# ON THE USE OF MACHINE LEARNING WITH DESIGN OPTIMIZATION DATA FOR SYSTEM TOPOLOGY DESIGN

BY

TINGHAO GUO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Assistant Professor James T. Allison, Chair
Professor Harrison Kim
Associate Professor Pingfeng Wang
Assistant Professor Jeremy S. Guest

ABSTRACT

In this dissertation, several machine learning strategies are presented to advance solution capabilities for homogeneous and heterogeneous system topology design. The core contribution is to begin bridging the gap between data science and design science. The key principle is to extract meaningful knowledge and insights from design data, and to build machine learning models that enable effective design exploration and lead to generalizable design insights. This work provides an alternative perspective for system topology design, leveraging design data instead of designer intuition derived from experience or established gradient-based topology optimization methods.

As a preliminary study for this dissertation, the research literature for a relevant segment of the engineering design research community was analyzed using network analysis. This study was based on a collection of 1,668 articles published in the American Society of Mechanical Engineers (ASME) Design Automation Conference (DAC) from 2002-2015. Several methodologies were developed and used, and useful insights were provided. These analyses revealed several insights, including opportunities for strengthening the link between design automation methods, such as design optimization, and machine learning. This result serves as a basis and motivation for the remainder of the work presented here. The remainder of the dissertation concentrates on efforts to advance understanding of how to use machine learning effectively with design data generated using design automation methods.

A novel design framework using deep learning was developed for homogeneous system topology optimization. The application chosen here involves heat conduction, with competing objective functions of temperature and power density. Existing methods can solve related heat conduction problems efficiently (e.g., thermal compliance), but not combined temperature and power density. The strategy presented here seeks to use data generated from related easy-to-solve thermal compliance problems to support efficient solution of the

desired problem. An indirect design representation was constructed using a variational autoencoder (VAE), and was combined with a deep convolutional style transfer network to improve the quality of generated designs. The VAE maps the original large-dimensional design space onto a lower dimensional space (called the latent space). The heat conduction problem was solved by optimizing with respect to latent variables, and system performance was evaluated using full-dimension design representations. Several variants of the optimization formulation have been examined, and the Pareto-optimal solutions are presented. The method is shown to successfully navigate the design space and identify many non-dominated designs that outperform those found using conventional topology optimization.

Topology optimization of heterogeneous systems, sometimes referred to as synthesis, requires different design representations and solution techniques. Here we considered two classes of synthesis problems where existing methods and recent advances, such as efficient enumeration, are limited in practical solution capability. The first class of synthesis problems (Case 1) considered is where efficient enumeration methods can be used to list all unique, feasible system design topologies, but not all topologies can be evaluated in a practical amount of time due to computational expense. Active learning is investigated here as a strategy to select subsets of topologies for evaluation with the goal of finding high-performance designs without the need to evaluate all candidates. Active learning is a semi-supervised learning technique that interactively improves predictive model accuracy with strategically selected training examples. The predictive model used here is an ensemble method called random forest. Several active learning strategies are considered and results indicate that active learning is a promising strategy for solving Case 1 synthesis problems.

Case 2 synthesis problems involve systems where all topologies of interest can neither be enumerated nor evaluated in a practical time period. Here a new approach for Case 2 problems is introduced where machine learning techniques are used to generate topologies in a way that implicitly satisfies constraints and aids search for high-performance designs, in essence creating a targeted design space representation for efficient search. This eliminates

the need to enumerate all unique, feasible topologies, and supports approximate solution of Case 2 synthesis problems. Generative adversarial networks (GANs) are investigated as the design representation tool for this design automation process. Experiments were conducted to explore capabilities of GAN-based synthesis methods for electronic circuit synthesis. Multiple GAN strategies are investigated. The numerical results demonstrate that the improved Wasserstein GAN is capable of generating feasible circuit topologies efficiently. The GAN-based design framework may also be extended to more general design synthesis tasks.

*To My Parents*

# ACKNOWLEDGEMENTS

I would like to express gratitude to my advisor, Professor James T. Allison, for giving the opportunity to pursue my graduate study. His guidance and support helped me accomplish the Ph.D research. I would like to thank Professors Harrison Kim, Pingfeng Wang, and Jeremy Guest for being my committee.

A number of collaborators have contributed to success of this dissertation, including Prof. Max Yi Ren, Dr. Neal Davis, Dr. Zhi Han, Dr. Daniel Herber, Danny Lohan, Ruijin Cang, Jiarui Xu, Yue Sun, Yilin Dong, and Dongqi Su. The Engineering System Design Lab (ESDL) members also provided useful insights and invaluable feedback, including Dr. Anand Deshmukh, Albert Patterson, Yong Hoon Lee, Satya Peddada, Siyao Luan, Madhav Arora, Andrew Blanco, Jason McDonald, Jeff Arena, Lakshmi Rao, etc. Without them, I would not have completed this dissertation.

I am grateful to the Department of Industrial and Enterprise Systems Engineering (ISE) at the University of Illinois at Urbana-Champaign for providing me with graduate teaching assistantships these years. I had the opportunities to work with many instructors and TAs, including Dr. Douglas King, Prof. Yu-Ching Lee, Prof. Masooda Bashir, Dr. Yuan Zhao, Steven Lin, Dr. Hao Jiang, Dr. Ashish Khetan, etc. I really enjoyed working with them and cherish the time together. The ISE staff, Prof. Rakesh Nagi, Aleta Lynch, Barb Bohlen, Holly Tipsword Kizer (former ISE staff), and Amy Summers (former ISE staff) are kind and thoughtful. In particular, I would like to thank Prof. Ramavarapu Sreenivas for his support and encouragement.

I would like to acknowledge internship supervisors for improving my data science skills, including Hank Roark, Dan Peterson, Dr. Scott Ray, and Dr. Amy Wang. I truly appreciate my friends, Rongbo Lu, Dr. Jingnan Chen, Dr. Bill Roberts, Xiou Ge, Xinchen Pan, Haohua Wan, Dr. Meng Wang, Zihang Zhang, and many others, who have always been supportive and making me stronger.

Finally, thanks to my parents for your continuing love.

# Contents

**List of Figures** x

**List of Tables** xiv

**1 INTRODUCTION** 1

1.1 Overview . . . 1

1.2 Research Objectives . . . 4

1.3 Dissertation Scope and Overview . . . 11

**2 NETWORK ANALYSIS OF DESIGN AUTOMATION LITERATURE** 13

2.1 Introduction . . . 14

2.2 Background and Objectives . . . 15

2.3 Data . . . 19

2.4 Network Analysis Methods . . . 20

2.5 Results . . . 24

2.6 Discussion . . . 47

2.7 Conclusion . . . 50

**3 HOMOGENEOUS SYSTEM TOPOLOGY DESIGN** 53

3.1 Introduction . . . 53

3.2 Literature Review . . . 55

3.3 Background for Augmented VAE Indirect Representation . . . 64

3.4 Proposed Method . . . 68
3.5 Numerical Results . . . 73
3.6 Discussion . . . 89
3.7 Conclusion . . . 92

**4 HETEROGENEOUS SYSTEM TOPOLOGY DESIGN: CASE 1 95**

4.1 Introduction . . . 95
4.2 Literature Review . . . 98
4.3 Methodology . . . 108
4.4 Design Problem . . . 114
4.5 Results . . . 120
4.6 Discussion . . . 127
4.7 Conclusion . . . 129

**5 HETEROGENEOUS SYSTEM TOPOLOGY DESIGN: CASE 2 132**

5.1 Introduction . . . 132
5.2 Generative Adversarial Networks (GANs) . . . 139
5.3 Circuit Synthesis . . . 143
5.4 Numerical Experiment Design . . . 146
5.5 Numerical Results . . . 151
5.6 Discussion . . . 157
5.7 Conclusion . . . 160

**6 CONCLUSION 164**

**BIBLIOGRAPHY 170**

**A CHAPTER 3 200**

**B CHAPTER 5 204**

# List of Figures


2.1 Main topic frequency plot . . . . . 26
2.2 Sub-topic frequency plot . . . . . 27
2.3 Main topic chronological evolution . . . . . 28
2.4 Sub-topic chronological evolution . . . . . 29
2.5 Main topic correlation . . . . . 30
2.6 Sub-topic correlation . . . . . 32
2.7 Main topic and sub-topic correlation . . . . . 33
2.8 Average in-degree for main topics . . . . . 34
2.9 Average in-degree for sub-topics . . . . . 35
2.10 Citation network for sustainable energy system design . . . . . 36
2.11 Scatter plot for association rules of the main topics . . . . . 37
2.12 Group matrix-based visualization for main topics . . . . . 38
2.13 Scatter plot for association rules of sub-topics . . . . . 41
2.14 Group matrix-based visualization for DAC sub-topics . . . . . 42
2.15 Top 5 clusters using PM . . . . . 43

3.1 Design problem formulation framework for comparison of alternative design methods; SIMP and VAE topology optimization strategies are compared here 59
3.2 A schematics of an autoencoder . . . . . 65
3.3 Flowchart of the optimization scheme with respect to $\mathbf{z}$ . . . . . 68

3.4 The VAE model augmented by a style loss . . . . . 71
3.5 A sample of reconstructed topologies from a 20D latent space . . . . . 75
3.6 A sample of topologies generated from random samples in a 20D latent space 76
3.7 Parametric study results obtained by varying 2 of 20 latent variables . . . . 77
3.8 Locally-optimal solutions for the weighted sum method . . . . . 78
3.9 Locally-optimal solutions produced by the $\epsilon$-constraint method . . . . . 78
3.10 Pareto-optimal solutions produced by the $\epsilon$-constraint method . . . . . 79
3.11 Optimal solutions produced by the hybrid method (GA followed by gradient-based) . . . . . 80
3.12 Pareto-optimal solutions produced by the hybrid method . . . . . 81
3.13 Optimal solutions produced by the MOGA method (initial population = 200) 81
3.14 Select Pareto-optimal solutions produced by the MOGA method (initial population = 200) . . . . . 82
3.15 Optimal solutions produced by the MOGA solution method (#latent variables = 20) . . . . . 83
3.16 Four Pareto-optimal solutions produced by the MOGA solution method . . . 84
3.17 Comparison of Pareto-optimal solutions for the four methods (#latent variables = 20) . . . . . 85
3.18 Optimal solutions produced by the hybrid method (less initial population) . 86
3.19 Reduced space vs. original space using population size 800 (with data) . . . 90
3.20 Reduced space vs. original space using population size 5,000 (with data) . . 90
3.21 Reduced space vs. original space using population size 800 (without data) . . 91

4.1 Conceptual design formulation space used for analyzing differences in problem formulation strategies . . . . . 102
4.2 Conceptual design formulation space using different solution strategies . . . . 107
4.3 The active learning framework. . . . . 108
4.4 Some potential circuit component types [109] . . . . . 114

4.5 Different representations for the same circuit [109] . . . . . 115
4.6 Two different circuit topologies with twelve 2-port components [109] . . . . . 116
4.7 Comparisons between the different query methods . . . . . 119
4.8 Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 1) . . . . . 123
4.9 Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 2) . . . . . 124
4.10 The Kendall tau distances through the active learning iterations . . . . . 125

5.1 A Venn diagram for GAN-based methodology . . . . . 136
5.2 Design formulation analysis of the GAN-based methodology . . . . . 137
5.3 A basic GAN framework . . . . . 140
5.4 Examples of the heterogeneous circuit [109] . . . . . 144
5.5 Two common circuit templates [109] . . . . . 145
5.6 Procedure used to test GAN effectiveness at generating circuit topologies . . 150
5.7 The GAN circuit synthesis framework using predictive modeling . . . . . 151
5.8 Frequency response: generated circuit topologies (0–30 iterations) . . . . . 152
5.9 Frequency response: generated circuit topologies (100–1500 iterations) . . . . 153
5.10 Frequency response: GAN generation performance . . . . . 154
5.11 Frequency response: improved WGAN generation performance . . . . . 154
5.12 Frequency response: convergence of the improved WGAN . . . . . 155
5.13 Set 1: two high-performance circuit topologies given by the improved WGAN 161
5.14 Set 2: two high-performance circuit topologies given by the improved WGAN 162
5.15 Proposed framework using the GAN . . . . . 163
5.16 A linear quarter-car suspension model (standard passive system shown) . . . 163

A.1 A sample of dominated points for the weighted sum method . . . . . 200

A.2 Pareto-optimal solutions (#1 – #100) for the MOGA method (#latent variables = 20) . . . . . 201
A.3 Pareto-optimal solutions (#101 – #200) for the MOGA method (#latent variables = 20) . . . . . 202
A.4 Pareto-optimal solutions (#201 – #258) for the MOGA method (#latent variables = 20) . . . . . 203

B.1 Low-pass filter: generated circuit topologies (0 – 30 iterations) . . . . . 205
B.2 Low-pass filter: generated circuit topologies (100 – 1500 iterations) . . . . . 206
B.3 Low-pass filter: GAN generation performance . . . . . 207
B.4 Low-pass filter: improved WGAN generation performance . . . . . 207

# List of Tables


2.1 Summary of the co-authorship network . . . . . 25
2.2 Association rules for main topics . . . . . 39
2.3 Association rules for the sub-topics . . . . . 40
2.4 Top 10 influential articles in the DAC citation network . . . . . 44
2.5 Topic clusters in the ASME DAC citation Network . . . . . 45

3.1 Computational expense comparison . . . . . 86
3.2 Design space coverage analysis for the MOGA method . . . . . 87

4.1 The parametric study on the number of new samples taken at each iteration 126
4.2 The parametric study on parameter $A$ in the statistical lower bound approach 126

5.1 Frequency response: vanilla GAN . . . . . 148
5.2 Low-pass filter: vanilla GAN . . . . . 148
5.3 Frequency response: WGAN and improved WGAN . . . . . 148
5.4 Low-pass filter: WGAN and improved WGAN . . . . . 149
5.5 Frequency response: DCGAN . . . . . 149
5.6 Low-pass filter: DCGAN . . . . . 149
5.7 Frequency response: Parametric study on improved WGAN parameters . . . 156
5.8 Frequency response: performance of the GAN based method with active learning 157

B1 Low-pass filter: the parametric study on the parameters of the improved WGAN 204

# Chapter 1

# INTRODUCTION

## 1.1 Overview

Data science is an interdisciplinary and emerging field that studies the insights and knowledge extracted from data [61]. It is a complex domain that uses a wide range of techniques, including mathematics, statistics, information theory, and computer science. In particular, it is focused on the interface between optimization, machine learning, artificial intelligence, databases, data mining, databases, and visualization. Data science is often associated with terms such as "Big Data" and "Predictive Analytics" [247]. In Ref. [15], Baesens summarized data science applications in various settings, including marketing, risk management, government, Web, and logistics. Gandomi and Haider also presented analytical methods for data science: text analytics, sentiment analysis, audio analytics, video analytics, social media analytics, and predictive analytics [84]. Recently, engineers and designers are becoming interested in data science. Kusiak wrote an article about big data in mechanical engineering [142]. This article posited that big data could play an important role in the design of materials, products, and systems; it could even help reconfigure industry and businesses, including product design and development, manufacturing, and the energy industry. In addition, data science in supply chain management has been studied in Refs. [108,247]. Rel-

evant techniques—namely optimization, regression, classification, clustering, associations, and model visualization—are applicable to process science and manufacturing [28, 242].

Data science has been applied in several ways to engineering design and product development, but a recent literature analysis study [100] has identified that data science is not well-connected to many important topics within design research that stand to benefit from its application. Here we propose a new strategy for using data science in design: utilizing design automation methods not just to solve design problems, but to generate rich data sets for unprecedented engineering systems that can be used with machine learning to accelerate the generation of design knowledge.

Design Science is a relatively new research area; it aims to provide people with creative and innovative products and services using scientific methods [197]. Papalambros proposed three breaking developments in the future of the design research, one of which is to access 'brain data'. Brain data is also big data in neuroscience [145, 240, 243]. The collaboration between neuroscience and design research can be regarded as a way of delivering new research topics and insights in the future [197]. The use of data science, including machine learning and big data, may play an important role in design science. Papalambros also quoted several senior researchers' visions regarding the future of design research:

- Marco Cantamessa: "Among the scientific approaches that will probably cast a significant influence on design research over the next few years, I see great promise in cognitive neuroscience and in the use of big data."

- Sean Hanna: "The advent of machine learning, big data, parallel processing and similar computational tools on one hand, and the technology to see brain activity on the other, provide the tools both to model and to observe this messy process."

- Jordan J. Louviere: "Despite all the hype, so-called 'big data' largely applies to current products in current markets, with the possibility that real-time and/or quasi-continuous updating could provide feedback about markets where data are available."

- Chris McMahon: "The issue is again one of proprietary data and of accumulating and sharing engineering knowledge, but it is also a 'big data' challenge how can we learn from the very extensive data that we have about the performance of artifacts and the tools used to design them?"

- Wei Chen: "Future design science research, in understanding the role of the human (both as user and as designer), will not only build on the fundamental principles of design but will also exploit new and exciting research opportunities in crowdsourcing, social computing, web-based user analysis, human-centered design, network analysis, data mining, and many other fields."

The visions above indicate that data science may be a useful tool for design research. The gaps between data science and design research need to be explored. It may be promising to address the existing issues and difficulties associated with design research utilizing data science as a "design method" such that new insights and creative designs can be delivered.

In this dissertation, sub-domains of data science, including machine learning and data mining, are considered for fusion with design research. The first study presented here focuses on social network analysis of the ASME Design Automation Conference (DAC) literature. This study revealed that machine learning is not yet thoroughly integrated into the DAC literature. This result is aligned with the researchers' visions described above. These factors serve as a motivation and foundation for this dissertation. Here we address challenging topological engineering design problems using machine learning and design data generated using optimization and efficient enumeration methods. Two important system classes are examined: homogeneous and heterogeneous. An indirect design representation for topology optimization using a variational autoencoder (VAE) and deep style transfer network is used to study the homogeneous system for the heat conduction. Active learning and and generative adversarial networks (GANs) are proposed to address important challenges identified in heterogeneous system topology design. Two canonical circuit synthesis design problems are used in the studies presented here.

## 1.2 Research Objectives

The primary objective of this dissertation is to develop methods based on machine learning tools that leverage design optimization data to solve challenging engineering design problems that previously could not be solved in a practical way. To explain the motivation and reasoning, we would like to create data-driven design methodologies with similar capabilities to existing methods, but that are free from some important limitations (such as reliance on human intuition or expertise, or limits on problem formulation). We are aiming to solve problems that cannot be solved by established methods efficiently. This gives rise to a tradeoff between problem formulation and solution accuracy. We will present relevant theatrical framework and the tradeoff in each core chapter.

Two types of system topology design problems, homogeneous and heterogeneous systems, are considered. Homogeneity is defined as "the quality or state of being of a similar kind or of having a uniform structure or composition throughout" [177]. Heterogeneity refers to "the quality or state of consisting of dissimilar or diverse elements" [176]. For instance, in a high performance computer, homogeneous computing (also referred as CPU computing), consists of similar cores or units while the heterogeneous computing (also generalized as CPU-GPU computing) contains more than one type of processor or core [85].

Homogeneous and heterogeneous system design problems can be found across a wide range of applications. A small sample of these applications is referred to in this document. A number of studies have been carried out in microstructural materials systems [2, 40, 206, 256–258]. The topology design for genetic regulatory circuits was also investigated [96, 98]. However, the meaning of homogeneous and heterogeneous systems for topology design has not been well defined (and is a topic of ongoing work). In this dissertation, we define homogeneous systems to be those where each element (or node) has the same type of properties (material, functionality, representation, etc.). Heterogeneous systems are defined as those where two or more fundamentally distinct element types exist in the system. Several

questions will be considered in this dissertation:

- What is the nature and structure of research within the engineering design community (limited here to DAC)? How might machine learning be used in new ways to advance design knowledge?
- How can machine learning be used with design data generated using design optimization, instead of with historical design data? How might this strategy aid the more rapid development of fundamentally new system designs as opposed to incremental changes?
- When a design space is too large for conventional methods to be practical, how might a predictive model support targeted sampling and evaluation of design candidates?
- How might we leverage design optimization data to develop an intelligent strategy for generative design?

In Chapter 2, the first research question will be addressed. A collection of 1,668 articles from the ASME Design Automation Conference (DAC) (2002-2015) were analyzed using social network analysis. This chapter focuses on exploring new opportunities for research within the DAC community. More specifically, the DAC co-authorship network and collaboration patterns were analyzed using centrality metrics, and the networks were found to exhibit the small world property. Two topic modeling strategies were proposed: a frequency-based model and Propagation Mergence (PM). The latter is a hybrid approach that utilizes natural language processing, unsupervised learning, and network structure. Descriptive statistics are presented, such as topic trends, correlation, and citation analysis. Inductive statistics, including association rule learning and cluster analysis, are also discussed. The results uncover the major research areas within DAC, and also reveal potential research gaps. One important finding is that the DAC community has not looked comprehensively yet at the potentially valuable links between machine learning and AI with other important design

research topics. This observation is the starting point for this dissertation. Here it is posited that the concept of using machine learning in conjunction with design optimization-derived data for topological design of unprecedented systems is a promising unexplored link. While grammar rules and other similar methods have been used to inspire new system architectures, the approach proposed here is unique and may have the ability to scale well to large system design problems. The motivation behind this work is to provide an analysis and to identify potential research gaps between sub-domains in the design research. The proposed suggestions and recommendations may guide community directions (keeping in mind that changes in direction are made collectively by the community). The centrality metrics used in the co-authorship network help identify collaboration patterns, which is especially useful for researchers who are new to the community; topic modeling techniques reveal research areas that have received varying levels of attention. The insights and gaps obtained from these results helped inspire the research topics chosen for this dissertation.

Chapter 3 considers the second research question, and involves a data-driven framework for homogeneous system topology design. Here we consider a homogeneous heat conduction system design for compact power electronics. The objective of the proposed data-driven methodology is to overcome the shortcomings in the existing methods, especially with respect to problem formulation accuracy. For instance, the well-known density-based topology optimization method, Solid Isotropic Material with Penalization (SIMP) [20], uses a direct design representation to handle a large number of continuous continuous variables and multiple constraints, but often suffers from the sensitivity issues because of the zero-derivative of the objective function with respect to the design variables. SIMP can only solve the approximate problem (thermal compliance), rather than the desired problem here (i.e., power density and maximum temperature objectives). Generative design algorithms (GDAs) [134, 161] use an indirect representation but rely on the prior design knowledge, because design rules have to be pre-defined for the desired topologies. Here we are seeking an efficient encoding strategy that is independent of intuition or existing representations and avoids optimizing thousands

of design variables. It is speculated that learning a representation for the complex systems, with a purpose of dimension reduction, may be a good option. It is well recognized that Principal Component Analysis [200] can be used for dimension reduction, but often results in information loss due to the linear property. Recently, an autoencoder, as a nonlinear encoding technique, has become one of the most powerful AI concepts in 2010s. The autoencoder can encode design features into a short representation (i.e., a latent vector) through the nonlinear activation function, and these latent variables can represent the topological design.

Here we specifically investigate an augmented variational autoencoder (VAE) for the homogeneous heat conduction design. The proposed VAE-based design methodology contains two phases. In the first phase, the design data is produced using a density-based topology optimization approach. Designs are represented as binary images, where each pixel is either 0 or 1. A variational autoencoder (VAE) and a deep convolutional style transfer network were used with this design data as training data. The purpose of the VAE is to encode the original two-dimensional (2D) image topologies into a lower-dimensional latent space. Representations in the latent space can then be decoded back into high-dimensional images. This dimension reduction in the latent space creates an abstract design representation that supports efficient design space exploration. Reconstruction of the 2D topologies can be conducted, and one can even generate novel 2D topologies via the decoder by sampling from the latent space.

The second phase of this study involves multi-objective optimization based on both conventional topology optimization (high-dimension), and a new strategy where optimization is performed in the low-dimension latent space. While conventional topology optimization methods achieve computational efficiency by exploiting problem structure, this limits their applicability to certain classes of problems. Here we seek to extend solution capabilities to a wider range of problem classes, while achieving computational efficiency through targeted dimension-reduction approaches. Optimization algorithms search for optimal solutions in the

low-dimension latent space, and performance metrics are evaluated using the full-dimension representation after VAE decoding. The proposed framework is demonstrated using a heat conduction topology optimization problem. The optimal (non-dominated) designs are presented in the objective function space, and the various Pareto fronts derived from alternative solution strategies are compared. This data-driven design framework provides a new perspective of design process for topology optimization, which is fundamentally different from traditional approaches such as SIMP (Solid Isotropic Material with Penalisation) [20] or level-set methods [249].

The third question will be answered in Chapters 4. Here circuit synthesis, a type of heterogeneous system topology design problem, is considered. Problems similar to these have been solved using heuristic algorithms (such as evolutionary algorithms, or EAs), and efficient enumeration strategies [109]. Neither of the these existing methods, however, are scalable to large heterogeneous system design problems. Two challenges may arise using the efficient enumeration method for circuit synthesis. The first challenge occurs when the enumeration of circuit topology is achievable but quantitative evaluation of design candidates requires an impractical amount of time. Here we refer to this situation as a Case 1 synthesis problem. We have access to all candidate topologies, but we cannot rank them altogether, and can only evaluate a small subset of all candidate topologies.

In Chapter 4, we introduce active learning as a strategy to solve Case 1 synthesis problems approximately. Active learning is applied to circuit synthesis. Active learning is a semi-supervised machine learning technique where a learner can interactively query the user to obtain more accurate prediction with fewer training samples. The active learning method is motivated by the need to strategically select a small number of designs to evaluate to find high-performance solutions quickly. Since the true system performance of candidate circuits require continuous dynamic system optimization (sizing) to evaluate, obtaining a large number of system evaluations that would be required for conventional supervised learning can be impractical. Consequently, active learning is expected to reduce the number of evaluations

needed for effective learning. The learner used here is an ensemble method, known as random forest. Several query strategies are compared. The results indicate that active learning is a promising strategy in reducing the evaluation cost for the circuit synthesis problem, and could be used as a general solution approach to similar design problems.

Several factors motivate the active learning for solving this type of problem: 1) Since Case 1 problems involve computationally expensive topology evaluation, an inferred function is used to estimate evaluation of topologies with low computational expense. Supervised learning could be used to address this regression task, but it uses a set of fully-labeled or evaluated training examples, and thus will not be practical for Case 1 problems. 2) An alternative approach would be to utilize unlabeled or unevaluated data in conjunction with a small amount of labeled or evaluated data. This is a semi-supervised learning scheme, where the acquisition of labels or evaluation often requires a high cost (e.g. human agent, physical experiment, etc.). 3) Under the semi-supervised learning setting, can we query the labeled or evaluated data and update the small amount of the training set iteratively (i.e., the required number of training examples is lower than that the regular supervised learning), but attain a good model accuracy? That is how active learning strategy comes into play.

Chapter 5 addresses the fourth research question and presents work that aims to overcome the challenge of Case 2 synthesis problems, where it is impossible to enumerate all possible heterogeneous topologies corresponding to a large component catalog. Current circuit synthesis methods involve domain knowledge, evolutionary algorithms (EAs), and enumeration. While these methods have been applied successfully, they either rely on human expertise, or are difficult to navigate for the large topological design spaces. These issues motivate exploration of an intelligent design scheme that implicitly learns the design principles from the design optimization data and efficiently generates feasible topology designs using an indirect representation. Generative models have the ability to represent and manipulate the high-dimensional data, and a number of design tasks intrinsically require sample generation from a particular data set [90]. Consequently, it is speculated that the generative models

may be well-suited for the Case 2 problem.

