Evaluating Designer Learning and Performance in Interactive Deep Generative Design

Ashish M. Chaudhari[1]
Sociotechnical Systems Research Center,
College of Computing,
Massachusetts Institute of Technology,
Cambridge, MA 02139
e-mails: amchaudhari@mit.edu;
ashish.chaudhari@philips.com

Daniel Selva
Assistant Professor
Aerospace Engineering,
Texas A&M University,
College Station, TX 77840
e-mail: dselva@tamu.edu

Deep generative models have shown significant promise in improving performance in design space exploration. But there is limited understanding of their interpretability, a necessity when model explanations are desired and problems are ill-defined. Interpretability involves learning design features behind design performance, called designer learning. This study explores human–machine collaboration's effects on designer learning and design performance. We conduct an experiment (N = 42) designing mechanical metamaterials using a conditional variational autoencoder. The independent variables are: (i) the level of automation of design synthesis, e.g., manual (where the user manually manipulates design variables), manual feature-based (where the user manipulates the weights of the features learned by the encoder), and semi-automated feature-based (where the agent generates a local design based on a start design and user-selected step size); and (ii) feature semanticity, e.g., meaningful versus abstract features. We assess feature-specific learning using item response theory and design performance using utopia distance and hypervolume improvement. The results suggest that design performance depends on the subjects' feature-specific knowledge, emphasizing the precursory role of learning. The semi-automated synthesis locally improves the utopia distance. Still, it does not result in higher global hypervolume improvement compared to manual design synthesis and reduced designer learning compared to manual feature-based synthesis. The subjects learn semantic features better than abstract features only when design performance is sensitive to them. Potential cognitive constructs influencing learning in human–machine collaborative settings are discussed, such as cognitive load and recognition heuristics. [DOI: 10.1115/1.4056374]

Keywords: design space exploration, deep generative models, interpretability

1 Introduction

Deep learning methods have been applied to a variety of engineering design problems such as airfoil design [1], structural design [2–4], and metamaterial design [5]. Indeed, design space exploration (DSE) using deep learning, referred to as deep generative design, creates novel designs efficiently and shows improvements over traditional optimization methods [6,7]. Deep learning methods can effectively optimize well-defined design performance metrics and meet quantitative requirements under constraints [8]. Deep generative design helps by distilling high-dimensional input data into low-dimensional representations, which we call *features*. However, to be useful for deriving insights, the features need to be understood by designers, a key requirement for model interpretability. In this context, designer learning includes identifying successful designs, understanding the features behind successful designs and constraints, and knowing the analogical association between designs [9,10]. Knowing driving features can aid in directing the exploration of a large design space [11–13]. Understanding key features is also necessary when designers must explain design decisions to stakeholders, which requires rationales, especially in the early design phase. This learning process is a prerequisite when a design problem is ill-defined, and the knowledge from early design tasks needs to be transferred to subsequent design processes.

In this paper, we highlight that bringing a human designer into the DSE can potentially better balance a designer's learning and design performance compared to a "black-box" only optimization [14,15]. Existing approaches to human-in-the-loop design space exploration take designer inputs on design- and feature selection [16–18] and present feedback on performance metrics and the diversity of generated methods [3,19]. The interaction between a human designer and the computer includes visual graphical user interfaces [16], natural language processing interfaces for question answering and textual explanations [20], and tangible physical interfaces [14]. Despite this progress, there is a limited understanding of whether generative design methods are effective for human learning. Specifically, the evaluation of how and whether different types of features (semantic versus abstract) and modes of interaction improve designer learning and design performance are limited.

Existing approaches apply automation to different functions such as design search, design analysis, and design evaluation [21]. Within each part, the level of automation can vary from low to high, i.e., from manual to fully automatic. This paper explicitly analyzes levels of automation of design exploration. Also, the type of feature is associated with engineering significance. Features are information sets that refer to the form, function, material, or precision attributes of a part [22]. The semantic nature of a feature can exploit an individual's dense prior knowledge relative to abstract, data-driven features [23]. Therefore, the research objective of this paper is to quantify the effects of changing two interactivity-related factors in DSE: (i) the level of automation of the search function, e.g., whether a designer generates a design manually from its constituting parts (low automation), manually from predefined features, or automatically using a deep generative design method (high automation); and (ii) the semanticity of features, e.g., features can have a

[1]Corresponding author.
Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received June 12, 2022; final manuscript received November 18, 2022; published online January 10, 2023. Assoc. Editor: Mark Fuge.

Table 1 Dimensions and categories of existing interactive deep generative design techniques

Dimension	Categories	Examples
Input type	Design space	Choose desired designs from generated designs to guide further exploration
	Feature space	Parametric change or selection made to latent embedding or feature values
	Objective space	A user selects a desired range or values for a specific design objective
DSE outcomes	Performance-driven	Generated designs maximize fixed performance metrics or converge towards true Pareto front
	Diversity-driven	Designs are generated to increase diversity in decision/feature and objective spaces
	Learning-driven	Designs are generated to learn the main aspects driving the problem, such as sensitivities or features common among Pareto designs
Human–machine interface	Graphical user interface	Generated designs and/or features are visualized as images or graphs and objective/feature space are visualized as scatter plots
	Natural language interface	User asks questions through voice or a chatbox, and human–machine provides answers such as explanations about the design's performance
	Tangible interface	A user creates designs by manipulating a physical representation of it (e.g., wooden blocks on a tabletop) while visualizing the tradespace information on a computer screen

semantic meaning, or they can be abstract latent features that are an output of a generative design algorithm. The evaluative criteria of the automation and feature type level are designer learning and design performance.

The approach follows a human subject experiment and a quantitative measurement of designer learning and design performance. A conditional variation autoencoder (C-VAE) [24] enables the generative design of mechanical metamaterials with strength-based and density-based objectives. The human subject experiment instantiates variations of the C-VAE based on the independent variables under study. Overall, the experimental data include 42 subjects from a within-subject experiment. We measure design performance with established multi-disciplinary design optimization measures. Similarly, a questionnaire measures designer learning after each experimental condition separately [25,26]. An item response theory (IRT) model estimates the subjects' feature-specific "abilities" based on the questionnaire responses [27].

This study contributes critical behavioral insights and an IRT model for assessing learning in interactive deep generative design.

(1) The analysis is the intertwined nature of design performance with designer learning, a hypothesis promoted by Sim and Duffy [9,28,29]. For example, the semi-automated generated method with a high level of automation adversely impacts the subjects' feature-specific learning and overall design performance. Barriers to feature learning likely diminish the designer's ability to generate better designs.
(2) The study identifies behavioral patterns in how individuals learn about feature importance in interactive deep generative design. The positive influence of semanticity on how much the subjects learn and their performance depends on the features' performance sensitivity. The higher the performance sensitivity due to a feature, the higher the related learning. These insights can help design better interactive and learning-focused deep generative design tools.
(3) The paper also contributes a unique approach combining experiments and IRT to evaluate component-level learning of features. There have been qualitative approaches to assess feature understanding [30,31]. But the presented method is the first in developing a quantitative IRT model for evaluating learning in a human–machine collaborative setting. Provided that a questionnaire is implemented and relevant design features are embedded in test questions, the IRT model can scale to other design problems for evaluating feature-specific abilities.

The rest of the paper is structured as follows. Section 2 reviews existing interactive methods for generative design and related research studies. Section 3 presents the mathematical details of the C-VAE-based interactive tool, the IRT model, and the experiment design. Section 4 provides results from the analysis of the experimental data. Section 5 explains the main findings and suggests future design support tools. Section 6 presents the conclusion. The developed tool is available.[2]

2 Related Work

2.1 Interactive Generative Design Methods. Generative design refers to computational design methods that can automatically conduct DSE under constraints [4,8]. Generative design methods create multiple optimal designs by varying the weights of multi-objective and design parameters using gradient-based (e.g., stochastic gradient descent) or gradient-free (e.g., genetic algorithms) techniques. This paper focuses on deep generative design, which refers to algorithms that generate new designs using deep learning [32]. Deep neural networks (DNNs) and convolutional neural networks (CNNs) are frequently used to build surrogate models for engineering problems due to their high performance in learning patterns from images to recognize objects. The CNN consists of convolution and pooling layers, with fully connected layers at the end. Variational autoencoders use multiple CNNs. An encoder network transforms inputs into low-dimensional latent features. A decoder network is to reconstruct a design from features that maximize similarity to the original inputs [33].

There are many variations in the functional allocation between humans and algorithms in the optimization. This paper includes one option which is the user selecting the step size, but other options could be the user repairing designs to make them feasible, the user giving feedback to the agent about its designs, the user providing additional constraints to the optimizer in real-time, etc. More formally, existing interactive generative design approaches vary along the dimensions of input type, knowledge outcomes, and the type of human–machine interface. Table 1 presents a description of these dimensions. The input type pertains to how user feedback is incorporated. A user might steer design exploration by choosing a desired set of designs or design variable values [34], select design parameters or features [17,35], or set a range for desired objective values [15,19]. Using high-level rather than low-level features can reduce the number of input commands. In the context of detailed parametric tasks, generative design methods utilize form features, material features, precision features, or primitive features [22]. Furthermore, generative design typically attempts to solve multi-objective problems with multiple conflicting design criteria. Any desired DSE outcomes must be represented as a particular loss function. Existing generative design methods can optimize predefined performance metrics like structural compliance [36,5,4], maximize the diversity of generated designs [3,7,8], or learn driving features behind selected designs [17,35]. Finally, the mode of interaction between a user and the underlying tool can be a graphical user interface, natural language interface, or tangible interface. The human–

[2]https://github.com/amchaudhari/daphne-mm

machine interface also provides feedback to the user through visual representations of generated designs [37–40], explanations about evaluation models and driving features [19], and evaluations of physical prototypes [14]. This paper adopts a methodology where a user makes parametric changes in the feature space to optimize performance and learning objectives using visual design representations on a graphical user interface.

