Supplementary information for variational operator learning: a unified paradigm for training neural operators and solving partial differential equations

Tengfei Xu[1], Dachuan Liu[1], Peng Hao [*1], and Bo Wang[1]

[1]State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China

S1 Notations

Table 1: Notations used in this Article.

$\mathbb{R}^d$	Euclidean space of dimension d
$\mathbf{A}$	Operators of governing equations in PDEs
$\mathbf{BC}$	Operators of boundary conditions in PDEs
$\mathbf{u}$	Variable of PDEs to be solved
$\mathcal{C}_{\mathrm{c}}^{\infty}(\Omega)$	Test function space of all compactly supported and infinitely differentiable functions from Ω to $\mathbb{R}$
$a(\cdot,\cdot)$	Coercive continuous bilinear forms
V	Hilbert space with the inner product $a(\cdot,\cdot): V\times V\to\mathbb{R}$ and norm $\|\mathbf{v}\|_V = a(\mathbf{v},\mathbf{v})^{1/2}$
L	Continuous linear forms on V
$\mathbf{x}=(x,y,z)$	Coordinates (x,y,z) in the global physical coordinate system
$\mathbf{r}=(r,s,t)$	Coordinates (r,s,t) corresponding to (x,y,z) in the parameter coordinate system of an element
$\mathbf{a}^e=[a^e_{j,i}]^{\mathrm{T}}$	Nodal solution of an element, where $\cdot^e_{j,i}$ refers to ith degree of freedom of jth node of an element. Note $\mathbf{a}^e$ is always treated as a vector conventionally in FEMs. For example, if all nodes of the elements have three degree of freedom, then $\mathbf{a}^e=[a^e_{1,1},a^e_{1,2},a^e_{1,3},a^e_{2,1},a^e_{2,2},...,a^e_{M,1},a^e_{M,2},a^e_{M,3}]^{\mathrm{T}}$. To write $a^e_{j,i}$ with two subscripts is just for convenience.

Continued on next page

*Corresponding author: haopeng@dlut.edu.cn

Table 1: Notations used in this Article. (Continued)

M	Number of nodes in an element
$\mathbf{N}$	Shape function matrix of an element
Π	Functional of an system
$\widetilde{\Pi}$	Approximation of Π
$\mathbf{B}$	Strain-displacement matrix of an element
$\mathbf{C}$ and $\mathbf{D}$	Elastic constitutive matrix
$\mathbf{G}^e$	Node index mapping matrix between element and global computational mesh
$\mathbf{K}$	Coefficient matrix of linear system (global stiffness matrix in context of FEMs)
$\mathbf{P}$	Right-hand side of linear system (global load vector in context of FEMs)
n_g	Number of Gauss points in the volume of an element
$\mathbf{J}^e$	Jacobian matrix of an element
$\mathbf{J}_m^{\mathcal{S}_\sigma^e}$	Jacobian matrix of mth boundary of an element
Subscript l	Quantities assigned with subscript l takes value at the lth Gauss point in the volume of an element
H_l	Weight of the lth Gauss point
$n_{\mathcal{S}_\sigma^e}$	Number of the boundaries of an element
n_m	Number of Gauss points of the mth edge of an element
Subscript ml	Quantities assigned with subscript ml at the lth Gauss point of the mth edge of an element
I_{ml}	Weight of the lth Gauss point of the mth edge of an element
n_f	Total number of degrees of freedom of all nodes in the computational mesh
δu	Variation of u (virtual displacement in context of solid mechanics)
$\mathcal{B}$	A linear elastic body in $\mathbb{R}^3$
$\mathcal{S}_\sigma$	Traction boundary of $\mathcal{B}$
$\mathcal{S}_u$	Displacement boundary of $\mathcal{B}$
$\mathcal{S}_{\mathrm{N}}$	Neumann boundary
$\mathcal{S}_{\mathrm{R}}$	Robin boundary
$\boldsymbol{n}$	Unit normal vector
Ω	Solution domain
$\boldsymbol{\sigma} = [\sigma_{ij}]$	Stress tensor of $\mathcal{B}$
$\mathbf{f} = [f_i]$	Body force tensor of $\mathcal{B}$
$\bar{\mathbf{X}} = \left[\bar{X}_i\right]$	Traction force tensor of $\mathcal{B}$

Continued on next page

Table 1: Notations used in this Article. (Continued)

$\mathbf{u} = [u_i]$	Displacement tensor of $\mathcal{B}$
$\boldsymbol{\varepsilon} = [\varepsilon_{ij}]$	Strain tensor of $\mathcal{B}$
θ	Fiber angle field
$\otimes$	Element-wise product

S2 Details of problem settings

In the VOL, we keep the same mesh settings with the corresponding commercial FEM solvers. All experiments in this Article are implemented with Pytorch [1].