Here an abstract design representation based on generative adversarial networks (GAN) and data derived from representative enumeration and design optimization is presented as a solution. GANs are a class of generative models in unsupervised learning, focusing on learning the distribution of a statistical data set. Two models are used in a GAN: one is called the generator, and is tasked with producing new data samples ("fake" data); the other model is a discriminator, and is tasked with evaluating sample authenticity (i.e., it determines whether a sample comes from the real data set or not). The generator works to produce passable samples without being caught by the discriminator. The role of the discriminator is to identify the samples coming from the generator as fake, thus applying pressure to the generator to improve output quality. This mechanism is equivalent to playing a minimax zero-sum game. The GAN-based methodology for circuit synthesis is based on design data generated using a recently-developed efficient enumeration strategy for synthesis problems. The GAN is constructed using this data, and, similar to VAEs, the circuit topologies are represented in terms of a latent vector drawn from a standard multivariate normal distribution. A new circuit topology can be produced via the generator using this latent vector. We explored a number of GAN architectures and focused on two case studies: frequency response and low-pass filter circuit design. A comparative study was performed. It was found that the improved WGAN has the ability to synthesize feasible circuit topologies efficiently. The GAN-based methodology for circuit synthesis may provide a foundation for other circuit synthesis tasks. For example, the active learning strategy developed for Case 1 problems could be combined GAN feasible design generation for Case 2 problems. In addition, the GAN framework could be utilized in sizing optimization and other heterogeneous topology optimization applications, such as active vehicle suspensions [109] or other mechatronic systems.

## 1.3 Dissertation Scope and Overview

This dissertation focuses on using machine learning methods to advance system topology design capabilities and understanding. One unique element is that data for these methods is obtained through systematic numerical experiments using design optimization tools and predictive physics-based models. Previous efforts in design research using machine learning have often focused on extracting knowledge from historical design data (which is fixed and unalterable) [81], whereas data obtained from design optimization is more flexible (data sets can be designed, and new data can be generated as needed). Designers can specify targeted design optimization studies to generate the data needed to obtain the desired knowledge. Learning from historical data is descriptive, whereas learning from design optimization data is normative. It addresses how engineers should design systems, as opposed to answering how engineers have previously designed systems.

The task described here is a significant undertaking. Developing a flexible and reliable design optimization implementation that can be used to generate a wide range of optimal designs under a variety of design conditions by itself requires significant effort. To keep the scope of this dissertation reasonable, the focus is on the intersection between design optimization and machine learning. The case studies are based on mature design optimization applications that have been developed by others. Readers are referred to the appropriate references for details regarding application models, performance metrics, specialized solution methods, and other specifics. Here these applications and solution tools created by others are leveraged to generate the data needed for the studies presented here. In addition, a detailed analysis of the distinction between homogeneous and heterogeneous topology design problems is outside the scope of this dissertation, and is a topic of ongoing parallel work by others.

The dissertation is organized as follows. Chapter 2 focuses on network analysis of the design automation literature. Chapter 3 introduces a VAE-based indirect representation

for homogeneous system topology design. Chapter 4 describes the active learning learning strategy for evaluation cost reduction in the heterogeneous system topology design (Case 1 synthesis). Chapter 5 presents the GAN-based design methodology for intelligently generating heterogeneous circuit topologies (Case 2 synthesis). The last chapter summarizes the dissertation, including conclusions, contributions, and future work. The appendices follow the bibliography.

# Chapter 2

# NETWORK ANALYSIS OF DESIGN AUTOMATION LITERATURE

**Collaborative Acknowledgement:** As explained in the previous chapter, much of this work has leveraged the contributions of others. At the beginning of each chapter that involved collaboration, the contributions made by others are summarized to help make clear what portions are independent work. This chapter in particular involved long-term contributions from a large number of other individuals. Obtaining and analyzing literature network data was very effort-intensive, and the overall effort lasted several years. A number of the following contributors are co-authors in associated publications.

**Jiarui Xu** focused on the computation of the co-authorship network; **Yue Sun** helped with model construction; **Yilin Dong** focused on data processing; **Prof. Neal E Davis** reviewed the chapter and provided important feedback, especially with respect to network analysis theory and research community analysis. In addition, we would like to acknowledge UIUC undergraduate Computer Science (CS) student, **Dongqi Su** for developing the Propagation Mergence (PM) approach and the interactive visualization application for the DAC citation network. We also would like to acknowledge the numerous graduate and undergraduate students at UIUC who contributed to this effort, including Industrial and Enterprise

Systems Engineering (ISE) alumni **Ehsan Salimi** and **Tuganai Borina**, and computer science (CS) alumni **Ethan Chan**, **Arin Tsikretsis**, **Varun Berry**, **Gabby Chuchro**, **Nick Nordeen**, **Zach Halasz**, and **Yijun Zhang**.

## 2.1 Introduction

In the previous chapter, a brief motivation was provided for using machine learning with design optimization-derived data. Part of this motivation was derived from observations made during a recent study of a subset of the engineering design research literature. In this chapter, we present a summarized analysis of this literature. One of the observations was that machine learning is not well-linked with other relevant topics within the engineering design research community. This dissertation aims to address this gap in part.

In this chapter, we present the results of a study of citation and co-authorship networks for articles published at the ASME Design Automation Conference (DAC) during the years 2002-2015. Two topic-modeling methods are presented for studying the DAC literature: A frequency-based model was developed to explore DAC topic distribution and evolution, as well as citation analysis for each core topic. Correlation analysis and association-rule mining were used to discover relationships between topics. A new unsupervised learning algorithm, *propagation mergence*(PM), was created to address identified shortcomings of existing methods, and applied to study the existing DAC citation network. Influential articles and important article clusters were identified and effective visualizations created. We also investigated the DAC co-authorship network by identifying key authors and showing that the network structure exhibits small-world-network properties. The resulting insights, obtained by the both the proposed and existing methods, may be beneficial to the engineering design research community, especially with respect to determining future research directions and possible actions for improvement. The data set used here is limited; expanding to include additional relevant conference proceedings and journal articles in the future would offer a

more complete understanding of the engineering design research literature.

## 2.2 Background and Objectives

The ASME International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE) are leading international research meetings in design and engineering [14]. IDETC provides an opportunity for researchers to share the latest research and to build social networks, both in academia and industry. One conference that is part of IDETC, the Design Automation Conference (DAC), focuses on the development and analysis of rigorous methods for designing engineered systems. The DAC research community celebrated its fortieth anniversary recently, precipitating community reflection on past progress and future directions [120]. Since 2012, ongoing efforts, led by some authors of this chapter and others, have sought to understand the nature of the DAC research community through established quantitative network analysis methods. After initial analysis of the DAC literature, Allison presented a lightning talk at 2014 ASME IDETC/CIE [10,45,120] during the keynote DAC session. This chapter presents a significant extension of this work, including application of distinct frameworks for topic modeling that are scalable and applicable to more general citation networks.

Network analysis, sometimes also referred to as *structural analysis*, examines social structures as a network or graph [195,253]; specifically, citation and co-authorship networks highlight patterns and connectivities within scientific literature. A citation network represents articles as nodes (vertices), and directed edges indicate article citations. A co-authorship network represents authors as notes, which are connected when the two authors have co-authored at least one article together. In a citation network, each article is represented as a node (vertex); a directed link (edge) represents the citation of one article by another. The citation network is directed and acyclic because an article can only cite its predecessors. Citation analysis began in the 1960s with Price's study of the social structure of scientific

literature [202, 203]. His discovery that the degree of a citation network follows a power law was the first demonstration of the scale-free network property [203]. Price's growth model for citation networks [202] is a special case of what is now known as the preferential attachment process [16, 188]. More recently, researchers have come to rely on citation analysis as an index of community structure. For example, Chen and Redner investigated the citation network of Physical Review publications from 1893–2007 and identified major communities using modularity maximization [46]. A patent citation network was studied to understand the mechanisms of knowledge transfer in nanoscale science and engineering from 1976–2004 [154]. Kajikawa et al. used citation analysis to identify fifteen research clusters in the field of sustainability science [128]. A subset of other fields analyzed include medicine [37, 93], engineering [128, 129], physics [113, 210], semantics [261], and the Nobel Prize literature [139].

A co-authorship network is distinct from citation networks in that each author is represented by a node, with an edge linking authors who co-author an article [187]. Co-authorship network analysis has attracted interest due to its topological features. For instance, Newman found that collaborative networks in medical research, physics, and computer science exhibit the so-called *small-world* effect, present when the degree distribution of authors and articles follow a power-law distribution [184]. Newman's later work explored collaboration patterns in article databases [186]. Ding focused on exposing the collaborative behavior of productive and highly-cited authors available on the Web of Science (WOS) active from 1956–2008 [62]. Liu et al. studied author impact in digital library conferences [158]. Further work, such as co-authorship structures and link prediction problems, can be found in Refs. [25, 156, 232, 233].

A variety of approaches and algorithms are used in network analysis. *Centrality measures* and *clustering coefficients*, for example, are often used. Degree centrality was used in macro-disciplinary evolution [191] and bipartite networks [147]. Betweenness centrality has been applied to study wireless sensor networks [155], biological networks [183], and metro

systems [60]. Structural or topological clustering methods were also used to study these networks [123, 169, 223, 259]. Newman has developed a series of algorithms for detecting and evaluating network community structures [187, 189]. Link analysis algorithms explore associations between nodes in web search ranking, the most well-known examples being Google's PageRank [32, 196] and HITS (Hypertext Induced Topic Selection) [138]. Other authors have explored path-based search algorithms (Sun et al. [233, 234]) and link prediction problems (Liben-Nowell [156]). A wide range of other studies has addressed for analyzing network structures [8, 52, 74, 185]. These structured-based approaches lay a foundation for development of the proposed approaches in this chapter.

The foregoing approaches rely on network structure alone, and are limited thereby. Document content and topical features could also be considered. Latent Dirichlet Allocation (LDA), a Bayesian network model, is a generative statistical model that analyzes features for topic discovery (Blei et al. [23]; Pritchard [204]), with the assumption that each document can be represented as a mixture of different topics. The author–topic model was introduced by Rosen-zvi et al. [215] as an extension of LDA. Similarly, a probabilistic method known as Author-Conference-Topic (ACT) enables one to conduct topic modeling for articles, authors and publication venues (Tang et al. [238]). Ding also utilized the ACT model for scientific collaboration analysis [62]. Fu et. al revealed the inherent structural forms of the US patent database using a methodology combining a Bayesian model with latent semantic analysis (LSA) [80]. Topic analysis has also gained popularity, including the use of topic groups [17], topic extraction [35, 37], content analysis [42], and cluster analysis [128, 129].

The objective of the work presented here is to address the following questions: 1) *What knowledge can we gain from the DAC literature networks?* and 2) *Can we generate observations and insights with the potential to benefit future efforts in this research community?* In our efforts to answer these questions, we made two core contributions: 1) We developed two strategies for DAC topic exploration: a frequency-based model and propagation mergence (PM). The latter is fundamentally different from existing strategies in that it combines

network and textual information. 2) Using both the proposed and existing methods, we completed the first thorough (but not exhaustive) study of the DAC citation and co-authorship networks. While similar analyses have been performed for other scientific domains, this is the first presentation network analysis results specifically for the DAC community.

Several aspects of the ASME DAC corpus were analyzed. We identified the most collaborative authors using the co-authorship network, and demonstrated that the DAC co-authorship network possesses the small-world property [8,250]. We used the frequency-based model to study DAC topic distribution and evolution. Citation analysis within each topic field was conducted, and topic relations were explored using correlation and association rule learning. The second model applied was the PM approach [231]; it is distinct from existing structure-based approaches or topic modeling techniques in that it is an unsupervised learning approach that accounts for both topology (citation structure) and text content for topic and cluster analysis. PM was used to identify highly-influential DAC articles, as well as to group DAC topic clusters. The creation of PM was motivated by the subjectively inadequate results generated by existing topic modeling methods for DAC articles, such as LDA [23].

The material presented here is the result of a long-term effort that builds upon work that was presented in 2014 [10,45], which included network visualization, a phrase timeline, chronological streamgraphs, topic modeling and visualization, and a recommender system based on author/article similarity. The work here explores several additional aspects of the DAC networks, and significant improvements were made to DAC topic modeling efforts by using new strategies, and more intuitive and precise outcomes resulted. We believe this work is an important initial milestone for understanding the engineering design research community via literature network analysis.

The remainder of the chapter is organized as follows. Section 2.3 describes data collection and preparation. Section 2.4 introduces methodologies used in the chapter, including the frequency-based model, correlation analysis, association rule learning, and PM. Knowledge and insights gained from our DAC network analysis are reported in Section 2.5. In Sec-

tion 2.6, we summarize the results and propose future work. Conclusions are presented in Section 2.7, including recommendations and new potential research activities for the DAC community. Data and code are available online [97].

## 2.3 Data

The American Society of Mechanical Engineers (ASME) keeps records of DAC articles, that were presented and archived in the proceedings of IDETC/CIE since 2002 [14]. ASME aims to archive all past proceedings, but at present published articles can only be verified since 2002; this limits the data set available for this study. We parsed the ASME conference proceedings website to extract bibliographical information including titles, authors, keywords and abstracts. This website does not include lists of references for each DAC article, so citation information was extracted directly from DAC articles using a mostly-automated strategy. The similarity between every pair of distinct author names was measured using a modified implementation of the Levenshtein distance [151] as a strategy to identify articles written by the same author with slight differences in name representation (e.g., middle initials included or not). This method considers the maximum edit distance between the two names, as well as individual name lengths and discrepancies of name initials to avoid false positives. If two names were identified as being from the same author using this automated approach, these matches were then checked by a human evaluator before merging to enhance name disambiguation accuracy. A total of 1,668 DAC articles were retrieved from the period 2002–2015. This excludes articles that were accepted, but not presented (and therefore not published). Both citation and collaboration networks were constructed using this database of bibliographic records and citation information.

We acknowledge the limitations of this data set. It is not a comprehensive representation of the engineering design research literature, reflecting only a specific segment of this community. A larger effort would be required to compile a more representative data set that

includes other important conferences (both other ASME and non-ASME design research conferences), as well as engineering design journals (such as the ASME *Journal of Mechanical Design (JMD)* and the *Journal of Computing and Information Science in Engineering (JCISE)*). Challenges involved with a more comprehensive effort include not only significantly larger and more heterogeneous data gathering, but also the presence of articles in journals such as JMD that are not from the engineering design research community. Important questions regarding the impact of design research, and influences on design research, could be answered in part if articles citing design research articles, or external articles cited by design research articles, were part of the data set. For this first step in literature network analysis for engineering design research, we limited our selected dataset to DAC articles to maintain reasonable scope while providing the possibility of useful results, and because of familiarity with this community and recent community interest in self-reflection and analysis. If expanded efforts are possible in the future, outcomes and methods from this initial study could be expanded to a more comprehensive analysis.

## 2.4 Network Analysis Methods

### 2.4.1 Co-authorship Analysis

We performed co-authorship network analysis using NetworkX [102]. We found the most collaborative authors based on node degree. Two graph metrics, the local clustering coefficient and average path length, were used to analyze the small-world behavior exhibited by the DAC co-authorship network [250]. Given a graph $G$, the local clustering coefficient $C(v_i)$ is a structural property that quantifies how closely the neighbors of node $v_i \in G$ tend to congregate in $G$. For a node $v_i$ with $k_{v_i}$ neighbors and $\Gamma_{v_i}$ edges between its neighbors, the local clustering coefficient is: $C(v_i) = \frac{\Gamma_{v_i}}{k_{v_i}(k_{v_i}+1)}$ [127]. The average clustering coefficient $\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C(v_i)$ is the mean value of $C(v_i)$ across all $n$ nodes in $G$ [250]. The average path length $\mathcal{L}$ refers to the average of number of edges in the shortest path between all

possible pairs of connected nodes in the network [250]. This robust topological metric has been well-studied and used widely for analysis of random graphs, scale-free, and small world networks [8].

### 2.4.2 Citation Analysis

#### Frequency-Based Model

Given a collection of 1,668 DAC abstracts, we first used ToPMine (Topic Phrase Mining) [69] and RAKE (Rapid Automatic Keyword Extraction) [214] to discover and extract key phrases. The ToPMine algorithm transforms a corpus into a "bag of phrases" and screens for phrases that appear more frequently than a certain threshold. This framework supports the high-quality production of key phrases. RAKE is an unsupervised, domain- and language-independent algorithm that automatically extracts key phrases from individual documents. It allows users to impose restriction on phrase frequency, the number of words in a phrase, and the length of each word while selecting acceptable key phrases. RAKE also filters out meaningless words as determined by the user. Each keyword is assigned a score in terms of word frequency, word degree, and ratio of degree to frequency. A set of candidate key phrases was generated using both approaches for the DAC abstracts.

As not every key phrase reflects topic information, human expertise can help refine the set of key phrases. For example, the phrase "recent years"—extracted by RAKE—is too general to convey meaningful information. The Natural Language Toolkit (NLTK) Python library was used to standardize phrases with certain part-of-speech (POS) patterns [164], thus transferring each key phrase to a root form. A field of ten main topics and fifty subtopics was provided by a human expert after examining automatically generated key phrases. Each key phrase was then assigned up to 3 topics by the human expert. Each abstract was searched for key phrases in the topic list, and assigned corresponding topics when a match occurred. Multiple topic assignments were allowed for each abstract. The resulting Boolean topic

matrix catalogued 1,668 abstracts against 10 main topics and 50 subtopics.

### Topic Exploration

Correlation functions as a simple but useful metric that can indicate the predictive relationship between topics. We computed the correlation matrix characterizing the linear relationship between the 10 main topics and 50 sub-topics. We also investigated each topic based on the citation network structure. The average degree measures the activity of citation in each DAC topic. (Recall that the citation network is a directed graph; the average degree refers to the average in-degree per node in the network.)

Association rule learning (ARL) is a well-established data-mining technique that reveals relationships between database entries [5]. In our case, ARL can be applied to topic relationships. The `apriori` algorithm in the R software package `arules` was used to mine association rules for DAC literature topics [5, 104]. Agrawal et al. summarized the problem of mining association rules as follows [4]: Let $I = \{i_1, i_2, \ldots, i_n\}$ denote the set of $n$ binary attributes (or *items*) and $T$ the set of transactions (or *database*). Each transaction $t \in T$ is a binary vector. Define $t_k = 1$ if transaction $t$ contains item $i_k$, and $t_k = 0$ otherwise. An association rule has a form of $X \Rightarrow Y$, where item $X$ is a subset of item set $I$ and $Y$ is a single item in $Y \in I$ and $Y \notin X$. Items $X$ and $Y$ are referred as the *antecedent* (left-hand-side or LHS) and the *consequent* (right-hand-side or RHS) of the association rule [4, 104]. Three critical metrics are considered in association rules: *support, confidence*, and *lift*. For given items $X$, the support of $X$, denoted as $\text{supp}(X)$ is the proportion of transactions in the database that contains items $X$. The confidence of a rule is computed as $\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$, where $\text{supp}(X \cup Y)$ is the support of the intersection of $X$ and $Y$, equivalent to the joint probability $P(X \cap Y)$ [103]. The confidence is an estimated conditional probability $P(Y|X)$, indicating how likely item $Y$ is to appear also in the transactions, given the proportion of transactions that contain items $X$. The *lift*, defined as $\text{lift}(X \Rightarrow Y) = \frac{\text{conf}(X \Rightarrow Y)}{\text{supp}(Y)} = \frac{\text{conf}(Y \Rightarrow X)}{\text{supp}(X)} = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)\text{supp}(Y)}$, measures the ratio between

occurrences of $X$ and $Y$ and the support expected if $X$ and $Y$ were statistically independent [103, 104]. Association rules with lift greater than one imply strong dependence, and are deemed to be useful to predict RHS in the data set [104]. LHS and RHS are considered independent if the lift is equal to one [103].

### Propagation Mergence (PM)

The PM approach was proposed by Su as an unsupervised learning algorithm considering both citation structure and content representation simultaneously [231]. Su also developed an online interactive visualization application using WebGL (Web Graphics Library) [11,41,198] in which PM was applied to the DAC citation network [230]. PM first identifies a set of most influential articles (also referred to as *source nodes*) using a TF-IDF representation [150] and a weight for the directed citation link. A PageRank-based algorithm scores the articles [196]; a set of influential articles is formed using this scoring and a user-specified threshold value. Source nodes (articles) propagate credit and information to non-source nodes in the citation network, leading to an initial cluster assignment; the number of initial clusters is equal to the number of the source nodes. The merging step combines clusters as necessary to eliminate redundancy, since a single article may be insufficient to reflect the whole cluster correctly. Alternatively, two or more clusters may be merged. PageRank- and HIT-based approaches have been introduced previously for network analysis [47, 63, 64]. For instance, Yang et al. presented a strategy combining a content-based method with the multi-type citation network and performed heterogeneous link analysis using topic PageRank, but the main focus was to improve ranking performance and author reputation [260]. PM is distinct because it groups DAC articles, into clusters by addressing citation network and content representation. Further technical details about the PM approach can be found in Ref. [231].

## 2.5 Results

### 2.5.1 Co-authorship Network Analysis

Table 2.1 summarizes DAC statistics from 2002 to 2015. A total of 1,668 articles were collected, associated with 2370 distinct authors, with an average of 119 articles each year. The top five collaborative authors are Timothy W. Simpson, Wei Chen, Janet K. Allen, Farrokh Mistree, and Panos Y. Papalambros. These were obtained using degree centrality. No specific pattern was found in the number of articles or authors each year. The average number of authors per article ranges from 2.20 to 2.67; this number increases slightly each year, indicating that researchers have tended to become more collaborative over time. The average clustering coefficient $\overline{C}$ and the average path length $\mathcal{L}$ are also reported. The average clustering coefficient has increased from 0.63 to 0.79, indicating that authors tend to form persistent cliques. One DAC author needs to communicate through fewer than two people (except 2006) to get acquainted with another DAC author if there exists a connection. The overall (all years) average clustering coefficient $\overline{C} = 0.71$ and overall shortest path length $\mathcal{L} = 5.90$ were calculated. These imply that the co-authorship network exhibits the small-world property, as $\overline{C} \gg C_{\text{rand}}$ and $\mathcal{L} \gtrapprox \mathcal{L}_{\text{rand}}$ [241, 250], where $C_{\text{rand}} = 0.011$ and $\mathcal{L}_{\text{rand}} = 4.59$ were computed using a random graph with the same number of nodes (2,370) and an average degree of $k = 3.81$. Albert and Barabási summarized a number of small world networks including several types of co-authorship networks with local clustering coefficients $0.066 \sim 0.76$ and the average path length $4.0 \sim 9.7$ [8]. The DAC co-authorship network exhibits similar characteristics. Overall, the DAC co-author network exhibited sparse connectivity. Many edges correspond to advisor-student research teams, with less frequent faculty collaboration. Later analysis results in this section indicate opportunities for additional collaborative work, which could help advance intellectual diversity and vitality [50, 163, 181].

Table 2.1: Summary of the co-authorship network

| Year | No. articles | No. Authors | Avg. No. Authors/article | $\overline{C}$ | $\mathcal{L}$ |
|---|---|---|---|---|---|
| 2002 | 117 | 278 | 2.38 | 0.63 | 1.22 |
| 2003 | 143 | 319 | 2.23 | 0.67 | 1.51 |
| 2004 | 115 | 275 | 2.39 | 0.67 | 1.43 |
| 2005 | 128 | 281 | 2.20 | 0.63 | 1.61 |
| 2006 | 118 | 272 | 2.31 | 0.71 | 2.76 |
| 2007 | 125 | 285 | 2.28 | 0.67 | 1.35 |
| 2008 | 119 | 308 | 2.59 | 0.75 | 1.40 |
| 2009 | 122 | 307 | 2.52 | 0.70 | 1.75 |
| 2010 | 112 | 273 | 2.44 | 0.76 | 1.69 |
| 2011 | 110 | 282 | 2.56 | 0.79 | 1.32 |
| 2012 | 123 | 302 | 2.46 | 0.74 | 1.55 |
| 2013 | 114 | 266 | 2.33 | 0.72 | 1.43 |
| 2014 | 108 | 269 | 2.49 | 0.75 | 1.48 |
| 2015 | 114 | 304 | 2.67 | 0.79 | 1.35 |
| Overall | 1668 | 2370 | 1.42 | 0.71 | 5.90 |

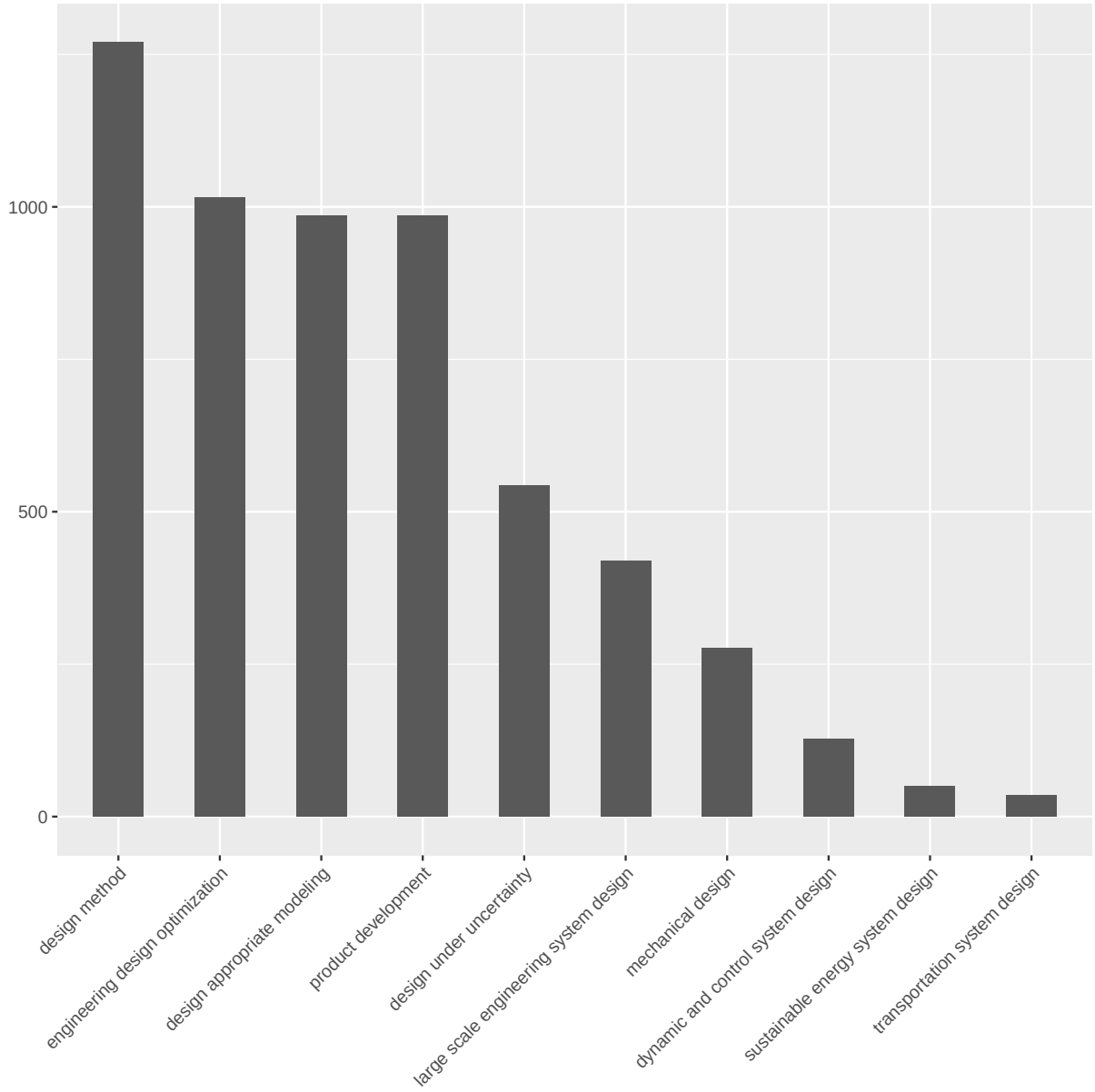



Figure 2.1: Main topic frequency plot

### 2.5.2 Frequency-Based Model

Figures 2.1 and 2.2 illustrate DAC topic frequencies. Each key phrase can be associated both with main and sub-topics. For instance, a key phrase "optimization method" may be connected to the main topics *engineering design optimization* and *design method*, and the sub-topics *search, numerical methods*, and *design process*. The top five main DAC topics

