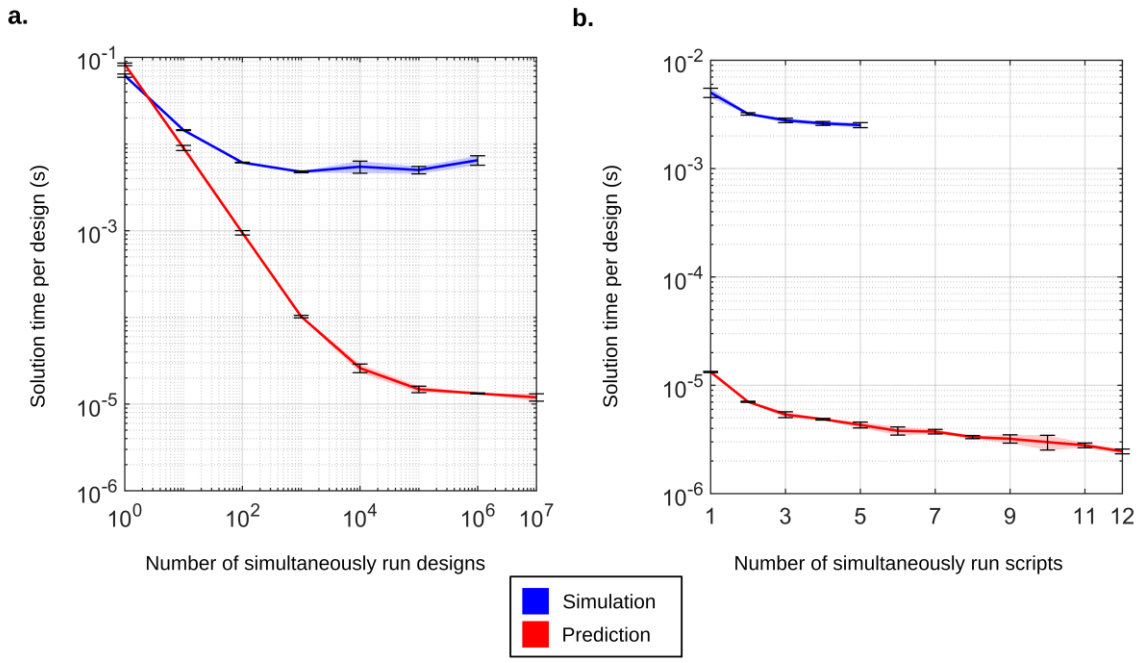
Supplementary information for

Deep learning for the rare-event rational design of 3D printed multi-material mechanical metamaterials

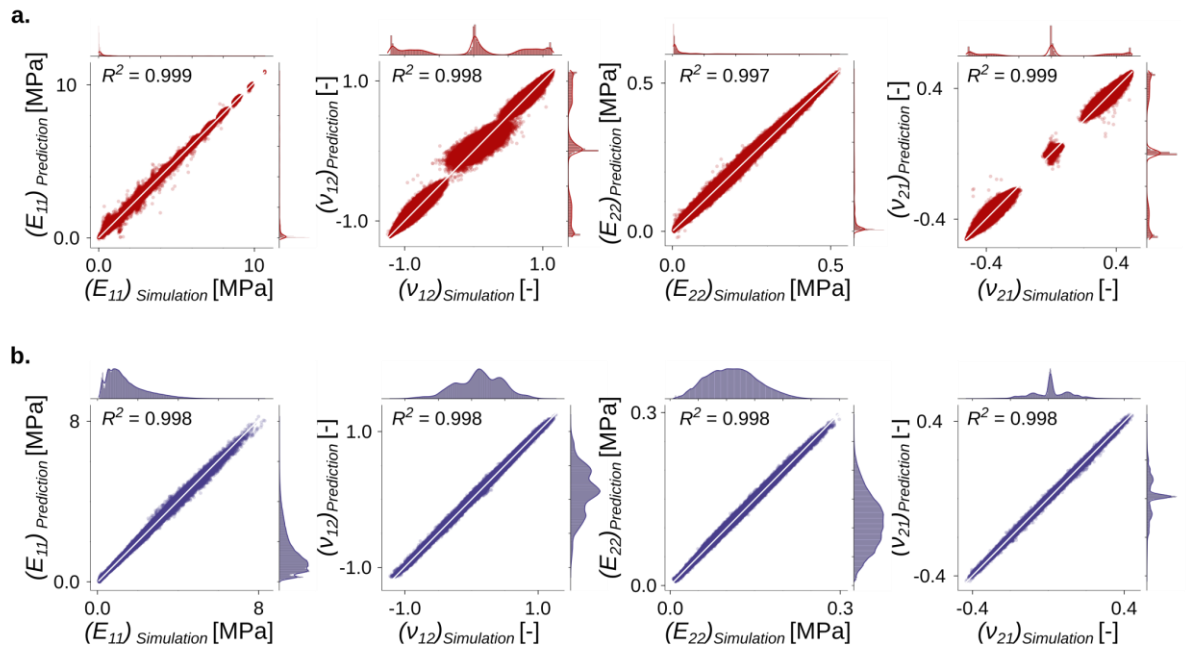
Helda Pahlavani[a,*], Muhamad Amani[a], Mauricio Cruz Saldívar[a], Jie Zhou[a], Mohammad J. Mirzaali[a], Amir A. Zadpoor[a]

