

REDUCING EVALUATION COST FOR CIRCUIT SYNTHESIS USING ACTIVE LEARNING

Tinghao Guo*, Daniel R. Herber†, James T. Allison‡
Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
{guo32, herber1, jtalliso}@illinois.edu

ABSTRACT

In this article, an active learning strategy is introduced for reducing evaluation cost associated with system architecture design problems and is demonstrated using a circuit synthesis problem. While established circuit synthesis methods, such as efficient enumeration strategies and genetic algorithms (GAs), are available, evaluation of candidate architectures often requires computationally-expensive simulations, limiting the scale of solvable problems. Strategies are needed to explore architecture design spaces more efficiently, reducing the number of evaluations required to obtain good solutions. Active learning is a semi-supervised machine learning technique that constructs a predictive model. Here we use active learning to interactively query architecture data as a strategy to choose which candidate architectures to evaluate in a way that accelerates effective design search. Active learning is used to iteratively improve predictive model accuracy with strategically-selected training samples. The predictive model used here is an ensemble method, known as random forest. Several query strategies are compared. A circuit synthesis problem is used to test the active learning strategy; two complete data sets for this case study are available, aiding analysis. While active learning has been used for structured outputs, such as sequence labeling task, the interface between active learning and engineering design, particularly circuit synthesis, has not been well studied. The results indicate that active learning is a promising strategy in reducing the evaluation cost for the circuit synthesis problem, and provide insight into possible next

steps for this general solution approach.

1 INTRODUCTION

Synthesis of electronic circuits is a process of configuring an assembly of electronic components to achieve a desired circuit behavior. Synthesizing a circuit is a challenging task because it involves both circuit topology and sizing. Circuit topology is determined by component selection and connections between components. The sizing problem is to identify the circuit component parameter values that optimize system performance. A number of methodologies have been proposed for circuit synthesis. Domain knowledge is often used [1–3], but requires significant expertise to apply successfully, and has limited effectiveness for complicated circuits. Methods that require minimal initial design knowledge are desirable in that novel topologies can be generated without demanding human expertise requirements [1–5]. For example, evolutionary algorithms [2, 4–9] and simulated annealing [10] have been applied to the circuit synthesis problem. Grimbleby used genetic algorithms to synthesize novel and effective circuits that satisfy both frequency- and time-domain specifications [4]. Das and Vermuri developed an automated circuit synthesis framework for an *RLC* circuit design problem [5].

Despite these achievements, evolutionary-based approaches are limited because 1) a number of parameters need to be tuned, including crossover probability and initial population size; 2) poor quality of the initial population may result in the convergence and robustness issues [5]; 3) direct genotype representations are not often successful in making progress toward improved designs for larger-scale systems. Global solutions are

*Graduate student, address all correspondence to this author

†Postdoctoral research fellow

‡Assistant professor

not guaranteed with evolutionary algorithms. Indirect generative design representations, such as grammar-rule based approaches, may be used to map design representations to circuit topologies, but these approaches so far are not generalizable, as they can only be applied successfully when an existing generative design algorithm is congruent with the system topologies to be generated, or when expert intuition can be used to define the generation rules.

Enumeration-based synthesis methodologies generate and test all possible circuit topologies under certain specifications, so global optimality is guaranteed. Enumeration-based approaches have a long history and have been used widely in electronic circuits [11, 12], hybrid powertrains [13], gear trains [14], and enzyme network topologies [15]. Naïve enumeration is simple, but impractical when the number of possible candidates is significant. Recent advancements developed by Herber et al. in efficient system architecture enumeration theory and algorithms, based on perfect matching theory and intelligent brute-force search methods, have made practical the use of enumeration methods for generation of all unique and feasible topologies [16]. The majority of designs produced via naïve enumeration are either isomorphic or violate network structure constraints (NSCs), but the generation of these non-value-added designs is minimized using the aforementioned strategies, making possible the enumeration of topologies for architecture design problems that are much larger than previously thought.

Efficient enumeration has been applied to circuit synthesis [17], and was shown to identify not only the topologies found earlier using GAs, but a much richer set of design data, including many previously unknown, non-dominated designs [16]. This study supported fair comparisons between unique, feasible topologies by using rigorous evaluation of the candidates [16, 17]. A dynamic system model is automatically generated for each candidate circuit topology, and then is used in solving a component sizing optimization problem [17]. Efficient enumeration strategies can be extended to other system architecture design problems, such as active vehicle suspensions [18].

While these recent advances have improved solution capabilities for circuit synthesis, they are still limited in scale to moderately-sized component catalogs that define the architecture space of interest. Here we seek to use circuit synthesis design data (obtained via efficient enumeration) along with machine learning techniques to enable approximate topology optimization of larger-scale circuit synthesis problems. Consider two possible cases where complete enumeration and evaluation is impractical [19]. *Case 1*) All unique, feasible topologies can be enumerated in a practical amount of time, but evaluation (e.g., size optimization) is too computationally expensive to perform for all topologies. *Case 2*) a catalog is too large to enumerate all topologies in a practical amount of time. Here we focus on Case 1. Ongoing work addresses Case 2.

In the circuit synthesis problem, for the catalogs considered here, both enumeration and evaluation can be performed.

Circuit evaluation involves straightforward generation of single-input single-output transfer functions [17], and then finding the solution of a nonlinear fitting problem. Having a complete set of data (topologies and evaluation metrics) allows us to study various limited-sampling strategies, and compare against known globally optimal designs. A small extension of this test problem (e.g., inclusion of nonlinear circuit elements) would increase evaluation expense, resulting in a Case 1 problem. Further extension involving a larger catalog would result in a Case 2 problem. Here we present an initial analysis of methods appropriate for the approximate solution of Case 1 problems. Methods for Case 2 problems would have different requirements and are outside the scope of this article.

Here machine learning is applied by constructing a predictive model that approximates the mapping from *topological design descriptions to performance metrics (real-valued outputs)*. This model is then used to identify potentially desirable topologies to evaluate. This new data is then used to enhance the predictive model. This iterative strategy is known as active learning, which is a semi-supervised machine learning technique that aims to achieve good accuracy with fewer training samples by interactively sampling the data from which it learns [20]. In this situation, the dataset contains a number of unlabeled (un-evaluated) data, and labeling (classification) or evaluations (regression) are relatively expensive. The learning algorithms can choose actively which samples to label or evaluate with the goal of reducing the amount of labeling or evaluation. This strategy is distinct from conventional supervised learning, also referred to as “passive learning”. A number of research efforts have focused on active learning, including text mining [21–25], speech recognition [26], and computational biology [27]. Very limited research in this context has been focused on circuit synthesis or similar tasks, which motivates this investigation into the gap between active learning and circuit synthesis design.