S2.1 Variable heat source problem and Darcy flow problem

We are interested in diffusion equations with the following form

$$\begin{cases} -\nabla \cdot (\ \boldsymbol{\kappa} \nabla T\ ) = Q & \text{in } \Omega, \\ T = \bar{T} & \text{on } \mathcal{S}_{\mathrm{D}}, \\ (\boldsymbol{\kappa} \nabla T) \cdot \boldsymbol{n} = \bar{q} & \text{on } \mathcal{S}_{\mathrm{N}}, \\ (\boldsymbol{\kappa} \nabla T) \cdot \boldsymbol{n} = \bar{h}\,(T_{\infty} - T) & \text{on } \mathcal{S}_{\mathrm{R}}, \end{cases} \tag{1}$$

which may have different meanings in different physical backgrounds. Specifically, we consider a cubic $\Omega = [0, 1\mathrm{m}] \times [0, 1\mathrm{m}] \times [0, 1\mathrm{m}]$ for a variable heat source problem and $\Omega = [0, 1] \times [0, 1]$ for a Darcy flow problem. The conductivity tensor is isotropic in these two problems, $\boldsymbol{\kappa} = \begin{bmatrix} \kappa_x & \\ & \kappa_y \end{bmatrix}$, $\kappa_x = \kappa_y$. For the variable heat source problem, $\kappa_x = \kappa_y$ is set as a constant while Q is a function of coordinates x and y; for the Darcy problem, $\kappa_x = \kappa_y = \kappa\,(x, y)$ is a function of coordinates x and y, while Q is constant.

2.1.1 Variable heat source problem

In our problem, temperature is constant along the z-direction, and the top and bottom of the solution domain are considered adiabatic. Thus, even though the solution domain is 3D, we will just study its cross domain OACB. In this case, OACB is set as Dirichlet boundaries $T = 0$, as shown on the right of Fig. 1. Thermal conductivity is set as 0.01 $\mathrm{W}/\,(\mathrm{m} \cdot \mathrm{K})$. Heat source $Q\,(\mathbf{x}) = Q\,(x, y)$ is also implemented with Gaussian random field generated with the Karhunen–Loéve expansion:

$$\begin{aligned} Q\,(\mathbf{x}) &\sim \mathcal{GP}\left(0, k_l\left(\mathbf{x}, \mathbf{x}'\right)\right) \mathrm{W/m^2} \\ k_l\left(\mathbf{x}, \mathbf{x}'\right) &= \sigma e^{-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}} \mathrm{W/m^2} \end{aligned} \tag{2}$$

where $l = \frac{1}{2}, \sigma = 1$.

2.1.2 Darcy flow

Following [2], variable conductivity field is modeled according to $\eta \sim \mathcal{N}\left(0, (-\Delta + 9I)^{-2}\right)$:

$$\kappa(\mathbf{x}) = \begin{cases} 12 & \text{if } \eta(\mathbf{x}) \geq 0 \\ 3 & \text{if } \eta(\mathbf{x}) < 0 \end{cases} \tag{3}$$

Source term Q is set as 1.