[a] *Department of Biomechanical Engineering, Faculty of Mechanical, Maritime, and Materials Engineering, Delft University of Technology (TU Delft), Mekelweg 2, 2628 CD, Delft, The Netherlands*

[*] Corresponding author. Tel.: +31-613221032, *e-mail:* h.pahlavani@tudelft.nl



Supplementary Figure 1. The solution time per design. The solution time per design depends on the number of simultaneously run simulations/predictions as well as the number of scripts run in parallel. A comparison between the finite element simulation time (Matlab) and the deep learning prediction time (Python) for the single unit cell model (**a**) for the different numbers of simultaneously run simulations/predictions (without any parallelization) and (**b**) for the different numbers of in-parallel run scripts (10^5 simultaneously run finite element simulations and 10^6 simultaneously run deep learning predictions per script). The standard deviation of the reported times was calculated by repeating each experiment three times.



Supplementary Figure 2. The evaluation of the trained deep learning models. The prediction *vs.* simulation results and the coefficients of determination for the test datasets for a the single unit cell model and **b** four-tile model.

Supplementary Table 1. The total number of possible designs.

Group number	ρ_h [%]	Number of hard elements	Number of possible designs	Number of possible designs considering symmetry
1	5	8	5.25721×10^{12}	1.31400×10^{12}
2	10	15	1.62392×10^{20}	4.05981×10^{19}
3	20	30	3.21988×10^{31}	8.04970×10^{30}
4	30	45	4.41669×10^{38}	1.10417×10^{38}
5	40	60	4.62150×10^{42}	1.15537×10^{42}
6	50	75	9.28261×10^{43}	2.32065×10^{43}
7	60	90	4.62150×10^{42}	1.15537×10^{42}
8	70	105	4.41660×10^{38}	1.10417×10^{38}
9	80	120	3.21988×10^{31}	8.04970×10^{30}
10	90	135	1.62392×10^{20}	4.05981×10^{19}
11	95	142	5.25721×10^{12}	1.31430×10^{12}
Sum			1.02070×10^{44}	2.55175×10^{43}
Total number considering three groups of unit cells with positive, zero, and negative values of the Poisson's ratio				7.65525×10^{43}

Supplementary Table 2. The coefficients of determination, the mean absolute error (MAE), and the mean squared error (MSE) for the outputs of the single unit cell and four-tile deep learning models.

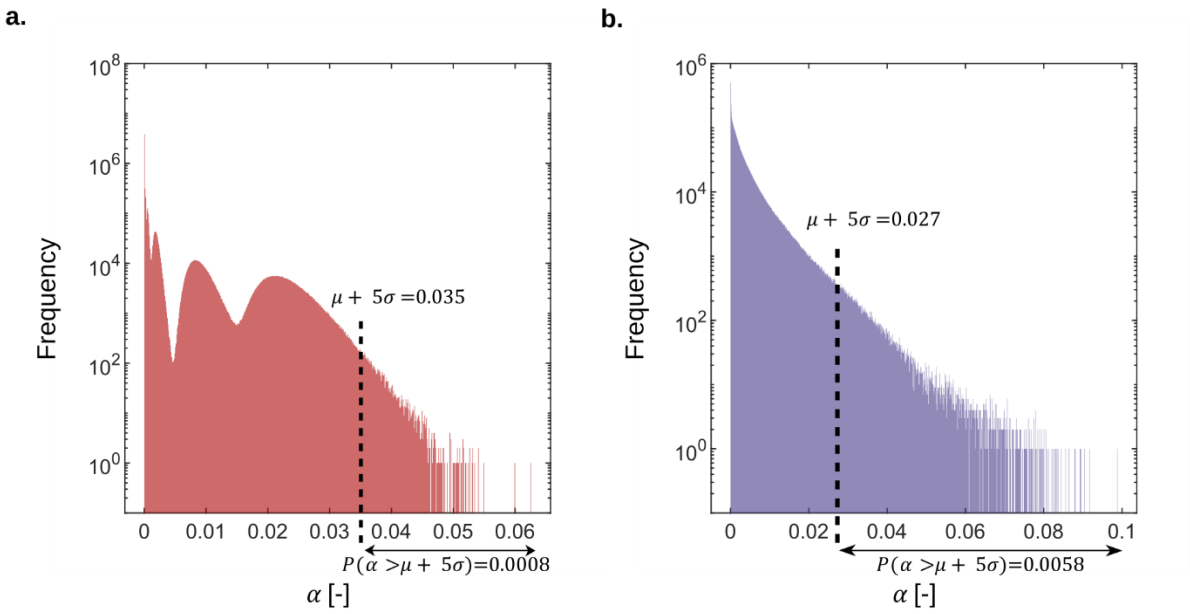
		Coefficient of determination (R^2)	MSE	MAE
Single unit cell model	**Total**	9.98×10^{-1}	1.14×10^{-4}	6.38×10^{-3}
	$\boldsymbol{E_{11}}$	9.99×10^{-1}	2.18×10^{-5}	2.65×10^{-3}
	$\boldsymbol{E_{22}}$	9.97×10^{-1}	8.96×10^{-5}	5.51×10^{-3}
	$\boldsymbol{\nu_{12}}$	9.98×10^{-1}	2.34×10^{-4}	1.04×10^{-2}
	$\boldsymbol{\nu_{21}}$	9.99×10^{-1}	1.11×10^{-4}	6.96×10^{-3}
Four-tile model	**Total**	9.98×10^{-1}	3.77×10^{-5}	4.58×10^{-3}
	$\boldsymbol{E_{11}}$	9.98×10^{-1}	2.96×10^{-5}	3.86×10^{-3}
	$\boldsymbol{E_{22}}$	9.98×10^{-1}	4.80×10^{-5}	5.30×10^{-3}
	$\boldsymbol{\nu_{12}}$	9.97×10^{-1}	4.92×10^{-5}	5.46×10^{-3}
	$\boldsymbol{\nu_{21}}$	9.98×10^{-1}	2.38×10^{-5}	3.69×10^{-3}

Supplementary Table 3. The range of the elastic properties for the single unit cell lattice structures corresponding to the data presented in Figure 1.