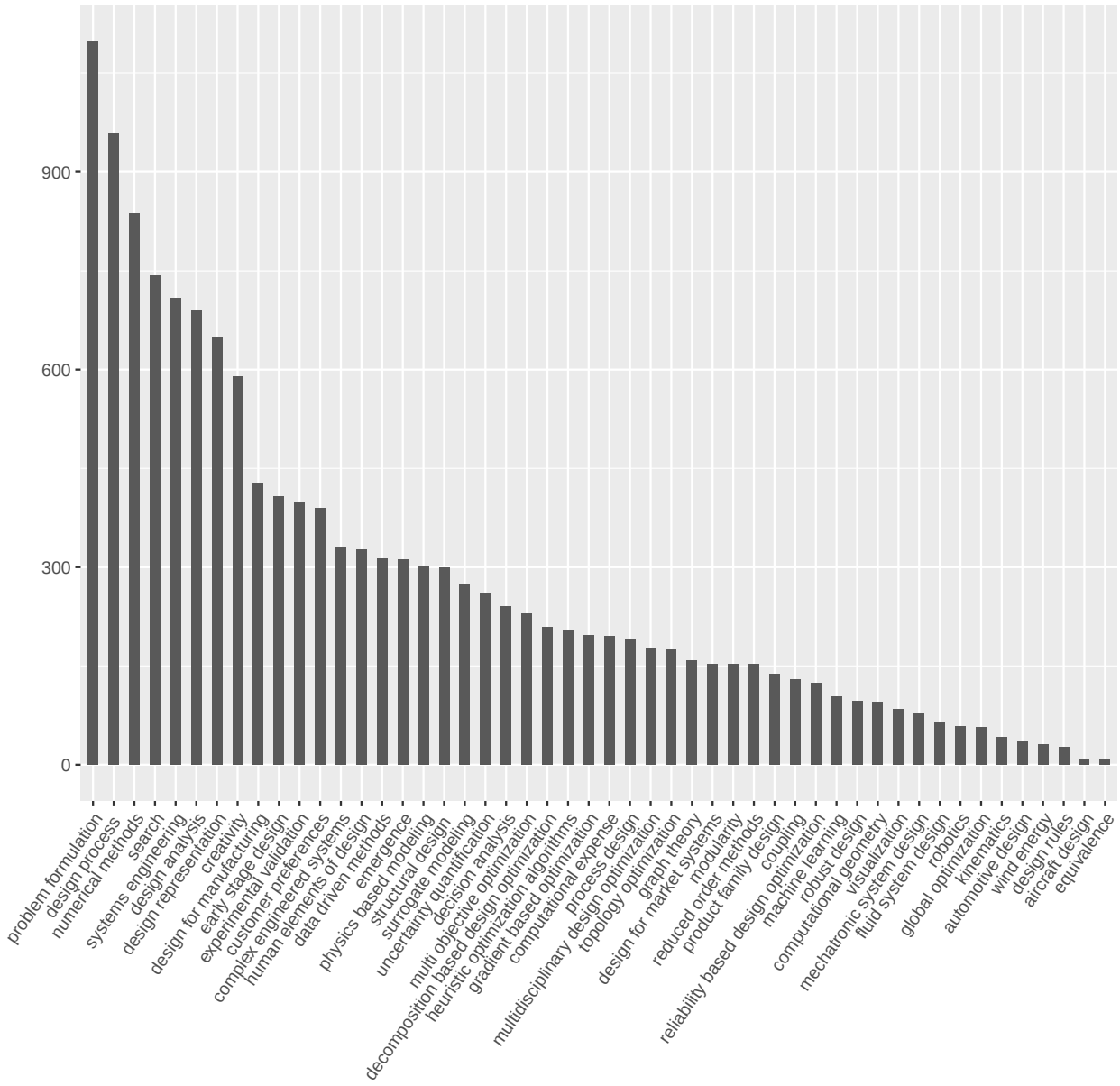



Figure 2.2: Sub-topic frequency plot

are design method, engineering design optimization, design-appropriate modeling, product development, and design under uncertainty. Note that some topics correspond to design methods or other general topics, whereas others correspond to specific application domains. Research areas such as sustainable energy systems and transportation system design are less frequent when considering the overall length of the study period (see Fig. 2.1). Either of these topics represents less than 3% of the collection. Chronological evolution of topics


Figure 2.3: Main topic chronological evolution

is illustrated in Fig. 2.3. From this visualization we can see that interest in sustainable energy systems is increasing in recent years, whereas interest in transportation systems has fluctuated. Visualizations for sub-topics are presented in Figs. 2.2 and 2.4. Aircraft design, design rules, and equivalence became less frequent in some years, but fluid system design and wind energy have increased in frequency in recent years. Wind energy, appearing first in 2009, accounts already for nearly 2% of the DAC article collection.

Correlation plots are shown in Figs. 2.5 and 2.6, where each cell grayscale density indicates the correlation strength between two topics. Four main topics, including design under uncertainty, design-appropriate modeling, engineering design optimization, and large-scale engineering system design are highly correlated with each other. Product development indicates a strong correlation with design method. Dynamic and control system design and transportation system design are also strongly correlated. It is natural to associate sustainable energy system design with both dynamic and control system design and transportation

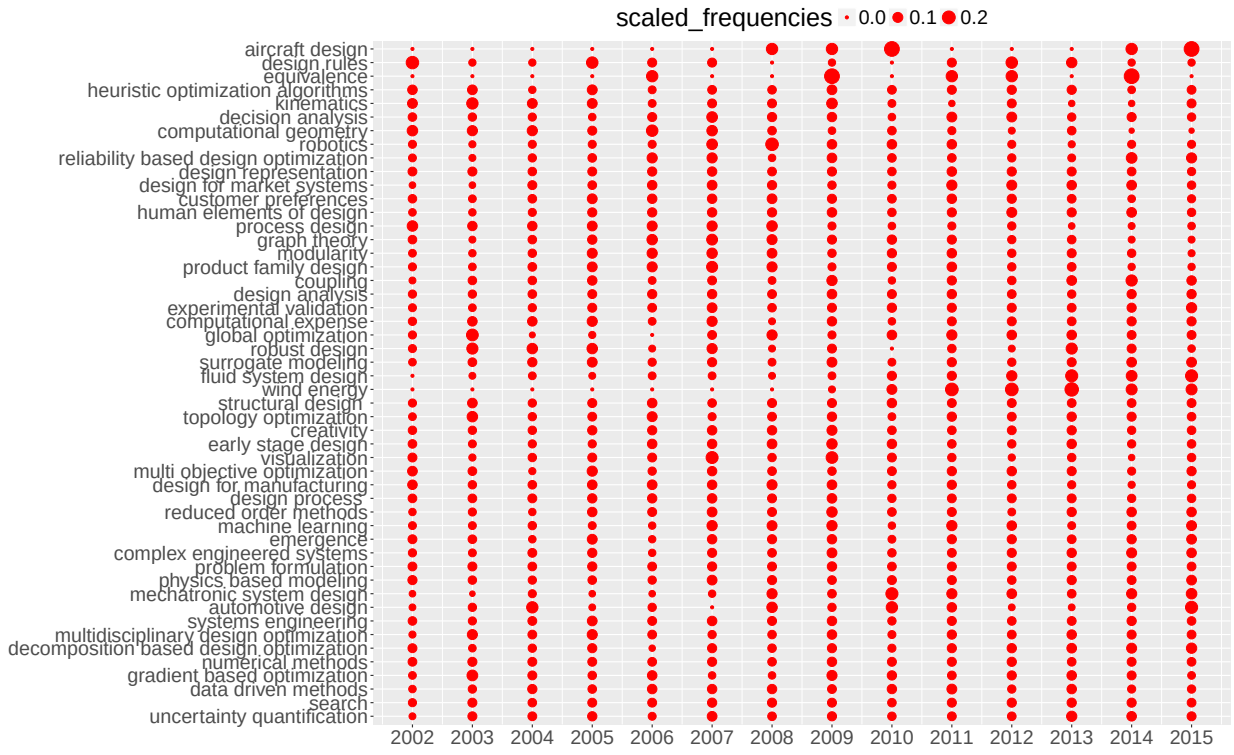



Figure 2.4: Sub-topic chronological evolution

system design, but this topic is relatively isolated from other topics in the DAC literature. Mechanical design is relatively isolated (weakly correlated with other topics). Research gaps and opportunities may be inferred from correlation matrices. For instance, product development and design under uncertainty are core topics, but have weak correlation. There is also weak correlation between transportation system design and design under uncertainty, which could be an opportunity for further investigation. These recommendations should be qualified; the presence of a weak correlation is a proxy for determining interfaces that the engineering design research community should investigate further. Good reasons may exist for not exploring a particular interface. In addition, lack of correlation in this dataset does not necessarily mean no correlation exists in engineering design literature in general. Additional articles that involve a particular interface may exist in other publication venues that are more appropriate for that interface than DAC (e.g., particular applications).

Sub-topic correlations are illustrated in Fig. 2.6. A strong correlation exists between

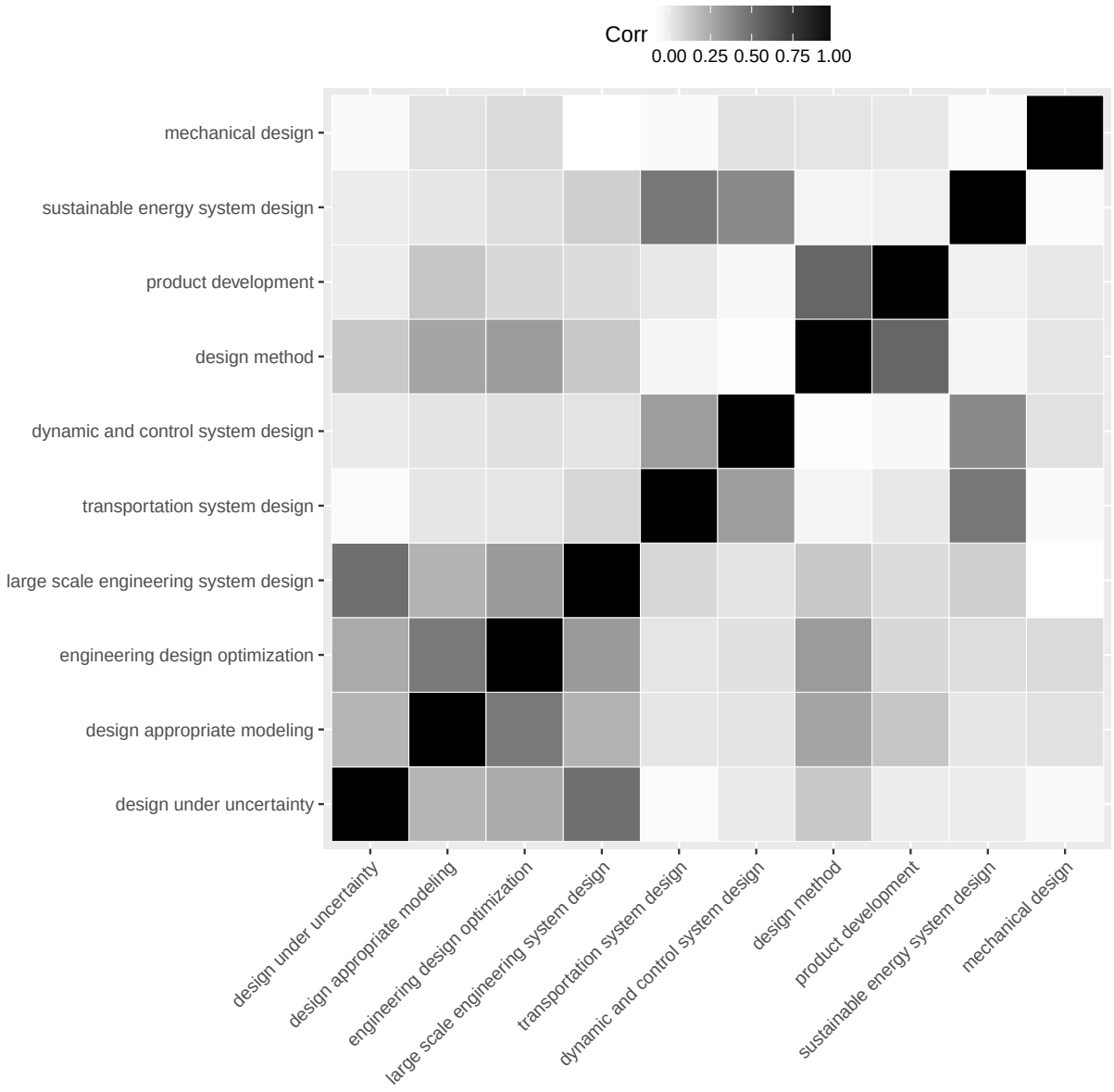



Figure 2.5: Main topic correlation

product family design, modularity, and graph theory. Strong correlations also exist between several pairs of subtopics, including: topology optimization/structural design, wind energy/fluid system design, and early stage design/creativity. Several subtopics have weak correlations, including robotics, kinematics, equivalence, design rules and aircraft design. Some of the weak correlations were unexpected, such as machine learning. General research interest in machine learning has increased in recent years, but only after 2015 began to

show a more significant presence in DAC. Based on our data, engineering design using artificial intelligence—particularly neural networks and deep learning—has not been studied extensively within the DAC community. A few strong contributors have made progress in this area. Topics related to machine learning that appear in the data set include classifiers, clustering, principle components analysis, and neural networks. Perhaps one of the more interesting features of the machine learning sub-topic is that it is not yet well-connected to other subtopics. There may be an important opportunity to create new links between machine learning techniques and other DAC topics of interest.

Figure 2.7 illustrates correlations between main and sub-topics. The topics of transportation system design, sustainable energy system design, and dynamic and control system design are relatively specialized; only a few sub-topics are correlated with them. A few entries with very low correlation levels (white/near-white in the figure), corresponding to sub-topics such as customer preferences and design for manufacturing, may be important interfaces to explore by DAC researchers.

For each topic, we calculated the average in-degree of the citation network and sorted these in a descending order (see Fig. 2.8). Mechanical design and transportation system design have low average in-degree values. The average in-degrees of sub-topics are plotted in Fig. 2.9. Topology optimization and structural design are well-established topics, but their citation strength is not high relative to topic maturity. Many optimization-based topics, including decomposition-based optimization, multi-objective optimization, multi-disciplinary design optimization, and global optimization, also have low average in-degrees. One factor may be some of these articles being revised and published as journal articles, and the journal version is cited instead of the DAC article. This cannot be verified from the current data set.

Wind energy has the ninth highest average in-degree, even though the overall topic frequency is low. In other words, DAC researchers in wind energy tend to cite DAC articles very well. Design rules, visualization, kinematics, machine learning, automotive design, and

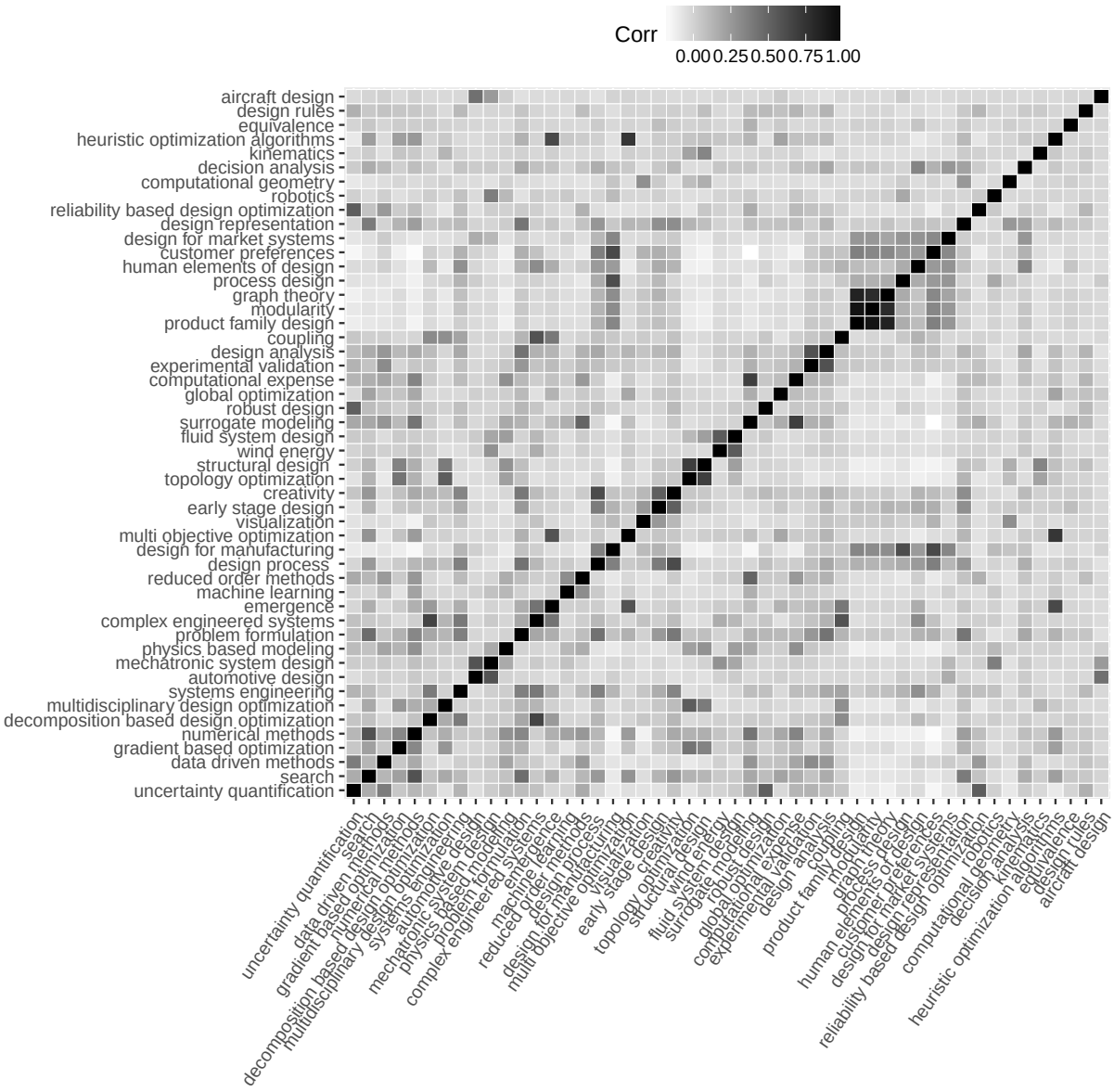



Figure 2.6: Sub-topic correlation

equivalence show a low intra-DAC citation rate.

Figure 2.10 shows the citation network for just the DAC articles belonging to sustainable energy system design. This topic has an average in-degree of 0.5, indicating a very sparse citation network. In general, a sparse citation structure was observed across the complete DAC literature network, indicating an opportunity for improvement. Some of this sparsity may be due to citation of journal versions of DAC articles.

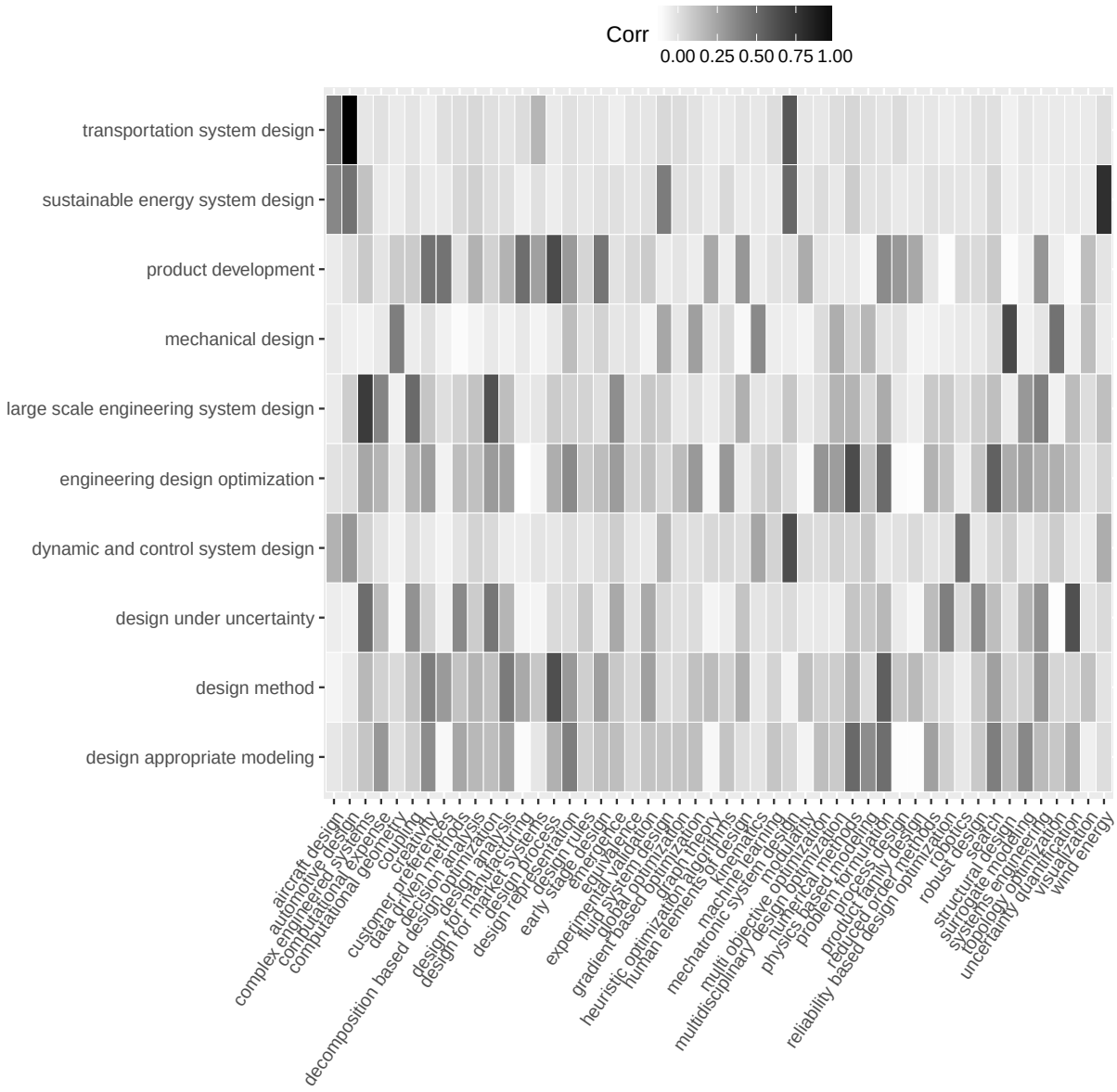



Figure 2.7: Main topic and sub-topic correlation

### 2.5.3 Association Rule Learning

Association rule learning can be applied to literature networks if each topic and article are treated as an item and a transaction, respectively. We set support and confidence thresholds at 0.004 and 0.5 for both main and sub-topics, generating 476 association rules for the main topics (Fig. 2.11). A grouped matrix-based visualization organizes the antecedent (LHS) and consequent (RHS) using a grouping of rules via clustering [105]. The ball plot of Fig. 2.12

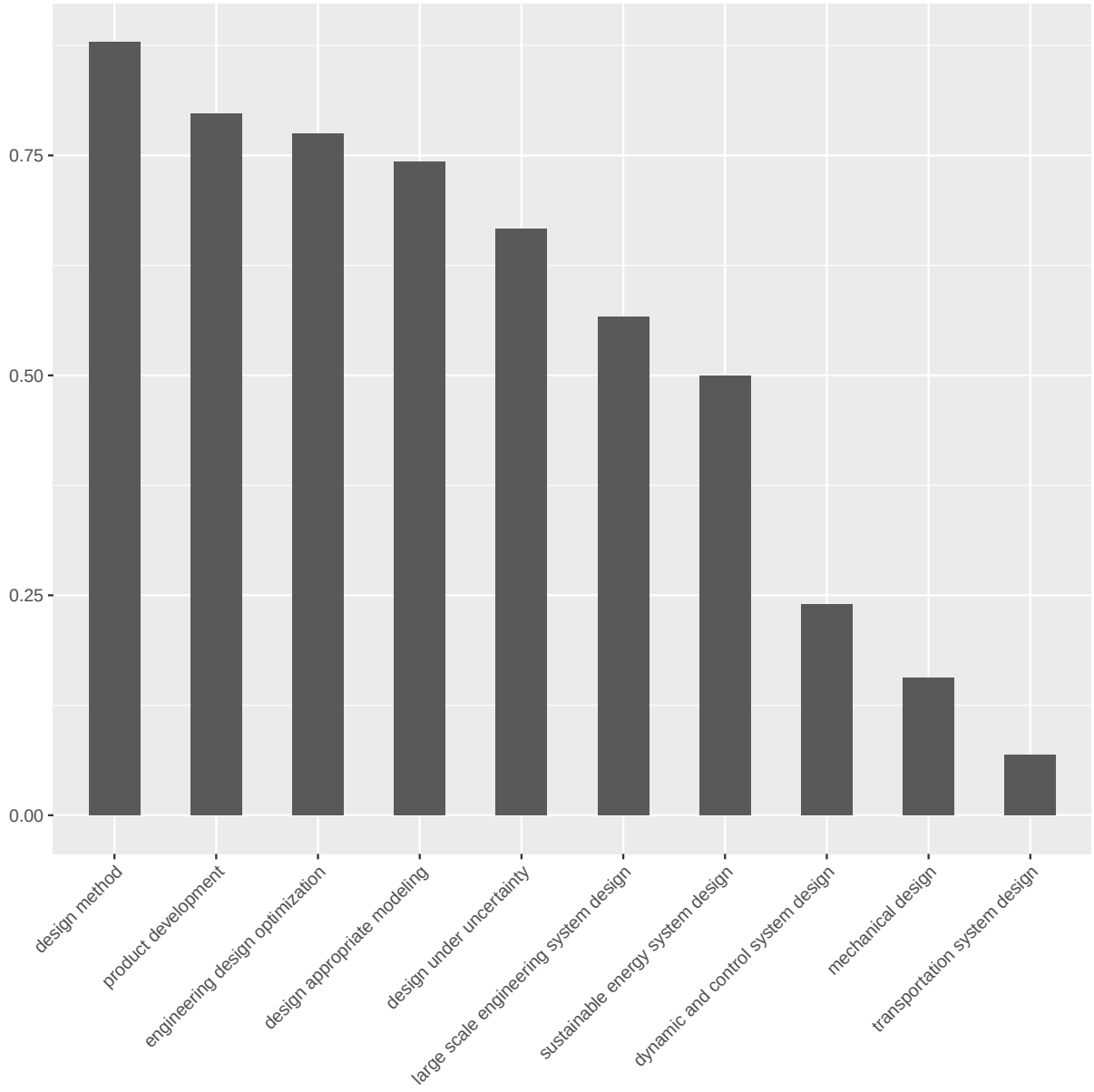



Figure 2.8: Average in-degree for main topics

includes antecedent groups (LHS) as columns and consequent groups as rows (RHS). Note that there are high lifts occurring at Rows 1–3, indicating a strong dependence between transportation system design, sustainable energy design, and dynamics and control system design. Topics in Rows 4–6 also have strong associations because of higher confidence and lift greater than one. The lift of product development is close to one, indicating some independence, also observed previously (Fig. 2.5).