Here the random forest algorithm is used to construct the predictive model in the iterative process, where model validation and query selection are performed. As an ensemble method, the random forest can prevent overfitting and reduce the variance by training on bootstrap samples of the data. Active learning, in general, considers three scenarios: membership queries, stream-based selective sampling, and pool-based sampling [20]. The pool-based sampling scheme is used here [28]. To decide whether to query or discard instances, several query strategies are examined. We tested the strategies using two real data sets from circuit synthesis problems, and identified the promising active learning strategy for circuit synthesis. This article involves three primary contributions: 1) we introduce active learning as a powerful tool to address the challenges in circuit synthesis: through comparing several query strategies, we were able to build a predictive model with fewer training examples, and thus further reduce true solution cost for circuit synthesis problems; 2) we generate new insights and identify the query strategy specifically for

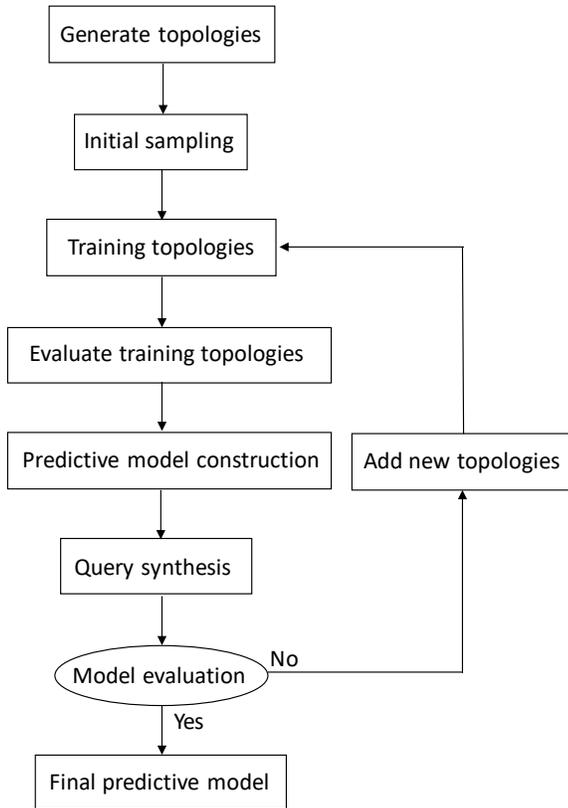


FIGURE 1: The active learning framework.

circuit synthesis via the analysis, such as ranking of the prediction; 3) we discuss how the proposed framework can be extended to other architecture design problems, such as active vehicle suspensions [17, 18] and fluid-based thermal management systems [29]. In addition, the relationship between active learning and adaptive surrogate modeling is clarified in this article.

The remainder of the article is organized as follows. In the next section, we describe the active learning framework for circuit synthesis. Section 3 describes the frequency response synthesis problem from which the dataset was obtained. Quantitative results are presented in Section 4. Discussion and conclusion are presented in Sections 5 and 6, respectively.

2 METHODOLOGY

In this section, the proposed active learning strategy for circuit synthesis is described. Figure 1 illustrates the overall active learning framework. The strategy assumes all possible N circuit topologies $X = \{\mathbf{x}_i\}$ for $i = 1, 2, \dots, N$ and $\mathbf{x}_i \in R^d$, are relatively easy to obtain, but evaluation of each circuit's performance metric $Y = \{y_i\}$ for $i = 1, 2, \dots, N$, is relatively expensive. The mapping $f : \mathbf{x}_i \mapsto y_i$ is the true response function.

2.1 Initial Sampling

All the possible circuit topologies are obtained using the efficient enumeration-based methodology introduced by Herber [16, 17]. Each circuit topology \mathbf{x}_i is represented by an adjacency matrix; the value of each matrix element indicates whether the two corresponding circuit components are connected (1) or not connected (0). A subset of N_I topologies are sampled randomly, denoted as an initial training set $X_I = \{\mathbf{x}_i\}$ for $i = 1, 2, \dots, N_I$. The training set elements are then evaluated using the true response function to obtain the performance $Y_I = \{y_i\}$, where $y_i = f(\mathbf{x}_i)$ for $i = 1, 2, \dots, N_I$. The unevaluated data set becomes (X_u, Y_u) , where $X_u = X \setminus X_I$ and $Y_u = Y \setminus Y_I$, respectively.

2.2 Predictive Model Construction

To approximate the performance y_i , a regression model is considered. Radial basis functions (RBFs) are often used, where the process can be seen as an artificial neural network to approximate the true unknown function [30]. Kriging is an approach for interpolation, where values are modeled using a Gaussian process [31–33]. Other methods may include response surface methodology (RSM) [34, 35], moving least-squares (MLS) [36–38], support vector regression (SVR) [39], adaptive regression splines [40], and inductive learning [41]. These approaches have been well-studied in surrogate-based optimization [42, 43], an engineering method in which a high-fidelity model cannot be used directly with optimization due to evaluation expense, but approximation models are constructed using samples from the original model. Optimization is then performed using the computationally inexpensive surrogate model to make solution of an approximate problem tractable. However, established surrogate modeling methods may not be applicable here because the design variables in the surrogate models are often continuous, rather than the discrete variables in the circuit synthesis that are represented in terms of the adjacency matrix.

Here we choose random forests as the model for approximating the response of the true circuit topology performance function. The random forest is an ensemble learning algorithm for regression and classification that can handle both continuous and categorical variables [44–46]. The random forest is composed of a number of decision trees at the training phase, and the output of the final prediction (regression) or label (classification) is based on the results given by the individual trees. The individual decision trees tend to have a low bias, but a high variance, resulting in overfitting to the training set. The random forest averages the output of multiple decision trees that were trained on different subsets of the training set. This strategy helps to reduce the overall variance and prevent overfitting [47]. The random forest uses bootstrap aggregating (also known as bagging) to improve the stability and accuracy of the learning algorithm [48]. For the given training set $T = \{(\mathbf{x}_i, y_i), i = 1, \dots, N_I\}$, bagging generates B bootstrap sample sets $T^b = \{(\mathbf{x}_i^b, y_i^b), i = 1, \dots, n^b, b = 1, \dots, B\}$ by

sampling n^b samples from T with replacement. Datum (\mathbf{x}_i, y_i) may appear multiple times or not at all in B data sets. A total number of B regression decision trees $\hat{f}_b(\cdot)$ are trained on T^b . For a new instance \mathbf{x} , the prediction \hat{y} takes the average of the B individual decision trees [48]:

$$\hat{y} = \hat{f}_B(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B \hat{f}_b(\mathbf{x}). \quad (1)$$

The estimated standard deviation s reflects uncertainty of the prediction:

$$s = \sqrt{\frac{\sum_{b=1}^B (\hat{f}_b(\mathbf{x}) - \hat{y})^2}{B-1}}. \quad (2)$$

The random forest algorithm has several tuning parameters, including the number of trees, depth of the tree, and so on. These hyperparameters can be determined through a Bayesian optimization during the training phase [49–51].

2.3 Query Synthesis

The query synthesis is a critical element of active learning. To determine whether a new instance \mathbf{x} should be queried or not, one needs to define a measure that characterizes \mathbf{x} . In other words, we should actively search the data that satisfies a certain criterion such that the training set can be updated iteratively using the query data. The choice of search is a process that involves the balance between exploration and exploitation over the topology space. One way of balancing exploitation of the prediction and exploration using estimated standard deviation is to find the instances with the smallest statistical lower bound (LB) [43]:

$$LB(\mathbf{x}) = \hat{y}(\mathbf{x}) - A s(\mathbf{x}), \quad (3)$$

where $\hat{y}(\mathbf{x})$ and $s(\mathbf{x})$ are the prediction and estimated standard deviation, respectively, given at \mathbf{x} quantified by Eqs. (1) and (2). Here parameter A controls the balance between exploitation and exploration; specifically, $A \rightarrow 0$ corresponds to pure exploitation, and $A \rightarrow \infty$ indicates pure exploration. Because $s(\mathbf{x})$ reflects the uncertainty of the prediction, it can be used as a utility measure for query selection, also referred to as *uncertainty sampling* [20]. Expected Improvement (EI) is also quantity that computes the amount of improvement to be expected given the mean $\hat{y}(\mathbf{x})$ and estimated standard deviation $s(\mathbf{x})$:

$$EI(\mathbf{x}) = (y_{min} - \hat{y}(\mathbf{x})) \Phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x}) \phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right), \quad (4)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and PDF of the standard normal distribution; y_{min} is the smallest observed value in the training set.