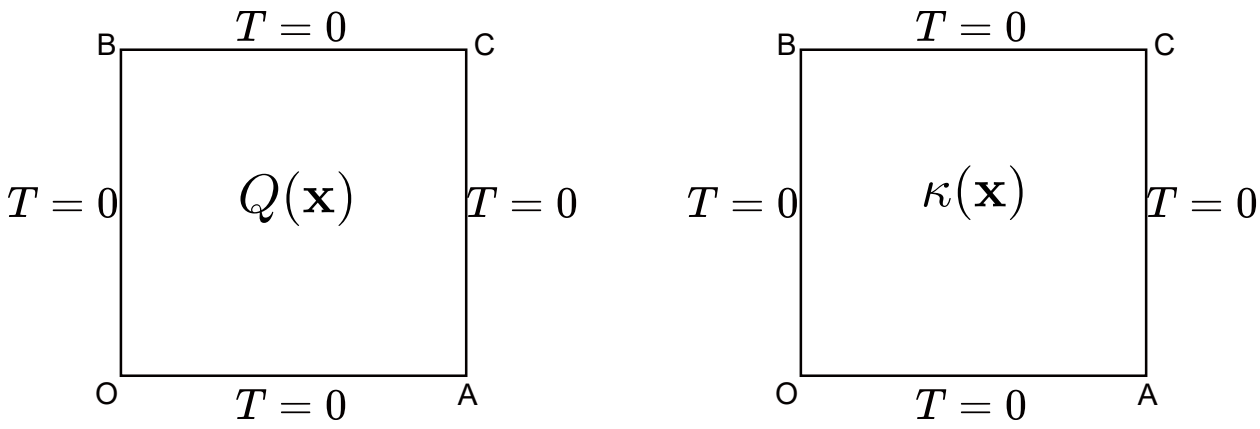

Figure 1: **Geometry and boundary conditions of Variable heat source problem and Darcy flow problem.** Left (variable heat source problem): The solution domain OACB is a cross-section of 1m×1m×1m unit cube. All boundaries are Dirichlet boundaries with $T = 0$. Right (darcy flow): The solution domain OACB is a 1×1 unit square. All boundaries are Dirichlet boundaries with $T = 0$.

S2.2 Variable stiffness elasticity

An elastic variable stiffness square plate with in-plane deformation is investigated in this section, and it is considered to be made of the fiber reinforced material, as shown in Fig. 2. The thickness of the plate is 0.125mm. Due to spatial variation of fiber orientation, the material property of the plate shows anisotropy. The material property matrix in x-y coordinate system $\mathbf{C}_{xy}$ can be written as:

$$\mathbf{C}_{xy} = \mathbf{T}^{-1}\mathbf{C}_{12}\mathbf{T}^{-\mathrm{T}} \tag{4}$$

where

$$\mathbf{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{5}$$

And $\mathbf{C}_{12}$ is the material property matrix in the principle material coordinates, which is not effected by the fiber angle. Eq. 6 gives the formulation of compliance matrix $\mathbf{S}_{12}$, i.e., the inverse of $\mathbf{C}_{12}$:

$$\mathbf{S}_{12} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & 0 \\ 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \tag{6}$$

We first calculate $\mathbf{S}_{12}$, then calculate the inverse of $\mathbf{S}_{12}$ to derive $\mathbf{C}_{12}$.

According to [3], AS4/3501-6 is chosen as the material of the plate. Mechanical properties of AS4/3501-6 used in this Article are given in Table 2. In this section, the fiber angle field $\theta = \theta(x, y)$ of the plate is characterized in two ways: (1) Linear 1-D variation. (2) B-splines surface. These two ways with their results are discussed separately in the section 2.2.1 (case 1) and section 2.2.2 (case 2). The boundary conditions of both cases are same, as shown in Fig. 2. The goal of the VOL is to learn the mapping between the fiber angle field space and the vector space of displacement components $[u_1, u_2]^{\mathrm{T}}$. In this Article, two neural operators are used to learn u_1 and u_2 separately. As a simple feature engineering, $\sin\theta$, $\cos\theta$, $sin2\theta$, $\cos 2\theta$ are fed to neural operators, rather than θ itself.