2.2 Interpretability Approaches for Deep Generative Models. Interpretability broadly refers to the ability to present and explain understandably the cause-and-effect relationship between inputs and outputs of a machine learning model [41,42]. The need for interpretability arises when predictions and calculated metrics do not suffice for making informed decisions. For example, in the conceptual design, the real-world objectives are challenging to quantify, decision risks or stakes are high, and there is a tradeoff between objectives. The reasons behind analyzing interpretability are to explain complex deep learning models, enhance the fairness of model results, create white-box models, or test the robustness/sensitivity of predictions [43]. The scope of interpretability is broader than that of explainability, which refers to explanations of the internal logic and mechanisms of deep learning models. We focus on evaluating the interpretability of existing deep learning models, given their prevalence in the current design literature.

The type of interpretability evaluation depends on the machine learning task and whether real humans are involved in experiments. Application-grounded, human-grounded, and functionally grounded evaluation are three types of evaluation approaches [42]. Application-grounded evaluation conducts experiments with domain experts within a real-world application, e.g., public testing of self-driving cars. Such evaluations require substantial time and effort, but are sometimes necessary for real-world validation. Human-grounded evaluations involve experiments with lay humans in the real-world with simplified tasks. Human experiments test hypotheses by questioning human participants about the preference between different explanations or by identifying correct predictions from the presented input/explanation. Finally, functionally grounded evaluations use a formal definition of interpretability for automated interpretations. This approach requires defining quantitative metrics as proxies for interpretability. Functionally grounded metrics are applicable when working with models that have been validated, such as by human experiments. The survey by Linardatos et al. [43] summarizes various model-specific and model-agnostic methods for functionally grounded evaluation. Some model-specific methods analyze gradients of outputs with respect to inputs to find salient features, e.g., sensitivity analysis for DNNs [44], Deep learning important features (DeepLIFT) [45], and visual explanations for CNNs [46]. Some model-agnostic methods compute importance values for input features within predictions, e.g., local interpretable model-agnostic explanations (LIME) [47] and shapley additive explanations (SHAP) [48].

In this study, we use human-grounded evaluation of interpretability for interactive deep generative design because such methods have limited behavioral validation. On a related point, Sec. 2.3 points to conflicting findings on the effectiveness of the deep generative design.

2.3 Learning and Performance Outcomes of Deep Generative Design. The effectiveness of interactive generative design tools may depend on task complexity, usability, and users' expertise. Viros i Martin and Selva [49] compare two versions of an human–machine agent with a natural language interface varying in functionality and level of pro-activeness. An "assistant" version only answers technical questions from the designer (e.g., querying databases and doing data analysis), whereas the "peer" version provides recommendations to improve design solutions. They find that more interactions with the tool in both versions improve the design performance and learning. Recent research finds that the collaboration of a human and a computer agent significantly improves design performance compared to human-only or agent-only processes [14,19,50–52]. A body of research on hybrid human–machine teams [53,54] finds that low-performing players benefit from the decision support, but this support can be overly conservative for high-performing players. The cognitive agent boosts the performance of low-performing teams in a changing problem setting but hurts the performance of high-performing teams [40]. Also, the evidence in Ref. [15] highlights differences in learning between expert designers and novices.

Despite being a crucial part of design decision-making, designer learning has received little attention in evaluating deep generative models. Recent studies have proposed approaches to measuring knowledge. One general approach is to pose questions testing individuals' specific learning about the problem at hand, as demonstrated by Bang and Selva [30] for tradespace exploration. Related to this, IRT offers a consistent approach to estimating concept-specific ability from observations of test responses. Mathematically, IRT defines the functional relationship between the individual's ability/knowledge on a topic and the likelihood that they correctly answer questions on the same topic [25]. The simplest IRT model uses a scalar ability parameter and the binary response, related through a Sigmoid function. More complex IRT models have been applied for estimating multi-dimensional ability levels, which can be either independent of each other [26] or interconnected in a Bayesian network [27]. This work presents a new variation of IRT for quantifying learning from design space exploration.

3 Methodology

Interactive design space exploration implicitly supports learning and performance goals by allowing visualization, generation, and evaluation of alternative designs. An example of a learning goal is identifying the driving features that make up good designs such as Pareto-optimal designs. An example of a performance goal is maximizing one or more design objectives.

3.1 Implementation of the Interactive Deep Generative Design. We use a conditional variational autoencoder (C-VAE) to represent the relationships between designs, features, and objectives. Figures 1(*a*) and 1(*b*) present the network structure of C-VAE. Suppose a vector or matrix $\mathbf{x}$ represents a design. Features $\mathbf{z}_1$ are predefined, deterministic functions of design $\mathbf{x}$ mainly representing mechanical and geometric features of designs such as shape and size. Grayscale image I denotes a visual representation of design $\mathbf{x}$, with each pixel taking a value between [0,1].

Two sequential neural networks, encoder and decoder, operate on image I as part of the variational autoencoder. The encoder network $E:\{I,\mathbf{z}_1\}\rightarrow\{\mu_a,\sigma_a\}$ converts image I and predefined features $\mathbf{z}_1$ into mean μ_a and standard deviation σ_a vectors that have the same length as latent dimensions. A latent feature vector $\mathbf{z}_2$ is a sample from a normal distribution with the same mean and deviation vectors $N(\mu_a,\sigma_a)$. Furthermore, the decoder network $D:\{\mathbf{z}_1,\mathbf{z}_2\}\rightarrow\hat{I}$ transforms the predefined features $\mathbf{z}_1$ and the latent features $\mathbf{z}_2$ into a reconstructed image $\hat{I}$. Furthermore, separate neural networks post-process the C-VAE outcomes. First, the adaptation network $A:\hat{I}\rightarrow\hat{x}$ reformats the grayscale image $\hat{I}$ into a binary array of the same size as $\mathbf{x}$, resulting in a reconstructed design $\hat{\mathbf{x}}$. Second, the regression network $R:\{\mathbf{z}_1,\mathbf{z}_2\}\rightarrow\hat{\mathbf{y}}$ predicts the design objective values $\hat{\mathbf{y}}$ from features. The network structures in Fig. 1(*b*) include operators such as 2D convolution (Conv2D), 2D transposed convolution (Conv2DT), linear transformation (Dense), and activation functions such as rectified linear units and Sigmoid function (Sigm) [55]. The dense layer does not have an activation function.

We propose three modes of human–machine collaboration concerning the level of automation of search decisions a designer must take:

(1) (*Manual design synthesis*) A user defines design x with all its constituent parts;

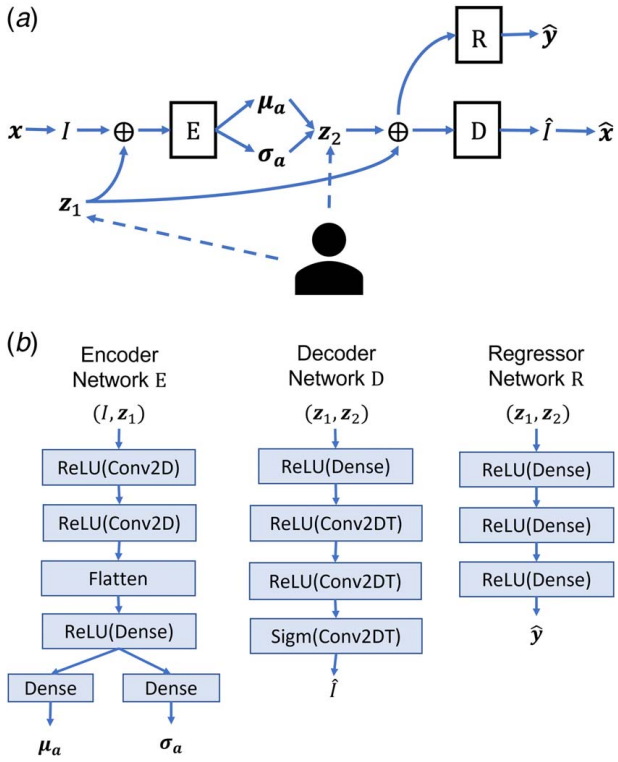

Fig. 1 (*a*) The structure of the conditional variational autoencoder. A user can directly manipulate feature inputs ($\mathbf{z}_1$, $\mathbf{z}_2$) on the decoder network D and (*b*) The structure of individual networks.

(2) (*Manual feature-based design synthesis*); and
(3) (*Semi-automated feature-based design synthesis*).

Sections 3.1.1, 3.1.2, and 3.1.3 present the operationalization of these three collaborative modes.

3.1.1 Manual Design Synthesis. The first collaborative mode involves a user manually creating a design $\mathbf{x}$ with all its constituting components and the C-VAE evaluating the design objectives $\hat{\mathbf{y}}$. A user only sees the design objective outputs $\hat{\mathbf{y}}$ and does not observe the intermediate latent features $\mathbf{z}_2$.

3.1.2 Manual Feature-Based Design Synthesis. In the second mode, a user manually selects features $\mathbf{z}_1$, $\mathbf{z}_2$ individually to generate designs with the decoder network *D*. Selecting feature values is done one feature at a time, either from predefined features or latent features. For every feature value adjustment, the C-VAE automatically generates a new design. A user can further decide whether to evaluate design objectives at any generated design or not. Suppose initial design $\mathbf{x}$ and its features $\mathbf{z}_1$ and $\mathbf{z}_2$ are given. A user makes $\Delta\mathbf{z}_2$ change to the latent features and evaluates the new design corresponding to the updated latent features $\mathbf{z}_2' = \mathbf{z}_2 + \Delta\mathbf{z}_2$. The output of this process is a newly reconstructed design $\mathbf{x}' = A(D(\mathbf{z}_1, \mathbf{z}_2'))$ and its design objective values $\mathbf{y}' = R(\mathbf{z}_1, \mathbf{z}_2')$.