Leave-one-out (LOO) cross-validation is an error-based approach that measures the leave-one-out prediction error using the training samples [52]. Given a new instance \mathbf{x} and training sample $(\mathbf{x}_i, y_i) \in T$, the infill function is defined as [53, 54]:

$$v(\mathbf{x}) = D(\mathbf{x}) \cdot E(\mathbf{x}) \quad (5)$$

where $D(\mathbf{x}) = \min_{\mathbf{x}_i \in X_l} d(\mathbf{x}, \mathbf{x}_i)$ is the distance between \mathbf{x} and its closest training point \mathbf{x}_c . The error estimate function is $E(\mathbf{x}) = |y_c - \hat{y}_c^{LOO}|$, where \hat{y}_c^{LOO} is the leave-one-out prediction at \mathbf{x}_c . The query points are the ones with the largest values of $v(\cdot)$. Here the distance term explains exploration, while the error function corresponds to exploitation. This measure is likely to query points with significant prediction uncertainty, helping to improve model accuracy [54]. Similarly, density-weighted heuristics take into account both content information and input space region density for the new instance \mathbf{x} [20]:

$$v(\mathbf{x}) = s(\mathbf{x}) \cdot \left(\frac{1}{|X_u|} \sum_{\mathbf{x}' \in X_u} d(\mathbf{x}, \mathbf{x}') \right)^\beta \quad (6)$$

where $|X_u|$ is cardinality of the current unevaluated samples. The first term uses uncertainty sampling measured by the estimated standard deviation; the second term is the information weight that is calculated by averaging the distances to all the instances in X_u ; β is a hyperparameter that balances the relative importance of both terms. The distance metric can be chosen as Euclidean distance or cosine similarity, depending on the problem.

2.4 Model Evaluation

The predictive model should be evaluated before updating the training set. Cross-validation is a model validation method for parameter and accuracy estimation. It partitions the samples into two subsets: the analysis is performed on one subset (training set), and the analysis is validated using the other subset (validation set). Cross validation measures how the model will generalize to an independent set and prevents overfitting. Due to the random forest properties, out-of-bag (OOB) error is used instead. OOB is computed as the average prediction of those trees in which instance (\mathbf{x}_i, y_i) did not appear in their bootstrap samples [47, 55]. The OOB error can be used to perform parameter estimation in the Bayesian optimization. The measure for model accuracy here is root mean square error (RMSE) [56]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}} \quad (7)$$

where m is the number of validation points. Other metrics such as maximum absolute error and R^2 may also be considered [42]. The true RMSE for the random forest model is approximated using a test set, composed of additional circuit topologies and the corresponding true performances. The error between the true outputs and the predictions given by the random forest approximates the true prediction error. In other words, the test set provides an unbiased evaluation of the final random forest model fit on the entire training set.

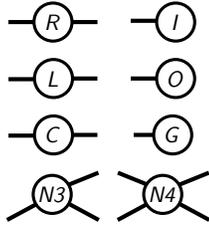


FIGURE 2: Some potential circuit component types.

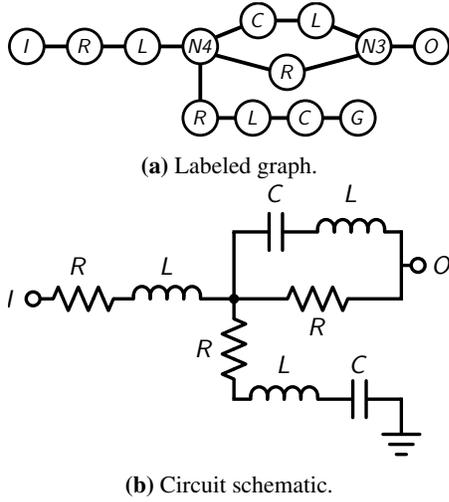


FIGURE 3: Different representations for the same circuit.

2.5 Training Data Set Update

Once the query data (X_q, Y_q) have been determined in the previous step, the training set and unevaluated set can be updated respectively:

$$(X_l, Y_l) := (X_l, Y_l) \cup (X_q, Y_q) \quad (8)$$

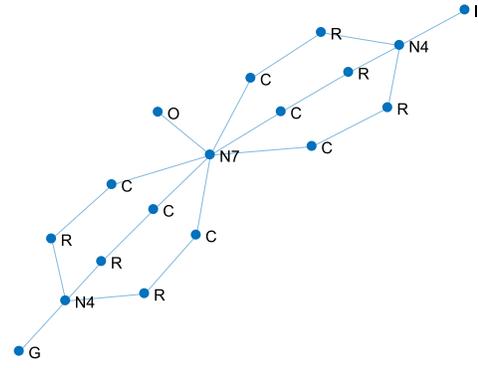
$$(X_u, Y_u) := (X_u, Y_u) \setminus (X_q, Y_q) \quad (9)$$

The stopping condition is either 1) *RMS E* falls below ϵ , or 2) the active learning process reaches a predefined iteration number n_{iter} .

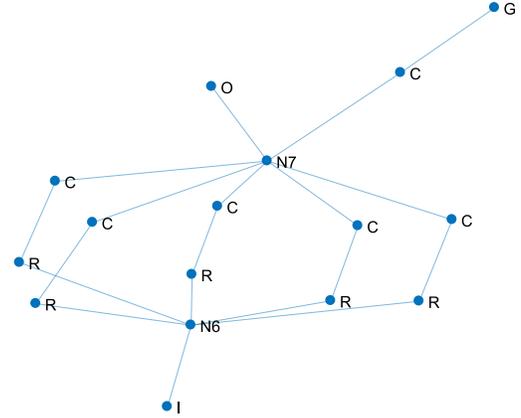
3 DESIGN PROBLEM

3.1 Problem Statement

Here we present the case study for circuit design synthesis. A set of circuit elements shown in Fig. 2 with replicates can be combined construct a circuit, such as the one shown in Fig. 3. Resistors (R), capacitors (C), and inductors (L) are 2-port components, meaning that they must have connections to two other components. The required 1-port components include the input (I), the output (O), and the ground (G). Various multiport (>2) common-voltage nodes are also available (e.g., the 4-port common-voltage node $N4$). Other component types are possi-



(a) Circuit example #1.



(b) Circuit example #2.

FIGURE 4: Two different circuit topologies with twelve 2-port components.

ble, but many circuits of practical importance, such as analog filters, can be constructed using this type of component catalog. Example circuit topologies are shown in Figs. 3 and 4. Figure 4a incorporates three 1-port nodes (I , O , G), one 7-port voltage node ($N7$), two 4-port voltage nodes ($N4$), and twelve 2-port nodes (R , C).

In the circuit synthesis problem, the circuit components (I , O , G) are fixed. A model may be constructed for a complete circuit by identifying the transfer function G between the input and output. We would like to synthesize practical circuits that satisfy the following target frequency response [4]:

$$|F(j\omega)| = \sqrt{\frac{2\pi}{10\omega}} \quad (10)$$

where the frequency range of interest is:

$$0.2 \leq \frac{\omega}{2\pi} \leq 5, \quad (11)$$

evaluated over 500 logarithmically-spaced points.