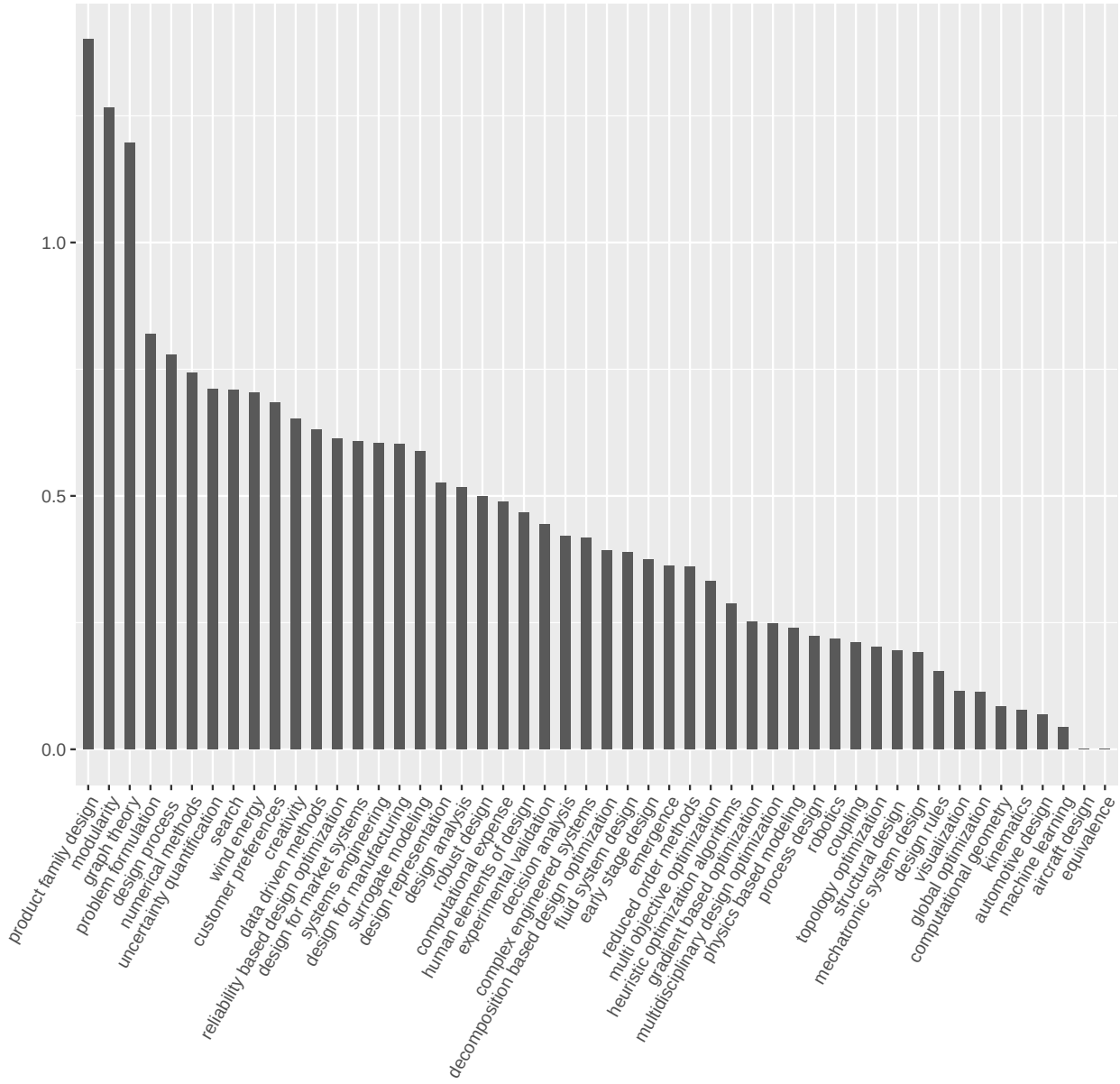



Figure 2.9: Average in-degree for sub-topics

Several rules with high confidence and lift are listed in Table 2.2. These rules cover topics with fewer occurrences (transportation system design, mechanical design, and sustainable energy system design). In Rule 4, LHS {large scale engineering design, design method, mechanical design} implies RHS {design under uncertainty}. While the support of LHS is only 0.0216, the confidence is 0.878, meaning 87.8% of the time LHS occurs in articles that also contain RHS. The lift of 2.697 indicates that the LHS and RHS have a strong

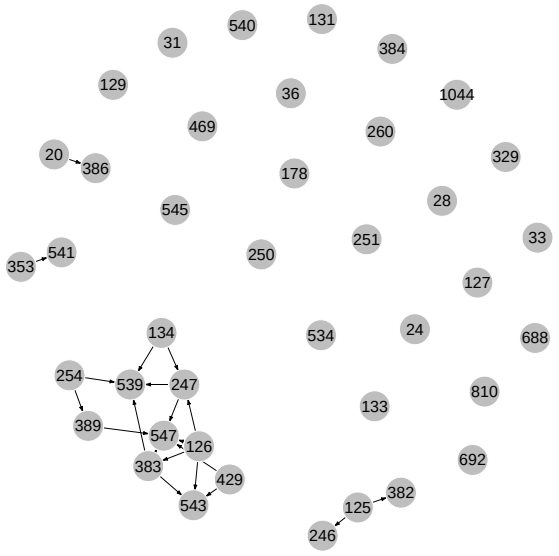



Figure 2.10: Citation network for sustainable energy system design

association. The probability of finding the RHS in articles that also contain the LHS is greater than the probability of finding RHS only by 169.7%. In other words, when one works on large scale engineering system design, design methods, and mechanical design, one may likely also consider design under uncertainty. The rest of the rules may be interpreted in a similar way.

The rules presented here can help extract insights not revealed via correlation plots. For instance, the paired correlation between transportation system design and product development is quite weak. Rule 1, however, implies that product development together with transportation system design and dynamic and control system design has a strong association with sustainable energy system design. Design-appropriate modeling (Rule 2) and engineering design optimization (Rule 3) also exhibit dependence on dynamic and control system design. In Rules 4 and 5, mechanical design co-occurring with certain other topics tends to indicate design under uncertainty. Another benefit of using association rules is to

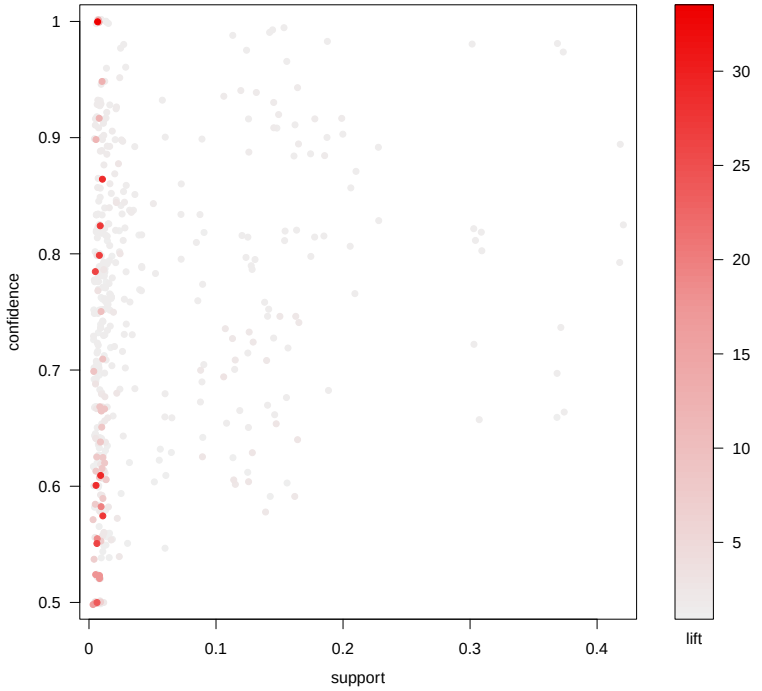



Figure 2.11: Scatter plot for association rules of the main topics

bundle the dependence among multiple topics for ready examination. Interested readers may find more rules extracted from the data set via corresponding online materials [97].

Sub-topics were analyzed, with the minimum support and confidence levels set at 0.004 and 0.5, respectively, for a total of 57,282 rules. Figures 2.13 and 2.14 are the scatter plots for the sub-topics and the group matrix visualizations. Among the sub-topics, the highest lift occurs for automotive design, wind energy, robotics, fluid system design, visualization, and mechatronic system design. In particular, wind energy has strong association rules with fluid system design and mechatronic system design. Machine learning associates not only with traditional optimization-based topics, but also decision analysis, early stage design, and process design. Design process has the least lift, but the median lift is still greater than 1, indicating a reasonable co-dependence with the other topics. We specifically examined rules relevant to the topics (equivalence, aircraft, design rules, kinematics, and global optimization) that performed either poorly in the correlation plot in Fig. 2.6, or were not shown in


Figure 2.12: Group matrix-based visualization for main topics

the grouped matrix. No rules were found on equivalence or aircraft design. A subset of 20 association rules on design rules, kinematics, and global optimization are reported in Table 2.3. Structural design can be inferred from kinematics and global optimization (Rules 1–2). The topic of design rules is not prevalent in the DAC literature, but it can produce useful insights by combining with other topics. For example, design rules could imply topics such as uncertainty quantification, computational expense, surrogate modeling, and so on (Rules 4–8). A number of topics can also set up relations with global optimization, as seen in Rules 8–20, although global optimization does not exhibit strong correlations in Fig. 2.6.

Table 2.2: Association rules for main topics

| | Rules | Support | Confidence | Lift |
|---|---|---|---|---|
| 1 | {transportation system design, dynamic and control system design, product development} => {sustainable energy system design} | 0.005 | 1.000 | 33.360 |
| 2 | {design appropriate modeling, transportation system design, sustainable energy system design} => {dynamic and control system design} | 0.007 | 1.000 | 13.031 |
| 3 | {engineering design optimization, transportation system design, sustainable energy system design} => {dynamic and control system design} | 0.008 | 1.000 | 13.031 |
| 4 | {large scale engineering system design, design method, mechanical design} => {design under uncertainty} | 0.022 | 0.878 | 2.697 |
| 5 | {engineering design optimization, large scale engineering system design, mechanical design} => {design under uncertainty} | 0.023 | 0.844 | 2.594 |

### 2.5.4 Propagation Mergence (PM)

Table 2.4 reports the ten most influential articles according to the PageRank-based algorithm in the PM approach. Design under uncertainty, reliability-based design, and product family design had strong impacts on the corpus. We also summarized twenty significant clusters identified by PM in Table 2.5. (Clusters formed by PM were labeled based on human judgment. While each cluster may correspond to multiple topics, we selected a consensus "most likely" label for each.) Cluster exchange represents interactions between clusters via the citation network.

Cluster 1 (design under uncertainty) spans a large structural space in the network. Reliability-based design typically involves both uncertainty and optimization, and hence this cluster is often exchanged with others.

Cluster 2 is product family design. Research efforts represented in this cluster favor a

Table 2.3: Association rules for the sub-topics

| | Rules | Support | Confidence | Lift |
|---|---|---|---|---|
| 1 | {gradient based optimization, kinematics} → {topology optimization} | 0.005 | 0.900 | 8.628 |
| 2 | {gradient based optimization, kinematics} → {structural design} | 0.006 | 1.000 | 5.579 |
| 3 | {data driven methods, structural design, design rules} → {reliability based design optimization} | 0.004 | 1.000 | 13.452 |
| 4 | {reliability based design optimization, design rules} → {uncertainty quantification} | 0.005 | 1.000 | 6.391 |
| 5 | {problem formulation, reduced-order methods, design rules} → {computational expense} | 0.004 | 1.000 | 8.554 |
| 6 | {reduced order methods, design rules} → {surrogate modeling} | 0.005 | 1.000 | 6.088 |
| 7 | {design process, design rules} → {design analysis} | 0.010 | 1.000 | 2.421 |
| 8 | {problem formulation, design rules} → {systems engineering} | 0.013 | 0.917 | 2.157 |
| 9 | {reduced order methods, design process, global optimization} → {computational expense} | 0.004 | 1.000 | 8.554 |
| 10 | {emergence, surrogate modeling, global optimization} → {heuristic optimization algorithms} | 0.007 | 1.000 | 8.137 |
| 11 | {global optimization, computational expense} → {surrogate modeling} | 0.010 | 1.000 | 6.088 |
| 12 | {global optimization, experimental validation} → {surrogate modeling} | 0.011 | 0.826 | 5.029 |
| 13 | {global optimization, heuristic optimization algorithms} → {emergence} | 0.015 | 0.893 | 4.789 |
| 14 | {multi-objective optimization, global optimization} → {emergence} | 0.014 | 0.857 | 4.597 |
| 15 | {design process , global optimization, design representation} → {early stage design} | 0.013 | 0.808 | 3.310 |
| 16 | {early stage design, global optimization} → {design representation} | 0.013 | 1.000 | 2.574 |
| 17 | {design process, global optimization} → {creativity} | 0.017 | 0.806 | 2.281 |
| 18 | {data driven methods, global optimization} → {search} | 0.012 | 1.000 | 2.245 |
| 19 | {creativity, global optimization} → {design representation} | 0.015 | 0.862 | 2.219 |
| 20 | {global optimization, experimental validation} → {design analysis} | 0.013 | 0.913 | 2.210 |

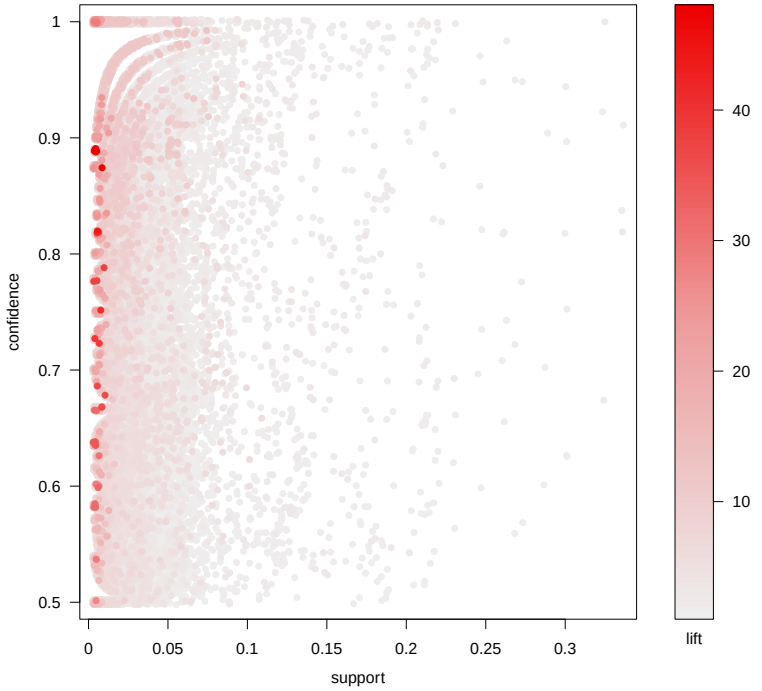


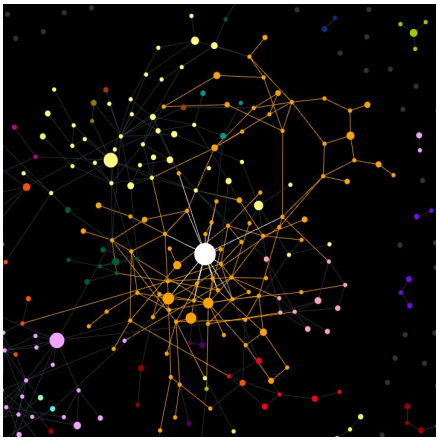

Figure 2.13: Scatter plot for association rules of sub-topics

variety of methodologies and analyses involving product family design, such as optimization (e.g., genetic algorithms), sensitivity analysis, data mining (e.g., cluster analysis), formal concept analysis, decomposition-based approaches, cost modeling, and others.

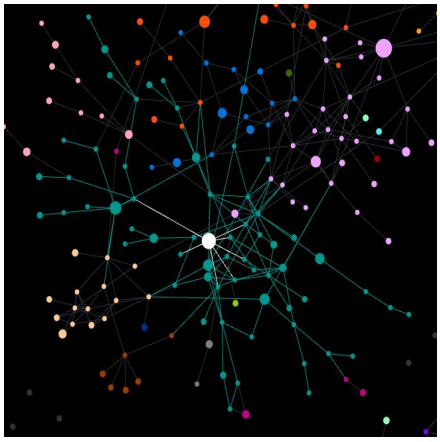
Cluster 3 (surrogate modeling) attempts to construct approximated models for use in design. Key phrases include Kriging, evolutionary algorithms, Monte Carlo simulation, and response surfaces. Reference [248] provided a comprehensive review of metamodeling techniques that has been extended to various optimization formulations including global, multi-objective, multidisciplinary design, and probabilistic optimization. Surrogate modeling actively interacts with a number of domains such as statistics, mathematics, computer science, and engineering design [248].

Cluster 4 (design for market systems) focuses on marketing, user preference, demand, and conjoint analysis. This cluster overlaps extensively with product design, as indicated by the key phrases. In Fig. 2.15(d), Cluster 4 (in pink) is close to Cluster 2 (product family

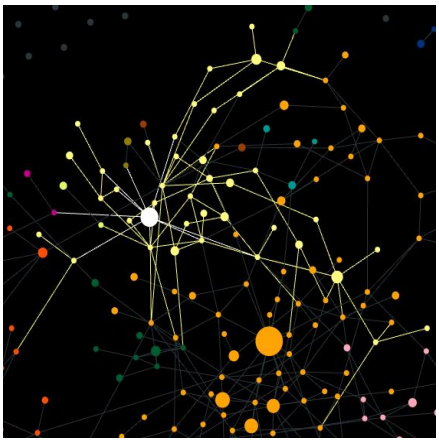

Figure 2.14: Group matrix-based visualization for DAC sub-topics



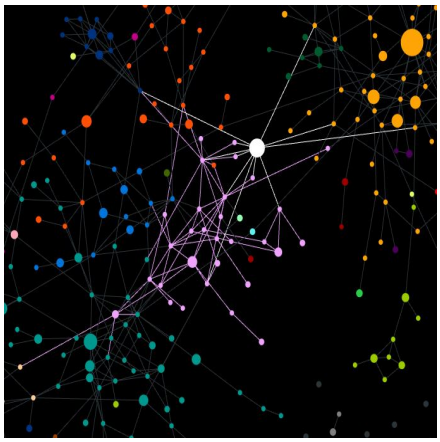
(a) Cluster 1: Uncertainty quantification



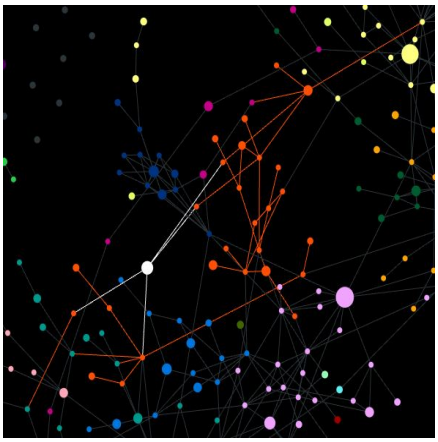
(b) Cluster 2: Product family design



(c) Cluster 3: Surrogate modeling



(d) Cluster 4: Design for market systems



(e) Cluster 5: Visualization

Figure 2.15: Top 5 clusters using PM

Table 2.4: Top 10 influential articles in the DAC citation network

| Year | Title | Author(s) | Score |
|---|---|---|---|
| 2004 | A Single-Loop Method for Reliability-Based Design Optimization | J.Liang, Z.P. Mourelatos, J. Tu | 19.94 |
| 2006 | Engineering Product Design Optimization for Retail Channel Acceptance | N. Williams, S. Azarm, P.K. Kannan | 12.30 |
| 2002 | On Sequential Sampling for Global Metamodeling in Engineering Design | R. Jin, A. Sudjianto, W. Chen | 11.42 |
| 2002 | A Quantitative Approach for Designing Multiple Product Platforms for an Evolving Portfolio of Products | C.C. Seepersad, F. Mistree, J.K. Allen | 9.93 |
| 2003 | Reliability-Based Design With the Mixture of Random and Interval Variables | X. Du, A. Sudjianto, B. Huang | 8.34 |
| 2007 | Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization | J.T. Allison, M. Kokkolaras, P.Y. Papalambros | 8.00 |
| 2004 | A Saddlepoint Approximation Method for Uncertainty Analysis | X. Du, A. Sudjianto | 7.55 |
| 2006 | Flexible Product Platforms: Framework and Case Study | O.L. de Weck, E.S. Suh | 7.33 |
| 2002 | Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design | X. Du, W. Chen | 7.06 |
| 2003 | Design Space Visualization and Its Application to a Design by Shopping Paradigm | G.M. Stump, M. Yukish, T.W. Simpson, E.N. Harris | 6.36 |

design, in green). Due to increased weight on marketing, this cluster was separated from product family design.

Cluster 5 represents visualization, but also has strong citation links to several other major clusters. In Fig. 2.15(e), red citation links are connected to Cluster 2 (product family design, in green), Cluster 3 (surrogate modeling, in yellow), and Cluster 4 (design for market systems, in pink). Cluster 5 is fairly cross-disciplinary as it involves tools for other topics.

Additional knowledge can be obtained by inspecting Clusters 6–20. For example, Cluster 6 addresses architecture design and configuration problems for product family design; Fig. 2.15(b) also shows that it is connected to Cluster 2 (product family design). Interested

Table 2.5: Topic clusters in the ASME DAC citation Network

| No. | Cluster label | Exchange | Extracted key phrases |
|---|---|---|---|
| 1 | design under uncertainty | 3,4,10 | reliability, uncertainty, error, random, reliability analysis, probabilistic, interval, probability, rbdo, input, confidence, variables, failure, analysis, simulation |
| 2 | product family design | 3,4,6 | product family, product, product platform, redesign, commonality, market, cost, components, variety, customer, pattern based, benchmark |
| 3 | surrogate modeling | 1,5 | kriging model, surrogate models, model, approximation,simulation, sequential sampling, response, expensive, computational, moga, metamodeling techniques |
| 4 | design for market systems | 1,2,7 | market systems, profit, product, consumer, price, competitive, retailer, demand model, choice, customer preferences, conjoint analysis |
| 5 | visualization | 2,3,4 | visualization, preference, user, content, product, user generated, interaction, data, clustering, dimensional, consumer, family, structure matrix, preference elicitation, efficient global |
| 6 | modularity | 2, 4, 19 | reconfigurable, reconfigurable system, transformation, adaptive, product, concept generation, system, concept, change, theory, state, changeable, architecture, sio, facilitate |
| 7 | customer preference | 2,4,5 | usage, buck, usage context, vehicle, seating buck, context, choice modeling, attributes, packaging, customer, appraisal, coverage, product, legacy |
| 8 | decomposition-based design | 1,3 | characteristics, assembly synthesis, decomposition, joint, deeper, product design, collaborative, in process, hierarchical, fundamental, decomposition based, optimization, product, product design optimization, adjustability |
| 9 | design process | 2,4 | decentralized, architecture, process architecture, mistakes, distribution, convergence, impulses, design process, process, subsystems, stability, design, group, equilibrium, solutions |
| 10 | robust design | 1,2 | robust design, robust optimization, interval uncertainty, variation, computer experiments, sequential quadratic programming, model uncertainty, sensitivity, parameters, mcro, tolerance, blade |
| 11 | topology optimization | N/A | heuristic gradient projection, mems, fuzzy, hybrid, stress, space frame, solar, frame, water, hgp, hdh, resonator, topology optimization, desalination system, semi isolated |
| 12 | wind energy | 1 | wind, farm, wind farm, wind farm layout, farm layout, turbine, landowners, land, wake, cost, onshore, power, extended, eps, shear |
| 13 | multi-disciplinary optimization | 2 | analytical target, target cascading, analytical target cascading, coordination, atc, network, enterprise, network decomposed, complementarity, subsystems, tolerance allocation, mdo, decomposition |
| 14 | multi-objective optimization | N/A | coordination, decomposition based, co design, plant, partitioning, wave, decomposition, wind, control, energy, subproblems, complementarity, multistage, optimization, power |
| 15 | structural design | N/A | meso structures, shear, pneumatic, non pneumatic, shear flexure, honeycomb, wheel, meta material, material, chiral, layer, metamaterial, properties, cell, wall |
| 16 | developing-world design | 2,6 | developing world, principles, world, water, pump, rural, modular product, demography, sustainable, village, irrigation, safe, communities, energy, alleviate poverty |
| 17 | N/A | 1 | graveyard, one to, consolidation, group, robust design, hybrid, ds, unequal, mapping, inequivalents, heterogeneous, members, heim, hypothetical, genetic algorithm |
| 18 | human element design | N/A | population, civilians, head, anthropometry, user population, restraint, stature, user, secular, body, virtual, dfhv, variables, dimension, accommodation |
| 19 | machine learning | 3,6 | Bayesian Network, classifiers, composite materials, set based, composite, protocol based, cooperative, inclusions, collaborative design, set, stiffness, satisfactory, multi agent, negative, heuristic |
| 20 | phase transitions | N/A | transition, saddle, crystal, nano, phase, surface, periodic, path, metamorphosis, search, nano design, review, energy, pathway, recent |

readers may find Ref. [72] useful, where Ferguson et al. discussed relevant concepts, summarized recent approaches for this topic, and proposed open questions for reconfigurable system design. It is not surprising to see that the key phrases in Cluster 7 (customer preference) overlap with Cluster 4 (design for market systems) and Cluster 2 (product family design). Clusters 8, 13, and 14 (decomposition-based design, multidisciplinary optimization, and multi-objective optimization) are focused primarily on optimization methods. Cluster 9 (design process) often deals with convergence and stability in decentralized design processes and large, complex systems. Cluster 10 (robust design) is very similar to Cluster 1 (design under uncertainty), as robustness is one of several desirable properties for designs involving various types of uncertainty. Clusters 11 (topology optimization) and 15 (structural design) address topology and structural optimization in engineering design. It is interesting to note that Cluster 12 (wind energy) has a citation relationship with Cluster 1 (uncertainty qualification). Cluster 16, adjacent to Cluster 2 (product family design), is developing-world design. An example of this topic is an article by Mattson and Wood that introduces nine design principles for the developing world [173]. Cluster 17 is labeled "undetermined"; distinctness in key phrases, including areas such as robust design, design process, and multi-objective optimization, make it difficult to assign a single cluster label.

One may notice that some clusters are isolated. To understand this observation, recall that we consider only citations within DAC. It is possible that these articles may have cited or been cited by other articles from outside the DAC corpus. For instance, human factors/ergonomics is well-studied in psychology. This helps to explain why Cluster 18 (human element design) is a minor cluster here. Machine learning (Cluster 19) is one of the smallest DAC clusters. This topic has had significant impact outside of DAC (e.g., data science and artificial intelligence). One important recent example is Google's Alpha Go [226]. However, as discussed in [101], the use of specific elements of artificial intelligence and deep learning in design is an emerging research topic within the DAC community, and increased effort toward understanding artificial intelligence in engineering design may be a

fruitful endeavor. More specifically, machine learning has been studied by a relatively small group of researchers publishing in DAC, and currently it does not have strong links to other DAC literature topics, but is emerging as a topic with increased interest and impact.

A number of clusters prefer to exchange with the top three clusters. Specifically, Cluster 2 (product design) is the most favored. The citation data reveal that DAC articles are often influenced by product family design. It is also worth noting that the boundary between two clusters may not be completely clear. As seen in Table 2.5, the key phrases in one cluster may also reflect other clusters. This is not an unexpected phenomenon because articles often involve a mixture of topics. For instance, Akundi et al. developed a multi-objective design optimization method for product design using genetic algorithms [7]; this article was grouped with Cluster 2 (product family design) because the most information in product family design was passed from it into the source nodes in the PM approach. One can determine what clusters are closer by viewing the exchange column.

## 2.6 Discussion

From the authorship network, we identified the five most collaborative authors using degree centrality, and demonstrated that the co-authorship network possesses the small-world property. The results show that DAC authors have become more inclined to collaborate with others as time progressed. Identifying highly-collaborative authors here may be helpful both to DAC newcomers (students, or scholars originally from other communities), and to others seeking to understand DAC network structure. While identifying highly-collaborative authors is an important feature of network structure and is an interesting result, similar information can be obtained via discussion with individuals familiar with the DAC community; we regard other results obtainable only through network analysis as more valuable outcomes of this study (e.g., gaps in possible links between topics, association rules, presence of persistent cliques, nature of collaboration and citation patterns, and temporal trends).

