NASA/TM-20205003868

Optimal Experimental Design With Fast Neural Network Surrogate Models

Joshua Stuckner
Glenn Research Center, Cleveland, Ohio

Matthew Piekenbrock
Wright State University, Dayton, Ohio

Steven M. Arnold and Trenton M. Ricks
Glenn Research Center, Cleveland, Ohio

February 2021

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM-20205003868

Optimal Experimental Design With Fast Neural Network Surrogate Models

Joshua Stuckner
Glenn Research Center, Cleveland, Ohio

Matthew Piekenbrock
Wright State University, Dayton, Ohio

Steven M. Arnold and Trenton M. Ricks
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

February 2021

Acknowledgments

The authors would like to acknowledge the support of the NASA Transformational Tools and Technologies (TTT) project during the course of this study. The authors also thank Dr. Pappu Murthy for performing a blind least-squares minimization to test the robustness of the proposed optimal design.

This work was sponsored by the
Transformative Aeronautics Concepts Program.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Optimal Experimental Design With Fast Neural Network Surrogate Models

Joshua Stuckner
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Matthew Piekenbrock*
Wright State University
Dayton, Ohio 45435

Steven M. Arnold and Trenton M. Ricks
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Summary

Designing optimal experiments minimizes the uncertainty of results and maximizes the efficient use of resources. Herein, machine learning surrogate models and the approximate coordinate exchange (ACE) algorithm are used to determine optimal experimental designs (OEDs) over large or arbitrarily restrictive design spaces. OED is particularly salient in materials science, where experiments are expensive and material properties must often be inferred indirectly. The proposed framework is demonstrated by finding optimal experiments with which the hidden constituent properties of composite materials can be most efficiently inferred from observable experimental outcomes. The OED is given by an information-theoretic criterion that maximizes the conditional mutual information between the hidden properties and the expected experimental outcomes. To perform tractable optimization, a neural network is trained as a surrogate model to mimic a physics-based simulation, which can calculate the expected experimental outcome based on a candidate experimental design and sampled constituent properties. The ACE algorithm is used to optimize over large design spaces with many tests and controlled parameters where an exhaustive search would be intractable even with the surrogate model. Using this approach, OEDs that are consistent with those produced by heuristic knowledge and established best practices are found; then optimal designs in larger design spaces where heuristic knowledge is unavailable are examined.

Nomenclature

Acronyms

ACE	approximate coordinate exchange
ANN	artificial neural network
CMC	ceramic matrix composite
CMI	conditional mutual information

*Summer Intern in Lewis' Educational and Research Collaborative Internship Project (LERCIP), graduate student at Wright State University.

GMC	generalized method of cells
MAC/GMC	Micromechanics Analysis Code with Generalized Method of Cells (software name)
MMC	metal matrix composite
MSE	mean squared error
OED	optimal experimental design
PMC	polymer matrix composite
ReLU	rectified linear unit
RUC	repeating unit cell

Symbols

$\mathbf{A}$	strain concentration matrix
a	output of neuron in neural network
b	bias term added to neuron output
D_{KL}	Kullback-Leibler divergence
E	elastic modulus
g	activation function
$\mathcal{H}$	experimental design space
I	conditional mutual information
k	number of controllable factors in experimental design
l	layer
ℓ	layup (number and orientation of plies in laminate)
n	number of tests in experimental design
p	parameter setting of kth parameter for nth test
Q	set of all possible hidden property values
q	hidden properties to be inferred
r	number of q values randomly sampled on each iteration
U	expected value of experiment
u	utility function that gives efficiency of experimental design
V	fiber volume fraction
Y	set of all possible experimental outcomes
y	experimental outcome
$\hat{y}$	expected outcome of the candidate experimental design
α	list of one or more angles of individual plies within laminate
ε	strain
η	experimental design containing n tests and k controllable factors
η^*	optimal experimental design
$\hat{\eta}$	candidate experimental design
Θ	weight of connections between neurons in neural network
θ	angle
ν	Poisson's ratio
σ	stress

Superscripts

PLS	proportional limit stress/strain
UTS	ultimate tensile stress/strain

Subscripts

c	number of repeats of α in laminate layup notation
f	fiber
m	matrix
S	indicates symmetric laminate if present in layup notation
x	indicates property is defined along loading direction
xy	indicates transverse direction

Introduction

Experiments that measure material properties are often expensive. To make matters worse, many properties cannot be measured directly and must be inferred from measurable experimental outcomes that are sensitive to hidden properties. Examples include inferring substrate properties from measurements on a deposited film (Ref. 1), inferring elastic properties from resonant response to ultrasonic waves (Ref. 2), and inferring fiber and matrix constituent properties from the bulk elastic properties of the composite material. Many spectroscopic techniques rely on indirectly inferring material properties from an observable response to some stimulus. Performing efficient experiments can save time and money while reducing the uncertainty of results. However, it is often difficult to determine the optimal experimental design (OED), especially when the inference calculation is computationally expensive. To this end, an efficient information-theoretic Bayesian design framework to select the optimal experiment to infer hidden properties from experimental observations is introduced. The framework uses a multilayer artificial neural network (ANN) as a computationally efficient surrogate model to accurately relate hidden properties to predicted experimental outcomes during the experimental design optimization where a complex physics-based model would be intractable. By considering the expected experimental outcome, designs are chosen that maximize the dependence of the observable properties on the hidden properties, which is a major advantage compared with traditional design of experiments. By utilizing the approximate coordinate exchange (ACE) algorithm (Ref. 3), this framework is tractable even when considering designs with a large number of tests and controllable design parameters. This general framework can be used to determine the most efficient experiment to infer hidden material properties, provided a model exists (or can be built) to map the relationship between the hidden and observable properties. The framework can handle arbitrarily complex systems, and the optimization is tractable even when the design space becomes large and the relationship model is computationally expensive.

Background

Optimal Experimental Design

OED can be used to determine a design that maximizes the expected value of an experiment. Lindley (Ref. 4) suggests the following general objective for experimental design:

$$U(\eta) = \mathbb{E}_{q,y|\eta}\left[u(q,y,\eta)\right] = \int_Y \int_Q u(q,y,\eta)\, p(q,y\,|\,\eta)\, \mathrm{d}q\, \mathrm{d}y \tag{1}$$

where $U(\eta)$ is the expected value, denoted by $\mathbb{E}$, of some utility function $u(q,y,\eta)$. Here, a utility function is used that gives the efficiency of an experimental design, η, in inferring hidden properties, q, from a particular set of experimental observations, y. The design, η, is an $n \times k$ matrix,

$$\eta = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nk} \end{bmatrix} \tag{2}$$

where n is the number of tests to perform, k is the number of adjustable design parameters, and each element, p_{nk}, is the parameter setting of the kth parameter for the nth test. Since q and y are not known before performing the experiment, the expected utility of a design, $U(\eta)$, is found by taking the utility at each q and y multiplied by the conditional probability of that particular q and y integrated over all possible outcomes, Y, and hidden properties, Q. That is, U is determined by taking the expectation of u over the joint probability distribution of q and y.

