Journal of Applied Physics ARTICLE scitation.org/journal/jap

Thermal transparency with periodic particle distribution: A machine learning approach

Cite as: J. Appl. Phys. **129**, 065101 (2021); doi: 10.1063/5.0039002
Submitted: 29 November 2020 · Accepted: 22 January 2021 ·
Published Online: 8 February 2021

Bin Liu,[1,a)] Liujun Xu,[2] and Jiping Huang[2]

AFFILIATIONS

[1]Department of Electronic and Material Engineering, LeShan Normal University, 614099 LeShan, China
[2]Department of Physics, State Key Laboratory of Surface Physics and Key Laboratory of Micro and Nano Photonic Structures (MOE), Fudan University, 200438 Shanghai, China

[a)]**Author to whom correspondence should be addressed:** liubin@fudan.edu.cn

ABSTRACT

The periodic interparticle interaction mechanism has been previously proposed for heat management, especially the practical application of thermal transparency. In our mechanism for engineering and manipulating thermal metamaterials, particles are arranged in periodic lattices with symmetric interactions. In this work, we relax the constraints in the previous work and allow rectangle lattice and arbitrary relative positioning between the two types of particles. We use a machine learning-based approach to solve the inverse design problem by training autoencoders to compress the dimensionalities of both the design space and the response space and training a neural network tailored for the inverse design problem. We employ the finite-element method for generating the training set for the neural network and for validating the calculated design parameters for a given thermal transparency problem. We also discuss the possibility of extending the machine learning-based workflow to other problems, such as thermal camouflage.

I. INTRODUCTION

Thermal metamaterial has been an emerging field with a number of intriguing applications,[1–3] such as thermal transparency,[4–7] thermal cloaks,[1,2,8–15] thermal concentrators,[8,16,17] and thermal camouflage[18–25] since 2008. Usually, these functionalities of thermal metamaterial are realized in the framework of asymmetric interactions between the background and the device.[4–7] The background is typically defined as the region with uniform physical characteristics or microstructures, excluding the region occupied by the device. Thermal transparency, by its definition, is achieving the effective thermal conductivity of the composite material or structure equal to the thermal conductivity of the surrounding background. The effectiveness of thermal transparency can be measured experimentally by using an infrared camera to take images of temperature distribution, as we did in our previous work.[26] Thermal transparency was previously realized by utilizing asymmetric interactions between the shell and the core. Another similar functionality, thermal invisibility, which also features the same thermal conductivities between the background and the device, employs asymmetric interactions between the matrix and inside particles. The region outside the shell and the region outside the matrix are considered the background in thermal transparency and thermal invisibility, respectively.

Although technically feasible, asymmetric interactions may hinder the application of thermal metamaterials, such as thermal invisibility, due to the noncommutability of matrices and inside particles required by asymmetric interactions.[26]

In our previous work,[26] we proposed a mechanism that employs symmetric interactions between periodic particles to realize thermal transparency. A periodic interparticle system (PIS) consists of an equal number of circular and elliptical regions with anisotropic and uniform thermal conductivity, respectively. Our mechanism avoids the aforementioned issue caused by asymmetric interactions. A brief summary of the mechanism is presented [Figs. 4(a)–4(d)]. Two types of particles (i.e., particle A and particle B) are embedded in a periodic manner in the background, which could be a pure material or uniform microstructure. The thermal conductivities and relative periodic positioning of particle A and particle B are tailored such that the influence of periodic particles on the thermal transfer of the background is removed. Thus, we named the mechanism "periodic interparticle interaction." Furthermore, we found the mathematical equivalence between the formula describing

periodic interparticle interaction and the Bruggeman formula. Our mechanism is an improvement or, at least, an alternative to previous mechanisms requiring asymmetric interactions.

However, in the mechanism proposed in our previous work, two constraints are placed to make the mathematical derivation feasible. The first constraint is that particles are arranged in the simplest lattice that exists: a square lattice. The second constraint is that particles are alternately positioned along both the horizontal and vertical directions. One can readily imagine that by relaxing these two constraints, anisotropy will be introduced, which would make the mathematical derivation considerably more complicated, if a closed form derivation exists.

In this work, we propose a completely different approach, a machine learning-based approach, to solve the problem of employing periodic interparticle interactions to realize thermal transparency while relaxing the lattice type from square to rectangular and allowing arbitrary relative positioning of particle A and particle B, as long as there is no overlapping between the particles. The essential difference between a square lattice and a rectangular lattice is that a rectangular lattice introduces geometric anisotropy, which leads to anisotropic thermal conductivity. Varying the relative positioning of particle A and particle B brings us the ability to manipulate the local thermal field. Therefore, generalizing the lattice type and allowing the relative positioning of particle A and particle B to vary presents at least two more degrees of freedom to manipulate temperature distribution or heat transfer behavior. Our work demonstrates that a machine learning approach can be highly effective in finding the thermal conductivities and relative positioning of particle A and particle B to realize the desired thermal transparency while allowing anisotropy, which could pose an intractable problem, by using a purely mathematical approach.

II. MACHINE LEARNING AS AN APPROACH TO THE INVERSE DESIGN PROBLEM

The problem that we solve in this work can be formulated as an inverse design problem. The parameters describing the thermal conductivity and positioning of particle A and particle B to form the lattice are considered design parameters. Then, the inverse design problem presented in this work can be described as finding a set or multiple sets of design parameters given the desired thermal transfer behavior so that the corresponding system produces the desired thermal transfer behavior. To make the investigation practical, one usually limits the measurement and calculation of thermal transfer behavior on a baseline. This is what we did in our previous work (see Fig. 3 in Ref. 26 and Figs. 7 and 8 in this paper), where the thermal fluxes were calculated on a baseline located at $x = -6$ cm relative to the origin, which is located at the center of the simulation box. In this work, the baseline is further discretized into 400 equally distanced points, on which thermal fluxes are evaluated. Therefore, the response is simplified into a 400-dimensional response vector.

Traditionally, inverse design problems are tackled by using either analytical (or semi-analytical) modeling,[27–33] such as the mechanism proposed in our previous work,[26] or a brute-force sweep over the design parameter space.[34] These two kinds of approaches can only address simple systems where analytical modeling is feasible, or a brute-force sweep is computationally tractable. To tackle more complex problems, evolutionary approaches[35,36] have been proposed to reduce the computational cost compared to the brute-force sweep. However, evolutionary approaches still suffer from drawbacks; for example, finding the global optimum of a problem is not guaranteed, and only one inverse design problem can be addressed at a time (i.e., a small variation in the design objective requires cold-starting the evolution process). Moreover, for large-scale inverse design problems, evolutionary approaches are still too computationally expensive.

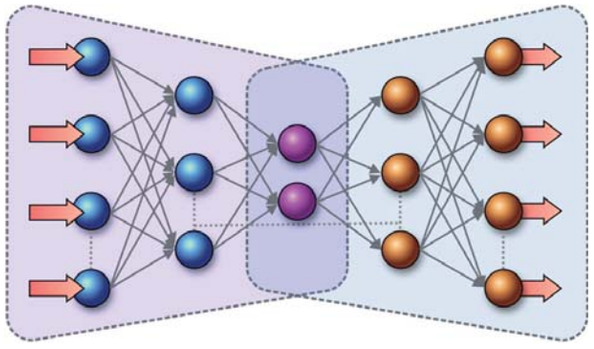

FIG. 1. A schematic diagram of a feed-forward NN. The leftmost layer serves as the input layer, followed by a series of hidden layers with various widths and an output layer on the right side.