Herber successfully enumerated and evaluated circuits for this problem with up to 6 general impedance elements and only (R , C) 2-port components [17]. All circuit topologies represented as labeled graphs were enumerated under a set of specifi-

cations using the enumeration algorithms in Ref. [16]. To evaluate the performance given a desired circuit topology, we consider the following minimization problem [17]:

$$\min_{\mathbf{z}} E = \sum |r_k(\mathbf{z})|^2 \quad (12a)$$

$$\text{subject to: } \mathbf{l} \leq \mathbf{z} \leq \mathbf{u} \quad (12b)$$

where \mathbf{z} is the vector of optimization variables, representing the coefficients for the 2-port elements (R , C); the individual residual $r_k = g(\omega_k, \mathbf{z}) - f(\omega_k)$ is the difference between the transfer function magnitude $g(\omega_k, \mathbf{z})$ and desired circuit magnitude response $f(\omega_k)$ specified in Eqn. (10); (\mathbf{l} , \mathbf{u}) are the upper and lower bounds for \mathbf{z} .

3.2 Data

To test the active learning framework, we first obtained two sets of the circuit synthesis data by specifying different simple bounds on the resistors and capacitors [17]:

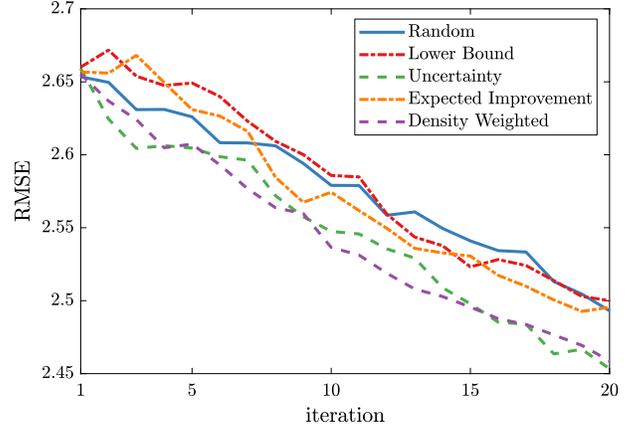
$$\text{(set 1)} \quad R \in [10^{-2}, 10^0] \Omega, \quad C \in [10^{-2}, 10^0] F$$

$$\text{(set 2)} \quad R \in [10^{-2}, 10^5] \Omega, \quad C \in [10^{-10}, 10^0] F$$

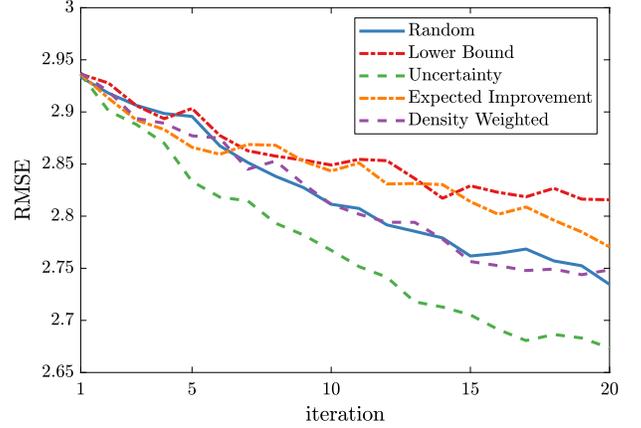
The circuit structure space was predefined by a collection of vectors, including distinct component types, the number of ports for each component type, the lower and upper bounds of replicates for each component type. Network structure constraints (NSCs) were used to define the feasibility for the graphs. A collection of 43,249 circuit topologies (denoted as X) with unique transfer functions were enumerated, and thus the corresponding circuit performance could be evaluated. For more details see Refs. [16, 17].

3.3 Data Preprocessing

Each circuit topology \mathbf{x}_i can be represented as a labeled-vertex graph and has a corresponding adjacency matrix. For a given catalog, the number of chosen components may vary across topology candidates. As a result, the adjacency matrix dimension can vary as well. Figures 4a and 4b share the same complexity number (12, computed from the number of 2-port components), but the total number of components is different. The adjacency matrices have dimensions 18×18 and 16×16 , respectively. In addition, adjacency matrix representation is not as compact as other representation possibilities. To address this, a preprocessing step was performed. We identified all possible components using a fixed-dimension adjacency matrix (large enough for the maximum number of each of the component types). However, this results in the corresponding matrix representation (feature space) being fairly sparse. Feature hashing, an efficient way of vectorizing features, is used to overcome this issue, turning the adjacency matrix feature space into a compact vector representation [57]. Feature hashing has been used widely to perform document classification tasks by hashing the features to their hash values. In other words, the *hashing-trick* transforms the high di-



(a) Set 1.



(b) Set 2.

FIGURE 5: Comparisons between the different query methods.

mensional vector into a lower dimensional feature space [58].

4 RESULTS

4.1 Comparison of Query Strategies

We randomly selected an initial training set of 10,000 circuits that were then evaluated to obtain the true response. Because the true response ranges from 10^{-5} to 10^2 , the natural logarithm was applied to scale performances. A set of 500 data points are queried to the training set in each iteration according to the criteria specified in Eqs. (3)–(6). The learning process is ended after $N_{\text{iter}} = 20$ iterations are completed. We then compare the different query strategies against a benchmark random sampling strategy. In random sampling, the set of 500 data points are randomly sampled and added to the training set during each iteration. In the lower bound, A was chosen as 2 to minimize the lower bound of the 95% confidence interval, and in the density weighted method, $\beta = 1$ was used.

Figure 5a presents the learning curve for Set 1. In the figure,

the blue solid line corresponds to the random sampling benchmark. The RMSEs for two of the query strategies in consideration, lower bound and expected improvement, are almost identical to random sampling after 20 iterations. The number of training samples needed is about 15,000 at RMSE = 2.55 at Iteration 10 for uncertainty sampling and weighted, but random sampling may require 17,000 samples at Iteration 14. Using one of these two more successful query methods can therefore reduce the number of true function evaluations by 2,000 to achieve the same desired RMSE. It is observed that the uncertainty and density weighted sampling methods are better than the others. The statistical lower bound and EI rely on both the prediction and variance, but perform worse than uncertainty sampling with the variance only. It may indicate that the prediction itself is less critical to the circuit synthesis problem. The space region information in the weighted density method may be useful in improving the model accuracy. Figure 5b in Set 2 exhibits different behaviors. Only uncertainty sampling outperformed random sampling. The other strategies, lower bound and expected improvement, perform worse than random sampling for Set 2. Because both sets were obtained by specifying different bounds on the components, these results also indicate that query strategy performance depends on data set properties.

4.2 Ranking Distance

While RMSE is a frequently used measure for the difference between the predicted and true observed values, the ordering or ranking of the predictions are also important. In the circuit synthesis problem, the ranking of the predicted values would facilitate the process of identifying high-performance circuit topologies. Here we specifically investigated the rank distance between the predicted values and observed values.