Type of unit cell	$\boldsymbol{E_{11}}$ [MPa]		$\boldsymbol{E_{22}}$ [MPa]		$\boldsymbol{\nu_{12}}$ [−]		$\boldsymbol{\nu_{21}}$ [−]	
	min	**max**	**min**	**max**	**min**	**max**	**min**	**max**
$\theta = 60°$	0.01	1.02	0.00	0.29	-1.24	-0.31	-0.53	-0.17
$\theta = 90°$	0.11	10.67	0.00	0.34	-0.70	0.78	-0.06	0.09
$\theta = 120°$	0.01	1.39	0.00	0.54	0.31	1.16	0.14	0.51

Supplementary Table 4. The probability of finding α values that are one, two, three, four, or five standard deviations higher than their corresponding mean value. To calculate α, we first normalized all the properties (i.e., E_1, E_2, ν_{12}, and ν_{21}) between 0 and 1.

	Model	$\boldsymbol{\mu+\sigma}$	$\boldsymbol{\mu+2\sigma}$	$\boldsymbol{\mu+3\sigma}$	$\boldsymbol{\mu+4\sigma}$	$\boldsymbol{\mu+5\sigma}$
$P(\alpha > \mu + x\sigma)$	Single unit cell	0.111	0.078	0.041	0.007	0.0008
	Four-tile	0.108	0.047	0.022	0.011	0.0058



Supplementary Figure 3. **The histogram of α.** These histograms are for the double-auxetic lattice structurs ploted in Figure 1: **a** single unit cell and **b** four-tile models.

Supplementary Table 5. The geometrical parameters of the designed single unit cell lattices.

θ [°]	$W[mm]$	$C[mm]$	$w[mm]$	$c[mm]$	$l[mm]$	$h[mm]$	$t[mm]$	$T[mm]$
60	56.25	56.25	11.25	12.5	6.50	9.50	1	10
90	56.25	56.25	11.25	12.5	5.63	6.25	1	10
120	56.25	56.25	11.25	12.5	6.50	3.00	1	10

Supplementary Methods

Hyperparameter tuning of single unit cell model:

For the training of the single unit cell model, we used a sequential model composed of a linear stack of fully connected layers. Before training the model, we configured the learning process by defining several parameters, including an optimizer (RMSprop), a list of metrics (Mean Squared Error (MSE) and Mean Absolute Error (MAE)), and a loss function (MSE), which is the objective that the model will try to minimize.

In this model, we systematically studied the effects of different hyperparameters (i.e., the number of hidden layers, the number of neurons in each hidden layer, learning rate, and activation function) on the performance of the deep learning model. To evaluate the performance of the model with different hyperparameter values and also to detect overfitting during the training process, we created a subset of the data known as the validation dataset. Therefore, we generated a dataset of 165,000 samples, and then we randomly selected 20% of this dataset as a testing dataset, 80% as a training dataset, and 20% of the training dataset as a

validation dataset for hyperparameter tuning. It should be mentioned that for hyperparameter tuning, we trained each model until 20 epochs. This means that during each epoch, the model was trained based on the training data, and was tuned with the results of the metrics (MSE, MAE) for the validation dataset. Therefore, during the training process, the model did not see the validation dataset, while the validation dataset indirectly affected the model. In this way, we tuned the hyperparameters of the model based on the results of metrics for the validation dataset. The following systematic approach explains how those hyperparameters were selected in our models.

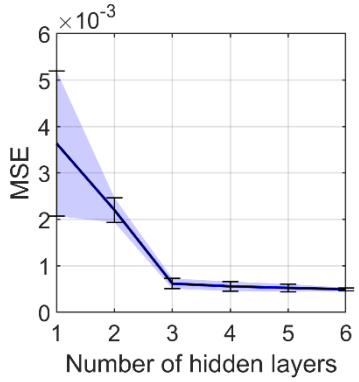
Number of hidden layers: At the first step, we studied the effects of the number of the hidden layers (i.e., 1, 2, 3, 4, 5, and 6) on reducing the loss function. Towards this aim, six models with different numbers of hidden layers were analyzed, while the rest of the hyperparameters for these models were kept unchanged*:*

- The number of neurons per each hidden layer: 128 (since we had 151 input parameters, we assumed the number of neurons in each hidden layer to equal 128).
- Learning rate: 0.0001.
- Activation function: ReLU.

Each model was trained three times to calculate the average and the standard deviation of the loss function (MSE) for each model. Supplementary Table 6 and Supplementary Figure 4 present the comparisons between the six models with different numbers of hidden layers.