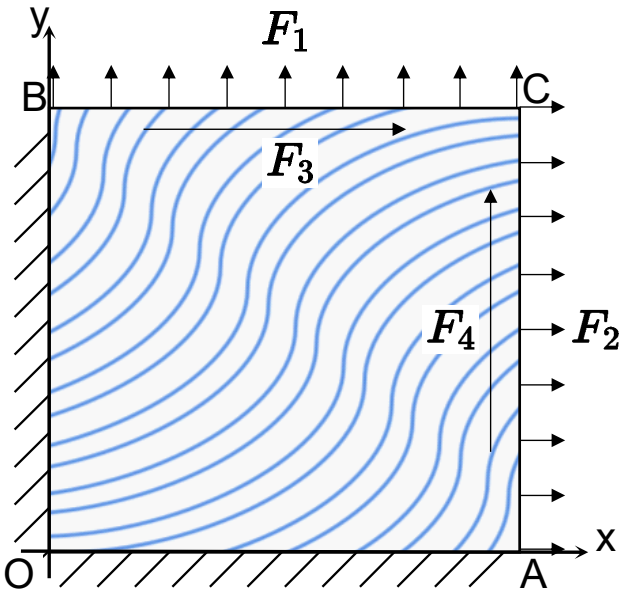

Figure 2: **Geometry and boundary conditions of variable stiffness elasticity problems.** A fiber-reinforced panel OACB is considered. OA=AC=CB=BO=100mm, OA and OB are displacement boundaries with $u_1 = u_2 = 0$, AC and CB are traction boundaries. F_1 and F_3 are line loads applied on boundary BC. F_2 and F_4 are line loads applied on boundary AC. F1=5 N/mm, F3=5*x N/mm, F2=5 N/mm and F4=5*y N/mm. The variation of fiber angle leads to variable stiffness property.

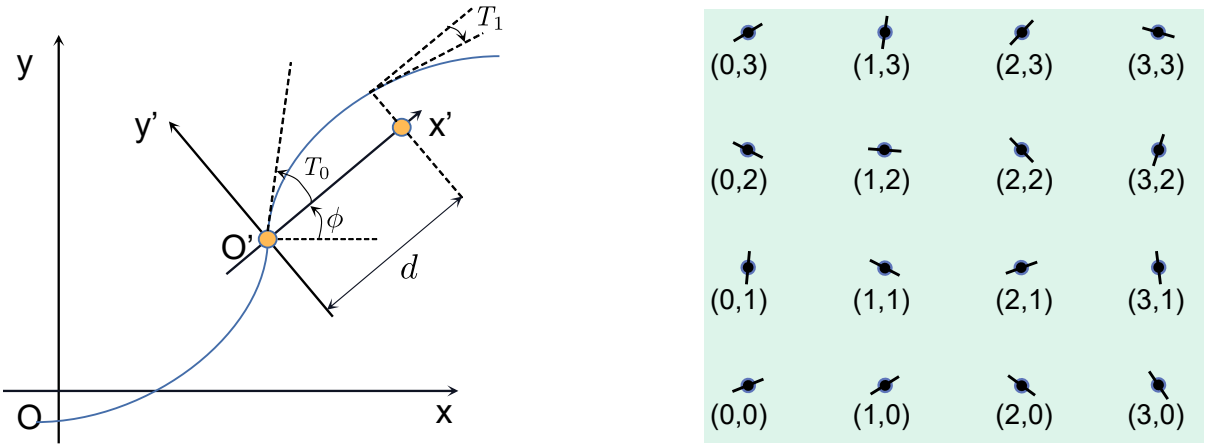

Figure 3: Left: Linear variation of the fiber orientation. Right: 4×4 control points with random angle values.

Table 2: Mechanical properties of AS4/3501-6 [3] used in this Article.

E_1 (MPa)	E_2 (MPa)	ν_{12}	G_{12}(MPa)
126000	11000	0.28	6600

2.2.1 Elasticity A: Linear 1-D variation

According to [4, 5], the fiber orientation angle along a reference path takes the form

$$\theta\left(x'\right) = \phi + \left(T_1 - T_0\right)\frac{|x'|}{d} + T_0 \tag{7}$$

where ϕ is the angle of rotation of the new coordinate system x'-O'-y' with respect to the original coordinate system x-O-y. x' is the x' coordinate in coordinate system x'-O'-y', L denotes the side length of the plate, T_0 is the fiber orientation angle according to x' axis at the panel center, T_1 is the fiber orientation angle according to x' axis at $x' = \pm d$, as shown on the left of Fig. 3. In this case, d is set as half of the side length of the panel. The fiber angle fields can then be sampled by sampling ϕ, T_0 and T_1. In this case, we use Latin hypercube sampling (LHS) to sample the tuple (ϕ, T_0, T_1). The bounds of ϕ, T_0 and T_1 are all set as $\left(-\frac{1}{2}\pi, \frac{1}{2}\pi\right)$. Total 100000 tuples of $\left(-\frac{1}{2}\pi, \frac{1}{2}\pi\right)$ are sampled for this case, and only the first 12005 samples are utilized for the generation of angle fields.