In order to infer q from y, the optimal design should be the experiment where y is most strongly influenced by q. If the hidden material properties did not influence the observable properties for a particular design, there would be no information in the experimental measurements with which to infer the lower length scale properties. To this end, Aggarwal et al. (Ref. 5) advocate using a utility function that reflects the gain in Shannon information in the quantities of interest (Ref. 6). In particular, they suggest using the relative entropy or Kullback-Leibler divergence, D_{KL}, from the posterior to the prior:

$$u(q, y, \eta) = D_{\mathrm{KL}}\left[p(q, y|\eta) \,\|\, p(q) \right] = \int_Q p(q|y,\eta) \log\left[\frac{p(q|y,\eta)}{p(q)} \right] \mathrm{d}q \tag{3}$$

Using Equation (3) as the utility function in Equation (1) expresses U in terms of the expected information gain for an experimental design, η. This is equivalent to the conditional mutual information (CMI) (Ref. 7), I, between the experimental observations and the hidden properties: $U(\eta) = I(q\,;\, y \,|\, \eta)$ (Ref. 5). Thus, the optimal design, η^*, which provides the highest expected value, is given by solving the following optimization problem over the design space, $\mathcal{H}$:

$$\eta^* = \underset{\eta \in \mathcal{H}}{\operatorname{argmax}}\, I\left(q; y \,|\, \eta\right) \tag{4}$$

Since the quantities are nonparametric, the CMI is estimated using a k-nearest-neighbors-based estimator (Ref. 8). Other CMI estimators are available, including those found in References 9 to 11. The optimization is performed using the ACE algorithm to maximize the expected utility over the design space. The ACE algorithm is a two-phase algorithm that pivots between a surrogate function to generate designs and a point exchange procedure to reduce the candidate set of designs. Since q is unknown, their values are sampled from a uniform distribution each time the utility function is called. The random q values and candidate designs, η, must be mapped to a prediction of the experimental outcome, y, by some model. Taking the mean of many evaluations of the stochastic utility function provides a Monte Carlo approximation to the expected utility for the specified design. With a sufficiently fast model and a CMI estimator for the utility function, the ACE algorithm yields a configurable yet reliable means to optimize a highly nonlinear model over a high-dimensional design space.

Neural Network Surrogate Model

Machine learning algorithms require training data to produce models that can make predictions without being explicitly programmed to do so (Ref. 12). Supervised learning is a subset of machine learning where the training data is a set of input/output pairs and the generated model learns to predict the desired output from a given input (Ref. 13).

An ANN is a machine learning algorithm loosely inspired by the biological brain (Ref. 14). ANNs contain an input layer and an output layer and may contain one or more hidden layers in between, with each layer containing one or more artificial neurons (Ref. 15). The input and output layers have one neuron for each data input and output, respectively. After the input layer, each neuron performs a simple computation based on the weighted output of neurons in the previous layer until the neurons in the output layer make the final model prediction. Neurons are connected to those in previous layers in a manner analogous to biological axons, and the strength of each connection is determined by an assigned weight. The output of the *i*th neuron in layer *l* is given by

$$a_i^l = g\left[\sum_{j=1}^{n}\left(a_j^{l-1}\Theta_{ij}^l\right) + b_i^l\right] \tag{5}$$

where Θ_{ij}^l is the weight of the connection between the *i*th neuron in layer *l* and the *j*th neuron in layer *l*–1, b_i^l, is a bias term added to the *i*th neuron in layer *l*, and *g* is an activation function that adds nonlinearity to the ANN. Without a nonlinear activation function, each neuron could only calculate a linear function and the ANN could not learn more complex functional mappings. Some popular activation functions include the sigmoid function, the hyperbolic tangent function, and the rectified linear unit (ReLU) (Ref. 16). The weights, Θ, and biases, *b*, are the model parameters that must be "learned" before accurate predictions can be made. These parameters are learned by minimizing a loss function, such as mean squared error (MSE), which is defined to represent the error in the model's predictions. During training, backpropagation is used to calculate the gradient of the loss function with respect to the model's parameters, and a gradient-based optimization algorithm iteratively updates the model parameters to minimize the loss function.

Given sufficient training data, ANNs can learn to model arbitrarily complex functions to a high degree of accuracy. They are also computationally efficient due to the highly parallel nature of the identical neurons. The preactivation of an entire layer of neurons can be calculated with a single dot product followed by an element-wise vector addition $(\mathbf{a}^{l-1}\boldsymbol{\Theta}^{l-1} + \mathbf{b}^l)$. Conversely, scientific simulations are often computationally expensive and must perform many sequential calculations to effectively model physics-based processes. A machine-learning-based surrogate model can efficiently mimic a physics-based model when trained using the input/output data pairs from the physics model as training data (Ref. 17). ANNs make ideal surrogate models due to their flexibility in mapping complex relationships and their high computational efficiency.

Micromechanics Analysis Code With Generalized Method of Cells

Training neural networks through supervised learning requires large amounts of labeled data. Recently, the use of high-fidelity simulations to create synthetic training data has generated models that can make accurate real-world predictions (Refs. 18 to 21). In this work, a surrogate model is trained to mimic Micromechanics Analysis Code with Generalized Method of Cells (MAC/GMC), a high-fidelity physics-based micromechanics simulation software package (Ref. 22) that can accurately predict the mechanical behavior of composite materials and laminates. The software is based on the generalized method of cells (GMC) micromechanics theory (Ref. 23) and has been extensively utilized over the past two decades. MAC/GMC provides both the effective composite response and the associated local fields within the composite material, thereby allowing incorporation of various deformation and damage constitutive laws. Within the context of the GMC, the microstructure of a periodic material is represented by a rectangular repeating unit cell (RUC) consisting of an arbitrary number of rectangular subcells, each of which may be a distinct material (Figure 1). In this work, representation of the microstructure is limited to a two-by-two RUC, but an ANN could mimic any arbitrary microstructure given the appropriate training data. More complex microstructures would increase the computational cost of generating training data, but the speed increase of the surrogate would likely be even more pronounced. Displacement and traction continuity is enforced in an average (integral) sense at each of the subcell interfaces and the periodic boundaries of the RUC. These continuity conditions are used to formulate a strain concentration matrix **A**, which gives the local subcell strains, $\boldsymbol{\varepsilon}_{\mathbf{subcell}}$, in terms of the global, average, applied strains $\boldsymbol{\varepsilon}_{\mathbf{applied}}$ ($\boldsymbol{\varepsilon}_{\mathbf{subcell}} = \mathbf{A}\,\boldsymbol{\varepsilon}_{\mathbf{applied}}$). The local subcell stresses, $\boldsymbol{\sigma}$, can then be calculated using the local constitutive law and the local subcell strains. Finally, the overall RUC stiffness is obtained utilizing the local constitutive law and the strain concentration matrix averaged over the RUC dimensions. These effective (RUC) properties are then employed for a given ply within a composite laminate configuration (Figure 1) using classical lamination theory. The detailed methodology of GMC and its formulation to be embedded within classical lamination theory is described thoroughly in Aboudi et al. (Ref. 24).