In recent years, machine learning-based approaches have been proposed to tackle inverse design problems with remarkable success.[37–53] Most of these works employ a neural network (NN) (Fig. 1) as the backbone to solve the problem. To comply with the standard nomenclature used in both academia and industry, we avoid the use of "deep learning" to describe the aforementioned machine learning-based approaches and the approach in this work, as deep learning typically involves NN with dozens of hidden layers. The NNs in the aforementioned machine learning-based approaches and in this work are not deep enough to qualify as deep NNs.

A typical workflow employing an NN to tackle inverse design problems involves first generating a training dataset (by numerical simulations, or experiments) that relates the design space (parametrized as design parameters) to the response space (usually represented as response vectors). Then, a feed-forward NN solving the forward problem is trained by using the design parameters as the inputs and the response vectors as the outputs. If the training of such a feed-forward NN converges, one can then perform a brute-force sweep over the design space to find optimal design parameter set(s) that can generate the desired response. A brute-force sweep over the design space using an NN is several orders less expensive in terms of the computational cost than a brute-force sweep over the design space by direct numerical simulations. However, a naïve and straightforward implementation of this approach can be problematic. For example, to accurately depict the response of a simple thermal metamaterial device or design, hundreds or even more sampling points need to be taken to represent the response space to capture the finest features in scale. This means that the width (i.e., the number of neurons) of the output layer of the NN has to be in the same order. Even with a shallow NN and a limited number of

design parameters, the large number of weights in the NN that need to be trained mandates a large training set, without which the NN would be prone to overfitting. However, generating a large training set for the NN would defeat the purpose of using machine learning to tackle the inverse design problem. Conversely, a feed-forward NN that solves the inverse problem can be trained by using the response vector as the inputs and the design parameters as the outputs. Once the training of such an NN converges, one can readily obtain a design parameter set by feeding the desired response vector as the inputs to the NN.

However, training an NN to solve such an inverse problem can be intrinsically infeasible, manifested in a training process that is difficult or impossible to converge. The root cause of this issue is that mapping the design space to the response space could be intrinsically many to one. As we will demonstrate in this paper, multiple design parameter sets can correspond to the same (or very similar) response vector. This many-to-one mapping from the design space to the response space translates into a one-to-many relationship between the inputs and the outputs of the NN for an inverse problem. Such an NN is generally not trainable, as the one-to-many relationship causes conflicting optimization objectives during the NN training process (i.e., the backpropagation algorithm is confused by whether it should adjust the weights of the NN to minimize the loss relative to one ground truth design parameter set or another). Several techniques have been proposed to alleviate the issue. Kabir *et al.* attempted to circumvent the problem by removing some training datasets that would otherwise cause conflicting optimization objectives during the NN training process.[45] This can be shown as essentially smoothing out the training dataset to make the mapping from the response space to the design space into an injective function.[54] Other types of NNs, such as tandem networks,[37] or generative adversarial networks (GANs)[39] were also proposed to address the many-to-one issue with some success.

Recently, Kiarashinejad *et al.* proposed a new approach[54] for the inverse problem of designing electromagnetic nanostructures, with remarkable success in addressing both the network-size issue and the many-to-one mapping issue. The essential innovation in their methodology is applying autoencoder-based dimensionality reduction to both the design space and the response space. With the design space converted into the reduced design space, and the response space converted into the reduced response space by applying dimensionality reduction, the many-to-one design space to response space mapping can be converted to a one-to-one reduced design space to reduced response space mapping, which essentially solves the many-to-one mapping issue and greatly facilitate the training of NN for the inverse problem. We draw a close analogy between the problems they tackled and the problem we face (i.e., the inverse design problem of realizing thermal transparency with periodic interparticle interaction) and find that their approach for the inverse problem of designing electromagnetic nanostructures can be readily transferable to solve our problem.

The relationships between the original and reduced design space and the original and reduced response space are illustrated in Fig. 2. The mappings from the original design space to the reduced design space, as well as to the response space, are many-to-one. Each mapping between the reduced design space, the reduced response space, and the original response space is one-to-one. Therefore, the issue of many-to-one mappings between the original design space and the original response space is tackled by removing the redundant dimensions of the original design space, which can be represented as the nonlinear combinations of the incompressible dimensions of the reduced design space.

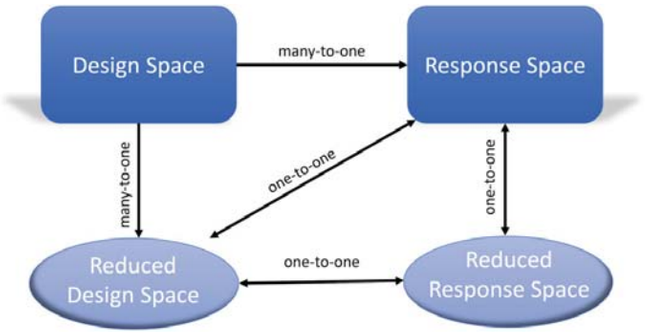

FIG. 2. The relationships between the original and reduced design space and the original and reduced response space.

III. NN ARCHITECTURES FOR THE INVERSE DESIGN PROBLEM

An autoencoder[55] is a widely used type of NN, which is composed of an encoder and a decoder [Fig. 3(a)]. The encoder compresses the inputs into an efficient latent representation which is decompressed by the decoder into the outputs. For an autoencoder, the input and output layers always have an equal number of neurons, while the middle, hidden layer (the bottleneck layer) usually has fewer neurons. The compression of the inputs and subsequent decompression produces reconstructions of the inputs, as the autoencoder tries to reproduce the inputs while learning an efficient representation of the inputs. The cost function for training an autoencoder measures the difference between the inputs and reconstructions. An autoencoder comprised of multiple symmetrical hidden layers is called a stacked autoencoder. Intuitively, the larger capacity of a stacked autoencoder gives it the power to learn more complex structure of the data. However, an overly large (deep, wide, or both) stacked autoencoder can disastrously overfit the dataset.[56] In practice, one usually limits the depth of a stacked autoencoder to three hidden layers, which includes the bottleneck layer.[56]

An autoencoder can efficiently compress high-dimensional data to low-dimensional representations. This makes an autoencoder extremely useful for reducing the dimensionality of both the design parameter sets [Fig. 3(b)] and the response vectors [Fig. 3(a)]. Moreover, once the dimensionality reduction of both the design parameter space and the response space is achieved, the issue of many-to-one design space to response space mapping is essentially solved. Mapping from the reduced design space to the reduced response space is generally one-to-one, making training an NN to solve the inverse design problem feasible.