3.1.3 Semi-Automated Feature-Based Design Synthesis. In the semi-automated mode, an optimization algorithm searches the design space or feature space in a semi-automated manner to maximize design objectives. A designer chooses tuning parameter(s) such as step size in this optimization. The step size is interpreted as a measure of "desired change with respect to selected design." The higher the step size, the more exploration with respect to the selected design, whereas a small step size exploits a region near that design. This approach allows users to retain a high level of control of exploration versus exploitation, using the agent for local search, even for exploration with a larger step. The C-VAE performs a fixed number of steps of the gradient descent algorithm in the feature space. A user selects a step size γ from a given range. A larger step size indicates more significant changes in the selected features. Suppose $\mathbf{x}_0$ is an initial design and $\mathbf{z}_{1,0}$ and $\mathbf{z}_{2,0}$ are its features. A gradient descent iteration updates the features by an amount proportional to the gradient of design objectives with respect to features. This process repeats for a fixed N iterations $i = 1, \ldots, N$:

$$\begin{aligned} \mathbf{z}_{1,i} &= \mathbf{z}_{1,i-1} - \gamma \nabla_{\mathbf{z}_1} R(\mathbf{z}_1, \mathbf{z}_2) \\ \mathbf{z}_{2,i} &= \mathbf{z}_{2,i-1} - \gamma \nabla_{\mathbf{z}_2} R(\mathbf{z}_1, \mathbf{z}_2) \end{aligned} \tag{1}$$

Here, $\nabla_{\mathbf{z}_1} R(\mathbf{z}_1, \mathbf{z}_2)$ and $\nabla_{\mathbf{z}_2} R(\mathbf{z}_1, \mathbf{z}_2)$ are the Jacobian matrices of the regression network with respect to features $\mathbf{z}_1$ and $\mathbf{z}_2$, respectively. Out of the N iterations, the user is shown only the results from the iteration with the largest weighted sum of design objectives. All design objectives are normalized between [0,1] with larger values preferred, and they have equal weights—which introduces a bias towards designs in the central region of the Pareto front.

$$\mathbf{z}_1^*, \mathbf{z}_2^* = \arg \max_{i=1,\ldots,N} \sum_j R_j(\mathbf{z}_{1,i}, \mathbf{z}_{2,i}) \tag{2}$$

where R_j is jth component of regression network output. Accordingly, the user observes the newly reconstructed design $\mathbf{x}^* = A(D(\mathbf{z}_1^*, \mathbf{z}_2^*))$ and corresponding design objectives values $\mathbf{y}^* = R(\mathbf{z}_1^*, \mathbf{z}_2^*)$.

3.2 Measures of Design Performance and Designer Learning

3.2.1 Multi-objective Performance Metrics. We use three established performance measures calculated based on the values of the design objectives $\hat{\mathbf{y}}$. First, *hypervolume improvement* is a measure commonly used in multi-objective optimization [56]. Given a set S of points (e.g., the output of a design search process), the hypervolume indicator of S is the area (for a 2D case) of the union of the region of the objective space dominated by each point in S and limited by a user-defined reference point. The reference point is at or near the anti-utopia point, i.e., the smallest objective value for each objective, assuming the problem requires objective maximization.

Second, a metric based on *credit assignment strategies* from multi-armed bandit theory evaluates designs more locally [57]. If an initial design $\mathbf{x}$ is modified to produce a new design $\mathbf{x}$', then the value of $\mathbf{x}$' is determined based on whether or not $\mathbf{x}$' dominates $\mathbf{x}$. If the new design dominates the initial one, i.e., if it is better than the initial design in all objectives, it receives a value of 1. Conversely, if the initial design dominates the new design, the new one receives a score of 0. If no design dominates, the new design receives a score of 0.5.

Third, the *distance to the utopia point* is the closest distance between the generated designs set S and the utopia point. In maximization, the utopia point has coordinates equal to the largest possible objective values, whereas in minimization, the utopia point has the smallest possible objective values as its coordinates.

3.2.2 Designer Learning: Feature-Specific Abilities. After exploring the design space, we implement a psychometric assessment approach to measure designer learning. This approach involves multiple-choice questions and an IRT model to estimate the feature-specific ability from an individual's responses. The feature-specific ability measures the degree to which a designer understands the effect of that feature on design objectives. These features are the same as the predefined and latent features in the C-VAE in Sec. 3.1. We use two types of questions to assess a person's knowledge: (i) design comparison and (ii) feature identification [30]. A *design comparison question* includes two given designs (say *A* and *B*) and requires a person to choose the design they think has a higher value for a given objective. For a given pair of designs (say A and B), a subject selects one of four choices: (i) "Option A," (ii) "Option B," (iii) "Minimal difference," and (iv) "Not sure." The "Not sure" option reduces the likelihood of a false-positive response.

Furthermore, a *feature identification question* tests a person's ability to correctly identify a particular feature's effect on a design objective in the context of adding the feature to a specific design. The question assumes that only the given feature changes value while other features are kept constant. In response to how the given objective changes, the person chooses four options: "Increases," "Decreases," "Minimal change," and "Not sure."

We use two methods to estimate designer learning based on the answers to these questions. The first learning metric is simply the *fraction of correct responses* to a question, which is the number of correct responses divided by the total number of available responses. In the second method, the IRT model represents the link between feature-specific abilities and the correctness of responses to individual questions. We use a Bayesian inference technique to estimate the posterior distributions of feature-specific abilities, conditional on the binary data about the correctness of the subjects' question-specific responses. Let $\boldsymbol{\theta}$ be the vector of feature-specific abilities, where variable θ_k can take any real value. Higher values of θ_k suggest more accurate knowledge of the effect of the kth feature component on different objectives. The probability that a response to question q is correct increases with the individual's abilities. More specifically, one question may test the knowledge of more than one feature but only the features relevant to the question are included in the model. That is, an individual's question-specific ability is $\theta_{q,\text{avg}} = \sum_k w_{k,q}\theta_k$, where $w_{k,q} = 1$, if the knowledge of feature z_k is relevant for question q, or $w_{k,q} = 0$ otherwise. We assume that the prior distribution on every model parameter is a standard normal distribution with zero mean and unit variance. The following function gives the likelihood of a correct response:

$$f_q(\theta, c_q, \alpha_q, \beta_q) = \frac{1}{c_q} + \left(1 - \frac{1}{c_q}\right) Sigm(\alpha_q(\theta_{q,\text{avg}} - \beta_q)) \quad (3)$$

where α_p is a question-specific discrimination (slope) parameter, β_q is the question-specific difficulty (threshold) parameter, c_q is the number of choices available in the multiple-choice question q, and Sigm is the Sigmoid function. Larger α_q indicates a stronger distinction between the ability required to answer correctly versus incorrectly. Larger β_q indicates a need for a higher average ability to answer correctly. The quantity $1/c_q$ in Eq. (3) represents the likelihood of guessing the right answer by chance alone, called a pseudo-guessing parameter. The further scaling by an amount $(1 - (1/c_q))$ ensures that the output probability is constrained between 0 and 1. In the limit of $\theta_{q,\text{avg}} >> \beta_q$, the sigmoid approaches 1 and so does the probability of correctly answering the question. Conversely, for $\theta_{q,\text{avg}} << \beta_q$, the sigmoid approaches 0 and the probability of a correct answer is equivalent to guessing. For $\theta_{q,\text{avg}} = \beta_q$, the model predicts a probability exactly halfway between pure guessing ($f_q(.) = 1/c_q$) and certainty ($f_q(.) = 1$). Note that the responses to different questions may be correlated if they test the knowledge of the same features. Parameter c_q is fixed (equal to 3 for three available options for a correct answer) but others are learned during model training.

3.3 Human Subject Experiment

3.3.1 Mechanical Metamaterial Design Problem. The experimental task involves the design of 2D mechanical metamaterials. A mechanical metamaterial is an artificially engineered structure of lattice units replicated in all directions. The lattice topology exhibits unique and tunable properties. Surjadi et al. [58] discuss unique properties of metamaterials and their applications in structural design and additive manufacturing.

We consider designs consisting of a unit cell structure repeating horizontally and vertically. The unit cell structure consists of multiple links joining nodes in a 3×3 grid in the XY plane. This defines a design space of 2^{36} possible designs, which is reduced to 2^{28} unique designs if we account for duplicates due to the replication of the unit cell structure in 2D space. Such design problem presents the right level of complexity for student subjects to develop problem understanding.

A metamaterial is evaluated using two design objectives: maximize *vertical stiffness* (which relates to strength) and minimize *volume fraction* (which relates to weight). The default model of choice for computing stiffness is a Hooke's law-based truss stiffness model (termed as the "truss model"), taken from Ref. [59, Ch.9]. The 2D metamaterial is a truss structure with each member assumed to only experience axial forces. The individual stiffness matrices for each member are determined by solving Hooke's law relationship. In cases where the truss model fails due to isolated members, a lower fidelity fiber stiffness model (called "the fiber model") is employed based on Cox [60]. The fiber model considers each member as a fiber and computes a length-normalized approximation of the effective stiffness.

The design problem also requires a *feasibility constraint* that metamaterials should satisfy: No two links in a unit cell should intersect, except at nodes; and a resulting metamaterial should be connected in the sense of a network graph, i.e., it should not have any disconnected subcomponents.

3.3.2 User Interface. Figure 2 presents the interactive design exploration platform used in the experiment. On the top panel, the tradespace plot shows a collection of existing designs (denoted by circles) and user-generated designs (indicated by triangles). A user may click on any design in the tradespace plot to visualize its details, including the unit cell structure on the bottom left panel (design visualization panel). On the bottom right panel, a user generates a new design through one of the three modes of interaction described in Sec. 3.1: the manual design synthesis (labeled "Change Design" in the figure), the manual feature-based design synthesis ("Change Feature"), and the semi-automated feature-based design synthesis ("Auto Feature Changes"). The manual design synthesis allows users to create a metamaterial design by specifying a unit cell structure. Upon clicking on the "Test metamaterial" button, the tradespace plot displayed the objective values of the new design and the design visualization panel shows the newly tested design. In the manual feature-based design synthesis mode (see Fig. 3), a user selects a change in individual features and considers the effect of the feature change on the selected design and its objectives. A newly generated design is updated in real-time on the design visualization panel whenever there is a change in feature values. The user must click on the "Test metamaterial" button to evaluate the generated design. In the semi-automated feature-based design synthesis mode, a user selects the maximum amount of change desired with respect to the initial design, and the C-VAE predicts the best possible design within that neighborhood of the initial design, according to Sec. 3.1.3. A newly generated design and the differences in features for the set change are visualized in real-time. Here again, the user has to signal intentionally, clicking on the "Test Metamaterial" button, if they want to evaluate a newly generated design.