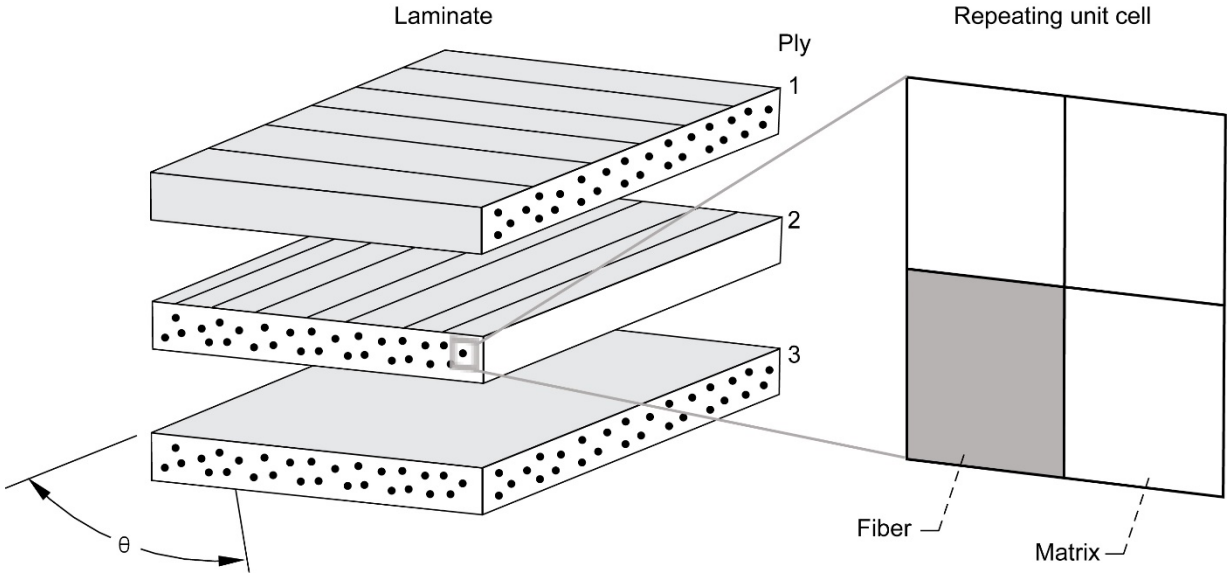

Figure 1.—Composite laminate made up of three stacked plies with diagram of two-by-two repeating unit cell (RUC). Each ply contains a doubly periodic array of parallel continuous fibers embedded in matrix material. Plies are oriented with the lengthwise direction of the fibers rotated by a specified angle, θ.

Application Example

Composites are increasingly being used in aircraft and automotive applications due to their higher specific properties when compared with traditional metallic materials. They are formed by combining two or more constituent materials, typically a fiber and matrix. Composites are divided into classes based on the matrix material and include polymer matrix composites (PMCs), metal matrix composites (MMCs), and ceramic matrix composites (CMCs). There is a wide selection of fiber and matrix materials as well as laminate stacking sequences to choose from, depending on an application's requirements. Typical matrix materials include epoxy, aluminum, and silicon carbide for PMCs, MMCs, and CMCs, respectively. The often stronger and stiffer reinforcing fiber can be made of materials such as carbon fiber, aramid fiber (Kevlar® (DuPont de Nemours, Inc.) and Twaron® (Teijin Aramid BV, LLC)), and boron fiber A composite ply is formed by embedding parallel or woven fibers into a matrix material. Multiple plies are stacked into a laminate (Figure 1) following some prescribed ply orientation design, known as a layup. Standard composite notation is used in this paper; a laminate's layup will be expressed as $[\alpha]_{cS}$, where α is a list of one or more angles of individual plies, c is the number of repeats of α, and the optional parameter, S, if present, indicates the layup should be made symmetric about the midplane. Laminates can be further divided into classes depending on the layup: unidirectional (all plies are at the same angle), cross-ply (plies alternate between 0° and 90°), angle-ply (plies alternate between a known plus/minus angle), and quasi-isotropic[1] laminates. In some manufacturing processes (e.g., those associated with CMCs), the fiber and matrix materials are cosynthesized into a composite ply, and therefore it is impossible to know the in situ material properties of the fiber and matrix constituents of the composite. These hidden constituent properties must be inferred by observing the loading response of the composite laminate.

The OED framework presented here is demonstrated by determining the optimal composite layup to be tested under uniaxial tensile loading to most efficiently infer the hidden constituent properties. Figure 2 shows the overall workflow of this framework. The goal is to find the OED, η^*, that can be used to most efficiently infer some hidden properties, q, from experimental measurements, y. First, training data containing input/output pairs is generated (indicated by the blue arrows). Generally, the training data can be produced by physical experiments or simulations (virtual data). Because ANNs often have high degrees of freedom, they require a large amount of training data compared with other types of machine learning and regression algorithms. Furthermore, the training data should span the entire domain of potential inputs, because ANNs tend to extrapolate poorly. Here, MAC/GMC is used to map the laminate's layup and constituent properties to the laminate's mechanical properties. The input data consists of the microscale fiber and matrix constituent properties and the microstructure of the fibers in each ply, including the fiber volume fraction, V, and the manner in which the plies are stacked into a layup, ℓ. The hidden constituent properties to be inferred are represented by $q = \left\{ E_{f,m}, \nu_{f,m}, \sigma_{f,m}^{\mathrm{UTS}} \right\}$, where E is elastic modulus, ν is the Poisson's ratio, and σ^{UTS} is the ultimate tensile strength of the specified constituent denoted by the subscripts f for the fiber and m for the matrix. The observable experiment outcome is the loading response of the laminate represented by $y = [E_x, (\sigma_x, \varepsilon_x)^{\mathrm{PLS}}, (\sigma_x, \varepsilon_x)^{\mathrm{UTS}}]$ where $(\sigma_x^{\mathrm{UTS}}, \varepsilon_x^{\mathrm{UTS}})$ is the ultimate tensile strength and corresponding strain of the laminate, $(\sigma_x^{\mathrm{PLS}}, \varepsilon_x^{\mathrm{PLS}})$ is the proportional limit[2] stress/strain, and E_x is the

[1]The laminate contains at least three different ply orientations with equal angle separation resulting in nearly isotropic in-plane laminate properties.

[2]The stress or strain point at which the loading curve first deviates from linearity. When dealing with real experimental (noisy) data, some offset criterion would need to be applied (e.g., 0.2 percent); however, when dealing with virtual data, no offset criterion is required.

effective elastic modulus of the laminate. The subscript x indicates the property is defined along the x-axis that is aligned with the loading direction.

Once sufficient training data is generated, a computationally efficient ANN can be trained as a surrogate model (indicated in purple in Figure 2) to mimic the MAC/GMC code to a high degree of accuracy. A high-speed surrogate model is a prerequisite to perform the OED optimization loop, shown in green, which often requires millions or even billions of model evaluations depending on the design space. During optimization, the ACE algorithm iteratively generates candidate designs, $\hat{\eta}$, which are used along with randomly sampled q values drawn from a uniform distribution as input to the ANN to predict the expected experimental outcome of the candidate design, $\hat{y}$. A total of r outcomes are generated for each potential design over the range of possible q values and the CMI, $I(q, \hat{y} \mid \hat{\eta})$, is determined in a Monte Carlo fashion and returned to the ACE algorithm to generate the next design. After a set number of iterations, the design that produces the maximum shared information between the hidden q values and the observed y values is determined to be the OED, η^*, which can be used to most efficiently infer the hidden properties with the least uncertainty.

