial s}{\partial Y} & \frac{\partial t}{\partial Y} \end{pmatrix} \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{pmatrix} \\ &= \mathbf{J}_{\mathrm{p}} \cdot \mathbf{J}_{Q} \cdot \mathbf{J}_{\mathrm{R}} \cdot \mathbf{J}_{\mathrm{s}} \end{aligned} \tag{23}$$

While Transformation S is identical as the one for straight-sided triangular elements, Transformation $\mathrm{R} : \{X,Y\} \rightarrow \{s,t\}$ transforms the curved triangle $\mathrm{T_e}$ with vertices (0, 0), (0, 1) and (1, 0) into the parametric space Ω_{pa}, written in natural coordinates as

$$\mathrm{R} : \begin{cases} s = (Y)s_1 + (1-X-Y)s_2 + (X)s_3 \\ t = (Y)t_1 + (1-X-Y)t_2 + (X)t_3 \end{cases}, \tag{24}$$

and

$$\mathbf{J}_{\mathrm{R}} : \begin{pmatrix} \frac{\partial s}{\partial X} & \frac{\partial t}{\partial X} \\ \frac{\partial s}{\partial Y} & \frac{\partial t}{\partial Y} \end{pmatrix} = \begin{pmatrix} s_3 - s_2 & t_3 - t_2 \\ s_1 - s_2 & t_1 - t_2 \end{pmatrix}. \tag{25}$$

Transformation $\mathrm{Q}: \{u, \zeta\} \rightarrow \{X, Y\}$ transforms the rectangular area $\Psi_e : \left[u_1, u_2\right] \times [0, 1]$ into a curved triangle $\mathbf{T}_e$; through the transformation, the upper edge and the two vertices of the lower edge of Ψ_e are mapped to the 3 vertices of the curved triangle, respectively, i.e.,

$$\mathrm{Q}: \begin{cases} X = \phi_X(u)(1-\zeta) \\ Y = \phi_Y(u)(1-\zeta) + \zeta \end{cases}, \tag{26}$$

and

$$\mathbf{J}_Q: \begin{pmatrix} \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} \\ \frac{\partial Y}{\partial \zeta} & \frac{\partial Y}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \frac{\partial \phi_X(u)}{\partial u}(1-\zeta) & \frac{\partial \phi_Y(u)}{\partial u}(1-\zeta) \\ -\phi_X(u) & -\phi_Y(u)+1 \end{pmatrix}. \tag{27}$$

Transformation $\mathrm{P}: \{\xi, \eta\} \rightarrow \{u, \zeta\}$ transforms the Gaussian quadrature space $[0, 1] \times [0, 1]$ into a rectangular area Ψ_e, i.e.,

$$P: \begin{cases} u = \frac{u_2 - u_1}{2}\xi + \frac{u_1+u_2}{2} \\ \zeta = \frac{1}{2}\eta + \frac{1}{2} \end{cases}, \tag{28}$$

and

$$\mathbf{J}_\mathrm{P}: \begin{pmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial \zeta}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial \zeta}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left(u_2 - u_1\right) & 0 \\ 0 & \frac{1}{2} \end{pmatrix}. \tag{29}$$