3.3.3 Experiment Design. Table 2 presents the experiment protocol, including the order of the design synthesis tasks and learning tests. The experimental conditions vary in two independent variables: (i) the level of automation and (ii) the semanticity of features. The level of automation involves the interaction between a designer and the conditional variational autoencoder (C-VAE). That is, a subject completes one of the following tasks at any given time: the manual design synthesis (task 1), the manual feature-based design synthesis (task 2), and the semi-automated feature-based design synthesis (task 3), as described in Sec. 3.1. In each task, the user can only generate new designs using one functionality. Furthermore, the predefined features z_1 in the C-VAE have semantic meanings (*semantic features*), whereas the latent features z_2 are mathematical variables (*abstract features*). Table 3 provides a brief description for individual features. We select five semantic features for the mechanical metamaterial design problem: horizontal lines, vertical lines, diagonal lines, triangles, and three-star nodes. The other five abstract features are the outputs from the encoder

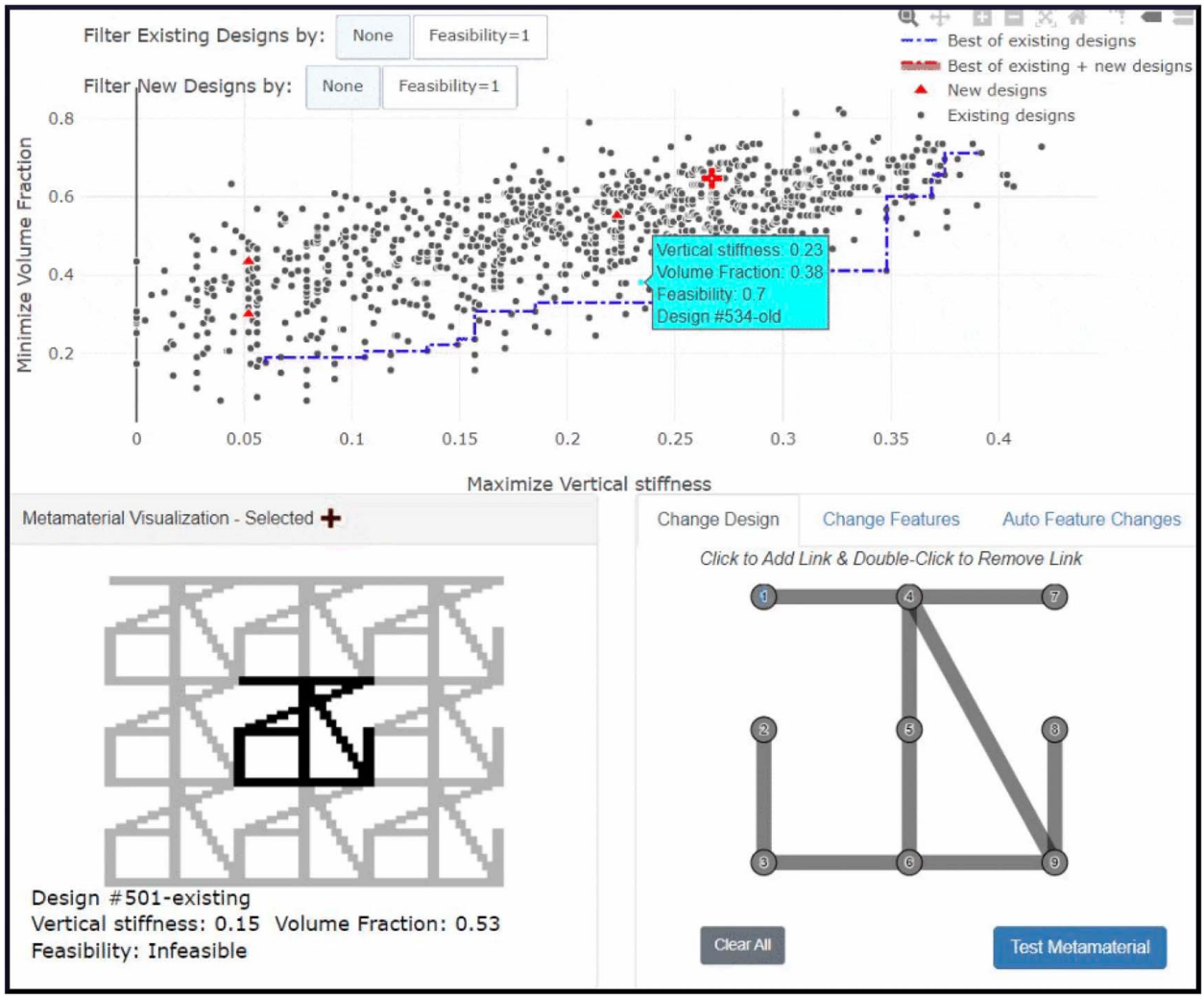

Fig. 2 A snapshot of the user interface showing interactive tradespace plot (top), visualization of selected metamaterial (bottom left), and the user-modified design (bottom right). The Pareto front in the tradespace plot is only for feasible designs. The picture shows infeasible designs but we have a filter button only to show feasible designs.

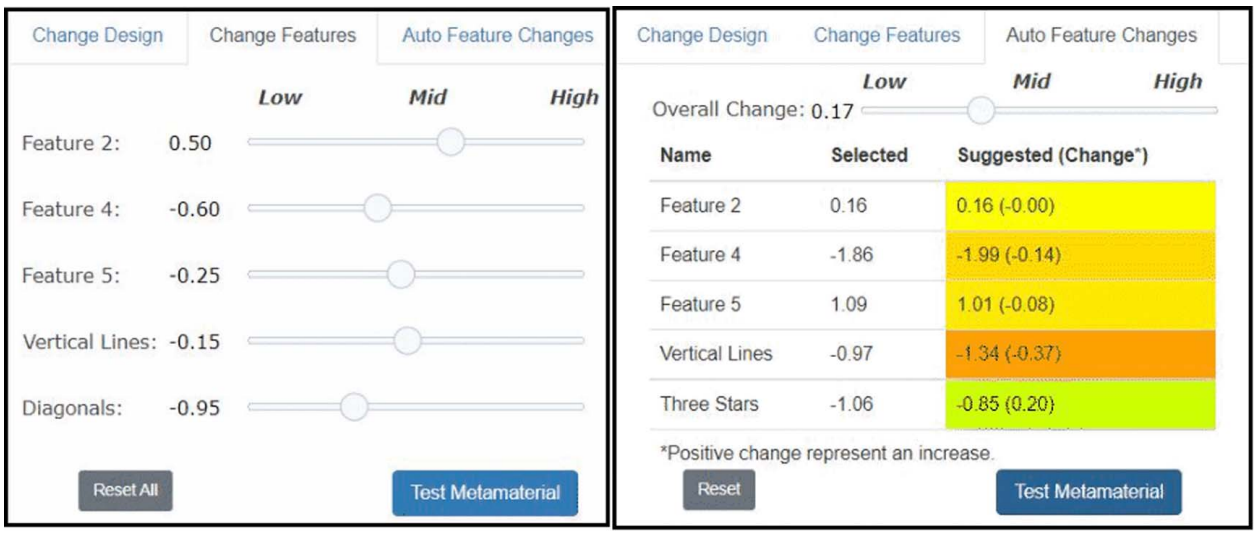

Fig. 3 Manual feature-based design synthesis (left) and semi-automated feature-based design synthesis (right); in both conditions, a generated design's real-time rendering is visible on screen, as depicted in the bottom left part of Fig. 2. But the objective values are shown only after a user has clicked the "Test Metamaterial" button.

Table 2 The order of design synthesis tasks and the learning tests in the experiment protocol

I	Design pretest	16 design comparison questions
II	Task 1 Design posttest	Manual design synthesis (8 min) 16 design comparison questions
III	Task 2 First feature test	Manual feature-based design synthesis (8 min) 20 feature identification questions
IV	Task 3 Second feature test	Semi-automated feature-based design synthesis (8 min) 20 feature identification questions

Note: The protocol also implements the reverse order (Part I, IV, III, and II) for approximately 20 of 42 subjects.

Table 3 The sensitivity of design objectives to the abstract and semantic features

Feature	Description	Volume fraction	Vertical stiffness
Feature 1	Derived from C-VAE	None	None
Feature 2	Derived from C-VAE	None	High
Feature 3	Derived from C-VAE	High	None
Feature 4	Derived from C-VAE	None	None
Feature 5	Derived from C-VAE	None	None
Horizontal lines	Number of horizontal links connecting two nodes	High	None
Vertical lines	Number of vertical links connecting two nodes	High	High
Diagonals	Number of inclined links connecting two nodes	High	High
Triangles	Number of three links (any orientation) connecting three nodes to each other	High	None
Three-stars	Number of three links connecting a single center node to three outer nodes separately	High	None

network, with the probability distribution approximately equal to the standard normal distribution for each one.

The experiment involves 42 junior-, senior-, and graduate-level students from engineering disciplines at Texas A&M University. Each subject completes the three tasks with different levels of automation. The order of three tasks, given by Table 2, is reversed for about half of the subjects pool so that no design synthesis task always follows the same task in both orders. This setup helps to counterbalance order effects. Task 1 with a pretest is always conducted at the start of the experiment. For every subject, a total of ten semantic and abstract features are randomly divided into two groups of five each, one for task 2 and the other for task 3. Because the features are randomly assigned to different tasks for each subject, all 10 features still appear in every task over the entire subject population. This within-subject design ensures that each subject completes all three tasks and sees five abstract features and five semantic features at some point between tasks 2 and 3. We can still partition the collected data into different levels of automation and types of features. The design allows us to study relative differences between different experimental conditions. The total experiment lasts about 45 min, and each subject receives a fixed payment of 20 USD at the end. The subjects must spend a minimum of 5 min on the instructions, which include textual and graphical details of the metamaterial problem and the user interface. The eight minutes of task duration provided extra time to familiarize themselves with the interface and was selected after pilot testing.