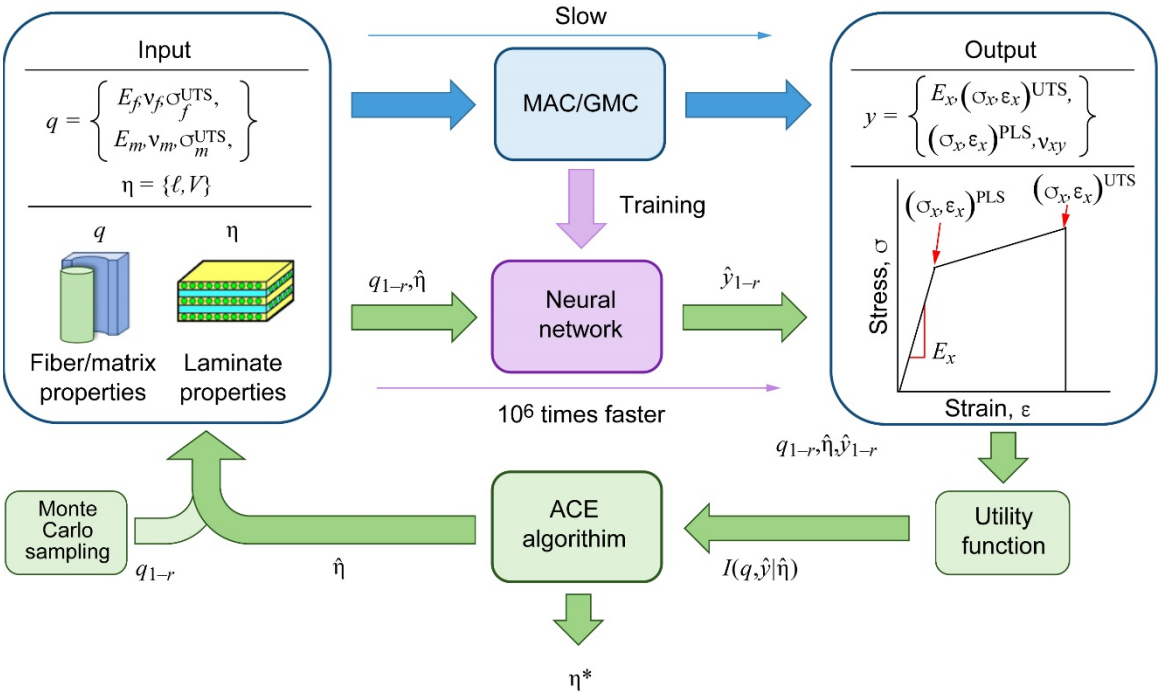

Figure 2.—Schematic of framework. A physics model (blue) generates training data for a surrogate model (purple), which is optimized over using the approximate coordinate exchange (ACE) algorithm (green) to generate the optimal experimental design, η^*. The controllable parameters of η are the layup, ℓ, and the fiber volume fraction, V. The hidden material properties, q, include the elastic modulus, E, the Poisson's ratio, ν, and the ultimate tensile strength, σ^{UTS}, of the fiber, f, and matrix, m. The experimental outcome, y, is the measured mechanical properties of the laminate and includes the elastic modulus of the laminate in the loading (x) direction, E_x, the ultimate tensile strength and corresponding strain (σ_x^{UTS}, ε_x^{UTS}), the proportional limit stress/strain (σ_x^{UTS}, ε_x^{UTS}), and the transverse Poisson's ratio, ν_{xy}. r is the number of q values randomly sampled on each iteration used to generate r predicted experimental outcomes, $\hat{y}$, for each candidate design, $\hat{\eta}$. I is the conditional mutual information.

Methods

Modeling Assumptions and Training Data

In the current study, the fiber and matrix are assumed to behave linearly elastically and to be isotropic, thus only a single E, ν, and σ^{UTS} need be specified for each constituent. A two-by-two RUC is used where the lower-left subcell is fiber and the other three are matrix material. The heights and widths of the subcells are adjusted based on V. Note, the idealized case of two-phase perfectly bonded constituents is assumed for all classes of composites. In reality, MMCs typically exhibit inelastic stress/strain behavior and fiber matrix debonding, whereas CMCs possess a third phase (i.e., a weak interface material between the fiber and matrix). Nonlinear behavior due to damage is accounted for with subcell elimination once the given failure criterion is reached; in the case of the fiber constituent, that criterion is maximum stress, whereas the matrix uses the Hashin failure criterion, which determines if the maximum tensile or compressive stress is achieved in any in-plane direction (Ref. 22). Subcell elimination represents the failure of a material in a subcell by setting the material stiffness of the subcell to approximately zero. Table I shows the range of material properties used for each composite class. The composite laminate is always assumed to be loaded in the x-direction under strain control, while all other force resultants are assumed to be zero. To simplify the design space, all laminates are assumed to be composed of 24 plies of equal 0.04167-mm thickness, so that the entire laminate thickness is 1 mm. Furthermore, 13 representative laminate layups are considered: $[0]_{24}$, $[15, -15]_{6S}$, $[30, -30]_{6S}$, $[45, -45]_{6S}$, $[60, -60]_{6S}$, $[75, -75]_{6S}$, $[90]_{24}$, $[0, 90]_{6S}$, $[0, -60, 60]_{4S}$, $[0, -45, 45, 90]_{3S}$, $[0, 45, 90, -45]_{3S}$, $[45, -45, 90, 0]_{3S}$, and $[0, 90, -30, 30, 60, -60]_{2S}$. Using categorical layup representations significantly reduced the input dimensionality, but an ANN could be trained using a continuous layup representation with appropriate training data. During virtual training, MAC/GMC outputs the laminate line load and midplane strain responses of the laminate. The line load in the loading direction can be converted to stress by dividing by the laminate thickness. From the resulting stress/strain curve, five pointwise properties were extracted: $y = \{E_x, (\sigma_x, \varepsilon_x)^{PLS}, (\sigma_x, \varepsilon_x)^{UTS}\}$. The ANN focused on predicting only these five extracted features, whereas MAC/GMC must first calculate the full stress/strain fields. This contributes to the efficiency of the ANN in addition to solving highly parallel vector operations.

TABLE I.—SUMMARY OF TRAINING DATA USED TO CREATE EIGHT TOTAL SURROGATE MODELS AND THEIR RESULTING ACCURACY

Dataset parameter		PMC[a]	MMC[b]	CMC[c]	PMC (any V[d])
No. of examples		26,000	26,000	26,000	65,000
V		0.4 to 0.7	0.2 to 0.4	0.15 to 0.30	0 to 1
E_f		60 to 110 GPa	380 to 415 GPa	280 to 380 GPa	60 to 110 GPa
σ_f		3,000 to 4,500 MPa	3,000 to 6,000 MPa	1,500 to 2,500 MPa	3,000 to 4,500 MPa
ν_f		0.2 to 0.3	0.2 to 0.3	0.2 to 0.3	0.2 to 0.3
E_m		3 to 5 GPa	50 to 250 GPs	350 to 420 GPa	3 to 5 GPa
σ_m		50 to 110 MPa	400 to 1,200 MPa	400 to 600 MPa	50 to 110 MPa
ν_m		0.33 to 0.38	0.3 to 0.4	0.2 to 0.3	0.33 to 0.38
Accuracy	With ν_{xy}	96.7 percent	96.8 percent	95.8 percent	99.0 percent
	Without ν_{xy}	97.6 percent	97 percent	96.0 percent	99.0 percent

[a]Polymer matrix composite.
[b]Metal matrix composite.
[c]Ceramix matrix composite.
[d]Volume fraction.