Kendall tau rank distance $K(\tau_1, \tau_2)$ is a metric that measures the number of pairwise disagreements or dissimilarity between two ranking lists (τ_1, τ_2) of size n [59]. If two lists are identical, then $K = 0$; and if two lists are opposite to each other, then $K = n(n-1)/2$. Often K is normalized by dividing by $n(n-1)/2$ so the normalized distance \tilde{K} lies in the interval $[0, 1]$. A value of $\tilde{K} = 0.5$ implies that the ordering of one of the lists was completely randomized.

Figures 6 and 7 report the normalized Kendall tau distances for both data sets with *uncertainty sampling* applied to the active learning strategy. The distances include the training set, test set, and all set. The training set contains the samples after 20 iterations, and the remainder of the data set is termed the test set. We are also interested in how well the predictive model performs on the ordering of all available 43,249 circuit designs (termed the all set); it is useful for designers to be able to select the best designs, which is possible if the rankings are preserved. In Set 1, a large portion of the ranking is kept, with the Kendall tau distances between 0.17077 and 0.24021 for the different sets.

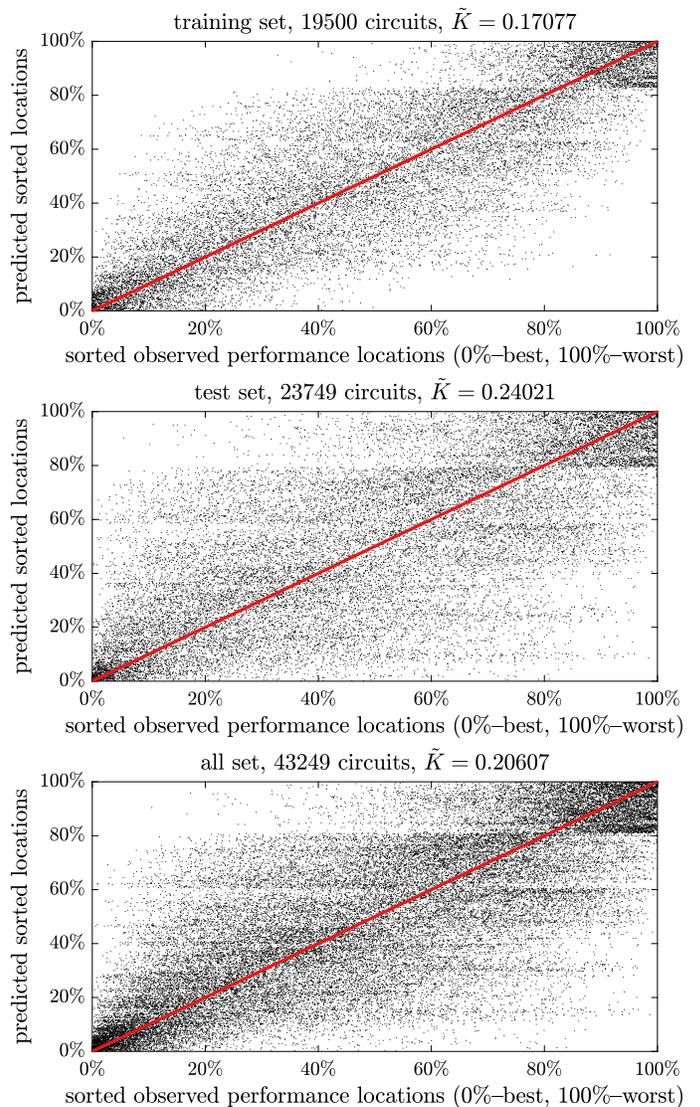


FIGURE 6: Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 1).

Similar observations can be found for Set 2 in Fig. 7. The Kendall distances through the iterations were also investigated in Figs. 8a and 8b. In both sets, the training process remains stable at Iteration 15, indicating a particular convergence property. One may further improve the training process by increasing the number of iterations, but in practice it will require more true circuit evaluations during the model construction. A cost-effectiveness analysis should be performed to assess potential benefit.

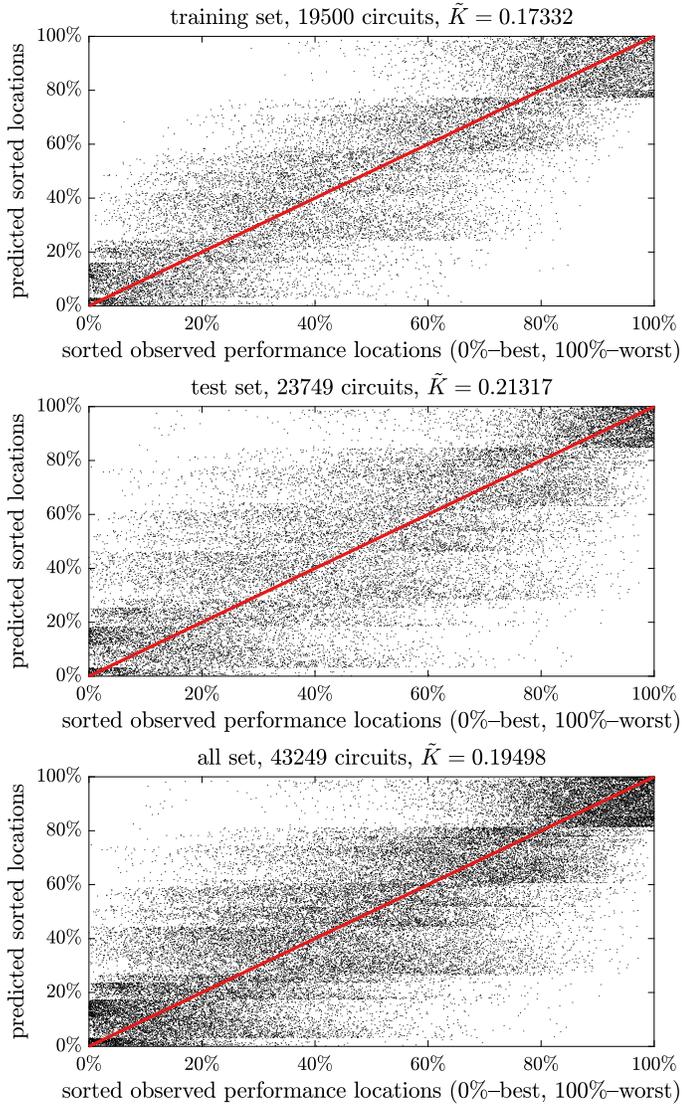


FIGURE 7: Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 2).