The topic modeling analysis revealed that DAC topics focus primarily on engineering design optimization (multi-disciplinary optimization, multi-objective optimization, decomposition-based optimization, heuristic optimization, global optimization, surrogate modeling, etc.), engineering design methods, product development (product family design, design for manufacturing, customer preferences, configuration, etc.), and design under uncertainty (reliability-based design, robust design, etc.). The overall patterns, trends, correlations, associations, and clustering are important observations, while the most collaborative authors and most influential articles provide complementary insights.

The two approaches for topic modeling used here have differing strengths and weaknesses. In the frequency-based model, human effort was required to complete the topic list. Based only on key phrases, a human expert assigned main topics and subtopics to each phrase. Some main and sub-topics were determined in advance, but others were created when new key phrases were encountered that did not fit the available topics, requiring another review of key phrases with the updated set of topics. Clearly this strategy depends on the particular experience of the human expert making the determinations, resulting in a strong potential for introducing bias. The advantage of the frequency-based model is that it enables simultaneous access to all 1,668 articles, which makes possible mining the topic information and association rules all at once. However, when new DAC articles are available, human experts must repeat the manual process of assigning labels.

The PM approach is an automated method that uses citation relations in addition to text content to improve results. Articles are clustered together using propagation via the citation network. As reviewed in Section 2.2, the current topic modeling methodologies focus mainly on either probabilistic models (natural language processing) or network-based approaches only [187, 189]. For instance, an initial set of DAC topics were explored in our early work using the ToPMine algorithm [69], but this approach was unable to capture the DAC core topics [10]. This motivated us to develop a fundamentally different algorithm (PM) that bridges the gap between natural language processing and citation networks. PM

has been compared with the established modularity-based topic modeling method described in Ref. [24]. Please see Refs. [230, 231] for details of this comparison. One possible weakness of PM is that it can only consider articles in the citation network. A subset of articles in the DAC dataset do not have citations from other DAC articles, and therefore are not present in the citation network, and cannot be considered by PM. However, this shortcoming can be compensated for by using the proposed frequency-based model. The two strategies are complementary to each other, providing a rich strategy for analyzing the literature from a research community.

Future work should include expanding the literature data set. Additions ideally would include articles from other relevant conferences and journals, as well as articles from other disciplines that cite or are cited by the engineering design research literature. Gathering and processing this data, as well as maintaining it, would require a much larger-scale effort than what is represented in this chapter. Such a resource, however, could provide valuable insights into not only internal characteristics of the engineering design research community, but also into understanding the historical impact of other scientific communities on the engineering design research community, and conversely, the impact that the engineering design research community has on other disciplines. A specific complementary analysis strategy that could be used for assessing impact of individual articles is article wake [139].

A related objective that could be supported using similar analytical tools is to learn how ideas evolve from initial observation or concept to accepted knowledge within a scientific community (i.e., formation of scientific consensus). For example, Herrara et al. present idea networks, where nodes represent scientific concepts, that can be used as a tool to understand evolution of scientific fields [113]. Centola presented techniques for understanding how ideas, norms, and practices propagate in social networks (such as research communities) [43]. An important element of forming scientific consensus is recognizing the possibility that some ideas accepted by the community may not be correct. Greenberg highlighted a pitfall of insular communities: the risk of becoming an echo chamber, wherein citation

bias, amplification, and invention combine to bolster "the conversion of hypothesis into fact through citation alone" [93]. As a case of "spontaneous hypothesis generation", the design community has seen persistent citation of Symon and Dangerfield [237] as stating that 70–75 percent of all product costs are decided during the conceptual design stage (emphasizing the importance of early-stage design). A detailed analysis presented by Barton et al., however, revealed that the validity of this claim cannot be supported by evidence [18]. How can we be confident that the design research community accepts ideas only when supported by evidence? After learning more about how ideas evolve within design research, as a community we could discuss whether different processes or norms could help improve research outcomes and impact.

## 2.7 Conclusion

In this chapter, we analyzed the ASME DAC authorship and citation networks for the years 2002–2015. This effort revealed DAC community structure, patterns of collaboration, trends through time, and potential opportunities for research activity. The data set was limited to DAC articles; a more complete analysis would include other conference and journal articles that are relevant to the engineering design research community. Such an analysis would require a significantly larger collaborative research effort. The methods and results presented here may serve as a foundation for future larger efforts.

We used two approaches to topic modeling. The first was a frequency-based model. Topic frequencies and topic evolution were described. Citation analysis was also performed for each topic. Correlation matrices and association rules helped to reveal topic relationships and dependence. The second topic modeling approach, PM, was created and used to identify influential articles and to group DAC literature topics into larger clusters, reflecting core research interests in this research community. The work presented in this chapter is descriptive in nature, and is intended for use by the DAC community to help inform collec-

tive decisions. Further study is needed to arrive at more complete knowledge of the DAC research community. These results are an early milestone in efforts to understand the nature of engineering design research. Here we offer a few candidate interpretations for community direction, with the caveat that these our intuitive opinions based on limited analysis, and that these interpretations should be revised as more information emerges and community discussion occurs.

We see trends in focusing efforts on applications and topics—such as an increased use of DAC theory and methods to advance sustainable energy systems and design for the developing world—as important developments. DAC should continue to identify important societal needs where advances in engineering design can enhance efforts to address these needs. We have identified missing links between existing DAC topics and parts of the community that, if attended to, may enhance community impact and lead to more thorough understanding of research topics. For example, transportation system design and machine learning are topics that have intuitive connections to other DAC areas, but at present are fairly isolated in the literature networks. It may make sense for researchers who study design for the developing world to collaborate with those with expertise in more general product design and development. Human factors is another topic that does not have a strong presence in the DAC literature, but may be important for realizing real-world impact of DAC research.

Data science is an area of intense research activity in recent years, and these advances could be leveraged in a variety of ways to advance design methods and accelerate generation of design knowledge. While machine learning and related topics have existed in the DAC literature for some time, they are not well linked (at least in the DAC network) to other topics. Efforts to integrate data-driven research efforts with both established DAC topics as well as broader advances in data science could enhance impact and lead to a richer understanding of engineering design. Others have identified leveraging new capabilities in artificial intelligence and data science as a promising path toward breakthrough developments in the engineering research community [197].

One of the primary objectives of this work was to provide information that could be used by DAC community members to help support collective efforts to guide community priorities and direction, make recommendations for adjustments to research activity, and to be used by interested individual researchers when evaluating their current and future contributions. This and similar analyses could be used within community visioning efforts, such as the recent NSF design and systems program workshops [70, 125]. This work is descriptive as opposed to normative. Value judgments are outside the scope of this chapter, and may be better addressed through collective efforts. Here we introduced new methods for topic modeling, and acknowledge that rigorous analysis and comparison is needed before definitive statements can be made regarding the nature and value of these methods. Literature network analysis in design research is an emerging topic, and there are a variety of rich research opportunities awaiting us. We hope the work here serves as a foundation for further efforts to analyze the nature of engineering design research, and invite the community to join with us in this work.

It has been observed that machine learning is not well-linked to other research topics within the engineering design community, as seen in Fig. 2.6 and Table 2.5. The aim of this dissertation is to address this research opportunity in part. In the next chapter, a homogeneous system topology optimization problem based on hear conduction will be presented, and it will be shown how a machine learning technique can be applied to extend design capabilities. Specifically, a variational autoencoder (VAE) will be used to develop an abstract design representation that supports solution of a problem that cannot be solved directly using established methods. The objective is to create a novel design framework based on design data generated from design optimization studies, rather than the conventional design approaches based on experience, intuition, or established optimization methods. It is hoped that in addition to producing novel designs and solving new problem classes, this new general approach will also help to accelerate the design process for a wider range of unprecedented systems.

# Chapter 3

# HOMOGENEOUS SYSTEM TOPOLOGY DESIGN

**Collaborative Acknowledgement:** In the work presented in this chapter, **Danny Lohan** provided the capabilities to generate heat conduction topological design and performance data; **Ruijin Cang** helped with the literature review and augmented VAE implementation; **Prof. Max Yi Ren** helped review the chapter and contributed several important ideas that improved the technical implementation.

## 3.1 Introduction

Through the topic analysis in the previous chapter, one can identify research gaps and opportunities in the engineering design research literature. While machine learning and artificial intelligence techniques have proven to be extremely useful in both academia and industry, they are not yet well-established in design research in an integrated way. Relatively few links to other topics were observed within the design research literature. This chapter addresses this gap in part by investigating a new approach for homogeneous system topology optimization that leverages existing machine learning methods in new ways.

In this chapter we propose an indirect low-dimension design representation to enhance

topology optimization capabilities. Established topology optimization methods, such as the Solid Isotropic Material with Penalization (SIMP) method, can solve large-scale topology optimization problems efficiently, but only for certain problem formulation types (e.g., those that are amenable to efficient sensitivity calculations). The aim of the study presented in this chapter is to overcome some of these challenges by taking a complementary approach: achieving efficient solution via targeted design representation dimension reduction, enabling the tractable solution of a wider range of problems (e.g., those where sensitivities are expensive or unavailable). A new data-driven design representation is proposed that uses an augmented Variational Autoencoder (VAE) to encode 2D topologies into a lower-dimensional latent space, and to decode samples from this space back into 2D topologies. Optimization is then performed in the latent space as opposed to the original discretized design space. Established topology optimization methods are used here as a tool to generate a data set for training by changing problem conditions systematically. The data is generated using problem formulations that are solvable by SIMP, and that are related to (but distinct from) the desired design problem. We further introduce augmentations to the VAE formulation to reduce unhelpful scattering of small material clusters during topology generation, while ensuring diversity of the generated topologies. We compare computational expense for solving a heat conduction design problem (with respect to the latent design variables) using different optimization algorithms. The new non-dominated points obtained via the VAE representation were found and compared with the known attainable set, indicating that use of this new design representation can simultaneously improve computational efficiency and solution quality.

## 3.2 Literature Review

### 3.2.1 Background

Over the past few decades, topology optimization methods have been applied to a wide range of domains for the purpose of material layout optimization. Examples include the early seminal work based on the homogenization approach [20], as well as various notable extensions for engineering problems with different governing equations and responses [225]. In the domain of heat transfer, topology optimization has been implemented for problems in conduction [88, 265], convection [3, 119], and conjugate heat transfer [9]. For a recent review of heat transfer topology optimization, readers are referred to Dbouk [55], where numerous examples of successful topological optimization studies are presented.

While topology optimization methods are becoming mature in many applications, some important limitations remain. Some notable shortcomings are reviewed in Sigmund et al. [225] and Deaton et al. [56]. For instance, solution performance typically depends on the choice of the starting material distribution, the spatial discretization, and the filtering mechanism [161, 225, 236]. In addition, efficient solution using established methods is often limited to specific problem types (e.g., objective functions that lend well to efficient sensitivity calculations). These and other limitations motivate additional complementary strategies for homogeneous system topology optimization.

Solid Isotropic Material with Penalization (SIMP) [20], an established density-based topology optimization strategy, capitalizes on problem structure and elegant sensitivity calculations to support efficient solution of large-scale problems. An alternative strategy to achieve computational efficiency, while expanding the set of solvable problem types, is to use indirect design representations. Instead of optimizing material distribution directly, we optimize parameters that determine material distribution through a particular mapping. If the indirect representation has lower dimension than the original direct representation, the

resulting reduced optimization problem dimension can help reduce solution time. A drawback of this strategy is that reduced-dimension representations may limit the set of designs that can be accessed, placing importance on identifying or creating mappings that provide targeted coverage of important design space regions.

Indirect design representations have been used recently for structural and thermal system optimization, using generative algorithms as a basis for design representation mapping. Khetan, Lohan, and Allison introduced a design abstraction for truss topology and geometry optimization using a cellular division algorithm based on map L-systems [134]. Lohan and Allison have proposed solutions to problems of scalability in design representation and localized design dependency [133,134,159–161]. Lohan et al. presented a generative design algorithm (GDA) as the design representation for generating dendritic topologies [161]. These early examples are promising, but all rely on creative identification or construction of an effective generative design representation. Reliance on expertise, intuition, and existing candidate representations limits generalizability. What happens in the case where established methods (such as SIMP) cannot solve the desired problem, and an effective indirect design representation cannot be readily identified? Here we propose a data-driven strategy where new indirect design representations are constructed based on design data, independent of intuition or existing representations.

Significant progress has been made toward extracting knowledge from design data. For example, Matthews et al. [172] developed a method for extracting and verifying design knowledge from design databases, including relationships between components. Fuge et al. [81] introduced a model to predict successful design approaches from features of design problems. Ren et al. discussed the potential to learn either a constrained design space [211], or a more efficient optimization algorithm [222] from crowdsourced design solutions. These previous efforts, however, are largely based on historical design data, and do not address the problem of constructing effective indirect design representations. Here we focus on design of potentially unprecedented systems (i.e., no historical data available), and on how to use

data-driven methods to construct effective indirect design representations.

Strategies based on historical data are descriptive in nature. For example, knowledge or heuristics derived from historical design data may reveal how expert designers reason through design decisions, but cannot provide new insights into alternative designs that may perform better. Historical data is fixed; we cannot design experiments to generate new data that could provide better insights. Here we propose a strategy where a systematically-specified set of design optimization problems is used to generate design data for further analysis. This frees us from the limitations connected to data derived from existing designs, allowing us to address unprecedented systems, and supports a process that is more normative in nature (i.e., how *should* the system be designed, not how has it been designed). This strategy also supports the use of designed experiments to generate design data, opening up more possibilities for analysis.

The demonstration example used here involves design of a heat conduction system in a fixed-volume domain. The desired design problem is to maximize power density while satisfying maximum temperature constraints. A related problem is solved efficiently using SIMP (minimize thermal compliance, subject to conductive material volume constraints) with a range of different boundary conditions and volume constraints to produce a wide variety of conductive material topologies. These form a manifold in the topology space that is governed implicitly by the optimality conditions. When the number of parameters that control the boundary conditions is limited, one may expect the manifold to be low-dimensional in comparison with that of the topology space. An analytical characterization of the manifold will therefore be expected to reduce optimization solution expense by circumventing the scalability issue common to the calculation of design sensitivities with respect to a meshed topology.

A significant assumption made here is that the SIMP problem formulation is aligned well enough with the desired problem formulation. If this assumption holds, it is expected that SIMP-derived data may be used to construct targeted indirect design representations that

are useful for solving the desired problem. Results shown later in this chapter support the validity of this assumption. Not only can solution via the indirect representation identify non-dominated solutions obtained from the SIMP-derived data, but it is shown to result in new non-dominated solutions for the desired problem formulation.

### 3.2.2 Framework for Design Method Comparison

While the existing strategies (such as SIMP) have been used widely to solve topology optimization problems, they have some limitations as discussed above. Here we provide a framework that helps to relate the proposed VAE-based design method to established topology optimization methods, with the objective of explaining the motivation and reasoning behind this new strategy. In essence, we seek to utilize a more accurate problem formulation, but maintain reasonable computational expense by making concessions in solution accuracy.

One way of describing a design method is to explain how a design problem is formulated, as well as how the problem is to be solved. Problem formulation and solution decisions are coupled; the solution strategy often affects design problem formulation decisions, and vice versa. Figure 3.1 shows a conceptual three-axis design problem formulation space. Formulation decisions must be made along three dimensions: design representation (how will design candidates be expressed?), comparison metrics (how will design candidates be compared?), and predictive modeling (how can we assign values to comparison metrics for design candidates?). This problem formulation framework was first presented in Ref. [165]. The origin of this formulation space represents the substantive rationality solution to the *ideal*, but unattainable in practice, design problem formulation. Such a formulation would require the design representation to be free of restrictions or assumptions, a perfectly realistic comparison metric that expresses true design utility, and means to evaluate this comparison metric in a way that matches reality perfectly. Design formulations that closer to the origin are expected to result in better design outcomes, but are also expected to require greater

solution expense. This framework can used as a theoretical basis for the reasoning behind the indirect design representation method for topology optimization presented here.

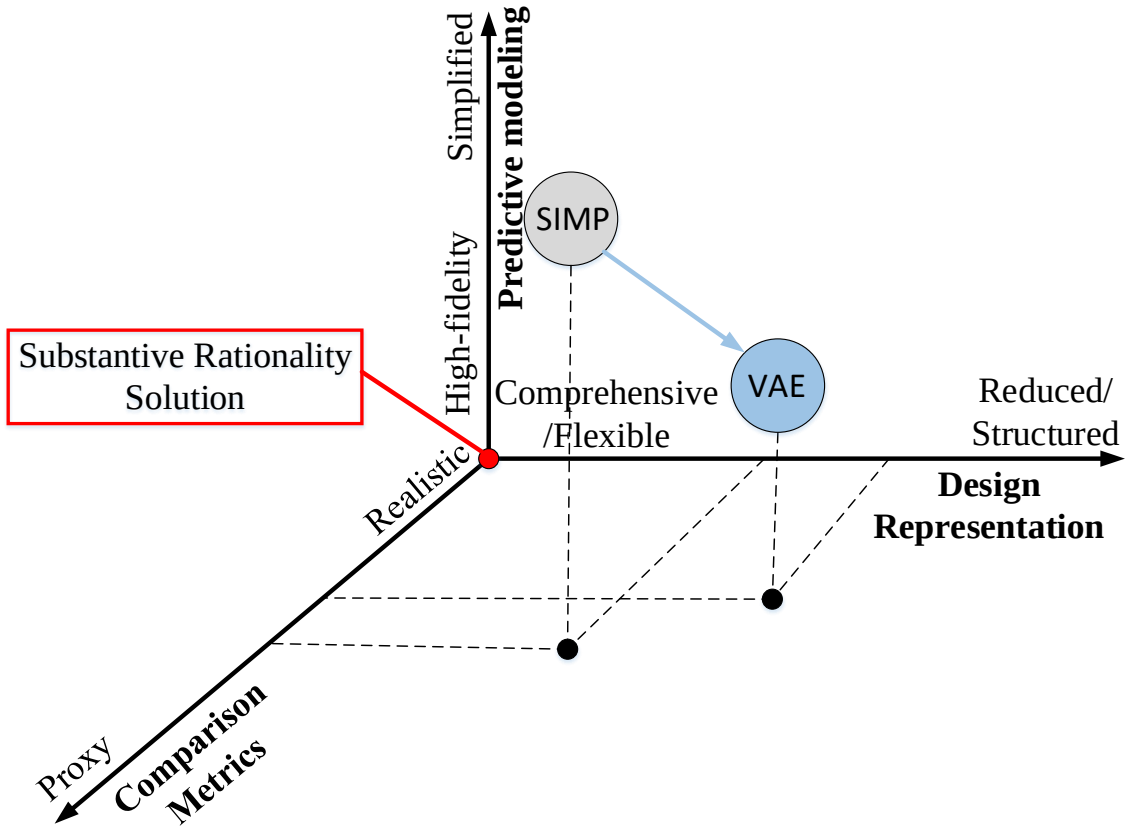



Figure 3.1: Design problem formulation framework for comparison of alternative design methods; SIMP and VAE topology optimization strategies are compared here

In the SIMP method, binary variables that describe material existence in each spatial voxel (or pixel for 2D problems) are transformed into continuous material distribution variables (relaxed), enabling use of efficient gradient-based methods. SIMP uses a direct design representation (moving towards the origin, especially for high-resolution discretizations), but the computational solution expense increases linearly with problem dimension when using a direct design representation as described. Efficient SIMP solution also requires the underlying optimization problem to have analytical design sensitivities, and thus are usually limited to problems with differentiable models. In addition, derivation of the sensitivity for

nonlinear problems (e.g., in their material constitutive models or due to large deformations) can be problem dependent and time consuming, often leading to simplification of models and thus inaccurate solutions.

Using the problem formulation framework illustrated in Fig. 3.1 as a basis for discussion, we can say that SIMP formulations are fairly close to the origin along the design representation axis (i.e., toward the left). Due to the limitations on predictive models, and therefore available comparison metrics, we can say also that SIMP formulations are farther away from the origin along the other two axes. This clarifies an opportunity for new types of design methods. Instead of making concessions on modeling and comparison metric choices, we would like to use certain objective functions and constraints directly that are not compatible with SIMP or other similar established topology optimization methods. In other words, we seek a formulation that moves down and back from the location of SIMP formulations in the conceptual formulation space. If possible it would be ideal to do this without degrading design representation. Due to the maturity of SIMP, it is unlikely that such a (solvable) formulation might be found.

The strategy proposed here first involves a transition from gradient-based to gradient-free solution methods, providing far fewer restrictions on predictive models and comparison metrics. It is well-known that such solution methods are computationally expensive for a given problem dimension. Here we aim to maintain reasonable solution expense by utilizing a reduced-dimension design representation that maps to the original high-dimension discretized topology description. Any time problem dimension is reduced, design space coverage is degraded (unless the original design representation possessed sufficient redundancy). The underlying assumption moving forward here is that if a reduced representation provides sufficiently targeted coverage of the original design space, high-quality design solutions can be obtained with reasonable solution expense.

How can one ensure both sufficiently reduced problem dimension, as well as sufficiently targeted design space coverage? Previous work relied on human expertise to identify gen-

erative algorithms or other indirect representations that were capable of accessing desired designs. But what if human expertise either cannot support creation of such indirect representations, or leads to representations where non-intuitive superior designs are not accessible? Here we introduce a data-driven strategy that overcomes these issues. Design data is generated by solving a related, easy-to-solve design optimization problem, and a machine learning strategy is used to construct a custom design representation that does not rely on human expertise. This move in the problem formulation space is illustrated conceptually in Fig. 3.1. The indirect design representation and gradient-free solution method supports moving closer to the origin from the SIMP formulation along the modeling and comparison metric axes, at the cost of moving to the right (strategically) along the design representation axis.

Existing effective machine learning techniques for constructing reduced-dimension representations of data, such as images and other geometric information, were reviewed as candidates for constructing indirect design representations in this investigation. We chose to use a variational autoencoder (VAE) that maps high-dimension training data to low-dimension latent variables, and then for a given set of latent variable values can map back to the original high-dimensional design space. The VAE technique is describe in detail later in this chapter. Information from SIMP-derived design data is leveraged to construct targeted indirect design representations. Other similar machine learning strategies may also be effective for this task, but the investigation of these is outside the scope of this chapter.

### 3.2.3 VAE-based Design Method

The design method introduced here utilizes an indirect design representation constructed using an augmented Variational Autoencoder (VAE), which *automatically* learns to generate topologies from a low-dimensional latent space. More specifically, this purely data-driven generative model attempts to find a two-way mapping from sample topologies (generated using SIMP) to a set of normally-distributed points in the latent space. To learn this mapping, we assume that the problem to be solved has a manifold of solutions that can be

approximated by a similar problem (called an "approximated problem") for which design sensitivities can be obtained with a much lower cost. For example, the problem of minimizing the maximum temperature of a thermal system requires subgradient calculations due to the max operation. The problem of minimizing thermal compliance, on the other hand, is differentiable and can be solved via SIMP easily. The solutions to these two problems, however, are similar in that both possess dendritic features that are useful for heat dissipation, suggesting that both are embedded in similar low-dimensional manifolds in the same topology space. With this heuristic, we propose to learn the latent space (for the solution manifold) through VAE using solutions collected from an approximated problem to achieve nonlinear dimension reduction. We will show that this VAE-based method offers a computationally viable solution to topology optimization problems where design sensitivities are unavailable or impractical to calculate, provided that an approximation problem can be leveraged. In addition, we further introduce augmentations to the VAE formulation to prevent generation of designs with scatterings of small material clusters that do not add value, while ensuring diversity of the generated topologies and coverage of important design space regions. Improvement in solution efficiency is demonstrated using a multi-objective optimization problem for heat conduction. The proposed design representation is abstract and is difficult to interpret in a physical sense, but it targets the design search accurately in areas that are likely to produce good performance. Leveraging information from design data obtained from the approximated problem supports efficient solution and identification of high-quality designs.

In this chapter, we claim three intellectual contributions to the existing literature of generative design. 1) We demonstrate that finding and solving approximation problems can lead to solution data that are useful for learning a solution manifold of an original problem for which sensitivities are impractical to obtain. The lower dimensionality of the learned manifold and the low-cost mapping from this reduced space to the original topology space enables accelerated solution searching for the original problem. 2) The application of the

proposed method on a heat-management problem suggests that significant dimension reduction and thus acceleration in optimization can be achieved when the underlying solutions are highly structured (e.g., dendritic structures are ubiquitous in solutions for different formulations of heat management problems). 3) The hypothesis that the reduced-dimension indirect design representation approximately covers the true Pareto set in the original design space is supported empirically via comparative study. Here we define a design space coverage metric that characterizes the degree to which non-dominated solutions are identified. The proposed methodology uses a more general solution method (i.e., multi-objective genetic algorithm, or MOGA) that does not require special problem structure, enabling the solution of more accurate problem formulations (i.e., closer to the origin along the comparison metric and predictive modeling dimensions in Fig. 3.1), and achieves solution efficiency via reduced design representation dimension.

The remainder of the chapter is organized as follows. Section 3.3 details the underlying concepts of the VAE and style transfer network used here. Section 3.4 presents the proposed VAE-based methodology. In this section, data is generated using SIMP, and VAE and style transfer networks are constructed using this data. The multi-objective optimization problem for heat conduction system design with respect to the latent space variables is formulated. Numerical results are reported in Section 3.5: the data reconstruction via VAE is presented; non-dominated solutions are identified using different solution methods for multi-objective optimization; computational expense results are compared and discussed; design space coverage is quantified and discussed using a proposed metric, including comparison with direct design representations. The chapter then finishes with discussion and conclusion sections. The Appendix figures follow the bibliography.

## 3.3 Background for Augmented VAE Indirect Representation

This section provides the background both for VAEs and style transfer networks used in the proposed indirect design representation. Features of these techniques are identified that are beneficial for constructing such representations. We also note here that other simpler dimension-reduction strategies, such as principle components analysis (PCA) were tested, but it was discovered that simpler linear strategies were unable to simultaneously reduce representation dimension and support identification of high-quality design solutions. VAEs were investigated due to their nonlinear dimension-reduction performance in other domains, and style transfer networks were identified as a strategy to better target design space coverage provided by VAEs.

### 3.3.1 Variational Autoencoder

A variational autoencoder (VAE) [137] is an extension of the standard autoencoder [21]. Both models are composed of an encoder $\mathbf{z} = f(\mathbf{x})$ that converts the input $\mathbf{x}$ to a latent vector $\mathbf{z}$, and a decoder $\hat{\mathbf{x}} = g(\mathbf{z})$ that produces a reconstruction $\hat{\mathbf{x}}$ from $\mathbf{z}$. A schematic of an autoencoder is shown in Fig. 3.2. The model is trained by minimizing the reconstruction loss between a set of inputs and the corresponding outputs. Variants of the autoencoder (e.g., denoising [245], sparse [168], and contractive [212]) have been developed to learn concise representations from high-dimensional input data, and are adopted widely for data compression [148], network pre-training [22], and feature extraction [213, 255].