We use the standard architecture for a stacked autoencoder to compress the response vector from 400 to 10 dimensions. As we will demonstrate, the architecture of the stacked autoencoder to compress

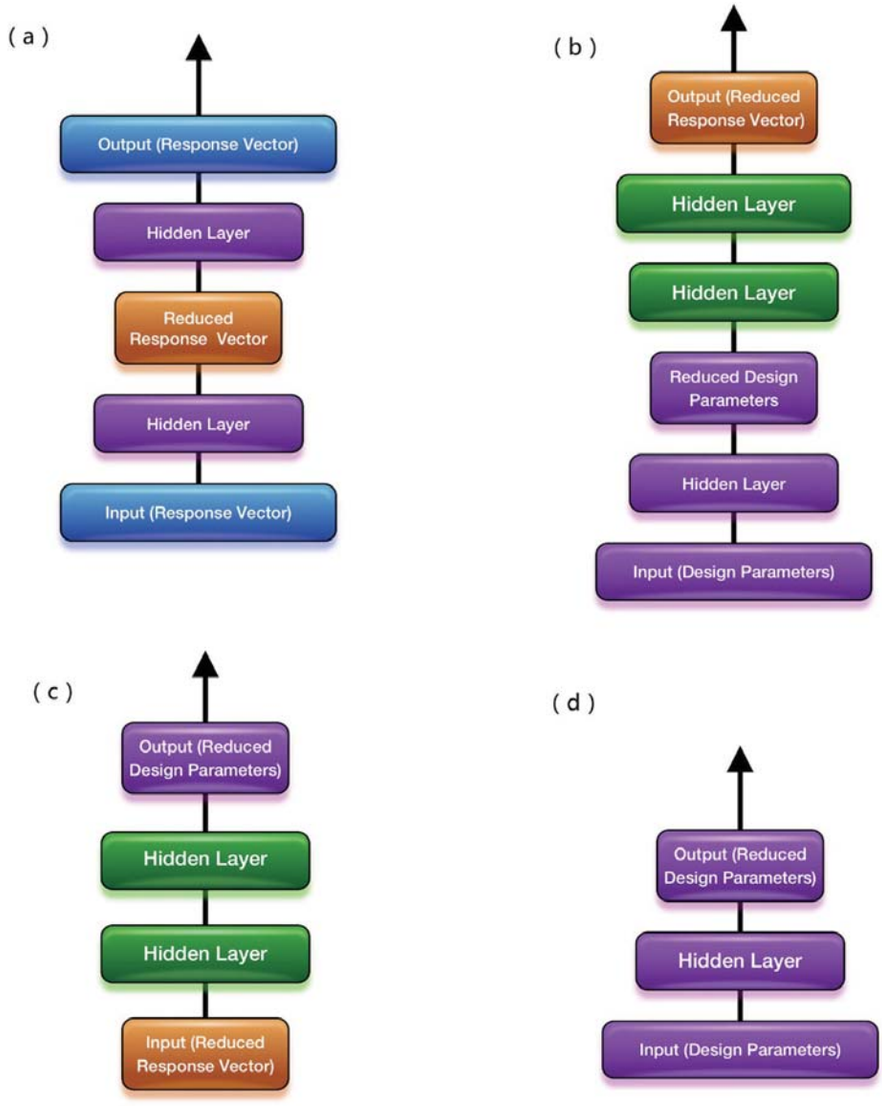

FIG. 3. Schematic diagrams of the NN architectures. (a) An autoencoder to map the response vector to the reduced response vector by performing dimensionality reduction. (b) An NN to map the design parameters to the reduced design parameters by performing dimensionality reduction. (c) An NN to solve the inverse design problem, with the reduced response vector serving as the inputs, and the reduced design parameters serving as the outputs. (d) The encoder part of (c), which is used to perform a grid search of the design parameters to find the design parameters that can map to the desired reduced design parameters.

the response space vector is found by employing a grid search over the hyperparameter space of the width of each hidden layer.

As for the dimensionality reduction for the design space, a stacked autoencoder with a different architecture is used [Fig. 3(b)]. Instead of using the design parameters as the output of the autoencoder, the reduced response vectors obtained from the bottleneck layer of the autoencoder performing the dimensionality reduction for the response space are used. Again, the architecture of the stacked autoencoder for the design space dimensionality reduction is searched by performing a grid search over the varying width of each hidden layer.

With the datasets in the reduced design space and the reduced response space in hand, we are ready to train an NN to solve the inverse design problem. As there is no many-to-one issue in training the NN for the inverse design problem, the standard procedure for training a feed-forward NN (a multilayer perceptron) is applied [Fig. 3(c)]. Now, given a response vector, the corresponding one-to-one reduced design parameter set can be obtained by first using the encoder in Fig. 3(a) to generate the reduced response vector, then feeding it as the inputs to the feed-forward NN in Fig. 3(c) to output the reduced design parameter set.

The final step involves restoring the reduced design parameter set to the original design parameter set. To accomplish this task, the encoder in Fig. 3(b) is extracted from the original stacked autoencoder to form a standalone NN [see Fig. 3(d)]. The inputs for the standalone NN are the original design parameter set, and the outputs are the reduced design parameter set. Given a reduced design parameter set, we perform a grid search over the original design space to find the design parameter set(s) that maps to the given reduced design parameter set through the NN. The grid search over the

original design space can be performed in an embarrassingly parallel fashion by employing a modern multicore CPU or a GPU, as each point in the grid search is independent.

In summary, the procedure of solving the inverse design problem of finding the optimal design parameter sets given a response vector can be divided into three steps. First, dimensionality reduction for the original design space and the original response space is performed by training stacked autoencoders. The bottleneck layers of the trained autoencoders represent the reduced design parameter sets and the reduced response vectors, respectively. Then, a feedforward NN is trained by using the reduced response vectors as the inputs and the reduced design parameter sets as the outputs.