5 DISCUSSION

The numerical results show that, for some query strategies, active learning can reduce evaluation cost compared to random sampling for the circuit synthesis problem. Among the tested query methods, *uncertainty sampling* performs the best. Here the estimated standard deviation given by the random forest algorithm captures the information for the query points. A number of other query strategies could be investigated with the potential for further evaluation cost reduction. For instance, query by disagreement (QBD) [60, 61] and query by committee (QBC) [62, 63] may be appropriate for the ensembling method. A clustering-based method could also be considered [64, 65]. It

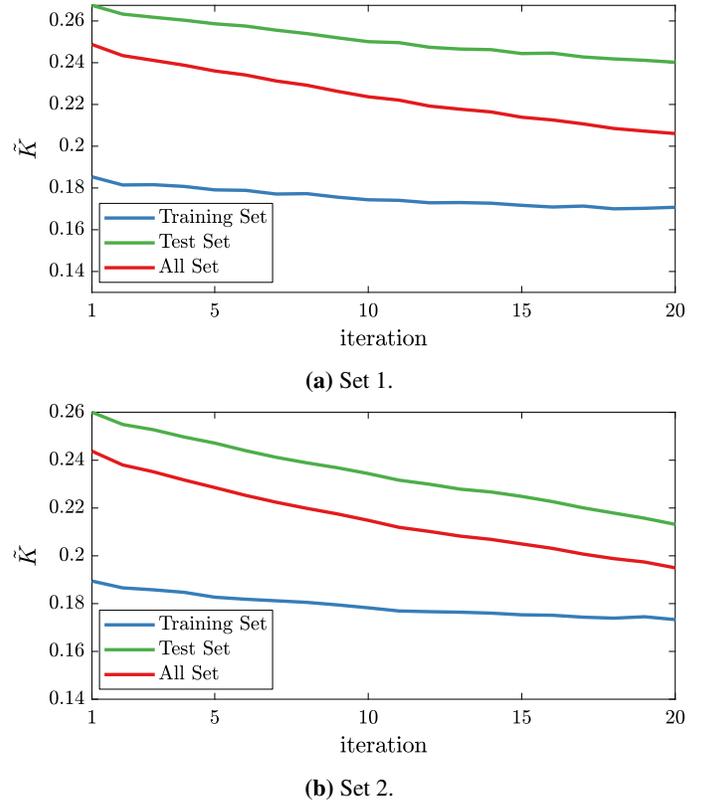


FIGURE 8: The Kendall tau distances through the active learning iterations.

might be useful to cluster the evaluated data X_u before applying either uncertainty sampling or the density weighted approach to each cluster. These methods in the past have primarily been used for classification problems in text mining. It is still worth exploring whether these methods are applicable to certain engineering design problems, such as circuit synthesis.

One drawback to consider is the additional computational cost required by active learning during the query phase; this cost may not be worth the gains. For instance, the LOO error in Eq. (5) is computationally expensive to evaluate cost (a topic of ongoing work), because an additional N_l random forest models must be retrained to obtain leave-one-out prediction in each loop. It is similar to the expected error reduction, where the future error for each query must be estimated, and a new model has to be retrained for every possible query over the entire unevaluated pool [66]. Variance reduction, often known as *optimal experimental design*, is also limited [67, 68]. It can only be applied to certain types of models such as linear/nonlinear and logistic regression, and hence is not generalizable. Whether or not the variance reduction method can be extended to tree- or nearest neighbor-based machine learning algorithms is still an open question [20]. Moreover, the variance reduction method involves inversion and manipulation of the Fisher information, and it turns

out to be slow and inefficient when a large number of parameters are to be estimated.

While the normalized Kendall tau distance indicates that a large portion of ordering has been preserved, improvements may be expected, as Kendall tau distances for both training sets are only around 0.17. Similar to the RMSEs, it is speculated that more query strategies or learners could be studied to further analyze the appropriateness of different methods for learning in engineering synthesis problems. For example, gradient booting is another ensemble method with the goal of reducing bias [69–71]; other learners such as a radial basis function network may be applicable [30]. These studies and analysis are left as future work.

It is observed here that active learning has some similarities with surrogate (or meta) modeling methods for design optimization. Wang and Shan summarized three classes of metamodel-based design optimization (MBDO) strategies [42]. The MBDO techniques also require sampling and construction of approximation models. In particular, adaptive MBDO and direct sampling approaches use an iterative mechanism to build, validate, and optimize metamodels. These MBDO strategies have been extended to multi-objective surrogate modeling [72–74]. However, there are also some distinctions. First, the vector of design variables \mathbf{x} in MBDO are often assumed to be continuous, and they serve as the solutions to the global optimization problem. In active learning, instance \mathbf{x} is an observation drawn from a certain probability density distribution and the feature space could be either numerical or categorical. Second, in machine learning or active learning, it is often assumed that observations (X, Y) are available (i.e., they exist in terms of data, rather than design variables), or at least X is easy to obtain but Y involves a high computational expense. An initial set of training samples can be drawn from available data. Surrogate modeling often generates initial samples via space-filling design methods, such as Latin hypercube sampling [33, 75], and uses them as the training samples for approximate model construction. Finally, query selection in active learning samples points from the unevaluated data pool based on the predictions. Adaptive MBDO performs the optimization and validates the model in the loop to determine re-sampling (or obtain additional samples) and update the metamodels. The direct sampling approach samples toward the optimal solution given by the metamodel. Surrogate modeling utilizes the optimization in the loop to aid the re-sampling scheme and construction of the metamodel, whereas active learning updates the approximation model iteratively using information from the unlabeled or unevaluated data in the pool.

As part of ongoing work to address Case 2 synthesis problems discussed in Section 1, we are investigating techniques for generating circuit topologies that implicitly satisfy NSCs. This eliminates the need to enumerate all unique, feasible topologies, supporting approximate solution of synthesis problems where the catalogs are impractically large for enumeration. This approach narrows the search space through an intelligent mapping. A gen-

erative model is a probabilistic model capable of producing both the observations and targets in the data set. Restricted Boltzmann machines [76], variational auto-encoders (VAEs) [77], and generative adversarial networks (GANs) [78] are generative models used for modeling observations and targets drawn from a certain joint probability distribution. Guo et al. developed an indirect design representation for topology optimization in heat conduction design using VAEs [79]. For the circuit synthesis problem, GANs could be a promising option to generate circuit topologies by sampling using GANs.

6 CONCLUSION

In this article, we presented an active learning strategy for reducing topology evaluation cost for a circuit synthesis problem. We aimed to address the problem where it is possible to enumerate all unique and feasible topologies, but exhaustive evaluation is impractical (Case 1 synthesis problem). Here we constructed a predictive model using a *random forest* to approximate true circuit topology performance. The active learning strategy interactively queried informative samples from the training set to construct iteratively-improved approximations, while reducing the number of training samples required. A number of query strategies were tested and compared. The active learning strategy helps reduce the evaluation cost for circuit synthesis because 1) we can avoid using the true evaluation function for each circuit topology; 2) we can make more accurate predictions given by the predictive models. The numerical experiments indicate that the uncertainty sampling query strategy is most effective among the tested methods for circuit synthesis. Through the active learning experiments, we found that uncertainty and topology structure may play critical roles in improving the appropriation model accuracy and make a significant contribution to reducing the system evaluation costs.

Future work should involve the exploration of more robust and efficient query criterion. While a number of query strategies have been presented, not all of the strategies outperform the random sampling benchmark. Recent studies indicate no query selection strategy is consistently or clearly the best [20], because it relies heavily on the learners and applications. For example, Jin et al. compared different metamodeling techniques and observed that the robustness and accuracy of various surrogate models depend on the non-linearity of the problems [80]. As a result, it is worth further investigation into the selection of the learners, as well as understanding the circuit synthesis and other heterogeneous topology optimization problems more deeply with respect to active learning solution methods. Even though only a specific test problem was investigated, it is clear that the active learning strategy could be extended to other architecture system design tasks. For instance, the active learning strategy is applicable to the low-pass filter problem [17]. The active vehicle suspension design can also be posed as an architecture design

problem using labeled graphs, and this active learning strategy may be suited for evaluation cost reduction [18]. In addition, other well-established methods, such as Bayesian inference and MBDO, have contributed to important advancements within and outside the design research community. A comprehensive comparison may be conducted where these methods are applied to the same circuit synthesis problem. The active learning strategies for circuit synthesis and other similar engineering problems have not yet been well-studied and we hope the work presented here will serve as a basis for productive future research.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) under Grant EEC-1449548.