Surrogate Model Training

In total, eight surrogate models were trained to mimic MAC/GMC and used to perform OED under differing conditions. For each case, the training data was split into an 80/10/10 training/validation/test split. Surrogate models were generated for each of three composite material classes: PMCs, MMCs, and CMCs. For each class, two models were trained in order to study the effect of using the laminate Poisson's ratio, ν_{xy}, as a target variable; one model included the Poisson ratio and the other did not. In physical experiments, measuring the laminate's transverse strain response in order to determine the Poisson's ratio requires additional setup and cost, but doing so can provide additional information relevant to determining q. In each case, the input training data was drawn from a uniform distribution of typical constituent property values and V for that composite class. A summary of the training data and final model accuracy is shown in Table I. The accuracy is calculated using the mean absolute percentage error of predictions on a test set. For these six models, q and V were sampled 2,000 times and used, along with each of the 13 layup configurations, as input to MAC/GMC to generate 26,000 input/output pairs for training data. The 13 layup configurations were one-hot encoded by representing the layup as 12 features, where each feature represents a particular layup when set to a value of 1 (with 1 layup represented by 12 zeros). One-hot encoding is necessary when training with categorical data so the ANN does not evaluate the layups based on the magnitude of an arbitrary label (i.e., 1 to 13). To study the effect of using V as a design parameter for OED, two additional surrogate models were generated using the PMC class composite (with and without using ν_{xy} as a target variable) by allowing V to range from 0 to 1 in the training data. For these last two models, 65,000 training examples were generated. Cross validation was not used because different potential folds of the training data had similar summary statistics, and cross validation would require more models to be trained during hyperparameter optimization. Before training, all input and output features were normalized so that each feature had a mean of zero and standard deviation of 1 for all examples in the training set. All ANN training and design was performed using Keras (Ref. 25) with TensorFlow (Ref. 26) in the R programming language (Ref. 27).

Manual Hyperparameter Search

For each of the six surrogate models using a narrow range of V, a manual hyperparameter search was used to select the model hyperparameters, such as the architecture and learning rate. The accuracy of these models (~97 percent) was deemed sufficient to perform OED. These ANNs had five layers with 80, 45, 45, and 25 neurons in the hidden layers, respectively. The output layer used to predict $y = \{E_x, (\sigma_x, \varepsilon_x)^{\mathrm{UTS}}, (\sigma_x, \varepsilon_x)^{\mathrm{PLS}}, [\nu_{xy}]\}$ had one neuron for each output prediction (five or six, depending on whether the model was used to predict ν_{xy}). The input layer had 19 neurons: 6 for q; 1 for V; and 12 from the one-hot encoding of the 13 possible layup configurations, ℓ. The ReLU (Ref. 16) activation function was used for nonlinearity. MSE was used as the loss function. Parameter optimization was performed using the Adam optimizer (Ref. 28) with Nesterov momentum (Ref. 29) and a learning rate of 0.002. Overfitting was prevented by using dropout (Ref. 30) as a regularization technique, with a dropout rate of 8 percent during training.

Bayesian Hyperparameter Optimization

At first, when using the same hyperparameters on the models where V was allowed to vary freely, a test accuracy of 97.7 percent was achieved (97.2 percent with ν_{xy}). Although this accuracy appeared acceptable, a closer examination revealed that the accuracy was 94.3 percent (93.7 percent with ν_{xy}) when

making predictions with a V less than 0.1 and higher than 0.9. Since one might expect an optimal experiment to include tests with a very high V (e.g., fiber only) and/or a very low V (e.g., neat matrix), a higher accuracy in this regime was desired. A hyperparameter search using Bayesian optimization was performed to build a model with higher accuracy. The upper confidence bound was sampled on each iteration over the following hyperparameters: number of layers (2 to 20), number of nodes in the first hidden layer (20 to 500), dropout rate (0 to 40 percent), learning rate (0.0001 to 0.01), batch size (16 to 1,024), activation (ReLU or tanh), and architecture layout (same nodes in each hidden layer or reduced nodes in deeper layers). After 250 iterations, the optimal hyperparameters when varying V were found to be 4 hidden layers, 400 nodes in each hidden layer, a learning rate of 0.001, a batch size of 512, a dropout rate of 7 percent, and the ReLU activation function. Using these hyperparameters along with learning rate decay and early stopping, a final model was trained that achieved an accuracy of 99.4 percent on the training set, 99.1 percent on the validation set, and 99.0 percent on the test set. Despite the low dropout rate, the small difference between the training and validation accuracy suggests little overfitting. For test set samples with extreme fiber V, this model had an accuracy of 97.7 percent. Similarly, a second optimized model trained in the same fashion but including ν_{xy} as a target variable had an overall test accuracy of 99.0 percent (and 97.5 percent on extreme V samples). When varying V, using optimal model architectures and hyperparameters found through Bayesian optimization produced final models that were much more accurate, especially for samples with extreme V.

Optimal Experimental Design

In all cases, the ACE algorithm was used to estimate the optimal design. Where possible, an exhaustive search was also used to evaluate the utility of all possible designs. With the exhaustive search, it was possible to determine the true optimal design and evaluate the ACE algorithm's convergence on a good estimate. Testing every possible design with an exhaustive search was only tractable when controlling one categorical parameter in the experimental design space.

Results

Table II and Table III show 24 OED results using only the layup as the design parameter (k = 1) and performing two to five tests per experiment (n = 2 to 5). Each of the three composite classes was used with and without measuring ν_{xy}. For select experiments, CMI matrices showing all the possible designs when n = 2 and a two-dimensional slice of the three-dimensional CMI matrices for when n = 3 are shown in Figure 3. When measuring the transverse laminate strain and performing two tests, the optimal design tested the $[0]_{24}$ and $[45, -45]_{6S}$ layups for the PMC, CMC, and MMC composites. These tests represent two of the most common laminate layups (or similar if using fewer plies) used for composite testing regardless of material system. When performing three tests, the $[0]_{24}$, $[90]_{24}$, and $[45, -45]_{6S}$ layups were deemed optimal for MMCs, which in practice is a common experimental set, based on heuristic knowledge and experience. This is because the $[0]_{24}$ layup response is highly dependent on the fiber properties, the $[90]_{24}$ on the matrix properties, and the $[45, -45]_{6S}$ on a good mix of both. In the case of CMCs, the optimal experimental set was found to be $[0]_{24}$, $[0, 90]_{6S}$, and $[45, -45]_{6S}$ layups, which was also expected from heuristics. In the CMC case, the composite matrix is extremely brittle, and thus the $[90]_{24}$ layup fails quickly, with little separation between the proportional limit stress/strain and ultimate tensile stress/strain points, causing the $[90]_{24}$ layup to have a slightly reduced experimental value. Meanwhile, the $[0, 90]_{6S}$ layup still allows for that separation. Interestingly, the optimal experimental set for the PMC case was the $[0]_{24}$, $[90]_{24}$, and $[30, -30]_{6S}$ layups, although Figure 3(b) shows that the $[0]_{24}$, $[90]_{24}$, and $[45, -45]_{6S}$ design was close to optimal as well. While the $[0]_{24}$, $[90]_{24}$, and $[45, -45]_{6S}$, layups