We administer four learning tests throughout the experiment, as shown in Table 2. Before any design synthesis task, part I includes a *design pretest* with 16 design comparison questions to test the prior knowledge of the subject about the mechanical metamaterial design problem. With the task ordering shown in Table 2, part II includes a manual design synthesis task and a *design posttest* with 16 design comparison questions to measure resultant learning. We do not repeat questions between design pretest and posttest to prevent the subjects from remembering answers. The questions in both tests still have a similar distribution of question complexity, as measured by the feature difference Δz_2 (see Fig. 4(*a*)). Parts III and IV, respectively, include the manual- and semi-automated feature-based design synthesis and first and second *feature tests*, which include 20 feature identification questions each. About half of all subjects complete the tasks in the reverse order of parts I, IV, III, and II to mitigate the impact of task order in the data.

3.4 Model Training. We train the conditional variational autoencoder on a dataset of 21,444 designs, which were generated from a greedy search using genetic algorithms [61]. The training data include 3×3 metamaterial lattice structures represented as 28×28 pixel grayscale images and 28-bit binary vectors, and the objectives vector for each design, made of vertical stiffness, volume fraction, and feasibility constraint. An image is generated from the binary vector representation of a metamaterial design. A pixel has a value of 1 if it falls on an active link or 0 otherwise. An image of an example unit lattice is highlighted in the bottom left part of Fig. 2. The image acts as an input I to the encoder network.

The loss function of the C-VAE comprises four terms. The reconstruction loss measures the difference between input designs and reconstructed designs. The Kullback–Leibler divergence (KLD) loss measures the difference in the posterior feature distribution and the standard normal distribution to reduce correlation among different features [62]. The KLD loss was weighted ten times the actual KLD loss. The regression loss compares the predicted and observed values of the objectives. Finally, a correlation loss term maximizes the correlation of feature 2 and feature 3, respectively, with vertical stiffness and volume fraction. This loss artificially introduces strong sensitivity between the design objectives and select abstract features to help with the assessment. The hypothesis is that if those features are strongly correlated with the design objectives, the user should be able to learn those features more correctly. Table 3 differentiates high versus low sensitivity features in the trained model based on the total-effect Sobol index. The semantic features are converted from integer numbers to normalized float values by centering with sample mean and standardizing with sample deviation. These values feed into the C-VAE as vector $\mathbf{z}_1$. We ran the Adam-based stochastic optimization algorithm for 50 epochs with a batch size of 128.

4 Results

The results present descriptive statistics and the posterior estimates from the item response theory model. The results use the aggregated data of both experimental task orders, as described in Table 2. We highlight the order-specific differences whenever relevant. The rest of the section is divided into the designer learning and performance outcomes.

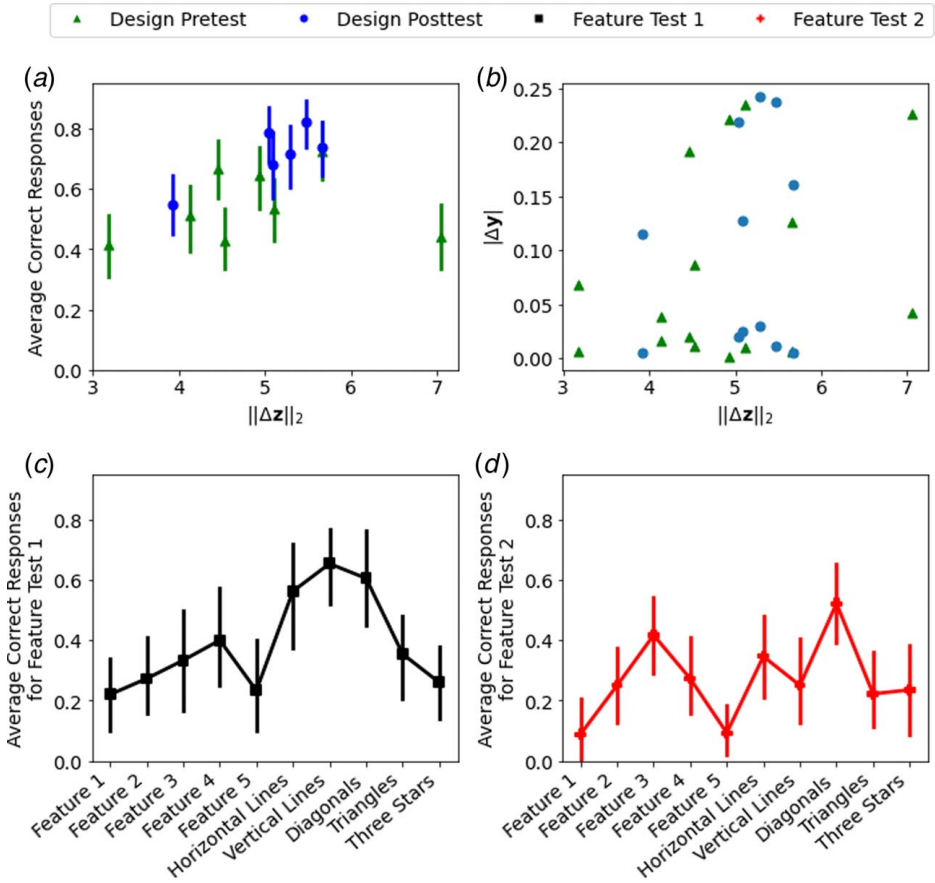

Fig. 4 (*a*) The accuracy of subjects' responses is proportional to the feature distance between design pairs in design posttest. This correlation is not significant in the design pretest, (*b*) intentionally, the correlation of the feature distance with the distance in the objectives space is insignificant for design pairs, (c) the correctness of responses is on average high for semantic features such as "horizontal lines," "vertical lines," and "diagonals" in task 2 (feature test 1), and (*d*) task 3 (feature test 2) produces lower accuracy of responses than task 2, especially for semantic features like "horizontal lines" and "vertical lines."

4.1 Designer Learning Outcomes. The experimental task increases the subjects' ability to differentiate designs based on design objectives in the design comparison questions asked. Figure 4(*a*) presents the average correctness of responses in the learning tests. We observe that the average correctness is higher in the design posttest than in the pretest (relative t-statistic = 4.12, two-sided p-value < 0.001, Cohen's $d = 0.78$). This difference is statistically significant irrespective of the task order. The average correctness of design posttest is higher during the forward order parts I, II, III, IV (relative t-statistic = 2.47, two-sided p-value = 0.018, Cohen's $d = 0.73$) and during the reverse task order (t-statistic = 2.58, two-sided p-value = 0.014, Cohen's $d = 0.84$). Furthermore, in the design posttest, the average correctness of response increases in proportion to the feature distance between the designs being compared (slope = 0.13 ($\pm$0.035), intercept = 0.05 ($\pm$0.18), r-value = 0.16, and a one-sided p-value < 0.001). Here, the feature distance $\|\Delta z\|_2$ defines the MSE distance between the features of two designs in a test question. The more different the two designs are, the easier it is expected for the subjects to predict the influence of features on a given objective correctly. Note that the correlation between the feature distance and the distance in the objective space ($|\Delta y|$) is statistically insignificant, according to the results in Fig. 4(*b*). Thus, $|\Delta y|$ is not expected to confound the effect of $\|\Delta z\|_2$ on the average correctness of responses.

Overall, the subjects most accurately learn the effect strength and direction for features with inherently significant and positive effects on the design objectives. The semanticity further improves the accuracy of responses. According to Fig. 4(*c*), high sensitivity semantic features such as "horizontal lines," "vertical lines," and "diagonals" collectively have higher mean correctness of response than the other semantic features (relative t-statistic = 4.5, two-sided p-value < 0.001, Cohen's $d = 0.69$), especially for task 2. Furthermore, these three semantic features have better correctness of response than the high sensitivity abstract features, i.e., "feature 2" and "feature 3" (relative t-statistic = 3.40, two-sided p-value < 0.001, Cohen's $d = 0.54$).

Task 3 produces a lower accuracy of responses compared to task 2. The "horizontal lines" feature has lower average correct responses in feature test 2 than feature test 1, according to Figs. 4(*c*) and 4(*d*) (t-statistic = 1.74, two-sided p-value = 0.08, Cohen's $d = 0.36$). A similar effect is observed for the "vertical lines" feature (t-statistic = 2.24, two-sided p-value = 0.03, Cohen's $d = 0.40$). The differences in other features are not statistically significant for the average correctness metric.

For a consistent comparison of feature-specific knowledge, Fig. 5 presents feature-specific abilities estimated using the item response theory model. A boxplot shows the first, second, and third quartiles as horizontal lines and the sample mean as a filled marker. The hollow circles outside a boxplot are sample outliers. Between design pretest and posttest, we observe that the subjects exhibit increased understanding of the effects of semantic features, except for the "diagonals" feature. In the design pretest, the subjects, on average, have a poor understanding of the influence of horizontal lines—*horizontal* lines do not influence the *vertical* stiffness. In the design posttest questionnaire, the most considerable estimated ability is for vertical lines.