REFERENCES

- [1] Mitea, O., Meissner, M., Hedrich, L., and Jores, P., 2011. "Automated Constraint-driven Topology Synthesis for Analog Circuits". In Design, Automation & Test in Europe Conference & Exhibition, pp. 1–4.
- [2] Gan, Z., Yang, Z., Shang, T., Yu, T., and Jiang, M., 2010. "Automated Synthesis of Passive Analog Filters Using Graph Representation". *Expert Systems with Applications*, **37**(3), pp. 1887–1898.
- [3] Sussman, G., and Stallman, R., 1975. "Heuristic Techniques in Computer-aided Circuit Analysis". *IEEE Transactions on Circuits and Systems*, **22**(11), pp. 857–865.
- [4] Grimbleby, J. B., 1995. "Automatic Analogue Network Synthesis Using Genetic Algorithms". In Genetic Algorithms in Engineering Systems: Innovations and Applications, pp. 53–58.
- [5] Das, A., and Vemuri, R., 2007. "An Automated Passive Analog Circuit Synthesis Framework Using Genetic Algorithms". In IEEE Computer Society Annual Symposium on VLSI, pp. 145–152.
- [6] Das, A., and Vemuri, R., 2008. "Topology Synthesis of Analog Circuits Based on Adaptively Generated Building Blocks". In ACM/IEEE Design Automation Conference, pp. 44–49.
- [7] Lohn, J. D., and Colombano, S. P., 1999. "A Circuit Representation Technique for Automated Circuit Design". *IEEE Transactions on Evolutionary Computation*, **3**(3), pp. 205–219.
- [8] Goh, C., and Li, Y., 2001. "GA Automated Design and Synthesis of Analog Circuits with Practical Constraints". In Congress on Evolutionary Computation, Vol. 1, pp. 170–177.
- [9] Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A., and Dunlap, F., 1997. "Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming". *IEEE Transactions on Evolutionary Computation*, **1**(2), pp. 109–128.
- [10] Ochotta, E. S., Rutenbar, R. A., and Carley, L. R., 1996. "Synthesis of High-performance Analog Circuits in ASTRX/OBLX". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **15**(3), pp. 273–294.
- [11] Bruccoleri, F., Klumperink, E. A. M., and Nauta, B., 2001. "Generating All Two-MOS-transistor Amplifiers Leads to New Wide-band LNAs". *IEEE Journal of Solid-State Circuits*, **36**(7), pp. 1032–1040.
- [12] Foster, R. M., 1932. "Geometrical Circuits of Electrical Networks". *Transactions of the American Institute of Electrical Engineers*, **51**(2), pp. 309–317.
- [13] Bayrak, A. E., Ren, Y., and Papalambros, P. Y., 2016. "Topology Generation for Hybrid Electric Vehicle Architecture Design". *Journal of Mechanical Design*, **138**(8), p. 081401.
- [14] Del Castillo, J. M., 2002. "Enumeration of 1-DOF Planetary Gear Train Graphs Based on Functional Constraints". *Journal of Mechanical Design*, **124**(4), pp. 723–732.
- [15] Ma, W., Trusina, A., El-Samad, H., Lim, W. A., and Tang, C., 2009. "Defining Network Topologies that can Achieve Biochemical Adaptation". *Cell*, **138**(4), pp. 760–773.
- [16] Herber, D. R., Guo, T., and Allison, J. T., 2017. "Enumeration of Architectures with Perfect Matchings". *Journal of Mechanical Design*, **139**(5), p. 051403.
- [17] Herber, D. R., 2017. "Advances in Combined Architecture, Plant, and Control Design". Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec.
- [18] Herber, D. R., and Allison, J. T., 2018. "A Problem Class with Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions". In ASME 2018 International Design Engineering Technical Conferences, no. DETC2018-86213.
- [19] Guo, T., 2018. "On the Use of Machine Learning with Design Optimization Data for System Topology Design". Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Aug.
- [20] Settles, B., 2012. "Active Learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **6**(1), pp. 1–114.
- [21] Settles, B., Craven, M., and Friedland, L., 2008. "Active Learning with Real Annotation Costs". In Proceedings of the NIPS Workshop on Cost-sensitive Learning.
- [22] Settles, B., and Craven, M., 2008. "An Analysis of Active Learning Strategies for Sequence Labeling Tasks". In Conference on Empirical Methods in Natural Language Processing, pp. 1070–1079.
- [23] Xu, Z., Akella, R., and Zhang, Y., 2007. "Incorporating Diversity and Density in Active Learning for Relevance Feedback". In European Conference on Information Retrieval, pp. 246–257.
- [24] McCallumzy, A. K., and Nigamy, K., 1998. "Employing EM and Pool-based Active Learning for Text Classification". In International Conference on Machine Learning.
- [25] Fujii, A., Tokunaga, T., Inui, K., and Tanaka, H., 1998. "Selective Sampling for Example-based Word Sense Disambiguation". *Computational Linguistics*, **24**(4), pp. 573–597.
- [26] Zhu, X., and Goldberg, A. B., 2009. "Introduction to Semi-supervised Learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **3**(1), pp. 1–130.
- [27] Smith, B. C., Settles, B., Hallows, W. C., Craven, M. W., and Denu, J. M., 2010. "SIRT3 Substrate Specificity Determined by Peptide Arrays and Machine Learning". *ACS Chemical Biology*, **6**(2), pp. 146–157.
- [28] Lewis, D. D., and Gale, W. A., 1994. "A Sequential Algorithm for Training Text Classifiers". In Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12.
- [29] Peddada, S. R. T., Herber, D. R., Pangborn, H., Alleyne, A. G., and Allison, J. T., 2018. "Optimal Flow Control and Single Split Architecture Exploration for Fluid-Based Thermal Management". In ASME 2018 International Design Engineering Technical Conferences, no. DETC2018-86148.
- [30] Broomhead, D. S., and Lowe, D., 1988. Radial Basis Functions, Multi-variable Functional Interpolation and Adaptive Networks. Tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom).
- [31] Wahba, G., 1990. *Spline Models for Observational Data*, Vol. 59. SIAM.
- [32] Cressie, N., 1988. "Spatial Prediction and Ordinary Kriging". *Mathematical Geology*, **20**(4), pp. 405–421.
- [33] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989. "Design and Analysis of Computer Experiments". *Statistical Science*, **4**(4), pp. 409–423.
- [34] Box, G. E. P., and Wilson, K. B., 1992. "On the Experimental Attainment of Optimum Conditions". In *Breakthroughs in Statistics*. Springer, pp. 270–