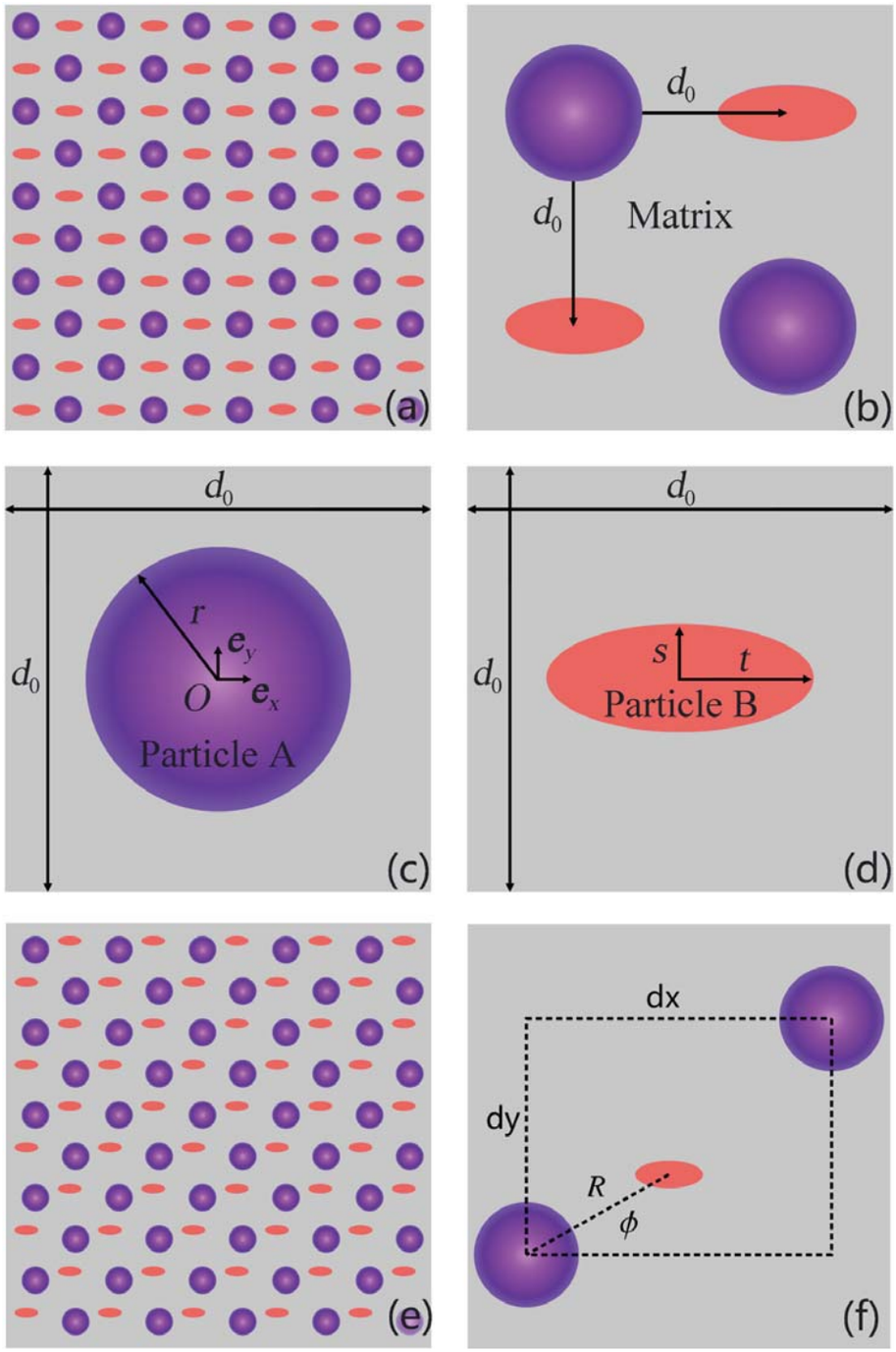

FIG. 4. (a)–(d) Periodic composite material and the basic structures in Ref. 26, where a square lattice is employed. (e) An example of the periodic structures employed in this work, in which particle A and particle B are in their respective rectangular lattices. (f) The relative positioning between a pairing particle A and particle B is parametrized by the distance R between their centers and the angular position ϕ relative to the x axis. (a)–(d) are adapted from Ref. 26.

TABLE I. Ranges of the design parameters, in which the design parameter sets for the training dataset are randomly generated. The range for R is selected to avoid overlapping between particle A and particle B.

Design parameter	Range	Design parameter	Range
dy	(0.8, 1.2) cm	e	(0.05, 0.2)
R	(0.67, 1.36) cm	$\kappa_{\rho\rho}$	(0, 1) $\mathrm{Wm^{-1}\,K^{-1}}$
ϕ	$(-\frac{\pi}{4}, \frac{\pi}{4})$	$\kappa_{\theta\theta}$	(0, 1) $\mathrm{Wm^{-1}\,K^{-1}}$
p_a	(0.05, 0.4)	κ_b	(0, 1) $\mathrm{Wm^{-1}\,K^{-1}}$
p_b	(0.05, 0.4)	κ_m	(0, 1) $\mathrm{Wm^{-1}K^{-1}}$

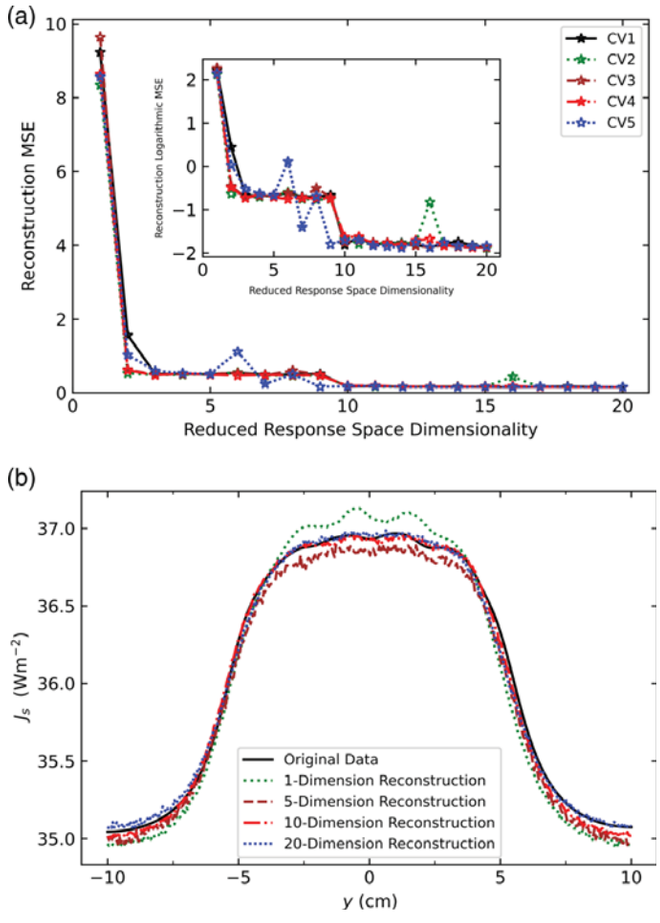

FIG. 5. The effect of the dimensionality of the reduced response space on the reconstruction MSE. (a) The reconstruction MSE for the response vector by the NN architecture in Fig. 3(a), as a function of the dimensionality of the reduced response space. The logarithmic reconstruction MSE is shown in the inset. The fivefold cross validation results are represented in different colors. (b) A sample system in the test set is selected to demonstrate the reconstructed response vectors for different dimensionalities of the reduced response space.