Supplementary Table 6. The comparison between the six models with different numbers of hidden layers.

layers	MSE			MSE average	Standard deviation
	Trial 1	Trial 2	Trial 3		
1	0.0034	0.0053	0.0022	0.003633	1.28×10^{-3}
2	0.002	0.0021	0.0025	0.0022	2.16×10^{-4}
3	0.00049	0.00067	0.00069	0.000616	9.01×10^{-5}
4	0.00052	0.00048	0.00067	0.000558	8.32×10^{-5}
5	0.00046	0.00049	0.00062	0.000524	6.81×10^{-5}
6	0.00046	0.00052	0.00049	0.000494	2.55×10^{-5}



Supplementary Figure 4. Tuning the number of hidden layers. The comparisons between the MSE of six models with different numbers of hidden layers.

The results of the comparison between the models with different hidden layers showed that the MSE values converged after considering five hidden layers. Also, the low standard deviation indicates that the model is not sensitive to the randomly selected training dataset. Therefore, we selected five hidden layers as an optimum number of hidden layers for our model.

Number of neurons in each hidden layer: In the second step, we optimized the number of neurons in each hidden layer using "Hyperparameter Tuning" with the HParams Dashboard of Tensorboard [44]. For this aim, we selected the possible number of neurons per layer:

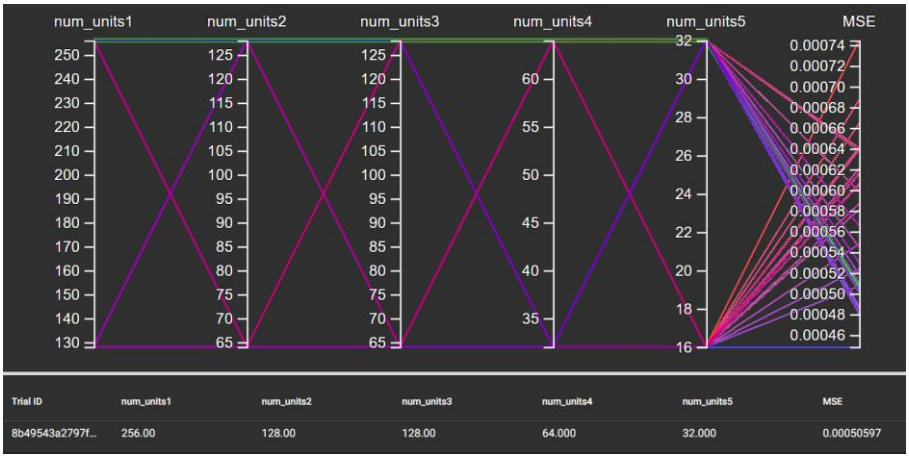
- Layer 1: 256 and 128 neurons
- Layer 2: 128 and 64 neurons
- Layer 3: 128 and 64 neurons
- Layer 4: 64 and 32 neurons
- Layer 5: 32 and 16 neurons

Since we had five layers and we assumed two possibilities for each layer, there were $2^5 = 32$ possible combinations that were studied to find the best combination. The model was trained for each combination, and the loss function (MSE) was calculated for each trained model. Also, we repeated the training process three times, and we compared the average and standard deviation of the loss function (MSE). As an example, Supplementary Figure 5 shows the parallel coordinate plot for the results of the second trial of the optimization of the number of neurons. Supplementary Table 7 summarizes the results of model training for different combinations of neurons per hidden layer. The results in this table were sorted based on the average and standard deviation of MSE. The best combination with the lowest averages of MSE was 256-128-128-64-32. This combination had also the lowest standard deviation of MSE among all the 32

combinations, meaning that the results of this combination were less sensitive to the selected training dataset

Supplementary Table 7. The comparisons between the trained models with different numbers of neurons per hidden layer.