Autoencoders, however, do not ensure that the model distribution (the distribution of model outputs) is congruent with the input data distribution, and thus may create outputs with drastically different properties from the input data. VAEs address this challenge by enforcing the distribution of the encoded inputs to match the sampling distribution in the

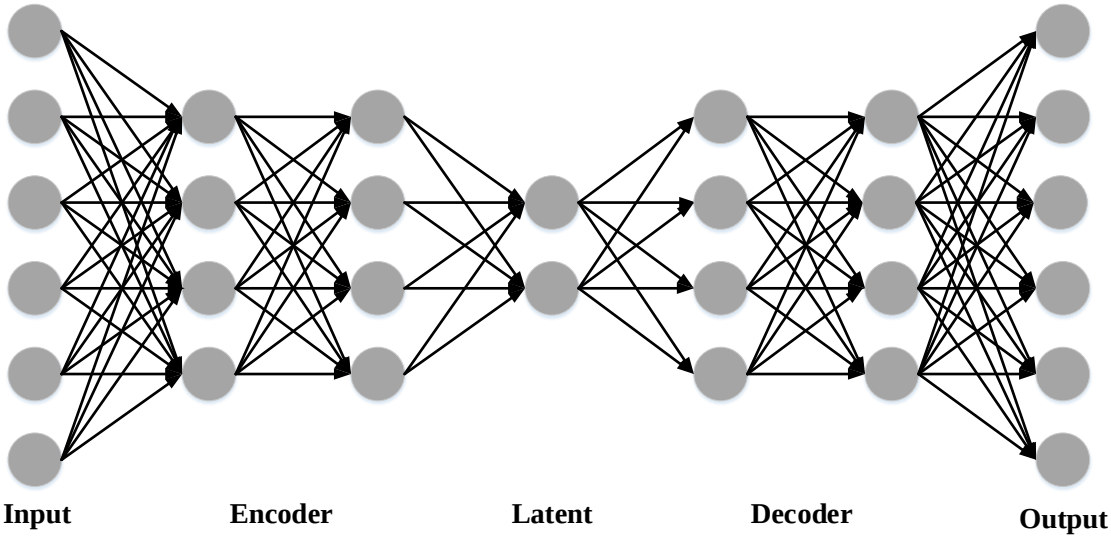



Figure 3.2: A schematics of an autoencoder

latent space.

We present a brief derivation of the VAE loss below, which is a function to be minimized with respect to encoder parameters $\phi$ and decoder parameters $\theta$. Let the dataset be $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, which defines the data distribution $p_{\text{data}}(\mathbf{x})$. Let the decoder define the probability of an output, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, conditioned on the latent vector $\mathbf{z}$. Matching the model distribution with the data distribution is equivalent to maximizing the marginal likelihood $\Pi_i^N p(\mathbf{x}_i)$, where

$$p(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} = E_{\mathbf{z}\sim p(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}). \tag{3.1}$$

In general, the computation of Eq. (3.1) and its gradient with respect to network parameters is expensive, due to numerical integration and decoder network complexity. It is noted, however, that for a particular $\mathbf{x}$, most $\mathbf{z}$ will lead to $p_\theta(\mathbf{x}|\mathbf{z}) \approx 0$, and thus have limited contribution to $p(\mathbf{x})$. Therefore, it is reasonable to sample only those $\mathbf{z}$ that are likely to produce $\mathbf{x}$, and use them to compute $p(\mathbf{x})$. To do so, a new function $q(\mathbf{z}|\mathbf{x})$ (the encoder) is introduced, which takes $\mathbf{x}$ and outputs a distribution of $\mathbf{z}$. Ideally, the space of $\mathbf{z}$ that are likely under $q$ will be much smaller than that under the prior $p(z)$, so that the marginal $\mathbb{E}_{\mathbf{z}\sim q}\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ becomes inexpensive to compute. Below we evaluate the difference between

the approximation ($\mathbb{E}_{\mathbf{z}\sim q}\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$) and the target ($p(\mathbf{x})$): We start by deriving the Kullback-Leibler divergence between two distributions in the latent space: the encoder distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$:

$$D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_{\mathbf{z}\sim q}\log q_{\boldsymbol{\phi}}(\mathbf{z}) - \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}). \tag{3.2}$$

By applying Bayes' rule to $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ and noticing that $\log p(\mathbf{x})$ is independent from $\mathbf{z}$, we have:

$$\log p(\mathbf{x}) - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{3.3}$$

The left hand side has the quantity we want to maximize: $\log p(\mathbf{x})$, and the KL-divergence term $D_{\mathrm{KL}}(\cdot)$ that ideally reaches 0. The right hand side of the equation contains the reconstruction likelihood ($\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$) and the KL-divergence between the encoder distribution and the prior distribution for sampling the latent space ($D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$).

These RHS terms can be maximized via stochastic gradient descent. Specifically, we model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu(\mathbf{x};\boldsymbol{\phi}), \Sigma(\mathbf{x};\boldsymbol{\phi}))$ as a normal distribution. The mean $\mu(\cdot,\boldsymbol{\phi})$ and the variance-covariance matrix $\Sigma(\cdot,\boldsymbol{\phi})$ comprise the encoder network. Similarly, we model the decoder outputs to follow $p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})} = \mathcal{N}(\mathbf{x}|f(\mathbf{z};\boldsymbol{\theta}), \sigma^2\mathbf{I})$ with mean $f(\mathbf{z};\boldsymbol{\theta})$ and variance $\sigma^2$. The function $f(\cdot;\boldsymbol{\theta})$ is thus the decoder network. $\sigma$ determines the importance of the reconstruction of $\mathbf{x}$ during the training of a generative model, and is set to 1 in the proposed model.

To summarize, the training of a VAE maximizes:

$$L(\theta,\phi,\mathbf{x}) = -\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})), \tag{3.4}$$

which represents a lower bound of the marginal $p(\mathbf{x})$.

### 3.3.2 Style Transfer Convolutional Neural Network

Through a preliminary study, we observed that the outputs from the VAE contain topologies with small disconnected phases. Disconnected conductive material reduces power density, and generally does not aid heat extraction, so it is desirable to avoid these features. Put another way, a VAE representation that produces designs with disconnected regions is inefficient because it retains access to less desirable designs. Here we improve targeted coverage by providing better access to topologies without disconnected regions.

As an implicit strategy to prevent the generation of topologies with disconnected conductive material, we introduce an augmentation to the VAE model. This augmentation forces topologies generated from arbitrary latent variable values to follow the topological style from the training samples. This approach of implicitly meeting design requirements via design representation is similar in spirit to the technique used by Khetan et al. [134] to guarantee generation of structurally stable trusses. The style transfer network was proposed originally to address the problem of texture transfer, i.e., to transfer image styles from source images to target contents. Synthesis of content and style historically has been a challenge in image processing [67, 68, 143, 251], but is now solved more easily since the recent development of deep convolutional neural networks (CNNs) [141,227] due to their ability to extract high-level semantic information from images. Gatys et al. [86] proposed a CNN-based style transfer network that can separate and recombine image content and style information, and then generate new images with the target content and styles. To be more specific, given a source image $\mathbf{s}$ containing a prescribed style and a target image $\mathbf{t}$, the network recovers an image $\mathbf{x}$ with content similar to $\mathbf{t}$ and texture from $\mathbf{s}$. The recovered image $\mathbf{x}$ can be obtained by solving a nonlinear least-squares problem:

$$\min_{\mathbf{x}} \|F(\mathbf{x}) - F(\mathbf{t})\|^2 + \|C(\mathbf{x}) - C(\mathbf{s})\|^2, \tag{3.5}$$

where $\|C(\mathbf{x}) - C(\mathbf{s})\|^2$ is the content loss, and content features $C(\cdot)$ are represented by

activations of deep hidden layers. Here $\|F(\mathbf{x}) - F(\mathbf{t})\|^2$ is the style loss, where style features $F(\cdot)$ are represented as the covariances among channel-wise hidden layer activations.

## 3.4 Proposed Method

In this section, we present a new method for solving heat conduction system topology design problems using a reduced-dimension latent variable design representation constructed using a VAE. This method involves two phases, as illustrated in Fig. 3.3. The first phase is to train a VAE using training topologies; the second phase is to perform multi-objective optimization with respect to the latent representation $\mathbf{z}$ learned by the VAE. Genetic, gradient-based, and hybrid algorithms are tested here for performing optimization in the latent space.

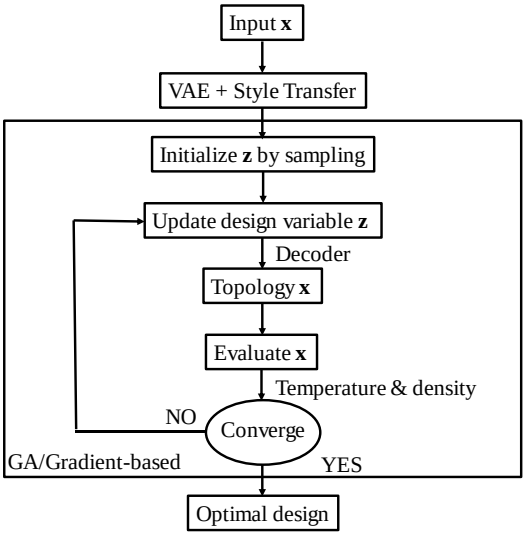



Figure 3.3: Flowchart of the optimization scheme with respect to $\mathbf{z}$

### 3.4.1 Data Collection

To generate data for VAE training, a density-based topology optimization approach is utilized. The topology optimization problem solved is:

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & C(\mathbf{x}) = \mathbf{U}^{\mathbf{T}}\mathbf{P} \\
\text{s.t} \quad & V(\mathbf{x}) \leq V_{\min} \\
& R(\mathbf{x}) \geq R_{\min} \\
& \mathbf{KU} = \mathbf{P} \\
& \mathbf{0} \leq \mathbf{x} \leq \mathbf{1},
\end{aligned}
\tag{3.6}
$$

where thermal compliance, $C(\mathbf{x})$, is minimized subject to a material volume constraint, minimum radius constraint, and Fourier's law for heat conduction. $\mathbf{U}$ and $\mathbf{P}$ represent the global displacement and force vectors, respectively; $\mathbf{K}$ is the global stiffness matrix [224]. The design variable, $\mathbf{x}$, represents material density. Each element of $\mathbf{x}$ specifies the material density of a corresponding voxel, and each element is bounded between 0 (void material properties) and 1 (solid material properties). To bias element densities towards a binary distribution, a power penalization (SIMP) is used. To enforce the minimum radius constraint, a density-based filter is implemented. Interested readers are referred to relevant references for more detail regarding topology optimization in heat transfer [159–161]. While 3D topology formulation was also discussed by Liu and Tovar [157], this chapter focuses on 2D thermal system topologies.

A set of problem parameters, including volume and radius bounds, as well as heat sink locations, are applied to Prob. (3.6). These problems are solved to obtain a set of topologies, each optimal with respect to the corresponding problem. The optimization process may produce infeasible solutions, but these were removed from the data set. Only the feasible and optimal designs were use for training the VAE. When using the SIMP method, each element of the spatial mesh has a corresponding design variable that varies continuously between 0 and 1. The penalty factor biases $\mathbf{x}$ values toward 0 or 1 during solution, but most values after convergence are close to 0 or 1 (not exactly binary). This necessitates a

post-processing step where a filter threshold value is used to convert each element into a binary digit. The resulting data set was split into training and test sets. Validation was performed by using the test data set to check image reconstruction quality.

### 3.4.2 The VAE and Style Transfer Network

Here multilayer perceptron (MLP) networks are used to construct the encoder and decoder of a VAE. The encoder contains two fully-connected layers with hidden layer sizes 1,000 and 100, respectively. The decoder architecture is symmetric to the encoder. The output dimension of the encoder is set to 5, 10, 15, 20, and 25, for the purpose of testing how latent space dimensionality affects topology optimization results. This comparison will be presented in the next section. Figure 3.4 illustrates an overview of the proposed model. During training, a batch of 40 random samples are generated via the latent space. The style transfer network has four layers, and the style loss is calculated with respect to the Gram matrices.

The loss function for training contains the reconstruction, KL divergence, style, and mode-collapse losses: $L_{total} = L_{recon} + L_{KL} + L_{collapse} + L_{style}$. Reconstruction and KL-divergence losses are standard in VAE implementations [137]; the style loss measures how well the generated images match the training data with respect to image style [86]; the mode-collapse loss prevents the VAE model from only producing similar samples [262], which is important for wide design space exploration. This loss is defined as:

$$f_{collaps}(S) = \frac{1}{N(N-1)} \sum_{i} \sum_{j \neq i} \left( \frac{S_i^{lT} S_j^l}{\|S_i^l\| \|S_j^l\|} \right)^2, \tag{3.7}$$

where $S$ denotes a batch of samples taken from the $l$th style transfer network.

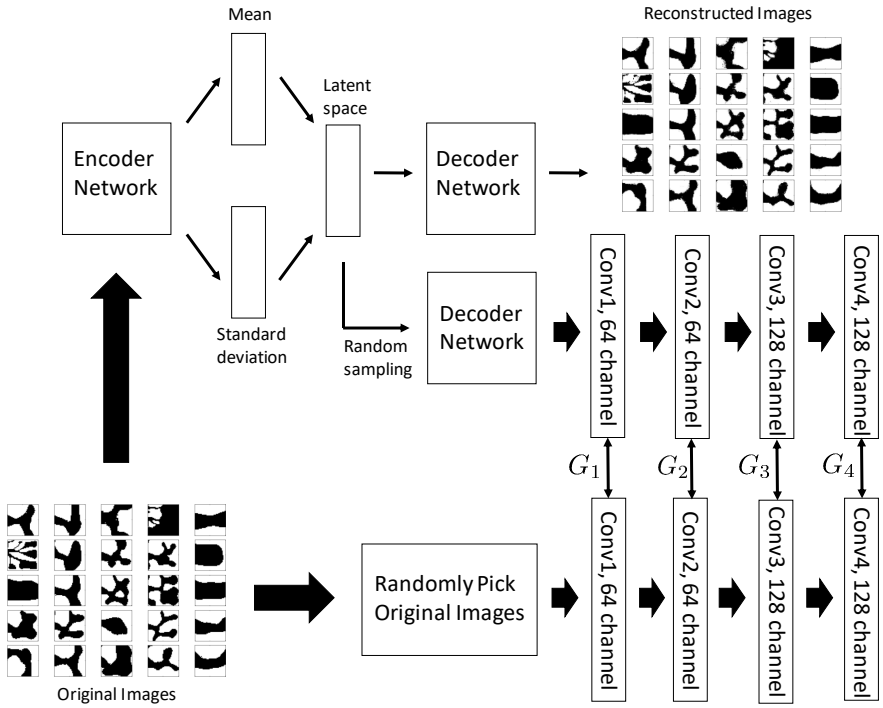



Figure 3.4: The VAE model augmented by a style loss

### 3.4.3 Heat Conduction Design Using Latent Variables

Here we discuss a topology design problem similar to Prob. (3.6), but with two key differences. First, instead of searching in the original topology space, we do so in the latent space derived from the VAE. Second, instead of compliance, we consider two objectives that are more closely aligned with underlying design intent: minimizing the maximum temperature $T_{\max}$, and maximizing the power density $\rho$. These objectives are conflicting: increasing power density typically increases maximum temperature. Thus, we formulate a multi-objective optimization problem where the objectives $T_{\max}(\mathbf{z})$ and $-\rho(\mathbf{z})$ are to be minimized with respect to the latent variables $\mathbf{z}$.

$$\min_{\mathbf{z}\in\mathcal{Z}} \quad \{T_{\max}(\mathbf{z}), -\rho(\mathbf{z})\} \tag{3.8}$$

This problem is inspired by heat spreading devices used to extract heat generated by

power electronic devices. It is desirable to pack as many value-added electrical components as possible into a given volume, while ensuring that temperatures do not exceed component or material limits. Adding in more electrical components increases the utility of the system, but reduces the volume available for conductive material that can be used to transfer heat from electrical components to the heat sink to keep temperatures manageable. We would like to find an arrangement (topology) of components and conductive material that helps us to maximize the number of useful electrical components while preventing thermal failure.

One measure of power density would be the amount of useful electrical power that can be routed through the electrical components, divided by the overall device volume. Because in this problem the volume is fixed, power density is proportional to the volume occupied by electrical components. For convenience, power density $\rho(\mathbf{z})$ is defined here in a unitless manner as the number of pixels (2D) that do not contain conductive material. This quantity is to be maximized, or equivalently, $-\rho(\mathbf{z})$ is to be minimized.

The problem can be solved with gradient-based or genetic algorithms. The initial guesses for gradient-based methods and the initial populations for genetic algorithms are drawn from the prior $p(\mathbf{z})$ defined by the VAE model.

The solution to a multi-objective optimization problem is a set of non-dominated (Pareto-optimal) solutions that quantify the tradeoff between objective functions. Four approaches for solving this multi-objective optimization problem are tested. The first, the weighted-sum method, is to sum the two objectives as shown in Prob. (3.9), with a weight $w$. We parametrically vary $w$ from 0 to 1 to produce and solve a family of single-objective problems using a gradient-based method. This produces a set of solutions that belong to the Pareto set (non-dominated designs).

$$\min_{\mathbf{z}\in\mathcal{Z}} \quad w \cdot T_{\max}(\mathbf{z}) + (1-w) \cdot (-\rho(\mathbf{z})) \tag{3.9}$$

In the second solution strategy, we select $-\rho(\mathbf{z})$ to use as a single objective function, and convert the temperature objective to a constraint with bound $T_{\text{allow}}$, as shown in Prob. (3.10). The upper bound $T_{\text{allow}}$ is varied parametrically to produce a family of optimization problems, the solution of which produces a set of non-dominated solutions. This approach is sometimes referred to as the $\epsilon$-constraint method [179].

$$\begin{aligned} \min_{\mathbf{z}\in\mathcal{Z}} \quad & -\rho(\mathbf{z}) \\ & T_{\max} \leq T_{\text{allow}} \end{aligned} \tag{3.10}$$

The third method is a hybrid implementation of the epsilon-constraint method. More specifically, for each value of $T_{\text{allow}}$ considered, Prob. (3.10) is solved first using a genetic algorithm (GA) to locate an approximate solution that is likely to be near the global optimum. The GA result is then used as a starting point for a gradient-based method, which then hones in rapidly on a precise local solution. On their own, gradient-based methods produce locally-optimal solutions, which may underperform globally-optimal solutions. Hybrid strategies improve the chances of, but do not guarantee, identification of global optima.

Finally, Problem 3.8 can also be solved directly using a multi-objective genetic algorithm (MOGA) [57] due to its population-based nature. While only one problem needs to be solved in this case instead of a set of problems, genetic algorithms tend to be computationally expensive. The Pareto-optimal solutions and frontier obtained using all four methods will be presented in the next section.

## 3.5 Numerical Results

In this section, we evaluate the proposed method for generative design through two steps. First, we show that the augmented VAE can learn a manifold of dendritic topologies rea-

sonably well by performing data reconstruction. We then assess the efficacy of the proposed dimension reduction method by quantitatively comparing the quality of the non-dominated solutions to a multi-objective heat conduction problem. To ensure a rigorous comparison, we employ four optimization approaches, with and without application to the reduced space introduced by the proposed augmented VAE method. The computational cost, measured as the number of physics-based analyses performed during the optimization, is reported for each of the four approaches and their variants.

We prepared a total of 15,000 topologies for training the VAE model and another 2,870 for validation. Five VAE models are trained with latent dimensions 5D, 10D, 15D, 20D, and 25D. For the multi-objective optimization, hyper-parameters (e.g. latent space dimension, initial population, etc.) were varied parametrically to more comprehensively explore design method properties.

### 3.5.1 Reconstruction and Generation of Topologies

We first show that the VAE is capable of data reconstruction. Figure 3.5 compares the original and reconstructed data for the test data set. Here a latent space of dimension 20 is used, and a sample of 30 topologies is visualized. We observed some isolated and blurry regions in the reconstructed topologies, indicating that the loss design of the VAE still has room for improvement.

In addition to reconstructing known topologies, the VAE can be used to generate new ones. Figure 3.6 illustrates a set of topologies created from random samples in the latent space. It is interesting to note that some of the generated topologies exhibit unexpected patterns. For instance, sample #45 has several separate chunks, whereas samples #73 and #76 resemble dendritic structures. For a particular generated topology, 2 out of 20 latent variables were selected and varied uniformly within the range of [-3,3]. The results of this parametric study are illustrated in Fig. 3.7, and show that significant, yet smooth, topology changes can be achieved by adjusting the VAE latent variables.

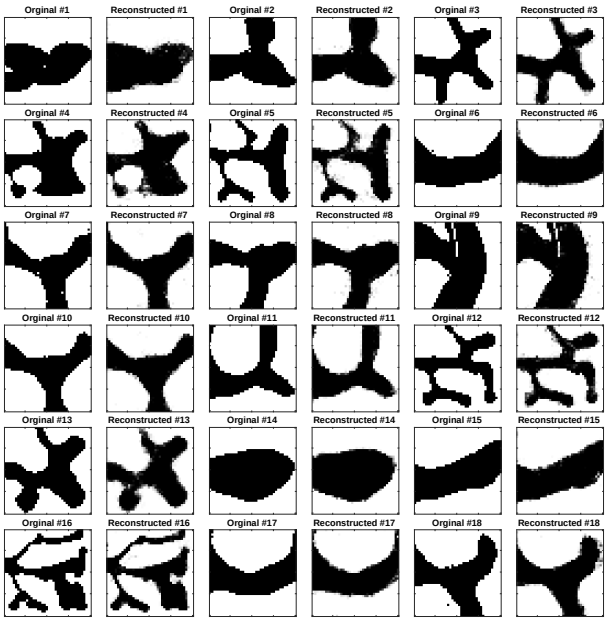



Figure 3.5: A sample of reconstructed topologies from a 20D latent space

### 3.5.2 Multi-objective Optimization Results

Multi-objective optimization results obtained using four different optimization strategies are reported and compared here.

#### Weighted Sum Method

A multistart strategy was used with the weighted sum method to find the Pareto-optimal solutions to Prob. 3.9 with an improved probability of finding global optima. Figure 3.8 shows the training data points along with the optimal solutions produced by the weighted sum method. The vertical axis is the negative of the square root of power density $\sqrt{\rho(z)}$, and the horizontal axis is the maximum temperature reached by the design. The reason for negating power density is to produce Pareto frontiers with the conventional orientation where moving to the lower left corresponds to improved designs. The square root of power density is taken to improve visualization. The temperature is measured in relative units of degrees Celsius. All similar plots of the objective function space use this strategy; moving down corresponds to increased power density, and moving to the right indicates a higher

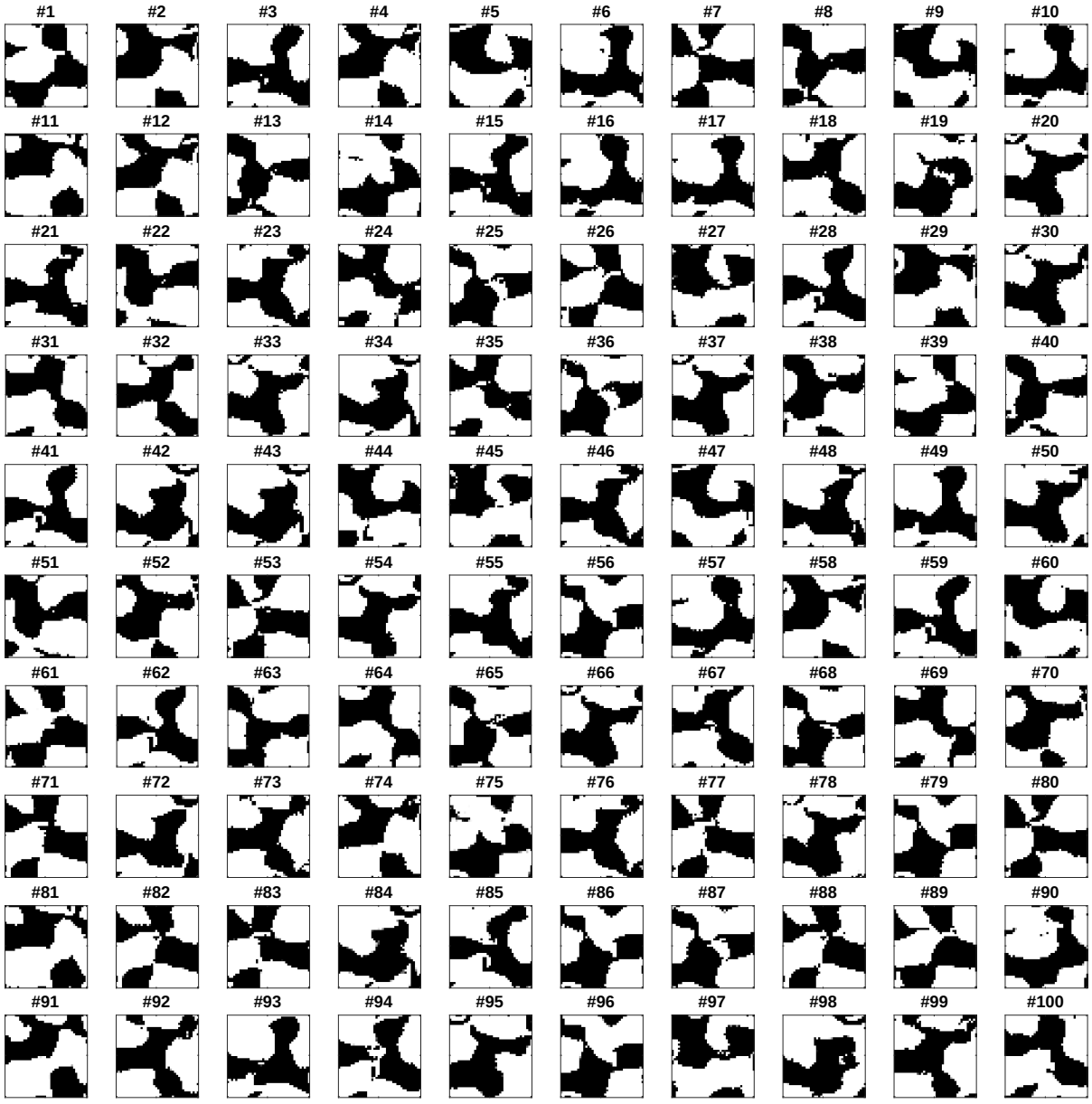



Figure 3.6: A sample of topologies generated from random samples in a 20D latent space

maximum temperature within the design domain.

Many of the points overlap on the figure (i.e., more individual problems were solved than the number of non-training points appear in the figure). The latent space dimension was varied parametrically; results from different latent space dimensions are labeled in the legend. The weighted-sum method did not perform well, as it did not identify points that were not dominated in the desired objective function space by training points obtained via SIMP.

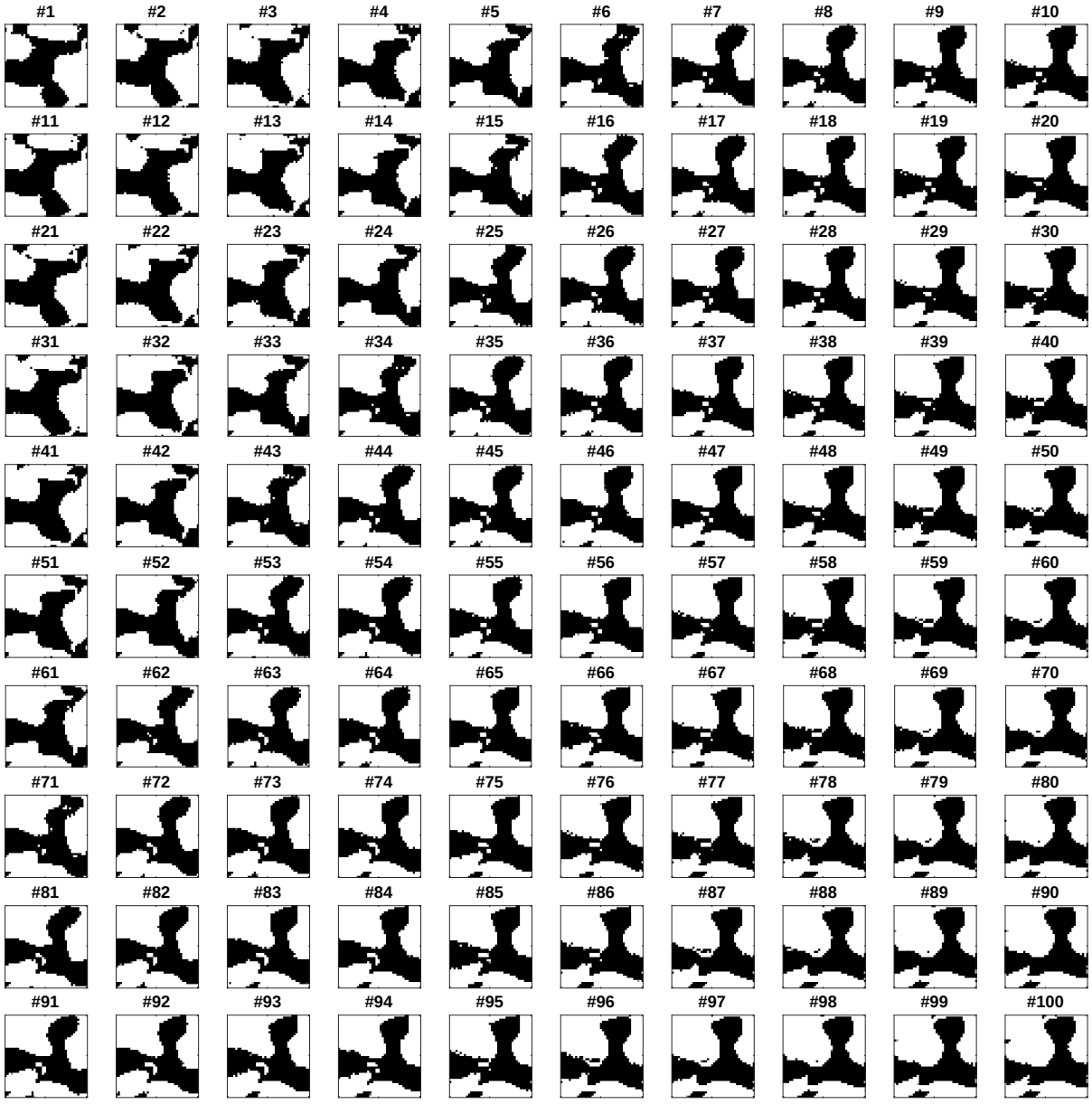



Figure 3.7: Parametric study results obtained by varying 2 of 20 latent variables

A sample of the local optimal solutions (dominated solutions) can be found in Appendix Fig. A.1.