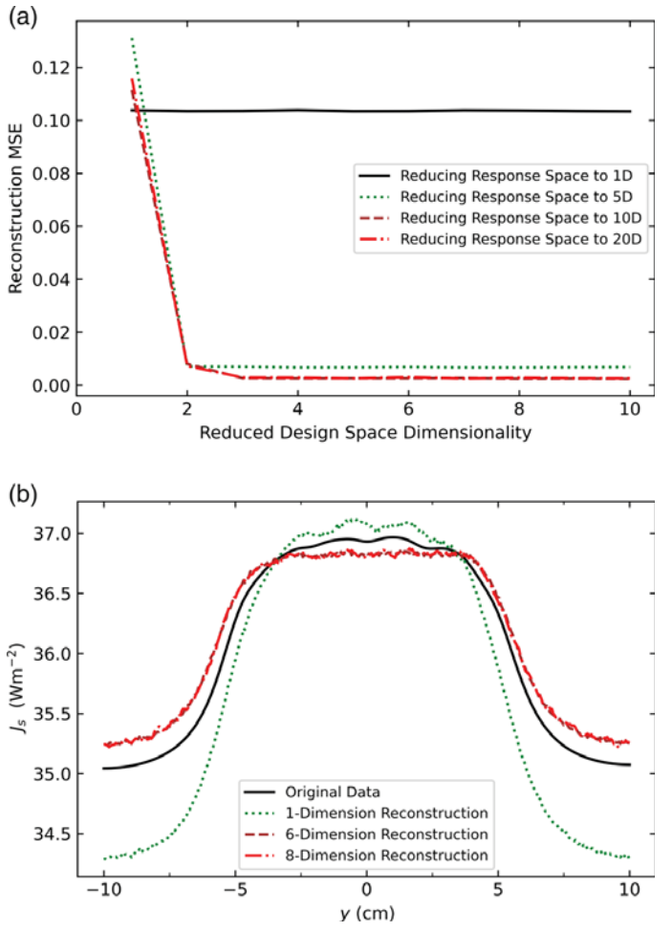

FIG. 6. The effect of the dimensionality of the reduced design space on the reconstruction MSE. (a) The reconstruction MSE for the response vector by the NN architecture in Fig. 3 as a function of the dimensionalities of both the reduced design space and the reduced response space. The architecture of the NN is formed by cascading the NN in Fig. 3(b) to the decoder part of Fig. 3(a). The variance of the reconstruction MSE between different NN models with the same combination of reduced design space and response space dimensionalities generated by the CV process is low (detailed data not included). And we only demonstrate the best NN models (i.e., with the least reconstruction MSE from the ground truth) selected by the CV process. (b) A sample system in the test set is selected to demonstrate the reconstructed response vectors for different dimensionalities of the reduced design space.

Finally, the reduced design parameter sets are restored to the original design parameter sets by performing a grid search over the inputs for the encoder part of the stacked autoencoder to reduce the dimensionality of the original design space.

IV. METHODS

Compared to our previous work[26] in which a square lattice for periodic particles is used, we relax the requirement into a rectangular lattice in this work [Fig. 4(e)]. Although it may seem, at first

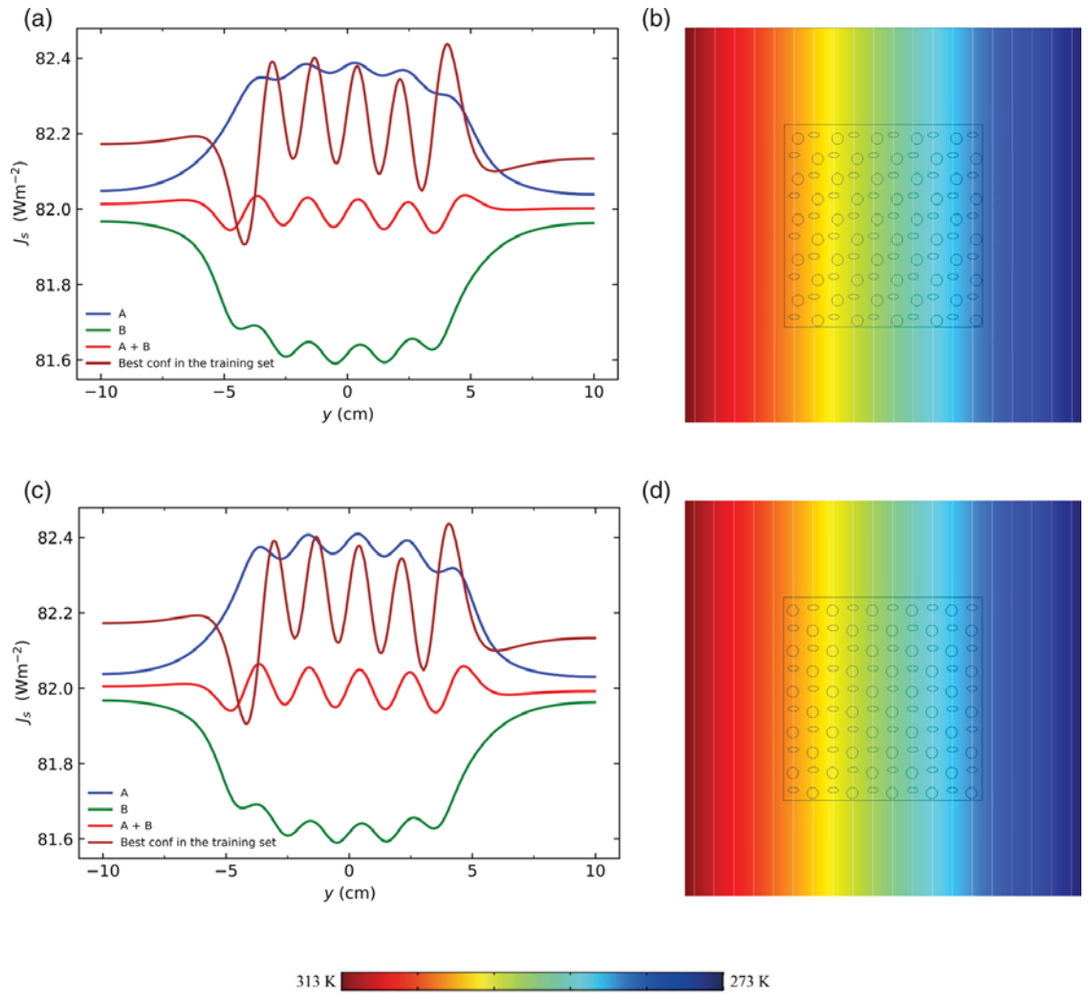

FIG. 7. Finite-element simulations of thermal transparency with heat fluxes $J_S \approx 82\,\mathrm{Wm}^{-2}$. The size of the simulation box is $20 \times 20\,\mathrm{cm}^2$, and that of the periodic composite material is $10 \times 10\,\mathrm{cm}^2$. The baseline for measuring heat fluxes is located at $x = -6\,\mathrm{cm}$ (not shown in the figures), and the origin is located at the center of the simulation box. White lines represent isotherms. [(a) and (b)] The design parameters are shown as system A in Table II. [(c) and (d)] The design parameters are shown as system B in Table II. [(b) and (d)] Particle A is represented as circular regions, while particle B is represented as elliptic regions. Blue, green, and red lines in (a) and (c) are simulated heat fluxes measured at the baseline with only particle A, only particle B, and particle A plus particle B. Brown lines in (a) and (c) are the heat fluxes measured at the baseline, generated by the system with the least MSE distance to the perfect thermal transparency with heat fluxes $J_S \approx 82\,\mathrm{Wm}^{-2}$, out of all the systems in the randomly generated training set.

glance, that two design parameters, dx and dy, are needed to describe the periodic particle lattice, only the ratio between the two design parameters matters. We fix the design parameter dx to 1 cm. More precisely, the ellipses (particle B) are fixed on a horizontally and vertically alternating rectangular lattice. The spacing between two horizontally adjacent ellipses is 2^*dx, and the spacing between two vertically adjacent ellipses is 2^*dy. The circles (particle A) are also arranged in a rectangular lattice in the same style. However,

TABLE II. The design parameters of two systems (A and B) to achieve thermal transparency at the baseline located at $x = -6$ cm with heat fluxes $J_S \approx 82\,\mathrm{Wm}^{-2}$. System A corresponds to Figs. 7(a) and 7(b), and system B corresponds to Figs. 7(c) and 7(d). dy and R are in (cm), $\kappa_{\rho\rho}$, $\kappa_{\theta\theta}$, κ_b, and κ_m are in ($\mathrm{Wm}^{-1}\,\mathrm{K}^{-1}$), and the other design parameters are unitless. The MSE between the two sets of design parameters is 1.05×10^{-02}.