Combination number	Number of neurons in each hidden layer					MSE			MSE average	MSE standard deviation
	1	2	3	4	5	Trial 1	Trial 2	Trial 3		
1	256	128	128	64	32	0.000493	0.000506	0.000502	0.0005	5.64×10^{-6}
2	256	128	128	32	32	0.00047	0.00048	0.000444	0.000465	1.52×10^{-5}
3	128	128	128	32	32	0.000562	0.000545	0.000606	0.000571	2.56×10^{-5}
4	128	64	128	32	32	0.000491	0.000504	0.000444	0.00048	2.58×10^{-5}
5	256	128	64	32	32	0.000554	0.000501	0.000486	0.000514	2.91×10^{-5}
6	128	64	128	64	16	0.000662	0.000581	0.000611	0.000618	3.32×10^{-5}
7	128	128	128	64	16	0.00051	0.000587	0.000579	0.000559	3.44×10^{-5}
8	128	64	64	64	16	0.000555	0.000616	0.000533	0.000568	3.50×10^{-5}
8	128	128	64	64	16	0.000618	0.000606	0.000686	0.000637	3.51×10^{-5}
10	128	64	128	32	16	0.000538	0.000605	0.000619	0.000587	3.54×10^{-5}
11	128	128	64	32	16	0.000532	0.000615	0.000619	0.000589	3.99×10^{-5}
12	256	128	128	64	16	0.000587	0.000524	0.000485	0.000532	4.22×10^{-5}
13	128	64	64	64	32	0.000606	0.000565	0.00049	0.000554	4.78×10^{-5}
14	256	64	128	32	32	0.000476	0.00053	0.000595	0.000534	4.84×10^{-5}
15	256	64	64	64	32	0.00046	0.00051	0.000585	0.000519	5.15×10^{-5}
16	256	128	64	32	16	0.000551	0.000448	0.000564	0.000521	5.18×10^{-5}
17	128	128	128	64	32	0.000435	0.000488	0.000572	0.000498	5.66×10^{-5}
18	256	64	64	32	32	0.00048	0.000599	0.000469	0.000516	5.87×10^{-5}
19	256	64	128	64	16	0.000587	0.000744	0.000677	0.000669	6.46×10^{-5}
20	128	128	64	64	32	0.000478	0.000638	0.00058	0.000565	6.59×10^{-5}
21	128	64	64	32	32	0.000659	0.00053	0.000497	0.000562	6.99×10^{-5}
22	128	128	128	32	16	0.000589	0.000639	0.000461	0.000563	7.48×10^{-5}
23	256	128	128	32	16	0.000436	0.000548	0.000637	0.00054	8.24×10^{-5}
24	256	64	128	64	32	0.000546	0.000641	0.000435	0.00054	8.42×10^{-5}
25	128	128	64	32	32	0.00046	0.000518	0.000683	0.000554	9.42×10^{-5}
26	256	128	64	64	16	0.000468	0.000641	0.000707	0.000606	1.01×10^{-4}
27	128	64	64	32	16	0.000773	0.00062	0.000517	0.000637	1.05×10^{-4}
28	256	128	64	64	32	0.000478	0.000565	0.000752	0.000599	1.14×10^{-4}
29	256	64	64	64	16	0.000479	0.000664	0.000763	0.000635	1.18×10^{-4}
30	128	64	128	64	32	0.00076	0.000485	0.000539	0.000595	1.19×10^{-4}
31	256	64	64	32	16	0.000883	0.000507	0.000586	0.000659	1.62×10^{-4}
32	256	64	128	32	16	0.000902	0.000687	0.000474	0.000688	1.75×10^{-4}



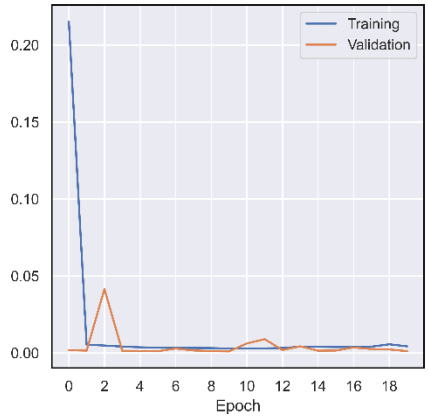
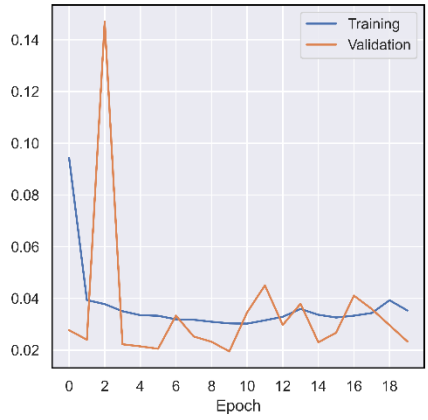
Supplementary Figure 5. Tuning the number of neurons in each hidden layer. The green line shows the following combination of neurons : 256-128-128-64-32.

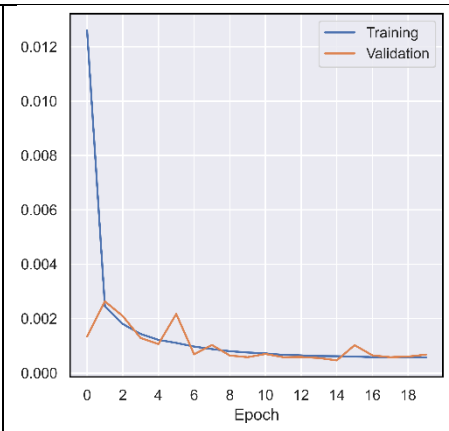
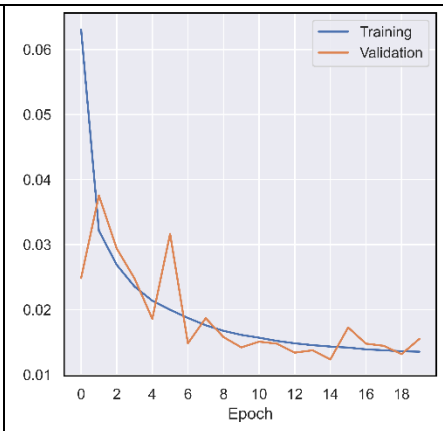
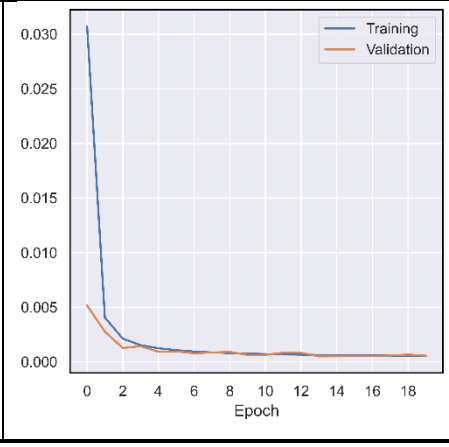
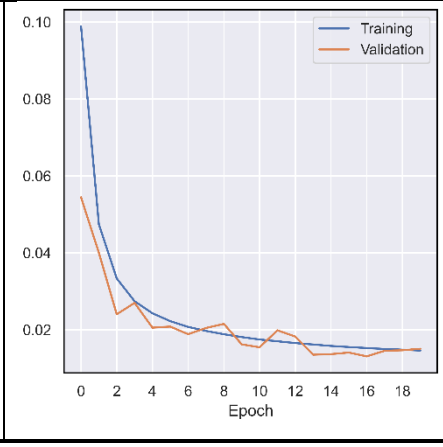
Learning rate: We analyzed the effects of the learning rate on the accuracy of the model. We selected three levels for the learning rate (i.e., 0.01, 0.001, and 0.0001) while keeping other hyperparameters unchanged:

- Number of hidden layers: 5.
- Number of neurons per each hidden layer: 256, 128, 128, 64, 32.
- Activation function: ReLU.
- Number of epochs: 20.

Supplementary Table 8 presents the comparison of models trained with different values of the learning rate. This highlights that the learning rate equal to 0.0001 provided the highest accuracy. It was, therefore, selected in our final model.

Supplementary Table 8. The comparison between the models trained with different values of learning rates.

Learning rate	MSE	MAE	Other parameters
0.01			R^2 for testing dataset: $R^2 = 0.958$ $(R^2)_{v_{12}} = 0{,}991$ $(R^2)_{v_{21}} = 0{,}995$ $(R^2)_{S_1} = 0{,}882$ $(R^2)_{S_2} = 0{,}963$ Training time: 4m, 16s

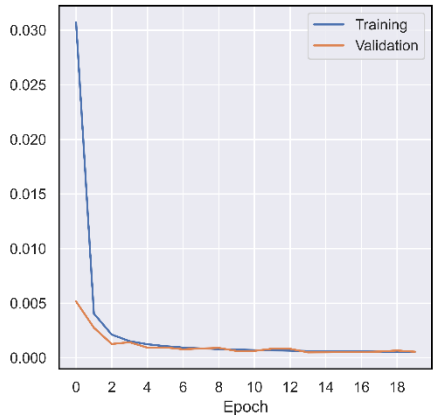
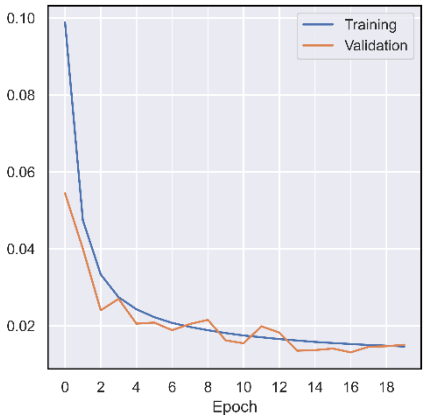
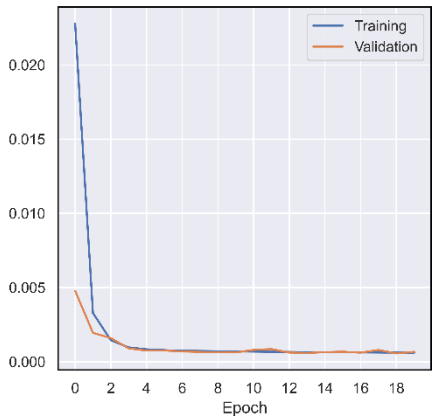
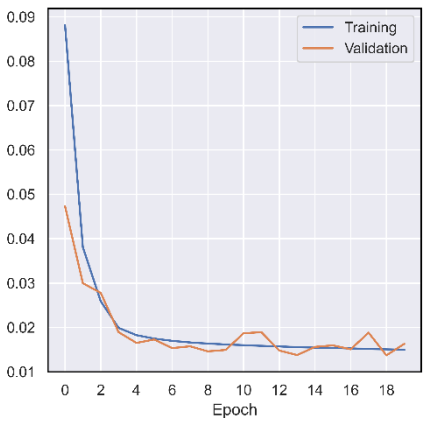
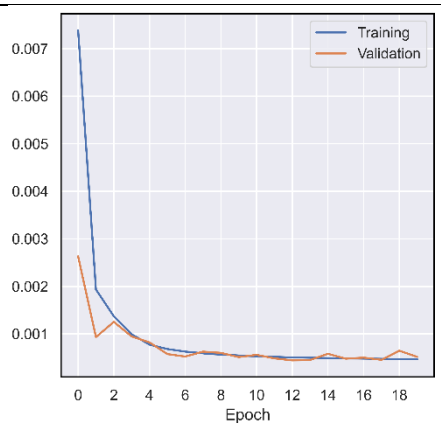
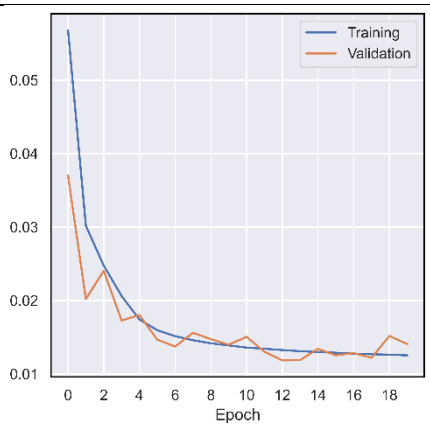
0.001			R^2 for testing dataset: $R^2 = 0.981$ $(R^2)_{v_{12}} = 0{,}992$ $(R^2)_{v_{21}} = 0{,}997$ $(R^2)_{S_1} = 0{,}958$ $(R^2)_{S_2} = 0{,}975$ Training time: 4m, 18s
0.0001			R^2 for testing dataset: $R^2 = 0.986$ $(R^2)_{v_{12}} = 0{,}992$ $(R^2)_{v_{21}} = 0{,}997$ $(R^2)_{S_1} = 0{,}969$ $(R^2)_{S_2} = 0{,}987$ Training time: 4m, 17s