TABLE II.—SUMMARY OF OPTIMAL POLYMER MATRIX COMPOSITE (PMC), METAL MATRIX COMPOSITE (MMC), AND CERAMIC MATRIX COMPOSITE (CMC) EXPERIMENTS (N = 2 TO 4)

Composite		Best 2		Best 3			Best 4			
PMC	With ν_{xy}	$[0]_{24}$	$[45, -45]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[30, -30]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[30, -30]_{6S}$	$[0, 45, 90, -45]_{3S}$
	Without ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[0]_{24}$	$[90]_{24}$	$[45, -45, 90, 0]_{4S}$	$[0]_{24}$	$[15, -15]_{6S}$	$[60, -60]_{6S}$	$[0, -60, 60]_{4S}$
MMC	With ν_{xy}	$[0]_{24}$	$[45, -45]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[45, -45]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$
	Without ν_{xy}	$[0]_{24}$	$[60, -60]_{6S}$	$[0]_{24}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$	$[0]_{24}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$	$[75, -75]_{6S}$
CMC	With ν_{xy}	$[0]_{24}$	$[45, -45]_{6S}$	$[0]_{24}$	$[0, 90]_{6S}$	$[45, -45]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[0, 90]_{6S}$	$[45, -45]_{6S}$
	Without ν_{xy}	$[0]_{24}$	$[45, -45]_{6S}$	$[0]_{24}$	$[0, 90]_{6S}$	$[45, -45]_{6S}$	$[0]_{24}$	$[90]_{24}$	$[0, 90]_{6S}$	$[45, -45]_{6S}$

TABLE III.—SUMMARY OF OPTIMAL POLYMER MATRIX COMPOSITE (PMC), METAL MATRIX COMPOSITE (MMC), AND CERAMIC MATRIX COMPOSITE (CMC) EXPERIMENTS (n = 5)

Composite		Best 5				
PMC	With ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$	$[0, -45, 45, 90]_{3S}$
	Without ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[45, -45]_{6S}$	$[60, -60]_{6S}$	$[75, -75]_{6S}$
MMC	With ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[30, -30]_{6S}$	$[45, -45]_{6S}$	$[60, -60]_{6S}$
	Without ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$	$[60, -60]_{6S}$
CMC	With ν_{xy}	$[0]_{24}$	$[90]_{24}$	$[0, 90]_{6S}$	$[15, -15]_{6S}$	$[45, -45]_{6S}$
	Without ν_{xy}	$[0]_{24}$	$[45, -45]_{6S}$	$[0, -60, 60]_{4S}$	$[0, 90, -30, 30, -60, 60]_{2S}$	$[0, 45, 90, -45]_{3S}$

represent commonly tested configurations, $[30, -30]_{6S}$ is seldom tested. This illustrates the potential for machine learning approaches to identify optimal designs that are not currently being considered in practice. Note the precise optimal angle may not actually be ±30, as the angle was not a free variable but rather a preselected angle within the set of 13 layups available to the algorithm. Based on these results, it is not surprising that testing the $[0]_{24}$, $[90]_{24}$, and $[45, -45]_{6S}$ layups (swapping the $[90]_{24}$ for the $[0, 90]_{24}$ with the CMC) is a heuristic "rule of thumb" across composite materials. However, using an information-theoretic approach for choosing the design can lead to even better design choices. It is likely that the stronger fiber/matrix property mismatch in the PMC, as compared with the MMC and CMC composites, favors a different optimal test. Further, as one might suspect, if the laminate Poisson's ratio, ν_{xy}, was not measured, the optimal experimental set changed for a specified number of tests, except in the case of CMCs (see Table II). From Figure 3(a), for the PMC case when the Poisson's ratio, ν_{xy}, is not provided, it is clear that although the optimal experimental set is the $[0]_{24}$ and $[90]_{24}$ layups, a nearly equivalent CMI is achieved for the $[0]_{24}$ and any other high angle-ply laminate. Alternatively, performing experiments on two similar layups (such as two quasi-isotropic or two high angle-ply laminates) resulted in the lowest CMI between q and y. In the case of three layups, there were many tests with equivalent utility when including the $[0]_{24}$ layup, any high angle-ply laminate, and any quasi-isotropic-ply laminate. Although the optimal design is given in Table II, the difference in utility of several slightly suboptimal designs may be insignificant due to the stochastic nature of the utility function. The fact that there were many equivalent designs when not measuring ν_{xy} and that maximum CMI increased when ν_{xy} was measured suggests that performing this additional measurement will likely lead to more reliable and accurate inference of the constituent properties when the optimal tests are performed.