2.2.2 Elasticity B: B-Splines surface

Following [6], the fiber angle fields can be characterized by the idea of the B-Splines. The B-Splines surface representation [7] is:

$$\mathbf{S}(u, v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\mathbf{P}_{i,j} \tag{8}$$

where $N_{i,p}(u)$ and $N_{j,q}(v)$ denote basis functions along u direction and v direction respectively, $\mathbf{P}_{i,j}$ denotes coordinates of control points. In this case, a linear mapping between the physics coordinates (x, y) and the parameter coordinates (u, v) is defined:

$$u = \frac{x}{L}, v = \frac{y}{L} \tag{9}$$

The control points of th B-spline are equally spaced in the x and y directions. We just fix all x and y coordinates of control points, and keep the z coordinates active. We do not use the actual B-spline surface to characterize the angle field. Instead, the fiber angle field is treated as the z axis component $Z(x_p, y_p)$ of the B-Splines surface. For a specific position (x_p, y_p) of the plate, the parameter coordinates of which are (u_p, v_p), we have the fiber angle $\theta(x_p, y_p)$ at the position (x_p, y_p)

$$\theta(x_p, y_p) = Z(x_p, y_p) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u_p)N_{j,q}(v_p)Z_{i,j} \tag{10}$$

Thus, the fiber angle fields can be sampled by sampling the z coordinates of all control points. In our case, $m = n = 3$, $p = 2$, i.e., a 4×4 mesh of control points and 2nd-degree B-spline basis functions are used. The control points are illustrated on the right of Fig. 3. We use LHS to sample the z coordinates of all control points, and all bounds of the z coordinates of all control points are set as $\left(-\frac{1}{2}\pi, \frac{1}{2}\pi\right)$. Like elasticity A, we also generate 100000 LHS samples and use the first 12005 samples to generate angle field.

S3 Metrics

1. *Relative L^2 error.* Consider a label $\hat{y}$ and its predition y, the relative L^2 error is defined as

$$\text{Relative } L^2 \text{ error} = \frac{\| y - \hat{y} \|_2}{\| \hat{y} \|_2}. \tag{11}$$

For a dataset $\mathcal{D}$, we calculate the average relative L^2 error on $\mathcal{D}$

$$\text{Average relative } L^2 \text{ error} = \frac{1}{N_{\text{data}}} \sum_{i \in \mathcal{D}} \frac{\| y_i - \hat{y}_i \|_2}{\| \hat{y}_i \|_2}. \tag{12}$$

2. *Residual norm.* The norm of residual tensor $\mathbf{R}$, which can be derived with Ritz approach or Galerin approach in this work. Note lower residual norm of one prediction does not necessarily indicate the prediction is closer to the ground truth. For all experiments in this work, we use L^2 norm of residual $\| \mathbf{R} \|_2$, and we mainly record average $\| \mathbf{R} \|_2$ per batch as the training error.

S4 Data generation

S4.1 variable heat source problem and Darcy problem

All labels for the test and the distribution-shift session of the output of these two problems are generated with the commercial FEM software COMSOL [8]. For both problems, the 4-node bilinear quadrilateral element is used for meshing. For variable heat source problem, we choose the mesh with size 255×255, so we have a single resolution 256×256 for variable heat source problem. For Darcy problem, we mesh the domain with various mesh sizes, 31×31, 63×63, 127×127, 255×255 and 511×511, corresponding to the resolution 32×32, 64×64, 128×128, 256×256 and 512×512 respectively, to get dataset with different resolutions.

S4.2 elasticity A and elasticity B problem

All labels for the test and the distribution-shift session of the output of these two problems are generated with the iteration solver we have written with Pytorch or the commercial FEM software ABAQUS [9]. For both problems, the 4-node bilinear quadrilateral element with reduced integration is used for meshing. For the iteration solver, we do not consider the hourglass effect [10]. For elasticity A problem, we mesh the domain with a mesh size 32×32, corresponding to the resolution 33×33. For elasticity B problem, we generate the dataset with mesh size 32×32, 64×64, 128×128, 256×256 and 512×512, corresponding to the resolution 33×33, 65×65, 129×129, 257×257 and 513×513 respectively.