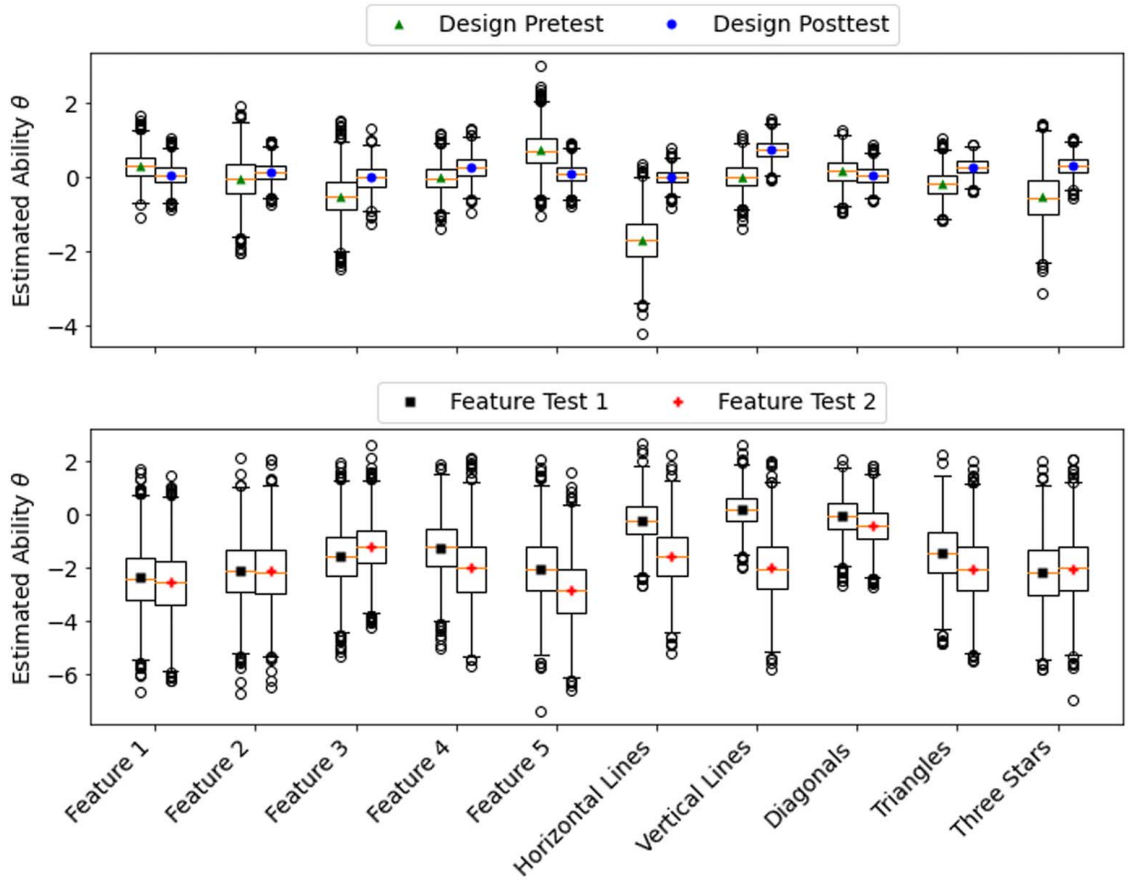

Fig. 5 The posterior samples of feature-specific abilities θ show that the subjects' knowledge about the semantic features improves in the design posttest compared to the design pretest, except for the "diagonals" feature. In the feature tests, the subjects, on average, appear to show higher ability levels in feature test 1 than in feature test 2, especially for "horizontal lines," "vertical lines," and "diagonals."

Figure 5 further shows the differences in the feature-based abilities measured from the feature tests. We observe that the estimated abilities for semantic features, such as horizontal, vertical, and diagonal lines, are higher than feature 2 and feature 3 combined (t-statistic = 71.60, p-value = 0.001, Cohen's d = 2.23) in feature test 1. And the subjects perform worse on the knowledge of horizontal lines and vertical lines in feature test 2 (task 3) than in feature test 1 (task 2). These results are consistent with the descriptive results presented in Fig. 4.

4.2 Design Performance Outcomes. The degree of performance improvement compared to the initial Pareto front varies across conditions. When comparing the overall performance based on all generated designs, the manual design synthesis (task 1) provides better mean performance than the other two conditions. Figure 6 presents the distribution of 1000 bootstrapped means for various performance measures. Bootstrapping allows hypothesis testing by resampling multiple sample sets from the experimental data [63]. Hypervolume improvement measures the improvement in the final generated Pareto front relative to the initial Pareto front. The higher the hypervolume improvement, the better. Since all design objectives are normalized between [0, 1], the hypervolume of 1 represents the utopia point. In Fig. 6(*a*), the mean hypervolume improvement is larger in task 1 than task 3 (t-statistic = 2.78, p-value = 0.008, Cohen's d = 0.43). At the same time, the smallest distance to utopia measures the distance between the final generated Pareto front and the utopia point ([1,0]). The smaller the distance, the better the performance. This metric is smaller in task 1 than in task 3 (t-statistic = 3.19, p-value = 0.003, Cohen's d = 0.45), as given in Fig. 6(*b*).

Task 3 performs better at the level of individual-generated designs when compared to task 1. The local dominance metric measures the improvement in each generated design relative to the initial design that it modifies. From Fig. 6(*c*), about 20% of generated designs in task 3 dominate their respective initial designs, compared to about 10% in task 1. The difference in the number of dominant generated designs compared to dominated generated designs in task 3 is large and statistically significant (t-statistic = 3.11, p-value = 0.004, Cohen's d = 0.74). However, a generated design in task 1 is likely to be twice as close to the utopia point as a generated design in task 3 if we kept the initial design the same (t-statistic = 6.08, p-value < 0.001, Cohen's d = 0.71), according to Fig. 6(*d*).

Among the semantic features, the changes made in the number of horizontal and vertical lines have a large, statistically significant correlation with the corresponding changes in overall hypervolume, as given in Table 4. Similarly, a significant correlation is observed for "Feature 4." Since the objectives are negligibly sensitive to "feature 4," this result could occur due to potential higher-order interaction effects. The subject population tested all features with similar frequency. Despite this effort, some features do not exhibit a high correlation with positive outcomes, possibly due to the relatively low influence of these features or the subjects' low feature-specific abilities.

5 Discussion

5.1 Positive Influence of Certain Semantic Features on Designer Learning. The results in Figs. 4 and 5 show that the effects of certain semantic features (e.g., horizontal and vertical

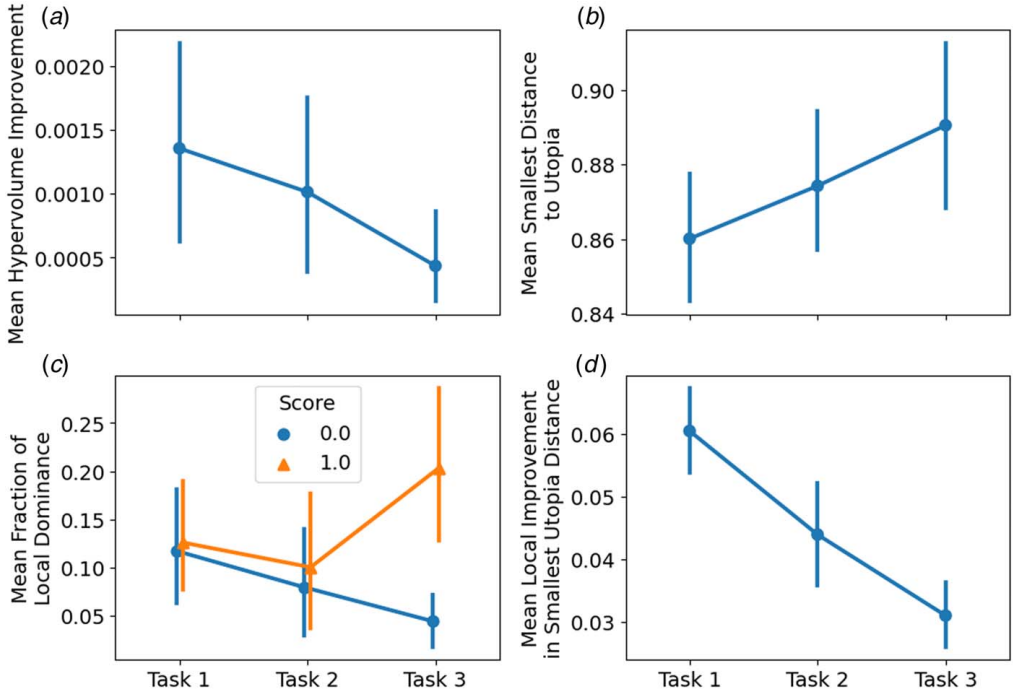

Fig. 6 Bootstrapped means of achieved design performance metrics: (*a*) the mean hypervolume improvement is biggest in task 1, (*b*) generated designs' smallest distance to the utopia point ([1,0]) is smallest on average in task 1, (*c*) the number of locally dominant designs, defined as the Pareto dominance of a generated design versus an initial design, is highest for task 3 with the average rate of 20%, and (*d*) however, the improvement in utopia distance for locally dominant generated designs in task 1 is two times as high as task 3

lines) are easier to learn for the subjects than other semantic and abstract features. These high-ability features are semantic and have considerable sensitivity due to the design objectives. As observed from Table 4, the subjects also learn to improve design performance by manipulating these features.

The constructs that may explain the above observations are the recognition of semantic features and intertwined effects of design performance and designer learning. First, the semantic nature can exploit an individual's dense prior knowledge [23], which the data-driven features lack, to explain feature behavior. The recognition from memory, i.e., recognition heuristic [64], places a higher value on identifiable features. The recognition heuristic might even be one of the first simple cues humans use to make decisions [65]. The recognition and simplicity could be a differentiating factor between single-link features (such as horizontal lines, vertical lines, and diagonals) and multi-link features (such as triangles and three-stars). Furthermore, retrospective learning triggers involve the need to learn from successful and failed designs [9]. The high sensitivity of certain semantic features likely allows the subjects to observe significant variations in objective values, thus triggering feature-specific learning for certain features. On the other hand, the low feature sensitivity does not provide a clear association between successful or failed designs and the changes in respective features.

Table 4 The correlation between the change in feature value and the change in overall hypervolume

Feature	Effort (feature changes)	Pearson's r
Feature 1	215	0.014
Feature 2	229	0.057
Feature 3	187	0.116
Feature 4	226	−0.189[a]
Feature 5	150	−0.182
Horizontal lines	154	−0.295[a]
Vertical lines	156	0.625[a]
Diagonals	161	−0.114
Triangles	186	0.038
Three stars	150	−0.0366

[a]Correlation coefficients are statistically significant with two-sided p-value < 0.005.