### $\epsilon$-Constraint Method

The results from applying the $\epsilon$-constraint method are shown in Fig. 3.9. It is clearly much more effective than weighted-sum, as it can access not only non-dominated solutions from

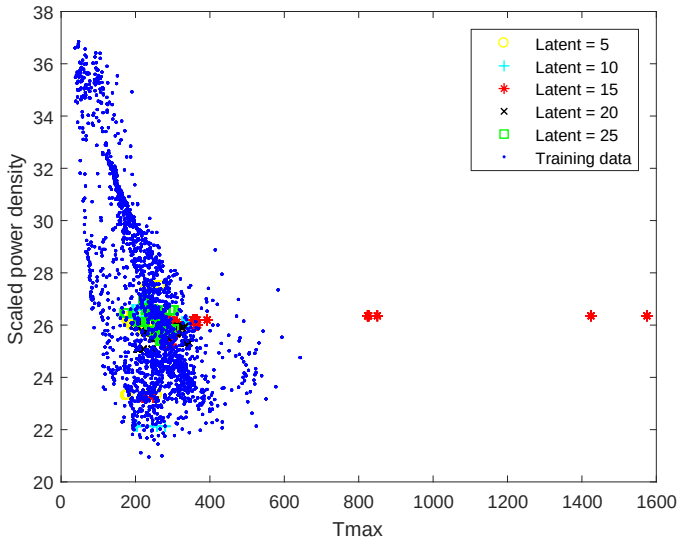



Figure 3.8: Locally-optimal solutions for the weighted sum method

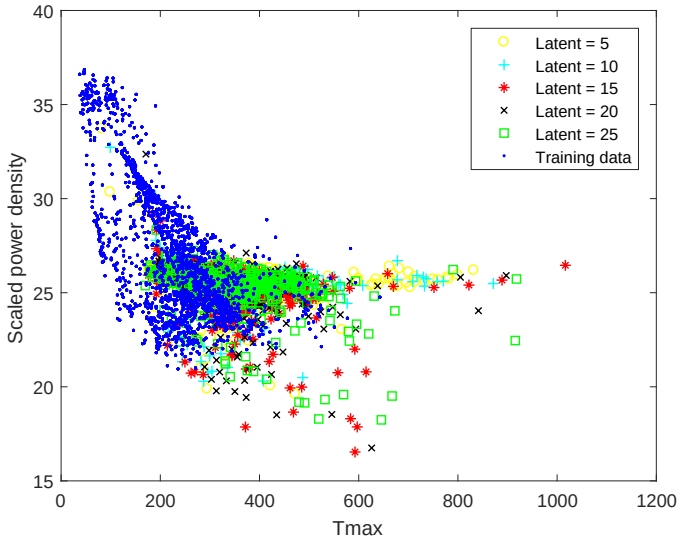



Figure 3.9: Locally-optimal solutions produced by the $\epsilon$-constraint method

the training data, but also new non-dominated points. The majority of the optimal solutions are around the center of the training data cluster. A few points with red stars and black crosses are on the Pareto frontier. Among these local solutions, eight non-dominated designs are illustrated in Fig. 3.10.

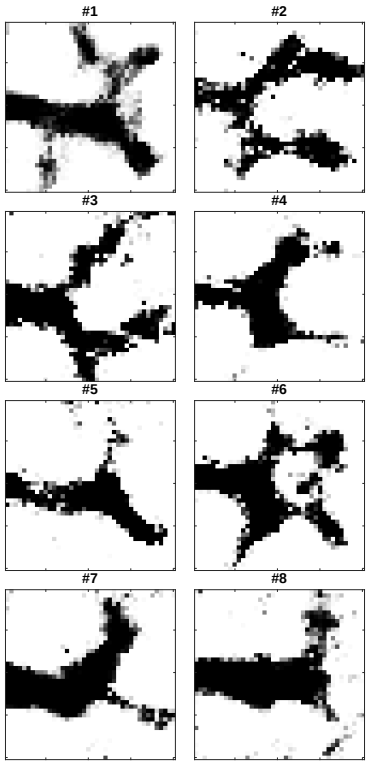



Figure 3.10: Pareto-optimal solutions produced by the $\epsilon$-constraint method

### Hybrid Method

The hybrid method uses a genetic algorithm (GA) with loose tolerances to find an approximate solution (ideally in the neighborhood of the global optimum) to use as a starting point for a gradient-based method. This sequential solution approach is applied to each single-objective problem defined using the $\epsilon$-constraint method. An initial population size of 200 was specified, and VAE models with different latent space dimensions were compared. Figure 3.11 displays the results of the hybrid method. Compared with the $\epsilon$-constraint method without starting points found via a GA, the hybrid method tends to identify more non-dominated designs; in particular, several new non-dominated solutions with $T_{\max}$ greater than 600 (shown using yellow circles and black crosses) were identified. These have significantly different performance from any of the training points, indicating the value of the VAE design representation in supporting broad exploration of the design space. Six Pareto-optimal topologies obtained using the hybrid method are shown in Fig. 3.12. While these

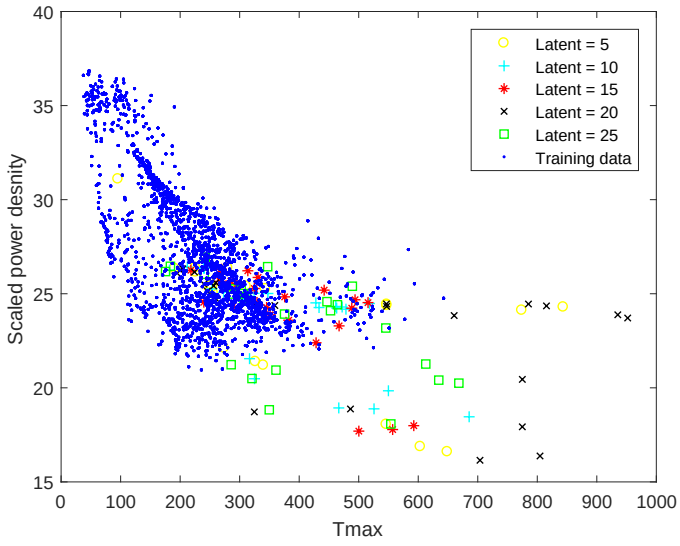



Figure 3.11: Optimal solutions produced by the hybrid method (GA followed by gradient-based)

points lie on the Pareto frontier, some could be improved. For instance, while solution #4 appears to be an intuitively reasonable design, solution #3 has several isolated material chunks. The power density of solution #3 could probably be improved while still keeping temperatures low by removing these chunks.

### Multi-Objective Genetic Algorithm

Multiobjective genetic algorithms (MOGAs) support direct solution of multiobjective problems due to their population-based nature. The implementation here uses an elitist strategy, a variant of NSGA-II algorithm [57]. The MOGA terminates if a metric, called *spread*, is less than a user-specified value. Here the *spread* is a measure of the movement of solutions on the Pareto frontier between the two most recent optimization iterations [171].

Two elements were explored using the MOGA method. First, different latent variable dimensions were tested using the same initial population size (200), reported in Fig. 3.13. A total of 96 Pareto-optimal solutions are shown in Fig. 3.14. They appear to be very similar to each other, but have some minor differences (such as material density). They appear to have three main solutions clusters: #1–#36, #37–#83, and #84–#96. The solution cluster

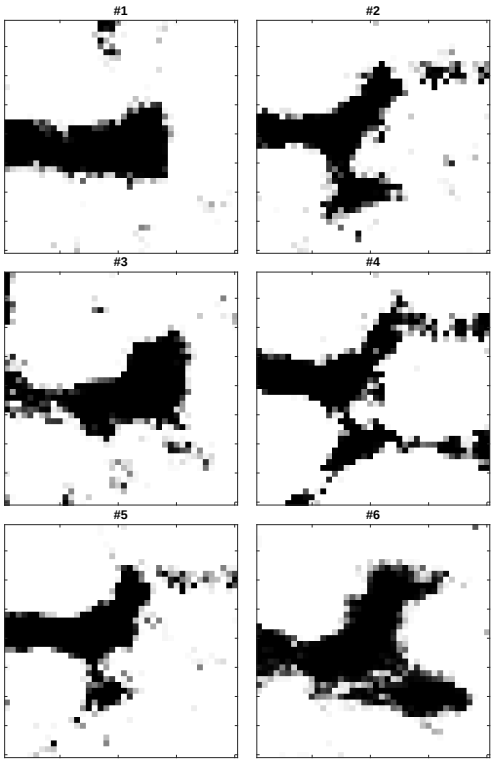



Figure 3.12: Pareto-optimal solutions produced by the hybrid method

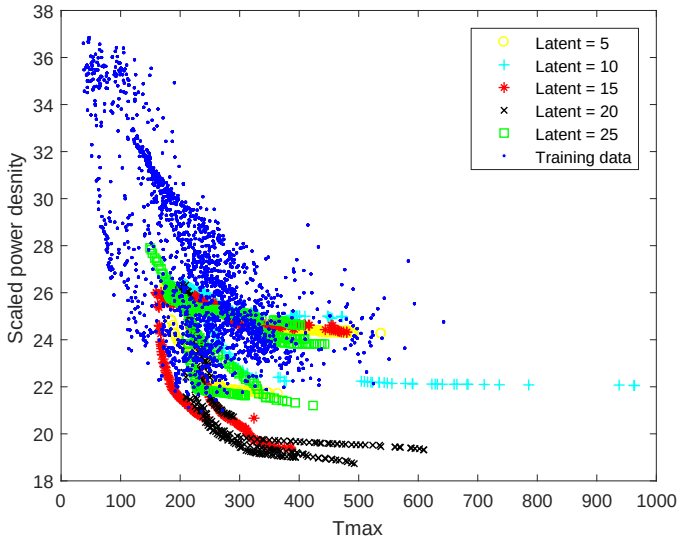



Figure 3.13: Optimal solutions produced by the MOGA method (initial population = 200)

#84–#96 has a number of isolated elements in the right corner of the topologies.

The second element tested using MOGA was influence of population size. We varied population size with a fixed latent variable size 20. In Fig. 3.15, the solutions with population

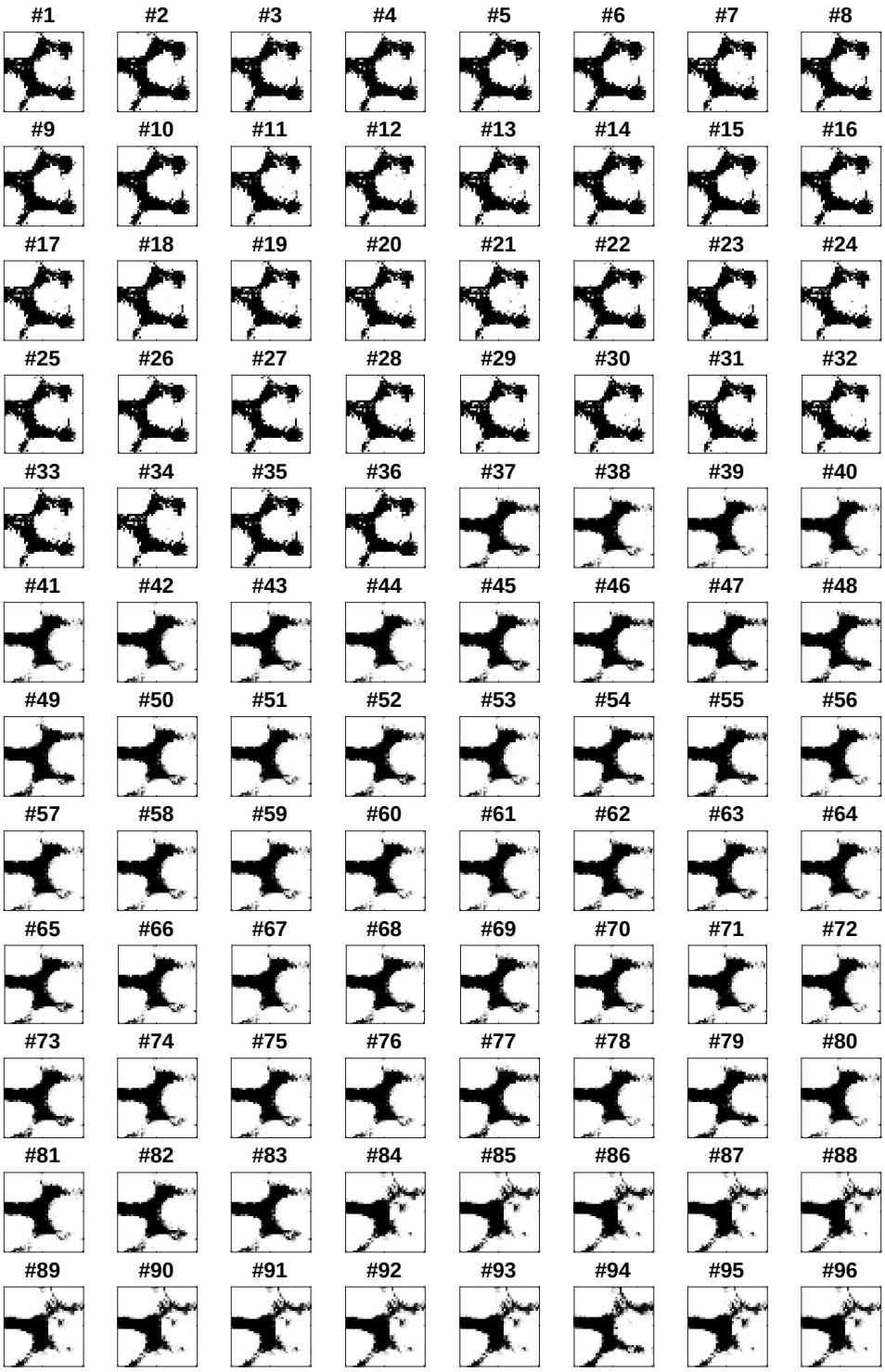



Figure 3.14: Select Pareto-optimal solutions produced by the MOGA method (initial population = 200)

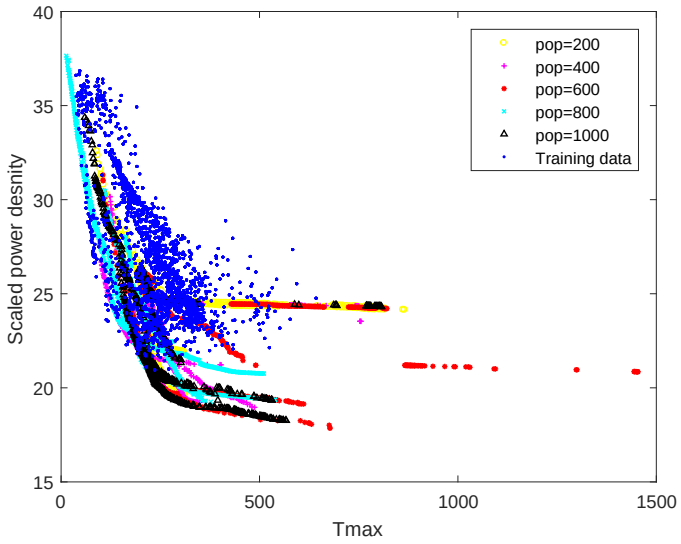



Figure 3.15: Optimal solutions produced by the MOGA solution method (#latent variables = 20)

sizes 600, 800, and 1000 are shown. The results show that for a 20D latent space, MOGA generally can access both non-dominated points from the training set, as well as new non-dominated points. This is one indication of effective design space coverage. Four optimal topologies are reported here, representing the basic topology structure; the other optimal topologies are shown in the Appendix in Figs. A.2–A.4. In Fig. 3.16, solutions #5 and #118 were found by MOGA; neither of these solutions were identified by other methods. It is interesting to note that a loop is formed in these solution. Solution #43 is a very dense solution. The structure of Solution #44 is very similar to those produced by the other three methods.

### 3.5.3 Solution Method Comparison

Not all four methods could identify all known non-dominated points (points on the Pareto frontiers). Among them, MOGA seems to perform the best at finding non-dominated points present in the training set, whereas the hybrid method performed well at identifying non-dominated points far from the Pareto set in the objective function space. We have included

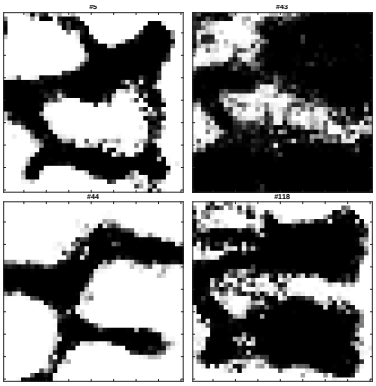


Figure 3.16: Four Pareto-optimal solutions produced by the MOGA solution method

illustrations of the Pareto frontiers and the corresponding topologies produced by the MOGA method in the Appendix. Here we present a comparison of the four methods using a latent space dimension of 20 in Fig. 3.17.

An important result observed is that many new non-dominated designs were found that go well beyond the attainable set estimated by the training data. In particular, many new non-dominated points were found using the indirect design representation (toward the lower-right portion of the plot). In other words, using the VAE indirect representation enabled identification of new solutions that for some design conditions are preferable to those that are reachable using conventional methods.

The MOGA with the indirect representation successfully identified the majority of the Pareto-optimal solutions, while other methods found smaller subsets of them. An important point to highlight is that several new non-dominated solutions were identified by the hybrid method corresponding to high power density designs, with the caveat that high temperature must be tolerated ($T_{\max} > 400$, black stars). GAs appear to be effective at identifying low-temperature solutions on the Pareto frontier, whereas the properties of gradient-based methods are utilized by the hybrid method to find non-dominated high power density solutions.

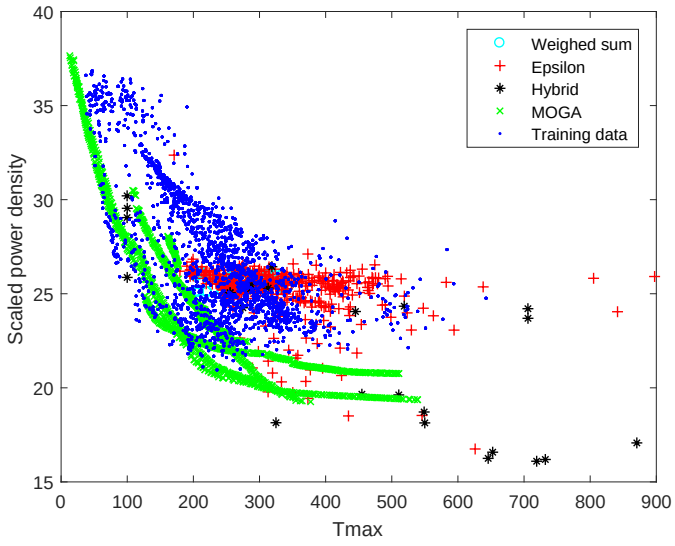



Figure 3.17: Comparison of Pareto-optimal solutions for the four methods (#latent variables = 20)

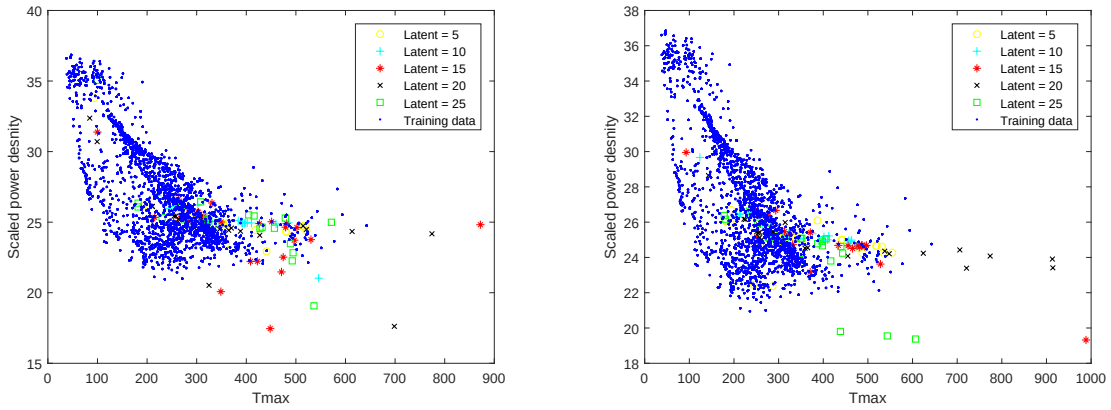
### 3.5.4 Computational Expense Comparison

The computational expense results for the four different solution methods is presented in Table 3.1. The *function count* was used as the metric here, and refers to the cumulative number of times the physics-based analysis was performed. These results are based on a latent space dimension of 20 and an initial MOGA population of 200. The $\epsilon$-constraint method required somewhat fewer function evaluations than the weighted sum method, and was successful in finding non-dominated solutions. The hybrid method was computationally expensive, but found new non-dominated solutions that were distinct from the training set and that were not accessible by any of the other methods. MOGA finds the Pareto set directly in a single algorithm execution, and appears to have the advantage of identifying a greater density of non-dominated points in the attainable set defined by the training set. MOGA-derived designs also tend to have fewer disconnected chunks. MOGA, however, did not perform well at accessing high-temperature non-dominated designs found using the hybrid method. The best solution method depends on which objective function has higher priority.

Table 3.1: Computational expense comparison

| | Weighted sum | $\epsilon$-constraint | Hybrid | MOGA |
|---|---|---|---|---|
| F-count | 4,351 | 4,303 | 75,905 | 30,601 |

An additional parametric study was performed to more fully explore the potential performance of the hybrid method. Algorithm settings were varied to adjust the balance of effort between the first and second solution phases. The GA population size and the maximum number of iterations allowed for the gradient-based search were varied. Figure 3.18 shows the non-dominated solutions identified by the hybrid method using the initial population 50 and 100 respectively. Both figures can still report a few non-dominated solutions (latent space 15D and 20D). In particular, the computational expenses are 9,435 and 29,782 respectively, compared to that of 75,905 in Table 3.1. This exploration demonstrates that the hybrid method has the potential to identify new non-dominated points with high computational efficiency when algorithm settings are adjusted appropriately.




(a) Initial population = 50

(b) Initial population = 100

Figure 3.18: Optimal solutions produced by the hybrid method (less initial population)

Table 3.2: Design space coverage analysis for the MOGA method

| Init. Pop. | w. Data or Not | Original | Reduced | Evaluation |
|---|---|---|---|---|
| 800 | Yes | 0.3388 | 0.3075 | 123,000 |
| 5000 | Yes | 0.3052 | 0.2712 | 780,000 |
| 800 | No | 0 | 0.1775 | 123,000 |

### 3.5.5 Pareto-Optimal Solutions Obtained via the Reduced Space

Here we test the hypothesis that the reduced design space $z$ is an effective approximation of the Pareto set in the original design space. The MOGA method is used for this test, as it identified more non-dominated solutions than the others (see Fig. 3.17). Two experiments were carried out using the MOGA method. Recall that GA solutions require an initial population specification. The numerical experiments make comparisons in three aspects: design space, initial population selection, and the initial population size. For the design space, we would like to see whether there is an advantage in finding the Pareto frontier using the reduced space (20D) compared to the original space (1,600D). The initial population may be randomly generated (called *without data*) or chosen from the training data set (called *with data*). The initial population size may affect the computational expense and optimal solutions, and sizes of 800 and 5,000 were chosen for evaluating this aspect. Design space coverage has been discussed and analyzed in structural optimization [133, 161] and biological networks [96]. Here a metric assessing design space coverage is defined as the ratio of number of the number of non-dominated solutions found and the population size. It is used here to measure how well different design representations cover the design space.

Figure 3.19 shows the Pareto frontiers (with data) for the reduced and the original high-dimension spaces. Both curves result from solutions using a population size of 800, where the initial population is sampled from the training set (with data). The computational expense is 123,000 evaluations. As illustrated in the figure, both the reduced space and original space approximately cover the true Pareto frontier in the training set. The design space coverages

for both the original and reduced spaces, as shown in Table 3.2, are 0.3388 and 0.3075, respectively. The Pareto curve for the reduced space is longer (red), but because the red curve only approximately covers the Pareto curves in the training set (e.g. some training points are still outside the boundary of the red curve), the design space coverage is slightly lower. It is also interesting to observe that one training data point (100, 26) is excluded by both curves. If the population size is increased to 5,000, as shown in Fig. 3.20, both curves move toward the bottom left and access that point successfully. As expected, increasing the population size improves the solution results, but with additional computational expense (780,000 evaluations). The design space coverage metric values do not show significant difference as a function of population size (800 vs. 5,000, reduced or original). The reduced design representation can still support identification of high performance non-dominated designs even with lower computational cost (123,000 evaluations), but with good design space coverage (0.3075).

Additional observations can be made from Fig. 3.21, which shows the results based on random initial populations of 800 (without data). A significant difference exists: the reduced space demonstrates a more robust ability to find the approximate Pareto frontier. More precisely, it can still generate an accurate Pareto set even when the initial population is less informative (with the design space coverage 0.1775 dropped from 0.3388 in Table 3.2). In contrast, the design space coverage using the original high-dimension space and random initial populations is 0. Several insights arise here. Under the same experimental specifications (without data and same initial population 800), the reduced design representation (20D) makes the design space more well-behaved (Pareto frontier is closer to the origin). Figure 3.21 indicates that using random initial populations weakens the ability to identify the non-dominated solutions, but when using the reduced design representation a significant number of non-dominated points are found. Both figures indicate that the reduced design representation requires less solution effort to identify non-dominated solutions.

In contrast to conventional design representations such as voxel- or rule-based topology

optimization, the VAE reduced space representation uses variational inference for latent representation learning, and involves a strong assumption concerning the distribution of the latent variables. The reduced space, even when containing only a few independent latent variables (e.g., here 20D assuming multivariate normal distribution), can access the non-dominated solutions with less effort. The VAE decoder learned helpful topological properties during the training phase when using SIMP-derived data. This intelligence has been embedded in the weights and biases of the decoder. Even if the initial population is generated randomly, the candidate topologies are meaningful and help narrow down the feasible solution search space. Figure 3.6) provided evidence of this, exhibiting a number of reasonable designs produced via random latent variable generation. The tradeoffs between different reduced representation dimensions (5D–25D) and non-dominated solutions can be seen in Fig. 3.14 (but with the initial population 200). Two few or too many dimensions (e.g. 5D, 10D, 25D) lead to challenges accessing the full Pareto frontier, but 15D and 20D representations tend to perform better. In other words, these analyses show that the reduced design representation preserves the important independent design degrees of freedom (DOF) as the latent variables without cutting off the meaningful and feasible candidate topologies. While the original space also has a number of independent DOF, results indicate that some redundancy in design variables exist, making the design space more difficult to navigate. Part of this difficulty is that many disconnected (low-utility) designs are included in the design space, making useful designs more difficult to identify.