Design parameter	dy	R	ϕ	p_a	p_b	e	$\kappa_{\rho\rho}$	$\kappa_{\theta\theta}$	κ_b	κ_m
A	1.02	1.20	−0.155	0.122	0.0534	1.99	0.446	0.446	0.337	0.410
B	1.02	0.979	−0.155	0.122	0.0534	1.99	0.446	0.446	0.337	0.410

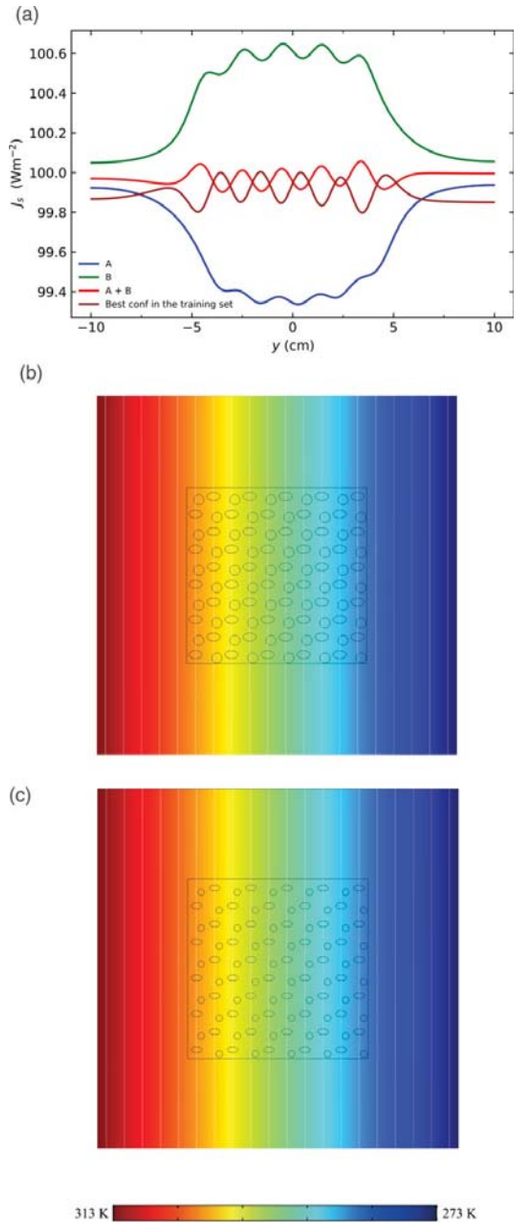

FIG. 8. Finite-element simulations of thermal transparency with heat fluxes $J_S \approx 100\,\mathrm{Wm}^{-2}$. The size of the simulation box is $20 \times 20\,\mathrm{cm}^2$, and that of the periodic composite material is $10 \times 10\,\mathrm{cm}^2$. The baseline for measuring heat fluxes is located at $x = -6$ cm (not shown in the figures), and the origin is located at the center of the simulation box. White lines represent isotherms. The design parameters for the red line in (a) and (b) are shown as system C in Table IV. The design parameters for the brown line in (a) and (c) are shown as system D in Table IV. [(b) and (c)] Particle A is represented as circular regions, while particle B is represented as elliptic regions. Blue, green, and red lines in (a) are simulated heat fluxes measured at the baseline with only particle A, only particle B, and particle A plus particle B. Brown lines in (a) are the heat fluxes measured at the baseline, generated by the system with the least MSE distance to the perfect thermal transparency with heat fluxes $J_S \approx 100\,\mathrm{Wm}^{-2}$, out of the systems in the randomly generated training set.

the relative positioning between adjacent circles and ellipses is allowed to vary. We denote the relative positioning between pairing circles and ellipses with a cylindrical coordinate R and ϕ [see Fig. 4(f)], with the origin located at the center of the ellipse. R denotes the distance between the centers of pairing circle and ellipse, while ϕ denotes the angle between the line connecting the centers of pairing circle and ellipse and the horizontal axis. The circle regions (particle A) have anisotropic thermal conductivities [a purple graded circle with a radius r, Fig. 4(c)], whereas the ellipse regions (particle B) are defined by isotropic thermal conductivities [a red ellipse with a semi minor axis s and semi major axis t, and $e = s/t$, Fig. 4(d)]. We denote the thermal conductivities of the circle regions (particle A), ellipse regions (particle B), and the matrix to be $\overleftrightarrow{\kappa}_a = \mathrm{diag}(\kappa_{\rho\rho}, \kappa_{\theta\theta})$, κ_b, and κ_m, respectively. $\overleftrightarrow{\kappa}_a$ is represented in the cylindrical coordinates (ρ, θ), whose origin is at the center of particle A. $p_a[=\pi r^2/(2d_x{}^*d_y)]$, $p_b[=\pi st/(2d_x{}^*d_y)]$, and $p_m = 1 - p_a - p_b$ are the area fraction of particle A, particle B, and matrix, respectively.

We generate the dataset to train the NNs by performing finite-element method (FEM) simulations using the commercial software package COMSOL Multiphysics.[57] A total of 50 000 FEM simulations were performed with randomly generated design parameters within predefined ranges, which are listed in Table I. Note that a randomly generated design parameter set is not necessarily valid, as overlapping between particle A and particle B may occur. A customized Python code is written to filter out invalid design parameter set. We perform a fivefold cross validation (CV) to find the best NN models as well as checking the robustness of NN training against different ways of splitting the training and test sets. In each fold, the design parameter sets and the response vectors generated from discretizing thermal flux in 40 000 simulations are used as the training set, with the remaining 10 000 simulations used as the test set.

We use TensorFlow[58] as the framework to train the NNs and use pandas[59] for data analysis. The trained NN models are also used in the grid search to restore the reduced design parameter sets to the original design parameter sets. The FEM simulations and NN training were primarily performed on a dual-socket 24 physical core Xeon Haswell-EP server.

V. RESULTS

First, we use the training set to perform a grid search for two hyperparameters for the autoencoder [Fig. 3(a)] by performing response vector dimensionality reduction [i.e., the number of hidden layers and neurons in the bottleneck layer (the bottleneck layer width)]. The range in the grid search for the number of hidden layers is 3–7, while the bottleneck layer width is in the range of 1–20. The reconstruction mean squared error (MSE) between the inputs and the outputs of a specific autoencoder architecture is used to find the optimal autoencoder architecture. The inputs for an autoencoder during training are 400 dimensional vectors from discretizing heat fluxes measured at the baseline $x = -6$ cm for the 40 000 simulated systems in the training set. We find that autoencoders with three hidden layers generally outperform the other architectures, which is consistent with the

TABLE III. The reduced design parameters of two systems (A and B) to achieve thermal transparency at the baseline located at $x = -6$ cm with heat fluxes $J_S \approx 82\ \mathrm{Wm}^{-2}$. System A corresponds to Figs. 7(a) and 7(b), and system B corresponds to Figs. 7(c) and 7(d). The reduced design parameters are inferenced by using the encoder part of the NN shown in Fig. 3(b). The MSE between the two sets of reduced design parameters is 4.57×10^{-07}.