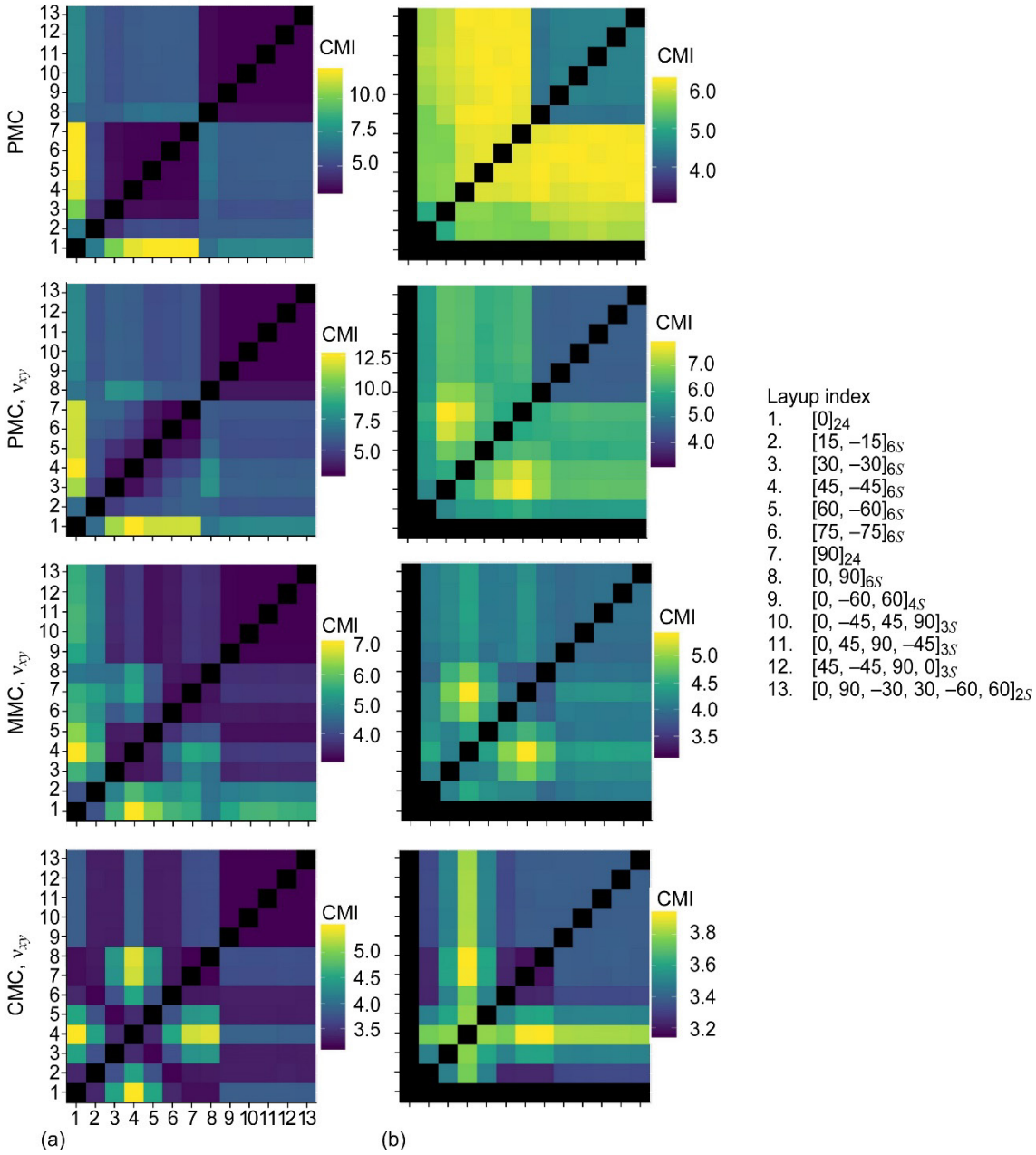

Figure 3.—Conditional mutual information (CMI) matrices for select experiments. Ceramic matrix composite, CMC; metal matrix composite, MMC; polymer matrix composite, PMC; laminate Poisson's ratio, ν_{xy}. (a) CMI of every possible combination of layups for experiments with two tests. (b) Slice of rank 3 CMI tensor of all possible combinations for three-test experiments where third test is $[0]_{24}$ layup.

TABLE IV.—SUMMARY OF OPTIMAL VOLUME FRACTION EXPERIMENTS (k = 2, n = 2 to 4)

Volume fraction		Best 2		Best 3			Best 4			
V = 0 to 1	With ν_{xy}	(0)	(1)	(0)	(0)	(1)	(0)	(0)	(0)	(1)
	Without ν_{xy}	(0)	(1)	(0)	(0.88)| $[15, -15]_{6S}$	(1)	(0)	(0.88)| $[15, -15]_{6S}$	(1)	(1)
V = 0.3 to 0.7	With ν_{xy}	(0.3)| $[45, -45]_{6S}$	(0.7)| $[0]_{24}$	(0.3)| $[90]_{24}$	(0.3)| $[30, -30]_{6S}$	(0.7)| $[0]_{24}$	(0.3)| $[90]_{24}$	(0.3)| $[30, -30]_{6S}$	(0.7)| $[15, -15]_{6S}$	(0.7)| $[0]_{24}$
	Without ν_{xy}	(0.3)| $[45, -45]_{6S}$	(0.7)| $[90]_{24}$	(0.3)| $[90]_{24}$	(0.7)| $[15, -15]_{6S}$	(0.7)| $[0]_{24}$	(0.3)| $[75, -75]_{6S}$	(0.7)| $[15, -15]_{6S}$	(0.7)| $[0, 45, 90, -45]_{3S}$	(0.7)| $[0]_{24}$

Controlling multiple parameters allows for more flexible and potentially more valuable experiments. This framework is robust enough to handle large design spaces, that is, those that contain many tests (high n) and multiple control parameters (k) simultaneously. Table IV shows the optimal experiments when both V and ℓ are controllable parameters (k = 2) and between two and four tests are performed (n = 2 to 4). A V of zero represents pure matrix, and a V of 1.0 represents pure fiber. In these cases, the pure material is not a composite; the ply orientation angles and laminate layup have no real meaning and are not reported. When two tests were performed, it was optimal to test the pure fiber and the pure matrix. This is a completely obvious result: when attempting to determine the properties of the fiber and matrix, one should measure the fiber and matrix directly if possible. When ν_{xy} was measured, repeated tests of the pure matrix were prescribed. It is interesting to note that when ν_{xy} was not measured, it was beneficial to measure a high-V laminate with a low-angle $[15, -15]_{6S}$ layup in addition to the pure components when performing three or more tests. Allowing the OED framework to test pure materials is an unrealistic and trivial example that does not fully demonstrate the framework's ability to find optimal designs with multiple design parameters. Table IV also shows the optimal experiments when restricting V to vary between 0.3 and 0.7. As expected, even when restricting V range, the optimal experiment always called for tests with extreme values within the allowed volume fraction range. Typically, low-V tests used higher angle layups than the high-V tests, leading to tests where both design parameters complemented each other in producing mechanical outcomes that were dominated by the effect of one constituent property or the other. Each optimal experiment contained tests that predominately experienced the effects of a different constituent material in nearly equal proportions.

Validation of Optimal Layup Selection

Least-squares minimization was used to evaluate the significance of using an OED versus a suboptimal design to infer the hidden constituent properties. The optimal design set, which had the highest CMI for PMC class composites when measuring ν_{xy}, consisted of tests on the $[0]_{24}$, $[90]_{24}$, and $[30, -30]_{12}$ layups. The chosen suboptimal design set, which had a low CMI, consisted of tests on the $[45, -45]_{12}$, $[60, -60]_{12}$, and $[75, -75]_{12}$ layups. Two virtual experimental sets were obtained by setting the "true" constituent properties to be inferred (in the typical range of PMC composites) and simulating the resulting mechanical response of the six specified layups (set 1, high CMI layup set, and set 2, low CMI layup set) using MAC/GMC. Least-squares minimization was used to iteratively estimate the true constituent properties by evaluating MAC/GMC using estimated constituent properties and the specified layups as input and comparing the resulting mechanical response to the true mechanical response. The minimization procedure for both the optimal (set 1) and suboptimal (set 2) experiments was allowed six iterations initiated from the same randomly chosen starting constituent properties. When using the OED, the minimization procedure estimated the target constituent properties with a mean absolute percentage error of 6.43 percent, whereas the suboptimal design had a mean absolute percentage error of 15.99 percent.

The minimization solver was able to converge on the target constituent properties more efficiently and with less error when using the experiment prescribed by the OED (set 1). The larger CMI between the constituent properties and the laminate mechanical response of the optimal experiment indicates that there is a larger mutual dependence between them, which suggests that a change to the constituent properties will have a larger effect on the mechanical response for the optimal layups than the suboptimal. That is, the minimization objective function likely has steeper gradients when solving for the constituent properties using the optimal design, leading to more efficient convergence. These results show that performing OED with this framework generates optimal designs that can significantly reduce the uncertainty in inferring hidden variables by choosing the most efficient experiment to conduct.

Discussion

MAC/GMC is a physics-based research code that performs numerous multiscale calculations to predict composite mechanical response. While the code is relatively fast, with a typical analysis running on the order of seconds, it still is too slow to be useful when performing OED optimization. For example, when considering 13 candidate layups and two tests ($n = 2$, $k = 1$), there are 78 possible designs to examine (by choosing 2 of the 13 layups without repeating), whereas when considering five tests ($n = 5$, $k = 1$) this rises quickly to 1,287 possible designs. For each design, the experimental outcome must be predicted for each constituent property sample point in a sufficiently large sample size (e.g., 1,000 times), in order to estimate the CMI across a range of possible hidden properties. Running on 80 cores, MAC/GMC required roughly 0.2 s per output, whereas the trained ANN required 21 μs per output on a single core. Using these values, generating an optimal design from an exhaustive search with five tests would require over 70 h using MAC/GMC, and only 30 s using the trained ANN. When the design space becomes continuous or includes multiple design parameters, an exhaustive evaluation of the CMI over the design space becomes intractable, even with the speedup of the ANN. In this case, the ACE algorithm could efficiently maximize the CMI over the design space when using the ANN within the utility function. To ensure that the ACE algorithm converged on the maximum CMI with the stochastic utility function, the optimal design generated by ACE was compared to the design with the maximum CMI found by an exhaustive search when the layup (represented by a single categorical variable) was the only design parameter for $n = 2$ to 5 tests. When considering ν_{xy} in the output, the ACE algorithm always converged on the same optimal design determined by the exhaustive search. As discussed earlier, when ν_{xy} was not considered, there were often many nearly equally optimal designs, and even the exhaustive search method did not always find the same optimal design, even when evaluating the stochastic utility function 1,000 times for each design. Nevertheless, the ACE algorithm always converged on the optimal or nearly optimal design.