Activation function: We also used different activation functions (i.e., ReLU, Sigmoid, and Tanh), while the rest of the hyperparamters were set as follows:

- Number of hidden layers: 5.
- Number of neurons per each hidden layer: 256, 128, 128, 64, 32.
- Learning rate: 0.0001.
- Number of epochs: 20.

Supplementary Table 9 shows the results of the comparison between the models trained with different activation functions. Based on these results, we selected the activation function ReLU in our single unitcell models.

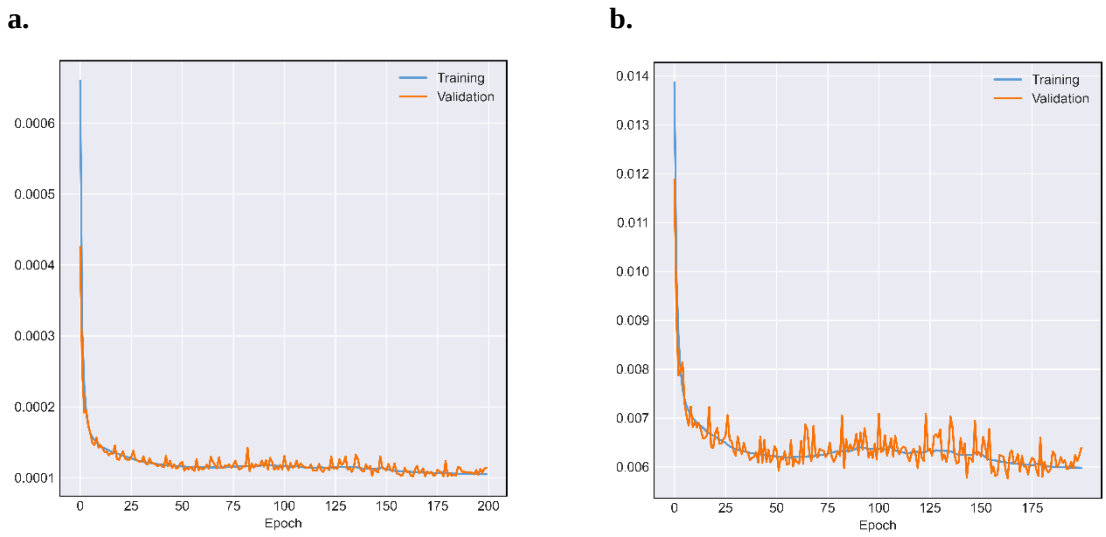
Supplementary Table 9. The results of the comparison between the models trained with different activation functions.

Activation function	MSE	MAE	Other parameters
ReLU			R^2 for testing dataset: $R^2 = 0.986$ $(R^2)_{v_{12}} = 0{,}992$ $(R^2)_{v_{21}} = 0{,}997$ $(R^2)_{S_1} = 0{,}969$ $(R^2)_{S_2} = 0{,}987$ Training time: 4m, 17s
Sigmoid			R^2 for testing dataset: $R^2 = 0.982$ $(R^2)_{v_{12}} = 0{,}993$ $(R^2)_{v_{21}} = 0{,}995$ $(R^2)_{S_1} = 0{,}964$ $(R^2)_{S_2} = 0{,}977$ Training time: 5m, 32s
Tanh			R^2 for testing dataset: $R^2 = 0.984$ $(R^2)_{v_{12}} = 0{,}994$ $(R^2)_{v_{21}} = 0{,}997$ $(R^2)_{S_1} = 0{,}962$ $(R^2)_{S_2} = 0{,}985$ Training time: 5m, 1s

We evaluated the performance of the model based on the coefficient of determination (R^2), MSE, and MAE values. The above-mentioned systematic analyses were, therefore, used to

determine the optimized architecture and hyerparameters of the model with the highest accuracy (see Table 2).

Once the hyperparameters were set, we increased the size of the training dataset and the number of epochs to improve the accuracy of our model further. For this purpose, we increased the size of training dataset by 100 times (i.e., a dataset of 16,500,000 lattice structures) and the number of epochs by 10 times (i.e., 200 epochs). Also, we assumed 20% of the training dataset as a validation dataset. In addition, we generated a dataset of 1,650,000 lattice structures as the testing dataset. MSE and MAE graphs for the single unit cell model are presented in Supplementary Figure 6. Quantitatively, the final optimized single unit cell model had a low MSE value of 1.05×10^{-4} and 1.14×10^{-4}, and a low MAE value of 6×10^{-3} and 6.38×10^{-3} for the training and validation datasets, respectively. Moreover, these graphs showed a good agreement between the training and validation datasets, confirming that the model was trained without overfitting.