Reduced design parameter	1	2	3	4	5	6
A	0.178	0.687	0.920	0.781	0.0599	0.995
B	0.178	0.687	0.919	0.782	0.0608	0.995

TABLE V. The reduced design parameters of two systems (C and D) to achieve thermal transparency at the baseline located at $x = -6$ cm with heat fluxes $J_S \approx 100\ \mathrm{Wm}^{-2}$. System C corresponds to Fig. 8(b), and system D corresponds to Fig. 8(c). The reduced design parameters are inferenced by using the encoder part of the NN shown in Fig. 3(b). The MSE between the two sets of reduced design parameters is 1.00×10^{-05}.

Reduced design parameter	1	2	3	4	5	6
C	0.178	0.687	0.920	0.781	0.0599	0.995
D	0.178	0.687	0.919	0.782	0.0608	0.995

general experience that deep autoencoders tend to overfit the training data.[56]

Figure 5(a) shows the reconstruction MSE between the inputs and outputs for a series of trained autoencoders with three hidden layers and varying bottleneck layer widths (1–20), evaluated by the CV process. It is clear that beyond a width of 10 for the bottleneck layer, the reconstruction MSE ceases to improve as the width increases. Furthermore, having 10 neurons in the bottleneck layer is sufficient for the purpose of dimensionality reduction for the response space, as evidenced by the low MSE. Figure 5(b) shows the original (ground truth) heat-flux curve and reconstructed curves with varying dimensionalities of the reduced response space for a simulated system picked randomly from the test set. The reconstructed curves with reduced response space dimensions of 10 and 20 can reproduce the shape of the original curve with satisfactory precision. Thus, we choose 10 as the dimension of the reduced response space, leading to the autoencoder architecture 200-50-10-50-200.

Then, we perform a similar grid search to find the optimal architecture for the NN [Fig. 3(b)] to reduce the dimensionality of the design space. The backbone of the architecture for the grid search is 10-20-20-X-20-30-Y-50-400, where X stands for the dimension for the reduced design space with a range of 1–10 and Y stands for the dimension for the reduced response space with 1, 5, 10, and 20. Figure 6(a) shows the reconstruction MSE as a function of the dimension of the reduced design space and the dimension of the reduced response space. Figure 6(b) shows the original (ground truth) heat-flux curve and reconstructed curves with varying dimensionalities of the reduced design space for the same simulated system depicted in Fig. 5(b). It is clear that the dimensionality of the design space can be reduced to 6 without significantly sacrificing the precision of reconstructing the response vector, and the dimensionality for the response space can be reduced to 10, as the reconstruction MSE for the response space dimensionalities of 10 and 20 almost overlap when the dimensionality in the design space goes beyond 4.

Now, we proceed to the final phase of NN training to solve the inverse design problem for thermal transparency. To accomplish this, we assemble an NN to use the reduced response vectors as the inputs and the reduced design parameter sets as the outputs [see Fig. 3(c)]. The reduced response vectors are derived from compressing the response vectors in the simulated systems in the training set by using the encoder part of the NN in Fig. 3(a) (400-50-10-50-400). The reduced design parameter sets are derived from compressing the original design parameters for the same simulated systems by using the encoder part of the NN in Fig. 3(b) (10-20-20-6). The training process quickly converges, indicating the absence of the issue of one-to-many mappings between the reduced response vector and reduced design parameters. Once the NN is trained to solve the inverse problem, we can feed a desired response vector as the inputs to the NN and get the corresponding reduced design parameters. The final part of the workflow involves restoring the reduced design parameters to the original design parameters. For this task, the encoder part of Fig. 3(b) for the dimensionality reduction of the design parameter space is extracted to form a separate NN [Fig. 3(d)]. Then, a grid search over the viable design parameter ranges (10 intervals for each parameter) is performed to find the corresponding original design parameter(s) to the reduced design parameter. The configurations of design parameters are ranked by using the MSE between their corresponding reduced design parameters and the desired reduced design parameters. The top 100 configurations of design parameters are simulated by FEMs, among which the best configurations for the purpose of thermal transparency are selected.

To demonstrate the power of the entire workflow, we solve two problems of realizing thermal transparency for heat fluxes $J_S \approx 82\ \mathrm{Wm}^{-2}$ and $J_S \approx 100\ \mathrm{Wm}^{-2}$ measured at the baseline $x = -6$ cm.

As for the problem of realizing thermal transparency for heat fluxes $J_S \approx 82\ \mathrm{Wm}^{-2}$, we find two configurations of design

TABLE IV. The design parameters of two systems (C and D) to achieve thermal transparency at the baseline located at $x = -6$ cm with heat fluxes $J_S \approx 100\ \mathrm{Wm}^{-2}$. System C corresponds to Fig. 8(b), and system D corresponds to Fig. 8(c). dy and R are in (cm), $\kappa_{\rho\rho}$, $\kappa_{\theta\theta}$, κ_b and κ_m are in ($\mathrm{Wm}^{-1}\,\mathrm{K}^{-1}$), and the other design parameters are unitless. The MSE between the two sets of design parameters is 3.59×10^{-02}.

Design parameter	dy	R	ϕ	p_a	p_b	e	$\kappa_{\rho\rho}$	$\kappa_{\theta\theta}$	κ_b	κ_m
C	0.978	1.12	−0.155	0.125	0.128	1.66	0.446	0.446	0.554	0.500
D	0.998	1.27	−0.189	0.0526	0.0759	1.74	0.785	0.746	0.385	0.499

parameters, as shown in Fig. 7 and Table II. To realize thermal transparency, the two systems with the design parameters we find are superior to the best one in the training set with randomly generated design parameters. We also calculate the MSE between the two sets of design parameters we found and the MSE between their corresponding reduced design parameters (Table III). The MSE for the reduced design parameters is four orders lower than the MSE of the original design parameters, indicating the efficacy of dimensionality reduction.

As for the problem of realizing thermal transparency for heat fluxes $J_S \approx 100\,\mathrm{Wm}^{-2}$, we find a system in the training set whose response vector is in close proximity to the perfect response vector. We also calculate a set of design parameters from our workflow of solving the inverse design problem. The two systems and their responses are shown in Fig. 8. The values for the design parameters and the reduced design parameters for the two systems are shown in Table IV. The system with the calculated design parameter performs slightly better than the coincidentally and randomly generated system. The most convincing evidence of the correctness and efficacy of our workflow for solving the inverse design problem is that two systems with significantly different design parameters have very similar reduced design parameters, as evidenced by the MSE between the reduced design parameters, which is more than three orders lower than the MSE between the design parameters (Table V). This is an overwhelmingly convincing demonstration of the capability of our trained NNs to map different configurations of design parameters to the same (or very similar) configuration of reduced design parameters for the purpose of realizing thermal transparency. Therefore, the most challenging issue (i.e., the many-to-one mapping from the design space to the response space) has been tackled by our NN approach.

VI. CONCLUSION

In this paper, we design a systematic workflow of employing the NN-based machine learning approach to realize thermal transparency by using thermal metamaterial-based periodic interparticle interactions. The problem itself is a generalization of the problem we have solved in our previous work, by relaxing the constraints and allowing a rectangle lattice and arbitrary relative position between the two types of particles. The key success of our approach originates from training autoencoders to reduce the dimensionalities of both the design space and the response space, solving the issue of many-to-one mappings between the design space and the response space which makes many previous NN-based approaches inutile, and the issue of training computational cost. Our approach successfully found design parameter set(s) for realizing thermal transparency with designated heat fluxes.