S5 Network settings

In this section, we detail the network settings for all problems. We use Fourier neural operator (FNO [2]) with 4 layers and GELU [11] activation function for all problems. We list the network settings details for all problem:

1. For variable heat source problem and Darcy problem, we use one FNO to learn the mapping, while for elasticity A and elasticity B problem we use two FNOs to learn u_1 and u_2 respectively, and the two FNOs have same amount of parameters, and share the same input.

2. For variable heat source problem, we use $Para\ |_{\text{G}}$ as input, while for other problems, we use $Para\ |_{\text{N}}$ as network input.

3. About feature engineering, for variable heat source problem and Darcy problem, we concat coordinate embedding to parameter tensor; for elasticity A and elasticity B problem, we have introduced in section 2.2.

4. For all problems, the width of each Fourier layer is 32. For Darcy problem, elasticity A and elasticity B problem, the number of origin modes and the number of transformed modes of Fourier layer are both 16. For variable heat source problem, we set number of the number of origin modes and the number of transformed modes of Fourier layer both to 14.

5. Lifting operation: A linear layer maps the input to 32 channels.

6. Projection operation uses such a sequence: Linear layer maps 32 channels to 128 channels - GELU [11] activation function - Linear layer maps 128 channels to 1 channel (output channel).

7. For all experiments, we keep data flow and parameters to type `float32`.

S6 Experimental details

For all experiments with VOL and the data-driven strategy, we use only 5 data as shift set, the training set only contains parameter tensor samples. For all experiments with neural operators, we run 5 times with different random seeds to guarantee the stability of our results, and we report mean and one standard deviation of the results.

S6.1 Scaling experiments

1. *Resolution.* The resolution of heat problem and Darcy flow problem is 256×256. For elasticity A the resolution is 33×33. For elasticity B the resolution is 257×257.

2. *Training setting.* For all cases and all sizes, we use Adamw [12] as optimizer. For the learning rate setting, we use a 0.5-cycle cosine annealing scheduler with warmup for the first $1/10$ iteration steps. For all cases and all sizes, we set the learning rate as 0.01. For all cases and all sizes, we use two-step update with conjugate gradient method (VOL+CG(2)). The batch size is set as 16. For elasticity A, elasticity B and variable heat source problem, the number of epochs is 50. For darcy flow problem, we just use 10 epochs for training.

3. *Metrics.* The metric of test error in scaling experiments is the *relative L^2 error*, see the definition in section 3.

S6.2 Resolution experiments

1. *Training setting.* For all cases and all resolutions, we use Adamw [12] as optimizer and use a 0.5-cycle cosine annealing scheduler with warmup for the first $1/10$ iteration steps. For all cases and all resolutions, we use two-step update with conjugate gradient method, the number of epochs is 50. For all resolutions of elasticity problem, the learning rate is 0.01, while for all resolutions of darcy flow the learning rate is 0.02. We set batch size as 8 for elasticity B problem at resolution 513×513, and set batch size as 16 for all other resolution experiments.

2. *Metrics.* The metric of test error in resolution experiments is the *relative L^2 error*, see the definition in section 3.

S6.3 Comparative experiments verifying generalization benefits

We conduct two experiments to verify generalization benefits of VOL. The first experiment is to compare VOL and the classical iterative method (conjugate gradient method with a restart procedure every epoch for these two experiments, CG(i)) in the same set, in which we compare the results of VOL and iteration solver

on the training set. The second experiment is to compare VOL and the classical iterative method in test set, in which we train VOL on a training set, and run test on a test set with a same size as training set, while the classical iterative method is run on the test set. The size of the training set in the first experiment, the size of the training and the test set in the second experiment are both 2000. We choose 1,2,5,10,25,50,100 as update steps for VOL and classical iterative method in both experiments. And for VOL, we use 50 epochs with 0.02 as learning rate, and use a 0.5-cycle cosine annealing scheduler with warmup for the first $1/10$ iteration steps. The batch size is set as 16. We conduct these two experiments on elasticity B problem at resolution 33×33, 129×129, 257×257. For CG(i), at the beginning of a new epoch, we restart the conjugate gradient method and then run it with the updated solution from the previous epoch. For the first experiment, we use average *relative* L^2 *error* on the training set per epoch as training error. For the second experiment we use average *relative* L^2 *error* on the test set as test error.