5.2 Mixed Influence of High Automation on Design Performance. From Fig. 6, we observe that higher automation in task 3 improves the local dominance of a newly generated design compared to an initial design. However, the local improvement in such a generated design could be half of that of a manually synthesized design, as seen from Fig. 6. This local dominance also does not necessarily translate into more significant hypervolume improvement. Potential explanations for this result could be related to (i) the low diversity of user-selected initial designs, (ii) the low amount of user-selected change (step size γ in Eq. (1)) in the initial design, (iii) the fixed number of iterations (≈50) used in the gradient descent algorithm, (iv) non-convex objective function, or (iv) the cognitive load in understanding model output.

Higher autonomy offers users more freedom in testing custom features and creating new designs. The subjects develop metamaterial designs based on the limited number of features in the experiment. Allowing users to define and test their features could facilitate learning [17]. Also, the cognitive load involved in parsing the automated suggestions should be a concern. A large amount of information on the user interface may complicate the comprehension and trustworthiness of results and could likely reduce the design performance of hybrid human–machine teams [40]. In task 3, the subjects view suggestions for five features simultaneously. However, in parametric design activities, a designer commonly evaluates one design variable at a time [66]. Besides the interpretability of information, human decision-makers are likely only to consider a single automated suggestion from a machine learning model at a time [67].

Also, higher model accuracy and complexity of design representation may influence the results. For example, more advanced deep

learning models could increase the design performance in task 3, but whether that improves designer learning is not guaranteed. The ease of use and complexity of design representation may influence how often the model gets used and, thus, designer learning. The semi-automated design synthesis may become more desirable for a complicated representation where manual design synthesis is not feasible.

5.3 Implications for Engineering Design Decision Support. Human–machine collaborative design requires that the human designer comprehend and trust the model results. Design decisions depend on the accurate understanding of model inputs, outputs, and the causal effects of the latent features. We observe that semantic nature and their effect size on design objectives influence the causal understanding of features. Semanticity can form a basis for human–machine collaboration and would scale up to more complex problems as long as meaningful semantic design features can be defined. While the method remains applicable, model training may require a larger amount of data for more complex problems. Accordingly, future design assistants should explicitly describe the underlying features in a roughly linguistic sort and clarify their effects on design objectives. The knowledge representation hypothesis by Brian Smith [68] similarly states that any mechanically intelligent system should embody a semantic representation of knowledge. The emerging approaches to achieving interpretability, such as transparency and post hoc explanations [41], offer additional ways to improve the designer's causal understanding.

The findings also highlight the role of designer learning in DSE and its effect on performance. Restricted learning due to a higher level of automation, cognitive overload from model outputs, or abstract features reduce the potential for higher design performance. Even though optimization using deep generative design can provide incremental improvements, global performance improvement is also a function of designer learning, especially in the human–machine collaborative setting.

5.4 Limitations and Future Directions. More validation with different deep learning methods, subject populations, and design problems is necessary to generalize the findings. This paper uses deep learning and the conditional variational encoder to focus on generative design. Some alternatives are evolutionary computation, adaptive or component-specific step size algorithms, or more advanced neural network architectures. Future work can evaluate such options in a human–machine collaborative setting. Additionally, we do not compare or test interpretability tools such as saliency maps, feature importance graphs, partial dependence plots, or specificity versus coverage plots. On the upside, the graphical interface and item response theory model provide a unique way to evaluate different alternatives in the future.

In data collection, the subjects did not have detailed domain knowledge and learned about mechanical metamaterials during the experimental task. While their engineering education forms a basis for their decisions, the lack of domain knowledge can drive their focus on certain semantic features. A more complicated design problem will need subjects with significant expertise to have external validity. However, laboratory experiments are still scalable to more complex problems. Recent research suggests that the representativeness in lab experiments depends not on matching subjects, tasks, and context separately, but rather on the behavior that emerges from the interplay of these three dimensions [69]. Moreover, the prevalence of open-source tools makes it easier to design user interfaces (oTree and MATLAB) and recruit lay subjects (Amazon Mechanical Turks). Future work can still validate the findings by comparing the patterns of designers' learning and performance outcomes between novices and experts. As with any experimental study, one needs to perform context-specific validation using more experiments when applying the insights to new settings.

6 Conclusion

The rise of deep learning applications in human–machine collaborative design necessitates the analysis of model interpretability, mainly to satisfy designer learning and performance goals. This paper facilitates such analysis by combining an interactive deep generative tool, human subject experiments, and a learning assessment based on item response theory. The findings provide essential mathematical tools and behavior insights for future design assistants. The subjects in our experiment appear to understand the sensitivity better for certain semantic features than abstract features. Cognitive factors such as cognitive load and semantic features are essential in mediating the overall design performance. If the findings hold, future interactive deep generative design platforms should emphasize discovering influential features and explaining them in the context of the problem definition. Interpretability measures would help maximize learning outcomes and performance while enlisting computational intelligence for design exploration.

Acknowledgment

The authors gratefully acknowledge the financial support from the US National Science Foundation (NSF) CMMI through Grant # 1907541.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Yonekura, K., and Suzuki, K., 2021, "Data-Driven Design Exploration Method Using Conditional Variational Autoencoder for Airfoil Design," Struct. Multidiscipl. Optim., **64**(2), pp. 613–624.

[2] Raina, A., Puentes, L., Cagan, J., and McComb, C., 2021, "Goal-Directed Design Agents: Integrating Visual Imitation With One-Step Lookahead Optimization for Generative Design," ASME J. Mech. Des., **143**(12), p. 124501.

[3] Valdez, S., Seepersad, C., and Kambampati, S., 2021, "A Framework for Interactive Structural Design Exploration," Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3B: 47th Design Automation Conference (DAC), Virtual, Online, Aug. 17–19, ASME, p. V03BT03A006.

[4] Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H., and Kang, N., 2021, "Integrating Deep Learning Into CAD/CAE System: Generative Design and Evaluation of 3d Conceptual Wheel," Struct. Multidiscipl. Optim., **64**(1), pp. 2725–2747.

[5] Wang, L., Chan, Y.-C., Ahmed, F., Liu, Z., Zhu, P., and Chen, W., 2020, "Deep Generative Modeling for Mechanistic-Based Learning and Design of Metamaterial Systems," Comput. Methods Appl. Mech. Eng., **372**, p. 113377.

[6] Chen, W., Fuge, M., and Chazan, J., 2017, "Design Manifolds Capture the Intrinsic Complexity and Dimension of Design Spaces," J. Mech. Des., **139**(5), p. 051102.

[7] Chen, W., and Ahmed, F., 2021, "Mo-padgan: Reparameterizing Engineering Designs for Augmented Multi-objective Optimization," Appl. Soft Comput., **113**(Part A), p. 107909.

[8] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," ASME J. Mech. Des., **141**(11), p. 111405.

[9] Sim, S. K., and Duffy, A. H., 1998, "A Foundation for Machine Learning in Design," AI EDAM-Artif. Intell. Eng. Des. Anal. Manuf., **12**(2), pp. 193–209.

[10] Grecu, D. L., and Brown, D. C., 1998, "Dimensions of Machine Learning in Design," AI EDAM, **12**(2), pp. 117–121.

[11] Hazelrigg, G. A., 2012, *Fundamentals of Decision Making for Engineering Design and Systems Engineering*, George A. Hazelrigg.

[12] Fillingim, K. B., Nwaeri, R. O., Borja, F., Fu, K., and Paredis, C. J., 2020, "Design Heuristics: Extraction and Classification Methods With Jet Propulsion Laboratory's Architecture Team," ASME J. Mech. Des., **142**(8), p. 081101.