ACKNOWLEDGMENTS

We acknowledge financial support by the National Natural Science Foundation of China (NNSFC) under Grant Nos. 11725521 and 12035004, by the Science and Technology Commission of Shanghai Municipality under Grant No. 20JC1414700, by the Education Department of Sichuan Province under Grant No. 18ZB0273, and by Leshan Science and Technology Bureau under Grant No. 15NZD100. We thank G. Dai for his contribution of Python codes and J. Huang for his assistance in figure plotting.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]C. Z. Fan, Y. Gao, and J. P. Huang, Appl. Phys. Lett. **92**, 251907 (2008).
[2]T. Y. Chen, C. N. Weng, and J. S. Chen, Appl. Phys. Lett. **93**, 114103 (2008).
[3]M. Maldovan, Nature **503**, 209 (2013).
[4]X. He and L. Z. Wu, Phys. Rev. E **88**, 033201 (2013).
[5]L. W. Zeng and R. X. Song, Appl. Phys. Lett. **104**, 201905 (2014).
[6]T. Z. Yang *et al.*, Adv. Mater. **27**, 7752 (2015).
[7]R. Z. Wang, L. J. Xu, Q. Ji, and J. P. Huang, J. Appl. Phys. **123**, 115117 (2018).
[8]S. Narayana and Y. Sato, Phys. Rev. Lett. **108**, 214303 (2012).
[9]R. Schittny, M. Kadic, S. Guenneau, and M. Wegener, Phys. Rev. Lett. **110**, 195901 (2013).
[10]H. Y. Xu, X. H. Shi, F. Gao, H. D. Sun, and B. L. Zhang, Phys. Rev. Lett. **112**, 054301 (2014).
[11]T. C. Han *et al.*, Phys. Rev. Lett. **112**, 054302 (2014).
[12]Y. G. Ma, Y. C. Liu, M. Raza, Y. D. Wang, and S. L. He, Phys. Rev. Lett. **113**, 205501 (2014).
[13]G. Dai and J. Huang, J. Appl. Phys. **124**, 235103 (2018).
[14]S. Yang, L. Xu, and J. Huang, J. Appl. Phys. **125**, 055103 (2019).
[15]S. Yang, L. Xu, G. Dai, and J. Huang, J. Appl. Phys. **128**, 095102 (2020).
[16]R. S. Kapadia and P. R. Bandaru, Appl. Phys. Lett. **105**, 233903 (2014).
[17]L. J. Xu, S. Yang, and J. P. Huang, Phys. Rev. E **98**, 052128 (2018).
[18]T. C. Han, X. Bai, J. T. L. Thong, B. W. Li, and C. W. Qiu, Adv. Mater. **26**, 1731 (2014).
[19]X. He and L. Z. Wu, Appl. Phys. Lett. **105**, 221904 (2014).
[20]T. Z. Yang, Y. Su, W. Xu, and X. D. Yang, Appl. Phys. Lett. **109**, 121905 (2016).
[21]R. Hu *et al.*, Adv. Mater. **3**, 1707237 (2018).
[22]S. L. Zhou, R. Hu, and X. B. Luo, Int. J. Heat Mass Transf. **127**, 607 (2018).
[23]L. J. Xu, R. Z. Wang, and J. P. Huang, J. Appl. Phys. **123**, 245111 (2018).
[24]L. J. Xu and J. P. Huang, Phys. Lett. A **382**, 3313 (2018).
[25]L. Xu, R. Wang, and J. Huang, J. Appl. Phys. **123**, 245111 (2018).
[26]L. Xu, S. Yang, and J. Huang, Phys. Rev. Appl. **11**, 034056 (2019).
[27]S. Molesky *et al.*, Nat. Photonics **12**, 659–670 (2018).
[28]A. Y. Piggott, J. Petykiewicz, L. Su, and J. Vučković, Sci. Rep. **7**, 1786 (2017).
[29]J. Lu and J. Vučković, Opt. Express **21**, 13351 (2013).
[30]L. Su, A. Y. Piggott, N. V. Sapra, J. Petykiewicz, and J. Vučković, ACS Photonics **5**, 301 (2018).
[31]L. F. Frellsen, Y. Ding, O. Sigmund, and L. H. Frandsen, Opt. Express **24**, 16866 (2016).
[32]A. Y. Piggott, Sci. Rep. **4**, 7210 (2014).
[33]D. Englund, I. Fushman, and J. Vuckovic, Opt. Express **13**, 5961 (2005).
[34]S. Y. Seidel and T. S. Rappaport, IEEE Trans. Veh. Technol. **43**, 879 (1994).
[35]A. Gondarenko and M. Lipson, Opt. Express **16**, 17689 (2008).
[36]Y. Ma *et al.*, Opt. Express **21**, 29374 (2013).
[37]D. Liu, Y. Tan, E. Khoram, and Z. Yu, ACS Photonics **5**, 1365 (2018).
[38]J. Peurifoy *et al.*, Sci. Adv. **4**, eaar4206 (2018).
[39]Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, Nano Lett. **18**, 6570 (2018).
[40]M. H. Tahersima *et al.*, Sci. Rep. **9**, 1368 (2019).
[41]T. Zhang *et al.*, Photonics Res. **7**, 368 (2019).
[42]W. Ma, F. Cheng, and Y. Liu, ACS Nano **12**, 6326 (2018), PMID: 29856595.
[43]Y. Qu, L. Jing, Y. Shen, M. Qiu, and M. Soljačić, ACS Photonics **6**, 1168 (2019).
[44]S. Inampudi and H. Mosallaei, Appl. Phys. Lett. **112**, 241102 (2018).

[45]H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang, IEEE Trans. Microw. Theory Tech. **56**, 867 (2008).
[46]G. Fujii, Y. Akimoto, and M. Takahashi, Appl. Phys. Lett. **112**, 061108 (2018).
[47]G. Fujii and Y. Akimoto, Appl. Phys. Lett. **115**, 174101 (2019).
[48]G. Fujii and Y. Akimoto, Int. J. Heat Mass Transf. **137**, 1312 (2019).
[49]G. Fujii and Y. Akimoto, Int. J. Heat Mass Transf. **159**, 120082 (2020).
[50]G. Fujii and Y. Akimoto, Phys. Rev. E **102**, 033308 (2020).
[51]R. Hu *et al.*, Phys. Rev. X **10**, 021050 (2020).
[52]R. Hu *et al.*, Nano Energy **72**, 104687 (2020).
[53]W. Sha, Y. Zhao, L. Gao, M. Xiao, and R. Hu, J. Appl. Phys. **128**, 045106 (2020).
[54]Y. Kiarashinejad, S. Abdollahramezani, and A. Adibi, npj Comput. Mater. **6**, 12 (2020).
[55]G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).
[56]J. Liang and R. Liu, "Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network," in *2015 8th International Congress on Image and Signal Processing (CISP)* (Curran Associates, 2015), pp. 697–701.
[57]See http://www.comsol.com/ for detailed descriptions for the features and usage of COMSOL Multiphysics.
[58]M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," Software available from tensorflow.org (2015).
[59]The Pandas Development Team, pandas-dev/pandas, Pandas, 2020.