Acknowledgements X. Zhuang and Q. Wang appreciate the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453) and the ERC Starting Grant COTOFLEXI (Grant no. 802205).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abdel-Hamid O, Mohamed A, Jiang H, Deng L, Penn G, Yu D (2014) Convolutional neural networks for speech recognition. IEEE/ACM Trans Audio Speech Lang Process 22(10):1533–1545
2. Abdollahi A, Arias I (2015) Constructive and destructive interplay between piezoelectricity and flexoelectricity in flexural sensors and actuators. J Appl Mech 82(12):121003
3. Abdollahi A, Peco C, Millan D, Arroyo M, Arias I (2014) Computational evaluation of the flexoelectric effect in dielectric solids. J Appl Phys 116(9):093502
4. Al Rahhal MM, Bazi Y, AlHichri H, Alajlan N, Melgani F, Yager RR (2016) Deep learning approach for active classification of electrocardiogram signals. Inf Sci 345:340–354
5. Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T (2019) Artificial neural network methods for the solution of second order boundary value problems. Comput Mater Contin 59(1):345–359
6. Awad M, Khanna R (2015) Support vector regression. In: Pepper J, Weiss S, Hauke P (Eds) Efficient learning machines. Springer, Berlin, pp 67–80
7. Benner P, Gugercin S, Willcox K (2015) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 57(4):483–531
8. Bhaskar UK, Banerjee N, Abdollahi A, Wang Z, Schlom DG, Rijnders G, Catalan G (2016) A flexoelectric microelectromechanical system on silicon. Nat Nanotechnol 11(3):263–266
9. Buhmann MD (2003) Radial basis functions: theory and implementations, vol 12. Cambridge University Press, Cambridge
10. Cai Y, Zhuang X, Augarde C (2010) A new partition of unity finite element free from the linear dependence problem and possessing the delta property. Comput Methods Appl Mech Eng 199(17–20):1036–1043
11. Cressie N (1990) The origins of kriging. Math Geol 22(3):239–252
12. Cross LE (2006) Flexoelectric effects: charge separation in insulating solids subjected to elastic strain gradients. J Mater Sci 41(1):53–63
13. Duda RO, Hart PE, Stork DG et al (2001) Pattern classification, vol 680. Wiley, New York
14. Ghasemi H, Park HS, Rabczuk T (2017) A level-set based IGA formulation for topology optimization of flexoelectric materials. Comput Methods Appl Mech Eng 313:239–258
15. Ghasemi H, Park HS, Rabczuk T (2018) A multi-material level set-based topology optimization of flexoelectric composites. Comput Methods Appl Mech Eng 332:47–62
16. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, vol 15, pp 315–323
17. Gómez A, Vila-Fungueiriño JM, Moalla R, Saint-Girons G, Gázquez J, Varela M, Bachelet R, Gich M, Rivadulla F, Carretero-Genevrier A (2017) Electric and mechanical switching of ferroelectric and resistive states in semiconducting batio3-δ films on silicon. Small 13(39):1701614
18. Goswami S, Anitescu C, Chakraborty S, Rabczuk T (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor Appl Fract Mech 106:102447
19. Hamdia KM, Ghasemi H, Bazi Y, AlHichri H, Alajlan N, Rabczuk T (2019) A novel deep learning based method for the computational material design of flexoelectric nanostructures with topology optimization. Finite Elem Anal Des 165:21–30
20. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, vol 2017, pp 2961–2969