S6.4 Comparative experiments on different optimization strategies

We use Darcy flow problem at resolution 512×512 for experiments about comparison on different optimization strategies.

1. *Training setting.* For all experiments, we use Adamw [12] as optimizer and use a 0.5-cycle cosine annealing scheduler with warmup for the first $1/10$ iteration steps. We train all expriments with a 0.02 learning rate, a batch size of 16 and 50 epochs.

2. *Metrics.* The metric of training error is average residual norm per batch, and the test error is the average *relative* L^2 *error* on the test set, see the definition in section 3.

S7 Configuration and wall clock time of experiments

S7.1 Configuration

All experiments were conducted on a single NVIDIA RTX 3090. The program was written in Pytorch [1].

S7.2 Wall clock time

We list wall clock time for scaling experiments in Table 3, resolution experiments in Table 4, two experiments verifying generalization benefits in Table 5, comparative experiments on different optimization strategies in Table 7. For comparative experiments verifying generalization benefits, we provide wall clock time of one iteration for VOL+CG(i) and CG(i), where we run 100 iterations to get the average and the standard deviation wall clock time, and we give the wall clock time record of the second experiments.

Wall time (s) / Problem \ Size of training set	100	200	500	1000	2000	5000	10000
Variable heat source problem	57.59±1.87	112.56±1.45	282.15±4.48	563.50±10.20	1117.92±15.24	2845.28±37.83	5670.28±83.80
Darcy problem	11.78±1.45	21.98±0.28	55.02±0.78	110.34±2.04	219.60±1.38	553.69±9.34	1104.75±15.92
Elasticity A	16.48±0.32	31.36±0.47	77.96±2.05	153.10±3.19	303.97±7.46	775.30±15.95	1524.67±29.31
Elasticity B	200.36±1.79	390.75±4.66	993.51±9.01	1969.70±16.58	2851.90±45.47	9902.78±75.40	19843.14±61.25

Table 3: **Wall clock time (s) of scaling experiments.** Average time with one standard deviation is shown.

Darcy flow	Resolution	32	64	128	256	512
	Wall clock time (s)	124.31±13.97	167.43±16.14	289.19±5.72	1163.78±11.48	4501.87±284.12
Elasticity B	Resolution	33	65	129	257	513
	Wall clock time (s)	398.48±1.36	527.42±2.15	1179.96±10.98	2851.90±45.47	16866.95±38.02

Table 4: **Wall clock time of resolution experiments.**

Wall time (ms) / Update step number i / Resolution	1	2	5	10	25	50	100
CG(i) at resolution 33×33	5.31±0.31	6.53±0.32	10.05±0.52	16.21±0.82	34.08±1.53	64.62±3.42	125.80±7.15
VOL+CG(i) at resolution 33×33	27.76±0.84	29.03±1.61	32.74±1.24	39.38±1.69	58.75±2.26	91.42±3.39	156.69±6.41
CG(i) at resolution 129×129	6.21±0.33	7.51±0.45	11.47±0.56	17.67±0.62	37.08±2.14	68.99±3.40	134.22±6.36
VOL+CG(i) at resolution 129×129	32.40±1.24	32.61±1.06	36.84±1.62	43.47±1.22	63.32±2.40	96.40±4.27	164.59±11.82
CG(i) at resolution 257×257	9.15±0.43	9.95±0.37	13.91±0.52	20.40±1.04	40.42±1.45	73.72±2.66	141.73±7.91
VOL+CG(i) at resolution 257×257	35.55±4.48	36.92±4.08	39.96±2.04	47.10±3.51	68.51±4.60	102.60±19.30	167.47±6.98

Table 5: **Wall time of one iteration of CG(i) and VOL+CG(i) with different numbers of update steps.**