It is not uncommon in materials science experiments to infer underlying material properties based on the results of physical measurements. Determining the optimal experiment to infer these properties can save significant time and money. This framework is especially important when (contrary to the current application) there are no predetermined best practices or heuristic knowledge to guide the experiment, or when there are restrictions on the design space due to some experimental condition. This method can determine the optimal design when restricting the design space in any arbitrary way, even with complex restrictions based on the settings of other parameters. It can also be used after some tests have already been performed to find the most efficient next set of tests to perform. In general, this framework can be used to determine optimal designs from a design space of arbitrary size, shape, and constraint, to most efficiently infer hidden properties as long as there is some way to map the relationship between the hidden properties and experimental observations.

Conclusions

A robust framework for determining the most efficient experiment with which to infer hidden material properties from observable outcomes was demonstrated. The framework can optimize over larger experimental design spaces with more tests and controllable parameters than previously possible by leveraging machine learning to build efficient surrogate models and the approximate coordinate exchange algorithm for optimization. This method was applied to finding optimal experimental designs with which to infer the hidden constituent properties (lower length scale) of composites from mechanical tests on the laminates (higher length scale). The resulting experimental designs often agreed with heuristic practices, with the exception of a small number of reasonable adjustments in specialized cases. This framework is general enough to be applied to any system where hidden properties are related to observable measurements and some model exists to map that relationship.

References

1. Aggarwal, R.; Demkowicz, M.J.; and Marzouk, Y.M.: Bayesian Inference of Substrate Properties From Film Behavior. Model. Simul. Mater. Sci. Eng., vol. 23, no. 1, 2015.
2. Maynard, Julian: Resonant Ultrasound Spectroscopy. Phys. Today, vol. 49, 1996, pp. 26–31.
3. Overstall, A.; and Woods, D.: The Approximate Coordinate Exchange Algorithm for Bayesian Optimal Experimental Design. Short Summary, 2015.
4. Lindley, D.V.; and Smith, A.F.M.: Bayes Estimates for the Linear Model. J. R. Stat. Soc., vol. 34, no. 1, 1972, pp. 1–18.
5. Aggarwal, R.; Demkowicz, M.J.; and Marzouk, Y.M.: Information-Driven Experimental Design in Materials Science. Springer Ser. Mater. Sci., vol. 225, 2015, pp. 13–44.
6. Ginebra, J.: On the Measure of the Information in a Statistical Experiment. Bayesian Anal., vol. 2, no. 1, 2007, pp. 167–212.
7. Cover, Thomas M.; and Thomas, Joy A.: Elements of Information Theory. Second ed., John Wiley & Sons, Hoboken, NJ, 2005.
8. Póczos, Barnabas; and Schneider, Jeff: Nonparametric Estimation of Conditional Information and Divergences. J. Mach. Learn. Res., vol. 22, 2012, pp. 914–923.
9. Kraskov, Alexander; Stögbauer, Harald; and Grassberger, Peter: Estimating Mutual Information. Phys. Rev. E, vol. 69, no. 6, 2004, pp. 066138–1—066138–16.
10. Rahimzamani, Arman; and Kannan, Sreeram: Potential Conditional Mutual Information: Estimators and Properties. Presented at the 55th Annul Allerton Conference on Communication, Control, and Computing, Monticello, IL, 2017, pp. 1228–1235.
11. Gao, Weihao, et al.: Estimating Mutual Information for Discrete-Continuous Mixtures. Adv. Neural Inf. Process. Syst., 2017, pp. 5987–5998.
12. Abramson, N.; Braverman, D.; and Sebestyen, G.: Pattern Recognition and Machine Learning. IEEE Trans. Inf. Theory, vol. 9, no. 4, 1963, pp. 257–261.
13. Russell, Stuart J.; and Norvig, Peter: Artificial Intelligence: A Modern Approach. Third ed., Pearson Education Limited, Essex, England, 2016.
14. McCulloch, Warren S.; and Pitts, Walter: A Logical Calculus of the Ideas Immanent in Nervous Activity. Systems Research for Behavioral Science: A Sourcebook, Ch. 11, Walter Buckley, ed., Taylor & Francis Group, Oxford, England, 2017, pp. 93–96.
15. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychol. Rev., vol. 65, 1958, pp. 386–408.

16. Nair, Vinod; and Hinton, Geoffrey E.: Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010, pp. 807–814.
17. Forrester, A.I.J.; Sobester, A.; and Keane, A.J.: Engineering Design via Surrogate Modelling: A Practical Guide. John Wiley & Sons, Chichester, England, 2008.
18. Andrychowicz, Marcin, et al.: Learning Dexterous In-Hand Manipulation. Int. J. Rob. Res., vol. 39, no. 1, 2019, pp. 3–20.
19. Comaniciu, Dorin, et al.: Image-Based Tumor Phenotyping With Machine Learning From Synthetic Data. U.S. Patent 10,282,588B2, May 2019.
20. Shrivastava, Ashish, et al.: Learning From Simulated and Unsupervised Images Through Adversarial Training. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, 2017, pp. 2242–2251.
21. Haj-Ali, Rami, et al.: Simulated Micromechanical Models Using Artificial Neural Networks. J. Eng. Mech., vol. 127, no. 7, 2001, pp. 730–738.
22. Bednarcyk, Brett A.; and Arnold, Steven M.: MAC/GMC 4.0 User's Manual: Keywords Manual. Vol. 2, 2002.
23. Aboudi, Jacob: Micromechanical Analysis of Composites by the Method of Cells—Update. Appl. Mech. Rev., vol. 49, 1996, pp. S83–S91.
24. Aboudi, Jacob; Arnold, Steven M.; and Bednarcyk, Brett A.: Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach. First ed., Butterworth-Heinemann, Oxford, England, 2013.
25. Chollet, Francois, et al.: Keras. 2015. https://keras.io/ Accessed Jan. 4, 2021.
26. Abadi, Martin, et al.: TensorFlow: A System for Large-Scale Machine Learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016, pp. 265–283.
27. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2017.
28. Kingma, Diederik P.; and Ba, Jimmy Lei: Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations Conference Track Proceedings, San Diego, CA, 2015.
29. Dozat, Timothy: Incorporating Nesterov Momentum Into Adam. Workshop Track ICLR 2016, 2016.
30. Srivastava, Nitish, et al.: Dropout: A Simple Way to Prevent Neural Networks From Overfitting. J. Mach. Learn. Res., vol. 15, no. 56, 2014, pp. 1929–1958.