[13] Suresh Kumar, R., Srivatsa, S., Silberstein, M., and Selva, D., 2021, "Leveraging Design Heuristics for Multi-objective Metamaterial Design Optimization," Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Volume 3B: 47th Design Automation Conference (DAC), Virtual, Online, Aug. 17–19, ASME, p. V03BT03A032.
[14] Law, M. V., Dhawan, N., Bang, H., Yoon, S.-Y., Selva, D., Hoffman, G., and Gero, J. S., 2019, "Side-by-Side Humanâ Computer Design Using a Tangible User Interface," *Design Computing and Cognition '18*, Springer Cham, Switzerland AG, pp. 155–173.
[15] Viros Martin, A., and Selva, D., 2019, "From Design Assistants to Design Peers: Turning Daphne Into an Ai Companion for Mission Designers," AIAA Scitech 2019 Forum, San Diego, CA, Jan. 7–11, p. 0402.
[16] Zhang, X. L., Simpson, T., Frecker, M., and Lesieutre, G., 2012, "Supporting Knowledge Exploration and Discovery in Multi-dimensional Data With Interactive Multiscale Visualisation," J. Eng. Des., **23**(1), pp. 23–47.
[17] Bang, H., Shi, Y. L. Z., Hoffman, G., Yoon, S.-Y., Selva, D., and Gero, J. S., 2019, "Exploring the Feature Space to Aid Learning in Design Space Exploration," *Design Computing and Cognition '18*, Springer Cham, Switzerland AG, pp. 195–212.
[18] Burnap, A., Hauser, J. R., and Timoshenko, A., 2019, "Design and Evaluation of Product Aesthetics: A Human–Machine Hybrid Approach," SSRN 3421771.
[19] Martin, A. V. I., and Selva, D., 2020, "Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions," IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens., **13**, pp. 30–48.
[20] Martin, A. V., and Selva, D., "Explanation Approaches for the Daphne Virtual Assistant," AIAA Scitech 2020 Forum, Orlando, FL, Jan. 6–10.
[21] Parasuraman, R., Sheridan, T. B., and Wickens, C. D., 2000, "A Model for Types and Levels of Human Interaction With Automation," IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum., **30**(3), pp. 286–297.
[22] Salomons, O. W., van Houten, F. J., and Kals, H., 1993, "Review of Research in Feature-Based Design," J. Manuf. Syst., **12**(2), pp. 113–132.
[23] Simon, H. A., 1981, *The Sciences of the Artificial*, MIT Press, Cambridge, MA.
[24] Sohn, K., Lee, H., and Yan, X., 2015, "Learning Structured Output Representation Using Deep Conditional Generative Models," Advances in Neural Information Processing Systems 28, Montreal, Canada, Dec. 7–12.
[25] Hambleton, R. K., Swaminathan, H., and Rogers, H. J., 1991, *Fundamentals of Item Response Theory*, Vol. 2, Sage Publishing, Thousand Oaks, CA.
[26] Hartig, J., and Höhler, J., 2009, "Multidimensional IRT Models for the Assessment of Competencies," Stud. Educ. Eval., **35**(2–3), pp. 57–63.
[27] Hans, A., Chaudhari, A. M., Bilionis, I., and Panchal, J. H., 2020, "Quantifying Individuals' Theory-Based Knowledge Using Probabilistic Causal Graphs: A Bayesian Hierarchical Approach," Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3: 17th International Conference on Design Education (DEC), Virtual, Online, Aug. 17–19, ASME, p. V003T03A014.
[28] Sim, S. K., and Duffy, A. H., 2003, "Towards an Ontology of Generic Engineering Design Activities," Res. Eng. Des., **14**(4), pp. 200–223.
[29] Sim, S. K., and Duffy, A. H., 2004, "Evolving a Model of Learning in Design," Res. Eng. Des., **15**(1), pp. 40–61.
[30] Bang, H., and Selva, D., 2020, "Measuring Human Learning in Design Space Exploration to Assess Effectiveness of Knowledge Discovery Tools," Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 8: 32nd International Conference on Design Theory and Methodology (DTM), Virtual, Online, Aug. 17–19, ASME, p. V008T08A017.
[31] Ross, A., Chen, N., Hang, E. Z., Glassman, E. L., and Doshi-Velez, F., 2021, "Evaluating the Interpretability of Generative Models by Interactive Reconstruction," CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, May, pp. 1–15.
[32] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2022, "Deep Generative Models in Engineering Design: A Review," ASME J. Mech. Des., **144**(7), p. 071704.
[33] An, J., and Cho, S., 2015, "Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability," Special Lecture IE, **2**(1), pp. 1–18.
[34] Khan, S., Gunpinar, E., and Sener, B., 2019, "Genyacht: An Interactive Generative Design System for Computer-Aided Yacht Hull Design," Ocean Eng., **191**, p. 106462.
[35] Bang, H., and Selva, D., 2016, "ifeed: Interactive Feature Extraction for Engineering Design," Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 7: 28th International Conference on Design Theory and Methodology, Charlotte, NC, Aug. 21–24, p. V007T06A037.
[36] Schulz, A., Xu, J., Zhu, B., Zheng, C., Grinspun, E., and Matusik, W., 2017, "Interactive Design Space Exploration and Optimization for CAD Models," ACM Trans. Graph., **36**(4), pp. 1–14.
[37] Simpson, T. W., Carlsen, D., Malone, M., and Kollat, J., 2011, "Trade Space Exploration: Assessing the Benefits of Putting Designers "Back-in-the-Loop" during Engineering Optimization," *Human-in-the-Loop Simulations*, L. Rothrock and S. Narayanan, eds., Springer, London, pp. 131–152.
[38] Raina, A., McComb, C., and Cagan, J., 2019, "Learning to Design From Humans: Imitating Human Designers Through Deep Learning," ASME J. Mech. Des., **141**(11), p. 111102.
[39] Raina, A., Cagan, J., and McComb, C., 2022, "Design Strategy Network: A Deep Hierarchical Framework to Represent Generative Design Strategies in Complex Action Spaces," ASME J. Mech. Des., **144**(2), p. 021404.
[40] Zhang, G., Raina, A., Cagan, J., and McComb, C., 2021, "A Cautionary Tale About the Impact of AI on Human Design Teams," Des. Stud., **72**, p. 100990.
[41] Lipton, Z. C., 2018, "The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery," Queue, **16**(3), pp. 31–57.
[42] Doshi-Velez, F., and Kim, B., 2017, "Towards a Rigorous Science of Interpretable Machine Learning." Preprint arXiv:1702.08608.
[43] Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S., 2020, "Explainable AI: A Review of Machine Learning Interpretability Methods," Entropy, **23**(1), p. 18.
[44] Simonyan, K., Vedaldi, A., and Zisserman, A., 2013, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." Preprint arXiv:1312.6034.
[45] Shrikumar, A., Greenside, P., and Kundaje, A., 2017, "Learning Important Features Through Propagating Activation Differences," International Conference on Machine Learning [PMLR, **70**, 3145–3153 (2017)].
[46] Zeiler, M. D., and Fergus, R., 2014, "Visualizing and Understanding Convolutional Networks," 13th European Conference on Computer Vision, Zurich, Switzerland, Sept. 6–12, Springer, pp. 818–833.
[47] Ribeiro, M. T., Singh, S., and Guestrin, C., 2016, "'Why Should I Trust You?' Explaining the Predictions of Any Classifier," KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, August, pp. 1135–1144.
[48] Lundberg, S. M., and Lee, S.-I., 2017, "A Unified Approach to Interpreting Model Predictions," NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, December.
[49] Viros i Martin, A., Selva, D., and Gero, J. S., 2022, *Design Computing and Cognition'20*, Springer Cham, Switzerland AG, pp. 655–665.
[50] Gyory, J. T., Soria Zurita, N. F., Martin, J., Balon, C., McComb, C., Kotovsky, K., and Cagan, J., 2022, "Human Versus Artificial Intelligence: A Data-Driven Approach to Real-Time Process Management During Complex Engineering Design," ASME J. Mech. Des., **144**(2), p. 021405 .
[51] Chong, L., Zhang, G., Goucher-Lambert, K., Kotovsky, K., and Cagan, J., 2022, "Human Confidence in Artificial Intelligence and in Themselves: The Evolution and Impact of Confidence on Adoption of AI Advice," Comput. Hum. Behav., **127**, p. 107018.
[52] Song, B., Soria Zurita, N. F., Nolte, H., Singh, H., Cagan, J., and McComb, C., 2022, "When Faced With Increasing Complexity: The Effectiveness of Artificial Intelligence Assistance for Drone Design," ASME J. Mech. Des., **144**(2), p. 021701.
[53] Bayrak, A. E., and Sha, Z., 2021, "Integrating Sequence Learning and Game Theory to Predict Design Decisions Under Competition," ASME J. Mech. Des., **143**(5), p. 051401.
[54] Song, B., Zurita, N. F. S., Zhang, G., Stump, G., Balon, C., Miller, S. W., Yukish, M., Cagan, J., and McComb, C., 2020, "Toward Hybrid Teams: A Platform to Understand Human–Computer Collaboration During the Design of Complex Engineered Systems," Proceedings of the Design Society: DESIGN Conference, Cavtat, Croatia, May, Vol. 1, pp. 1551–1560.
[55] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., and Killeen, T., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 33rd Conference on Neural Information Processing Systems, Vancouver, Canada, Dec. 8–14, Curran Associates Inc., pp. 8024–8035.
[56] Biscani, F., and Izzo, D., 2020, "A Parallel Global Multiobjective Framework for Optimization: Pagmo," J. Open Source Softw., **5**(53), p. 2338.
[57] Hitomi, N., and Selva, D., 2016, "A Classification and Comparison of Credit Assignment Strategies in Multiobjective Adaptive Operator Selection," IEEE Trans. Evol. Comput., **21**(2), pp. 294–314.
[58] Surjadi, J. U., Gao, L., Du, H., Li, X., Xiong, X., Fang, N. X., and Lu, Y., 2019, "Mechanical Metamaterials and Their Engineering Applications," Adv. Eng. Mater., **21**(3), p. 1800864.
[59] Jacob, F., and Ted, B., 2007, *A First Course in Finite Elements*, Wiley, West Sussex.
[60] Cox, H., 1952, "The Elasticity and Strength of Paper and Other Fibrous Materials," Br.J. Appl. Phys., **3**(3), p. 72.
[61] Chaudhari, A. M., Suresh Kumar, R., and Selva, D., 2021, "Supporting Designer Learning and Performance in Design Space Exploration: A Goal-Setting Approach," Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6: 33rd International Conference on Design Theory and Methodology (DTM), Virtual, Online, Aug. 17–19, American Society of Mechanical Engineers, p. V006T06A058.
[62] Prokhorov, V., Shareghi, E., Li, Y., Pilehvar, M. T., and Collier, N., 2019, "On the Importance of the Kullback–Leibler Divergence Term in Variational Autoencoders for Text Generation." Preprint arXiv:1909.13668.
[63] Kulesa, A., Krzywinski, M., Blainey, P., and Altman, N., 2015, "Sampling Distributions and the Bootstrap," Nature Methods, **12**(6), pp. 477–478.
[64] Goldstein, D. G., and Gigerenzer, G., 2002, "Models of Ecological Rationality: The Recognition Heuristic," Psychol. Rev., **109**(1), p. 75.
[65] Gigerenzer, G., and Todd, P. M., 1999, *Simple Heuristics that Make Us Smart*, Oxford University Press, New York.
[66] Hirschi, N., and Frey, D., 2002, "Cognition and Complexity: An Experiment on the Effect of Coupling in Parameter Design," Res. Eng. Des., **13**(3), pp. 123–131.
[67] Bastani, H., Bastani, O., and Sinchaisri, W. P., 2021, "Improving Human Decision-Making With Machine Learning." Preprint arXiv:2108.08454.
[68] Smith, B., 1982, "Reflection and Semantics in a Procedural Language." Technical Report, TR-272, MIT Laboratory for Computer Science.
[69] Chaudhari, A. M., Gralla, E. L., Szajnfarber, Z., Grogan, P. T., and Panchal, J. H., 2020, "Designing Representative Model Worlds to Study Socio-Technical Phenomena: A Case Study of Communication Patterns in Engineering Systems Design," ASME J. Mech. Des., **142**(12), p. 121403.