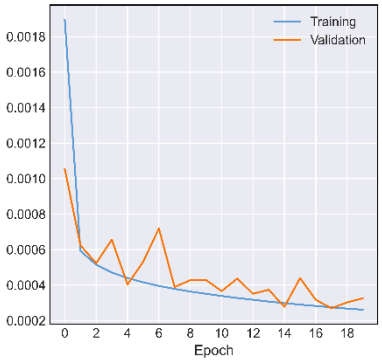
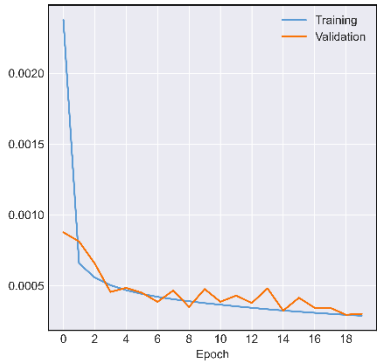
Supplementary Figure 6. a MSE graph and **b** MAE graph for the single unit cell model.

Hyperparameter tuning of four-tile model:

To design the architecture of the four-tile model, we selected the optimized hyperparameters obtained from the single unit cell model, namely five hidden layers (the number of neurons: 256-128-128-64-32), ReLU activation function, and a learning rate equal to 0.0001. Furtheremore, we trained another model with six hidden layers and the number of neurons was reduced from 256 to 8 (i.e., 256-128-64-32-16-8) to see if we could further improve the accuracy of the model. For these two models, we used a dataset of 250,409 lattice structures, and we randomly selected 20% of the dataset as a testing dataset, 80% as a training dataset, and 20% of the training dataset was randomly selected as a validation dataset. As compared to

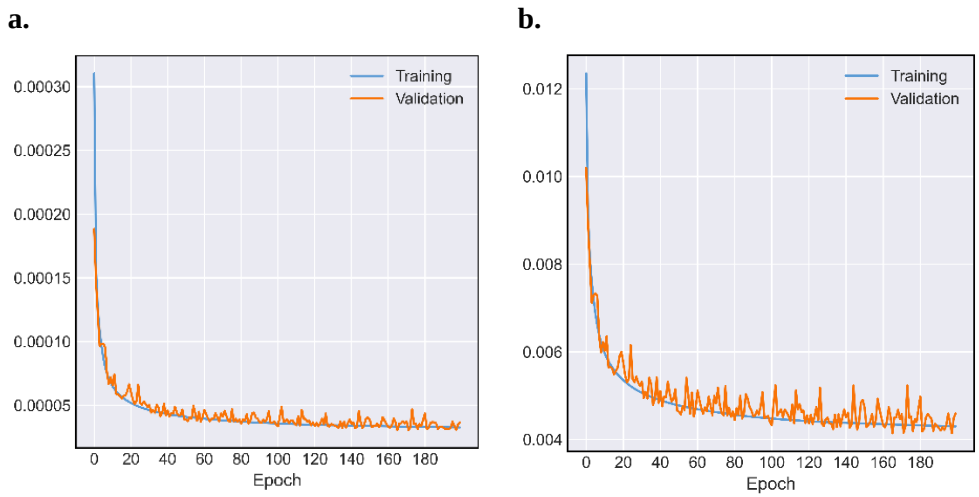
the model with five hidden layers, the model with six hidden layers had a slightly higher coefficient of determination for the testing dataset and a lower MSE value for the training and validation datasets with 20 epochs (see Supplementary Table 10). Since the six-layer model had low values of MSE and MAE and a high coefficient of determination, and also the MSE graph of this model showed a good agreement between the training and validation datasets to avoid overfitting, we stopped at this model and considered it to be our optimized four-tile model. Furthermore, we increased the size of the dataset to 15,331,140 lattice structures (60× larger) and the number of epochs by 10 times (i.e., 200 epochs). We randomly selected 90% of the dataset as the training dataset and the remaining 10% as the testing dataset. We also considered 20% of the training dataset for validation. The training procedure took 3569 minutes and 36 seconds. All the training parameters of the final four-tile model are listed in Table 2 of the manuscript. Quantitatively, the four-tile model had a high coefficient of determination of 0.998 and a low MSE value of 3.77×10^{-5} for the testing dataset, and respectively a low MSE value of 3.27×10^{-5} and 3.68×10^{-5} and a low MAE value of 4.3×10^{-3} and 4.6×10^{-3} for the training and validation datasets (see Supplementary Figure 7). Furtheremore, the MSE and MAE graphs showed a good agreement between the training and validation datasets, confirming that the model was trained without overfitting.

Supplementary Table 10. The comparison between the trained four-tile models with five and six hidden layers.

Architecture	5 hidden layers: 256-128-128-64-32	6 hidden layers: 256-128-64-32-16-8
Size of dataset	250,409	250,409
Epochs	20	20
Loss function (MSE)		
Coefficient of determination	$R^2 = 0{,}982$ $(R^2)_{v_{12}} = 0{,}967$ $(R^2)_{v_{21}} = 0{,}994$ $(R^2)_{S_1} = 0{,}984$ $(R^2)_{S_2} = 0{,}984$	$R^2 = 0{,}983$ $(R^2)_{v_{12}} = 0{,}966$ $(R^2)_{v_{21}} = 0{,}995$ $(R^2)_{S_1} = 0{,}983$ $(R^2)_{S_2} = 0{,}990$

Testing dataset MSE	3.21×10^{-4}	2.98×10^{-4}
Time	7m, 43s	7m, 21s

1



2 **Supplementary Figure 7: a** MSE graph and **b** MAE graph for the four-tile model.