Wall time (s) / Update step number i / Resolution	1	2	5	10	25	50	100
33×33	390.28±10.82	398.48±1.36	442.65±1.49	506.37±0.83	707.85±1.04	977.82±92.06	1408.21±7.77
129×129	1178.44±40.43	1179.96±10.98	1203.18±6.22	1269.53±3.73	1447.54±8.40	1742.36±5.39	2318.77±3.21
257×257	2773.74±25.46	2851.90±45.47	2894.66±20.62	3025.77±0.86	3537.62±3.90	4378.25±24.11	6119.27±17.73

Table 6: **Wall time of VOL+CG(i) in the second experiment verifying generalization benefits.**

Strategy	Data-driven strategy	VOL+CG(2)	VOL+CG(500)	VOL+DM
Wall clock time (s)	4608.92±43.90	4501.87±284.12	16339.63±173.32	4279.45±54.5

Table 7: **Wall time of experiments about comparison on different optimization strategies.**

S8 Further consideration about VOL+DM

In this section, we will show VOL+DM is equivalent to the direct iterative update on the normal equation of the original linear system in a way. Let's consider about the set 2-norm as our loss function $\mathcal{L} = \|\mathbf{R}\|_2 = \|(\mathbf{P} - \mathbf{K}\mathbf{a})^{\mathrm{T}}(\mathbf{P} - \mathbf{K}\mathbf{a})\|_2$. The gradient of the loss function with respect to the node solution prediction is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}} = \frac{\partial \|\mathbf{R}\|_2}{\partial \mathbf{a}} = \frac{1}{2\|\mathbf{R}\|_2} \frac{\partial[(\mathbf{P} - \mathbf{K}\mathbf{a})^{\mathrm{T}}(\mathbf{P} - \mathbf{K}\mathbf{a})]}{\partial \mathbf{a}} = -\frac{\mathbf{K}^{\mathrm{T}}(\mathbf{P} - \mathbf{K}\mathbf{a})}{\|\mathbf{R}\|_2}. \tag{13}$$

In the deep learning pipeline, the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{a}}$ will then be propagated back to the parameters of neural operator module, which would be updated with the gradient, network optimizer and learning rate scheduler. And From the equation above, we note that the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{a}}$ in VOL+DM is essentially a one-step direct iterative update on the normal equation of the original linear system. This might explain the gap between VOL+DM and other optimization strategies. Besides, This also inspires us to design other optimization strategies in our future work, for example, we might design a Jacobi-like strategy to update the parameters, to let the gradient be

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}} = -\mathbf{D}^{-1} \frac{\mathbf{K}^{\mathrm{T}}(\mathbf{P} - \mathbf{K}\mathbf{a})}{\|\mathbf{R}\|_2}, \tag{14}$$

where $\mathbf{D}$ is a matrix with the diagonal part of $\mathbf{K}^{\mathrm{T}}\mathbf{K}$.

References

[1] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[2] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[3] A. S. Kaddour, M. J. Hinton, P. A. Smith, and S. Li. Mechanical properties and details of composite laminates for the test cases used in the third world-wide failure exercise. *Journal of Composite Materials*, 47:2427–2442, 9 2013.

[4] Zafer Gürdal and Reynaldo Olmedo. In-plane response of laminates with spatially varying fiber orientations: Variable stiffness concept. *AIAA Journal*, 31:751–758, 1993.

[5] Zafer Gürdal, Brian. F. Tatting, and C. K. Wu. Variable stiffness composite panels: Effects of stiffness variation on the in-plane and buckling response. *Composites Part A: Applied Science and Manufacturing*, 39:911–922, 5 2008.

[6] Kunpeng Zhang, Dachuan Liu, Qun Wang, Peng Hao, Yuhui Duan, Hao Tang, and Bo Wang. Multi-level intelligent design of variable angle tow laminates via image-driven method. *Composite Structures*, 303, 1 2023.

[7] Les Piegl and Wayne Tiller. *The NURBS book*. Springer, 1997.

[8] COMSOL. http://www.comsol.com/products/multiphysics/.

[9] ABAQUS. https://www.3ds.com/products-services/simulia/products/abaqus/.

[10] Ted Belytschko, Jame Shau-Jen Ong, Wing Kam Liu, and James M. Kennedy. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43:251–276, 05 1984.

[11] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415*, 2016.

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.