- [35] Box, G. E. P., and Draper, N. R., 1987. *Empirical Model-building and Response Surfaces*. John Wiley & Sons.
- [36] Lancaster, P., and Salkauskas, K., 1981. “Surfaces Generated by Moving Least Squares Methods”. *Mathematics of Computation*, **37**(155), pp. 141–158.
- [37] Levin, D., 1998. “The Approximation Power of Moving Least-squares”. *Mathematics of Computation of the American Mathematical Society*, **67**(224), pp. 1517–1531.
- [38] Aitken, A. C., 1936. “IV.—On Least Squares and Linear Combination of Observations”. *Proceedings of the Royal Society of Edinburgh*, **55**, pp. 42–48.
- [39] Vapnik, V., 2013. *The Nature of Statistical Learning Theory*. Springer Science & Business Media.
- [40] Friedman, J. H., 1991. “Multivariate Adaptive Regression Splines”. *The Annals of Statistics*, **19**(1), pp. 1–67.
- [41] Langley, P., and Simon, H. A., 1995. “Applications of Machine Learning and Rule Induction”. *Communications of the ACM*, **38**(11), pp. 54–64.
- [42] Wang, G. G., and Shan, S., 2007. “Review of Metamodeling Techniques in Support of Engineering Design Optimization”. *Journal of Mechanical Design*, **129**(4), pp. 370–380.
- [43] Forrester, A. I. J., and Keane, A. J., 2009. “Recent Advances in Surrogate-based Optimization”. *Progress in Aerospace Sciences*, **45**(1–3), pp. 50–79.
- [44] Ho, T. K., 1995. “Random Decision Forests”. In International Conference on Document Analysis and Recognition, Vol. 1, pp. 278–282.
- [45] Ho, T. K., 1998. “The Random Subspace Method for Constructing Decision Forests”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(8), pp. 832–844.
- [46] Breiman, L., 2001. “Random Forests”. *Machine Learning*, **45**(1), pp. 5–32.
- [47] Hastie, T., Tibshirani, R., and Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2 ed. Springer.
- [48] Breiman, L., 1996. “Bagging Predictors”. *Machine Learning*, **24**(2), pp. 123–140.
- [49] Bull, A. D., 2011. “Convergence Rates of Efficient Global Optimization Algorithms”. *Journal of Machine Learning Research*, **12**(Oct), pp. 2879–2904.
- [50] Snoek, J., Larochelle, H., and Adams, R. P., 2012. “Practical Bayesian Optimization of Machine Learning Algorithms”. In Advances in Neural Information Processing Systems, pp. 2951–2959.
- [51] Gelbart, M. A., Snoek, J., and Adams, R. P., 2014. “Bayesian Optimization with Unknown Constraints”. *arXiv preprint*. arXiv:1403.5607.
- [52] Meckesheimer, M., Booker, A. J., Barton, R. R., and Simpson, T. W., 2002. “Computationally Inexpensive Metamodel Assessment Strategies”. *AIAA Journal*, **40**(10), pp. 2053–2060.
- [53] Kato, H., and Funazaki, K.-i., 2014. “POD-driven Adaptive Sampling for Efficient Surrogate Modeling and Its Application to Supersonic Turbine Optimization”. In Turbine Technical Conference and Exposition, p. V02BT45A023.
- [54] Iuliano, E., 2016. “Adaptive Sampling Strategies for Surrogate-based Aerodynamic optimization”. In *Application of Surrogate-based Global Optimization to Aerodynamic Design*. Springer, pp. 25–46.
- [55] James, G., Witten, D., Hastie, T., and Tibshirani, R., 2013. *An Introduction to Statistical Learning*, Vol. 112. Springer.
- [56] Hyndman, R. J., and Koehler, A. B., 2006. “Another Look at Measures of Forecast Accuracy”. *International Journal of Forecasting*, **22**(4), pp. 679–688.
- [57] Wu, W., and Benesty, M., 2016. *FeatureHashing: Creates a Model Matrix via Feature Hashing with a Formula Interface*. R package version 0.9.1.1.
- [58] Weinberger, K., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J., 2009. “Feature Hashing for Large Scale Multitask Learning”. In Annual International Conference on Machine Learning, pp. 1113–1120.
- [59] Kendall, M. G., 1938. “A New Measure of Rank Correlation”. *Biometrika*, **30**(1/2), pp. 81–93.
- [60] Cohn, D. A., 1994. “Neural Network Exploration Using Optimal Experiment Design”. In Advances in Neural Information Processing Systems, pp. 679–686.
- [61] Hanneke, S., 2009. “Theoretical Foundations of Active Learning”. Ph.D. Dissertation, Carnegie Mellon University.
- [62] Seung, H. S., Opper, M., and Sompolinsky, H., 1992. “Query by Committee”. In Annual Workshop on Computational Learning Theory, pp. 287–294.
- [63] Freund, Y., and Schapire, R. E., 1997. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. *Journal of Computer and System Sciences*, **55**(1), pp. 119–139.
- [64] Kang, J., Ryu, K. R., and Kwon, H.-C., 2004. “Using Cluster-based Sampling to Select Initial Training Set for Active Learning in Text Classification”. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 384–388.
- [65] Nguyen, H. T., and Smeulders, A., 2004. “Active Learning Using Pre-clustering”. In International Conference on Machine Learning, p. 79.
- [66] Roy, N., and McCallum, A., 2001. “Toward Optimal Active Learning through Sampling Estimation of Error Reduction”. In International Conference on Machine Learning, pp. 441–448.
- [67] Chaloner, K., and Verdinelli, I., 1995. “Bayesian Experimental Design: A Review”. *Statistical Science*, **10**(3), pp. 273–304.
- [68] Fedorov, V. V., 1972. *Theory of Optimal Experiments*. Elsevier.
- [69] Breiman, L., 1997. *Arcing the Edge*. Tech. Rep. 486, University of California, Berkeley.
- [70] Friedman, J. H., 2001. “Greedy Function Approximation: A Gradient Boosting Machine”. *Annals of Statistics*, **29**(5), pp. 1189–1232.
- [71] Friedman, J. H., 2002. “Stochastic Gradient Boosting”. *Computational Statistics & Data Analysis*, **38**(4), pp. 367–378.
- [72] Rao, L. G., Schuh, J. K., Ewoldt, R. H., and Allison, J. T., 2015. “On Using Adaptive Surrogate Modeling in Design for Efficient Fluid Power”. In ASME 2015 International Design Engineering Technical Conferences, no. DETC2015-46832, p. V02BT03A024.
- [73] Lee, Y. H., Schuh, J. K., Ewoldt, R. H., and Allison, J. T., 2018. “Simultaneous Design of Non-Newtonian Lubricant and Surface Texture Using Surrogate-Based Optimization”. In AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, no. AIAA 2018-1906.
- [74] Lee, Y. H., Cormann, R. E., Ewoldt, R. H., and Allison, J. T., 2017. “A Multi-objective Adaptive Surrogate Modeling-Based Optimization (MO-ASMO) Framework Using Efficient Sampling Strategies”. In ASME 2017 International Design Engineering Technical Conferences, no. DETC2017-67541, p. V02BT03A023.
- [75] Park, J.-S., 1994. “Optimal Latin-hypercube Designs for Computer Experiments”. *Journal of Statistical Planning and Inference*, **39**(1), pp. 95–111.
- [76] Hinton, G. E., and Salakhutdinov, R. R., 2006. “Reducing the Dimensionality of Data with Neural Networks”. *Science*, **313**(5786), pp. 504–507.
- [77] Kingma, D. P., and Welling, M., 2013. “Auto-encoding Variational Bayes”. *arXiv preprint*. arXiv:1312.6114.
- [78] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014. “Generative Adversarial Nets”. In Advances in Neural Information Processing Systems, pp. 2672–2680.
- [79] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T., 2018. “An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer”. In AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, no. AIAA 2018-0804.
- [80] Jin, R., Chen, W., and Simpson, T. W., 2001. “Comparative Studies of Metamodeling Techniques under Multiple Modelling Criteria”. *Structural and Multidisciplinary Optimization*, **23**(1), pp. 1–13.