21. Hughes TJ, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194(39–41):4135–4195
22. Jasim A, Wang H, Yesner G, Safari A, Maher A (2017) Optimized design of layered bridge transducer for piezoelectric energy harvesting from roadway. Energy 141:1133–1145
23. Jia X, Willard J, Karpatne A, Read JS, Zwart JA, Steinbach M, Kumar V (2021) Physics-guided machine learning for scientific discovery: an application in simulating lake temperature profiles. ACM/IMS Trans Data Sci 2(3):1–26
24. Kim HJ, Seo YD, Youn SK (2009) Isogeometric analysis for trimmed cad surfaces. Comput Methods Appl Mech Eng 198(37–40):2982–2995
25. Kim Y (2014) Convolutional neural networks for sentence classification. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp 1746–1751. https://doi.org/10.3115/v1/d14-1181
26. Kögl M, Silva EC (2005) Topology optimization of smart structures: design of piezoelectric plate and shell actuators. Smart Mater Struct 14(2):387
27. Krichen S, Sharma P (2016) Flexoelectricity: a perspective on an unusual electromechanical coupling. J Appl Mech 83(3): 030801
28. Kwon SR, Huang W, Zhang S, Yuan FG, Jiang X (2016) Study on a flexoelectric microphone using barium strontium titanate. J Micromech Microeng 26(4):045001
29. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
30. Lu H, Liu S, Ye Z, Yasui S, Funakubo H, Rappe AM, Gruverman A (2017) Asymmetry in mechanical polarization switching. Appl Phys Lett 110(22):222903
31. Merupo VI, Guiffard B, Seveno R, Tabellout M, Kassiba A (2017) Flexoelectric response in soft polyurethane films and their use for large curvature sensing. J Appl Phys 122(14):144101
32. Mo C, Davidson J, Clark WW (2014) Energy harvesting with piezoelectric circular membrane under pressure loading. Smart Mater Struct 23(4):045005
33. Myers RH, Montgomery DC, Anderson-Cook CM (2016) Response surface methodology: process and product optimization using designed experiments. Wiley, New York
34. Nanthakumar S, Lahmer T, Zhuang X, Park HS, Rabczuk T (2016) Topology optimization of piezoelectric nanostructures. J Mech Phys Solids 94:316–335
35. Nanthakumar S, Lahmer T, Zhuang X, Zi G, Rabczuk T (2016) Detection of material interfaces using a regularized level set method in piezoelectric structures. Inverse Probl Sci Eng 24(1):153–176
36. Nanthakumar S, Zhuang X, Park HS, Rabczuk T (2017) Topology optimization of flexoelectric structures. J Mech Phys Solids 105:217–234
37. Nguyen-Thanh N, Zhou K, Zhuang X, Areias P, Nguyen-Xuan H, Bazilevs Y, Rabczuk T (2017) Isogeometric analysis of large-deformation thin shells using RHT-splines for multiple-patch coupling. Comput Methods Appl Mech Eng 316:1157–1178
38. Nowack P, Braesicke P, Haigh J, Abraham NL, Pyle J, Voulgarakis A (2018) Using machine learning to build temperature-based ozone parameterizations for climate sensitivity simulations. Environ Res Lett 13(10):104016
39. Oliver MA, Webster R (1990) Kriging: a method of interpolation for geographical information systems. Int J Geogr Inf Syst 4(3):313–332
40. Rabczuk T, Belytschko T (2004) Cracking particles: a simplified meshfree method for arbitrary evolving cracks. Int J Numer Methods Eng 61(13):2316–2343
41. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol 2016, pp 779–788
42. Rupp CJ, Evgrafov A, Maute K, Dunn ML (2009) Design of piezoelectric energy harvesting systems: a topology optimization approach based on multilayer plates and shells. J Intell Mater Syst Struct 20(16):1923–1939
43. Sainath TN, Mohamed Ar, Kingsbury B, Ramabhadran B (2013) Deep convolutional neural networks for LVCSR. In: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, pp 8614–8618
44. Sevilla R, Fernández-Méndez S, Huerta A (2008) NURBS-enhanced finite element method (NEFEM). Int J Numer Methods Eng 76(1):56–83
45. Shu L, Liang R, Rao Z, Fei L, Ke S, Wang Y (2019) Flexoelectric materials and their related applications: a focused review. J Adv Ceram 8(2):153–173
46. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
47. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
48. Tesche C, De Cecco CN, Baumann S, Renker M, McLaurin TW, Duguay TM, Bayer RR 2nd, Steinberg DH, Grant KL, Canstein C et al (2018) Coronary CT angiography-derived fractional flow reserve: machine learning algorithm versus computational fluid dynamics modeling. Radiology 288(1):64–72
49. Timoshenko SP, Goodier JN, Abramson HN (1970) Theory of elasticity (3rd ed). J Appl Mech 37(3):888. https://doi.org/10.1115/1.3408648. https://asmedigitalcollection.asme.org/appliedmechanics/article-pdf/37/3/888/5450582/888_1.pdf
50. Tompson J, Goroshin R, Jain A, LeCun Y, Bregler C (2015) Efficient object localization using convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol 2015, pp 648–656
51. Wang S, Wang MY (2006) Radial basis functions and level set method for structural topology optimization. Int J Numer Methods Eng 65(12):2060–2090
52. Yegnanarayana B (2009) Artificial neural networks. PHI Learning Pvt. Ltd., Delhi
53. Yudin P, Tagantsev A (2013) Fundamentals of flexoelectricity in solids. Nanotechnology 24(43):432001
54. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, vol 2015, pp 649–657
55. Zhang X, Pan Q, Tian D, Zhou W, Chen P, Zhang H, Chu B (2018) Large flexoelectriclike response from the spontaneously polarized surfaces in ferroelectric ceramics. Phys Rev Lett 121(5):057602
56. Zhu Y, Zabaras N, Koutsourelakis PS, Perdikaris P (2019) Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. J Comput Phys 394:56–81