## 3.6 Discussion

In this study, deep convolutional neural networks were used both as the encoder and the decoder for thermal system design representations. A deep convolutional VAE with 16 layers and a style transfer network with 4 activation layers was first introduced. The filter size of the conventional layers for the encoder was $4 \times 4$, and the channel sizes were $32, 32, 64, 64, 96,$

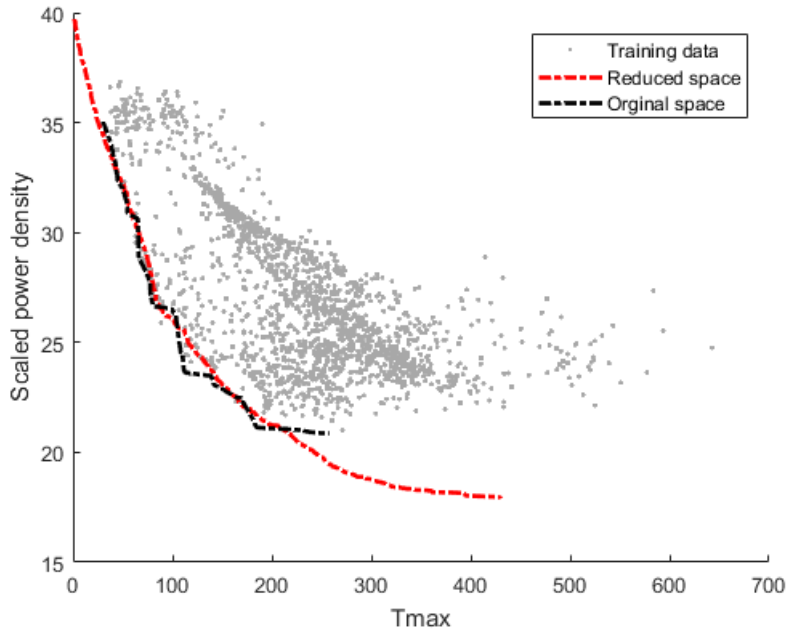



Figure 3.19: Reduced space vs. original space using population size 800 (with data)

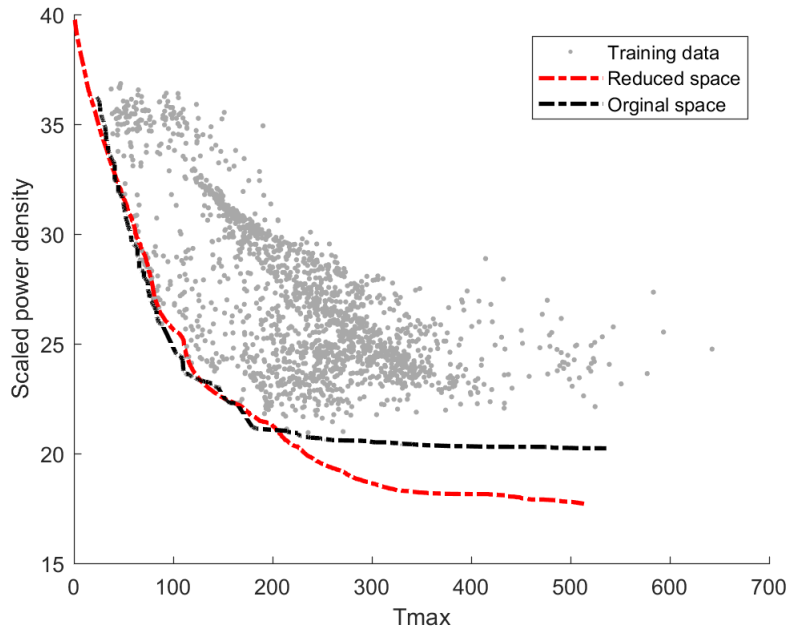



Figure 3.20: Reduced space vs. original space using population size 5,000 (with data)

and 96 for the six layers, respectively. A max-pooling layer was inserted after each of the two convolution layers to reduce the number of hidden units. Two fully-connected layers were used to encode the input $\mathbf{x}$ into the latent space. The decoder architecture was symmetric to

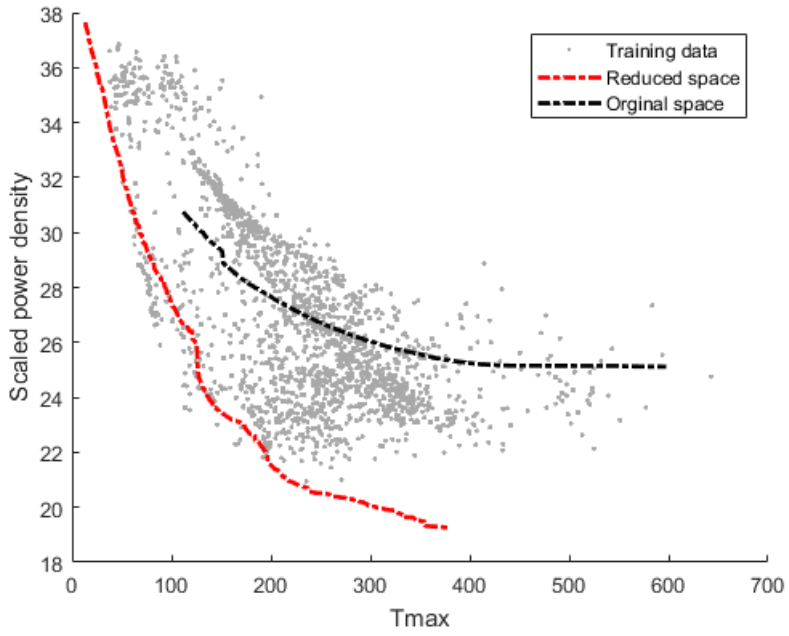



Figure 3.21: Reduced space vs. original space using population size 800 (without data)

the encoder. While this deep convolutional VAE worked well in conjunction with the style transfer network in the training phase, it increased the computational expense of optimization during the second phase. The preliminary results indicated infeasibility in the solutions for certain formulations. As a result, a multilayer perceptron was chosen in the end as the VAE. It is important to strike a balance between a simpler (less expensive, but less accurate) and a more complicated (more expensive, but more accurate) models.

The proposed design representation may be limited in some circumstances. The VAE model tends to produce blurry images, and not all details are preserved in the reconstructed or generated images due to the choice of the VAE loss function [65]. Improved models, such as sequential VAEs [263] and the modified VAE based on adversarial training [65], have been proposed and may help generate more realistic images (designs). Generative adversarial networks (GANs) could be another option in the first phase to generate sharper images [91]. Improved sampling techniques for obtaining training data may also increase method utility. Specifically, understanding how to plan systematic exploration of design problems for the generation of design data for unprecedented systems is an important question for

this emerging approach. Designing our strategy for obtaining training data is an option not available when relying on historical data, and could be an important advantage in using design optimization to generate design data for use with machine learning.

The data set was obtained by specifying several levels of boundary conditions, volume, and radius constraints (a full-factorial experiment was performed). Each training data point obtained using Eq. 3.6 is the result of a single SIMP solution. Because the volume constraint was varied, complete Pareto sets are contained in the training data shown in the figures above for the SIMP solutions for each boundary condition specified. An important result here is that some of the VAE results were successful in pushing down the Pareto frontier in the objective function space, i.e., new non-dominated points were found. The gap between the (initially) non-dominated training points and the non-dominated VAE points is relatively large in many of the figures. The training data may correspond to local optima, so it may be possible to narrow the gap between training data results and VAE results.

## 3.7 Conclusion

In this chapter, an indirect design representation based on a VAE was proposed for topology optimization. The notion of designed experiments based on a related problem to generate design data was presented. The VAE, augmented with a style transfer neural network, was trained using the design data. A description of the multi-objective optimization problem for the heat conduction, including the variants of the formulation, was discussed. The Pareto frontiers were presented using the proposed design representation. Using MOGA with the VAE representation was particularly effective at identifying a large majority of non-dominated solutions (when combining all design results together), whereas the hybrid method with the VAE representation was particularly effective at identifying new non-dominated points far from training points (high-temperature regions). One possible explanation is that GAs may first locate approximate solutions close to the Pareto frontier, and then gradient-

based algorithms are particularly effective at pushing solutions toward high power density in design space regions where temperature is high. Along the Pareto frontier, several types of solution topologies were found (see the Appendix for details).

Optimizing in the reduced design space was demonstrated to be successful in terms of 1) solving the desired problem efficiently using an indirect design representation constructed using training data obtained from a related (but easy-to-solve problem) problem; 2) leading to new, previously unreachable non-dominated designs identified by the hybrid and MOGA methods; 3) empirically supporting the hypothesis that the reduced-dimension design representation approximately access the true attainable Pareto set in the original design space based on the design space coverage metric.

This is a new perspective on how machine learning strategies may be utilized effectively in engineering design. The proposed method is data-driven, and has the potential to make practical the solution of topology optimization problems not amenable to established density-based methods. This could become an important new strategy for normative data-driven design methods, especially for design of unprecedented systems without historical data. Future work may include, but is not limited to addressing: 1) How should the design optimization experiments be conducted to produce effective training data? This open question is different from conventional design of experiments or sampling for surrogate modeling. The process depends on specific design tasks and objectives, such as constructing reduced design representations tailored to specific problem conditions. 2) The parametric study presented in Fig. 3.7 shows that generated topologies are sensitive to latent variable changes. Additional sensitivity analyses (e.g., impact on the objectives $T_{max}$ and $\rho$) may provide insights into strategies for improved VAE design representations. Such sensitivity analyses have the potential to be part of a general strategy for evaluating solution quality and robustness with respect to the generated topologies and objectives. 3) Design coverage quantification for indirect design representations is not yet well-studied, and should be explored further to support further advancements in approaches that rely on reduced design

representation dimension to provide computational efficiency. Additional coverage metrics should be proposed and evaluated.

In next two chapters, machine learning strategies will be applied to a different class of problems: heterogeneous system topology design. This different class of problems, sometimes referred to as synthesis, has different properties and challenges. The first heterogeneous system design study, presented in Chapter 4, will involve an active learning strategy. A predictive model is fit to data obtained via enumeration and simulation-based evaluation, and this model is used to strategically select small subsets of designs to evaluate. In Chapter 5 a generative adversarial neural network (GAN) is used to generate feasible designs, eliminating the need for complete enumeration in synthesis studies. In both chapters cases studies involving electronic circuit synthesis will be used. A core objective in the next two chapters is to extend the set of synthesis problems that can be solved effectively by leveraging design optimization data along with artificial intelligence, as well as to begin to understand the properties of this new class of design methods.

# Chapter 4

# HETEROGENEOUS SYSTEM TOPOLOGY DESIGN: CASE 1

**Collaborative Acknowledgement: Dr. Daniel Herber** made contributions to the work presented in this chapter, including creating the tools for enumeration and evaluation of electrical circuits, contributions to graph representations, and contribution of critical insights into ranking distance.

## 4.1 Introduction

Previous chapters provided motivation for the studies presented in this dissertation and demonstrated one approach for using artificial intelligence to extend solution capabilities for the first main class of problems: homogeneous system topology design. This and next chapter consider heterogeneous system topology design. Case studies are drawn from the literature on electronic circuit design synthesis. As with the thermal system topology optimization problem from the previous chapter, candidate circuit topologies may be represented using an undirected graph. The difference, here, however, is that nodes may be one of several types. This is a heterogeneous system topology design problem where candidate designs can be represented using a colored (labeled) graph. The design method presented here makes

use of design data generated from recently-developed efficient enumeration strategies for synthesis problems [109]. These enumeration methods have been applied recently to circuit synthesis problems defined by circuit component catalogs, and were shown to identify not only the topologies found earlier using GAs, but also a much richer set of design data, including many previously unknown non-dominated designs.

When enumerating heterogeneous system topologies, often duplicate or *isomorphic* graphs are generated. For example, two generated graphs that have distinct adjacency matrices may be shown to have the same topology after node renumbering. When enumerating, it is important to ensure no isomorphisms exist, i.e., all generated graphs are topologically unique. A second challenge involving labeled graphs that represent engineering system topologies is that some node connectivities might not make physical sense (e.g., electrical short circuits), or should be excluded for a variety of other reasons. During enumeration it is important to observe such connectivity constraints, referred to here as network structure constraints (NSCs). Finally, many engineering systems must be evaluated using a non-trivial model, and the structure of this model usually varies with system topology. For the circuit synthesis problems considered here, evaluation requires simulation of a system of ordinary differential equations, and a fair performance comparison between circuit topologies requires that each topology is optimized with respect to corresponding continuous variables (e.g., capacitance, resistance, inductance, etc.). To solve a circuit synthesis problem using enumeration requires automated construction of a corresponding dynamic system model for each candidate topology, and the automated generation and solution of a circuit component sizing optimization problem.

While recent advances in efficient enumeration have increased the size of synthesis problem that can be solved via enumeration, clear limits exist. Alternative methods are required to solve synthesis problems when component catalogs are too large. Here we observe that while enumeration may not be practical for these larger problems, enumeration can still be used to generate rich design data related to these problems. New methods are presented

here that leverage the availability of enumeration data and automated model generation/-sizing optimization and combine these resources with machine learning techniques to enable approximate solution of synthesis problems too large for enumeration. Consider the following two cases where efficient enumeration alone is insufficient to solve heterogeneous system topology design problems:

**Case 1:** A component catalog and set of NSCs results in topological design space where all unique and feasible circuit topologies can be enumerated in a practical amount of time, but evaluation (e.g., size optimization) is too computational expensive to perform for all unique, feasible topologies.

**Case 2:** The topological design space is even larger, to the point that complete enumeration, even with recent efficient enumeration algorithms, cannot be completed in a practical amount of time.

For completeness, we refer to the situation where it is possible to enumerate and evaluate all topologies of interest as Case 0 (i.e., efficient enumeration is sufficient). In this chapter we introduce a new type of method for solving Case 1 synthesis problems that relies on the availability of complete topological information. In the next chapter we address Case 2, where we do not have the luxury of access to all design topologies. Solution methods tailored for Case 1 synthesis problems will not be applicable to Case 2 problems due to the unavailability of the complete set of topological design data.

In this chapter, a Case 1 synthesis solution method is introduced, where an active learning method is used to sample limited sets of candidate designs strategically with the objective of identifying the optimal topology approximately. In other words, machine learning is used to capitalize on topological design space structure to maximize system performance without needing to test all designs exhaustively, reducing overall solution expense. This solution method 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 circuit topologies often requires computationally-expensive simulations, limiting the scale of solvable problems. Strategies are needed to explore topological 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 topological design data as a strategy to accelerates effective design search. Predictive model accuracy is improved incrementally using 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 with available data and design automation tools is used to test the active learning strategy; this application is of significant importance to mechatronic systems. While active learning has been used for structured outputs, such as sequence labeling tasks, the interface between active learning and engineering design has not been well studied. Results indicate that active learning is a promising strategy for reducing the evaluation cost for the selected circuit synthesis problems, and provide insight into possible next steps for generalization to a wider range of problems.

## 4.2 Literature Review

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 determining both circuit topology and component 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 [83, 180, 235], 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 significant human expertise requirements [53, 83, 94, 180, 235]. For example, evolutionary algorithms (EAs) [53, 54, 83, 89, 94, 140, 162] and simulated annealing [192] 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 [94]. Das and Vermuri developed an automated circuit synthesis framework for an *RLC* circuit design problem [53].

Despite these achievements, EA-based approaches are limited because 1) a number of algorithm parameters need to be tuned, including crossover probability and population size [94, 140, 162]; 2) effective solution may depend on using initial design populations with certain properties, resulting in potential solution convergence and robustness issues [53]; 3) direct genotype representations are not often successful in making progress toward improved designs for larger-scale systems [48]. It should also be acknowledged that global solutions are not guaranteed with EAs.

One strategy that can be used to improve the scalability of EAs to larger systems it to utilize effective mappings from indirect genotype encodings to phenotypes (detailed design specification). Indirect generative design representations, such as grammar-rule approaches or generative algorithms, may be used to facilitate indirect genotype encodings that map to direct design descriptions [38, 134, 208]. Indirect design representations that utilize neural networks [48, 99] or generative algorithms [134, 161] have been used successfully to reduce the computational expense of EA solution in application domains outside of circuit synthesis. Existing indirect encodings, however, cannot be generalized to problems where an existing appropriate generative design algorithm does not exist, or where human expertise cannot be used to define the generation rules.

Enumeration-based synthesis methodologies[1] generate and test all possible topologies

[1]Strictly speaking, enumerative strategies involve generation and selection of a design topology, as opposed to additive composition, and as a result some may argue that enumeration is not a synthesis activity. In addition, it could be argued that EA solution methods are also not synthesis depending on genotype definition. We acknowledge this issue, but use the term synthesis here when referring to the task of identifying the best-performing topology due to 1) its historical use in the circuit design literature, and 2) because in practice enumerative and EA circuit design methods have aims that are similar to design synthesis.

under certain specifications, so global optimality is guaranteed. Enumeration-based approaches have a long history and have been used widely in electronic circuits [34,75], hybrid powertrains [19], gear trains [58], and enzyme network topologies [166]. Naïve enumeration is simple to implement, but impractical when the number of possible candidates is significant. One example of Naïve enumeration would be to generate all possible adjacency matrices corresponding to graphs that represent candidate system topologies. When using naïve enumeration, the majority of designs produced are either isomorphic or violate network structure constraints (NSCs). It is desirable to focus design search efforts only on unique (non-isomorphic) designs that are also feasible with respect to component connectivity constraints (NSCs). Recent advancements by Herber et al. in efficient system architecture enumeration theory and algorithms, based on perfect matching theory and intelligent search methods, have made practical the use of enumeration methods for generation of all unique and feasible topologies for certain problem classes, including for design problems that are much larger than previously thought [112].

Efficient enumeration has been applied to circuit synthesis [109], and was shown to identify not only the topologies found earlier using EAs, but a much richer set of design data, including many previously unknown, non-dominated designs [112]. This study supported fair comparisons between unique, feasible topologies by using quantitative evaluation of the candidates [109,112]. A dynamic system model was automatically generated for each candidate circuit topology, and then was used in solving a component sizing optimization problem [109]. Efficient enumeration strategies have been extended to other system architecture design problems, such as active vehicle suspensions [110].

### 4.2.1 Framework for Design Method Comparison

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 topological design space of interest. Here we provide motivating context for a new circuit synthesis

design methodology that leverages a machine learning strategy to improve scalability, while providing reasonable computational expense and solution quality.

One approach for defining a design method is to describe how the design problem is formulated, as well as how the problem is to be solved. This context helps to describe fundamental differences between design methods, and how different methods may be useful in different situations. Design problem formulation and solution decisions are coupled. For example, the intended solution method may influence problem formulation decisions to leverage solution method properties, and different solution methods might be chosen based on desired formulations.

Figure 4.1 illustrates one possible approach for conceptualizing a design problem formulation space. Formulation decisions must be made along three dimensions: design representation (a precise way of articulating different design candidates), comparison metrics (an explicit approach for comparing design candidates), and predictive modeling (an approach for mapping design descriptions to comparison metric values). Moving closer to the origin represents an improvement in accuracy of the problem formulation with respect to design intent. The origin represents a hypothetically ideal or *substantive rationality solution* to the design problem formulation problem [165]. It is not attainable in practice because it would require a completely open-ended design representation (no restrictions or assumptions), a perfectly accurate predictive model, and comparison metrics that express true design utility with perfect accuracy. Moving closer to the origin is usually possible, but typically comes at the cost of increasing solution expense. We will discuss selected circuit synthesis strategies using this framework as a way to explain the reasoning behind the active learning circuit synthesis method introduced here.

One advantage of both enumerative and EA-based methods is that they do not inherently restrict comparison metric or predictive modeling choices. It is possible to use high-fidelity predictive models, even those with non-smooth or noisy responses. That said, solution expense may be prohibitive using these approaches. Using a reduced-dimension design rep-

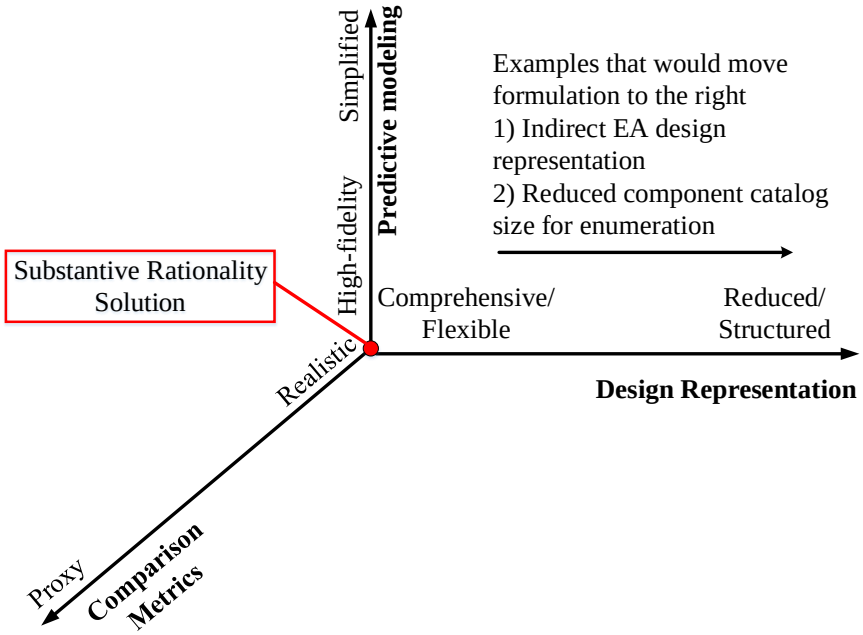



Figure 4.1: Conceptual design formulation space used for analyzing differences in problem formulation strategies

resentation (e.g., indirect EA genotype encoding) simplifies the design space, and can be conceptualized as moving right away from the origin in Fig. 4.1 along the design representation dimension. While solution may be easier, this modification eliminates from consideration a number of design candidates, thus making the specified problem formulation a less accurate approximation of the ideal problem formulation.

An important advantage of enumeration is the ability to identify a globally-optimal solution with respect to a given problem formulation, whereas EAs can only solve problems approximately. A disadvantage of enumeration, even with recent enhancements in efficient enumeration, is that the scale of problem that can be solved in a practical amount of time is quite limited. To define a problem that is solvable in practice using efficient enumeration, the complexity of the topological search space must be restricted. This can be done be reducing the number of items in the component catalog that can be used to compose topologies. This reduction, however, moves the problem to the right away from the origin in the

problem formulation space, as illustrated in Fig. 4.1. In general, there is a tradeoff between design problem formulation accuracy and solution accuracy. Enumeration improves solution accuracy, but requires concessions in formulation accuracy. EAs typically can be used to solve larger problems than can be solved using enumeration; this can improve formulation accuracy (more comprehensive design representation), but degrades solution accuracy due to the approximate nature of EAs with no guarantees of optimality.

It would be desirable to improve design representation (move left toward origin in the formulation space), while still allowing for high-fidelity predictive modeling and accurate comparison metrics (stay close to the origin in the other dimensions), and supporting high-quality solutions in a practical amount of time. Here we propose a new design method that does just this by combining efficient enumeration with tools from machine learning. It allows increased topological design space complexity, high-fidelity design candidate evaluation, and produces high-quality (approximate) solutions in a practical amount of time.

### 4.2.2 Active Learning Design Concept

Synthesis problems can be solved in a comprehensive way using efficient enumeration when the topological design space is sufficiently simple. Efficient enumeration methods help to expand the scope of problems that can be solved via enumeration, but still have limits. Consider two classes of problems that cannot be solved completely using enumeration:

**Case 1:** All unique, feasible topologies can be enumerated in a practical amount of time, but evaluation (e.g., size optimization/simulation) is too computationally expensive to perform for all enumerated topologies.

**Case 2:** A component catalog and set of NSCs results in a topological design space that is too large to enumerate all topologies in a practical amount of time.

Here we focus on Case 1. Chapter 5 addresses Case 2. Consider the case where $N$ is the total number of unique, feasible topologies generated using enumeration, $N_E$ is the number

of topological design candidate evaluations that can be performed within computational resource constraints, and $N \gg N_E$. It is not possible to find the globally-optimal solution using enumeration in this case due to evaluation limits. Here we would like to find an approximately optimal solution for this more complex synthesis problem (i.e., we support more accurate problem formulation by making concessions in solution accuracy). How best can we 'spend' the $N_E$ design evaluations to arrive at a high-quality solution? If we choose $N_E$ topologies to evaluate all at once, there is no way to leverage information available from these evaluations to improve solution quality further. A different strategy is to first choose a subset $N_1 < N_E$ of topologies to evaluate, and use the obtained results to inform selection of a new subset of $N_2 < N_E$ topologies to evaluate. If $N_i \ll N_E$, then this process can be iterated several times to adaptively improve selection of designs to evaluate. This type of iterative adaptive sampling technique has been studied in the context of other problems, and is known as *active learning*.

### 4.2.3 Electronic Circuit Synthesis Using Active Learning

The proposed active learning synthesis strategy is demonstrated and tested using an electronic circuit synthesis problem. This application was chose because it is well-studied in the literature, is relevant to mechatronic system design, and because recently-developed automated design enumeration and evaluation tools are openly available [109], supporting transparent investigation and replication. Test problems are chosen such that both complete enumeration and evaluation can be performed to support more comprehensive analysis of the method.

The type of circuit evaluation required here involves straightforward generation of single-input single-output transfer functions [109], and then finding the solution of a nonlinear fitting problem. Having a complete set of data (topologies and corresponding evaluation metric values) 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 synthesis problem where comprehensive evaluation cannot be performed in a practical amount of time. Further extension involving a larger catalog would result in a Case 2 problem, which is a topic of separate ongoing work involving fundamentally different solution strategies. Here we present an initial analysis of methods appropriate for the approximate solution of Case 1 problems. Methods for Case 2 problems have different requirements and are outside the scope of this chapter.

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, using a variety of strategies, potentially desirable topologies to evaluate. This new data is then used adaptively 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 [218]. In this situation, the dataset contains a number of unlabeled (unevaluated) 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 number of required labeling or evaluation tasks. This strategy is distinct from conventional supervised learning, also referred to as 'passive learning', that constructs a model in a single iteration. A number of efforts have focused on investigation of active learning for a variety of applications, including text mining [82,174,219,220,259], speech recognition [264], and computational biology [228]. Very limited research in this context has been focused on design synthesis or similar tasks, which is an additional motivating factor for this investigation of using active learning for circuit design synthesis.

Here the random forest algorithm is used to construct the predictive model in an iterative manner, including model validation and query selection steps. As an ensemble method, the random forest can prevent overfitting and reduce the variance by training on boot-

strap samples of the data. Active learning, in general, uses one of three types of sampling: membership queries, stream-based selective sampling, and pool-based sampling [218]. The pool-based sampling scheme is used here [153]. For the task of deciding whether to query or discard instances, several query strategies are tested here using two complete data sets from circuit synthesis problems, and identified the most promising active learning strategies for circuit synthesis.

Similar to adaptive surrogate modeling strategies common in design optimization [149, 248], the active learning strategy aims to find the best topologies by iteratively querying circuit topologies. The main difference between the active learning strategy presented here and surrogate modeling is that the inputs to the predictive model here have topological differences, whereas in surrogate modeling the inputs typically have only continuously-varying differences.

As discussed earlier in this section, the active learning strategy presented here aims to support the use of accurate comparison metrics and high-fidelity models. Suppose we want to solve a Case 1 synthesis problem where we can enumerate all unique and feasible topologies, but we cannot enumerate them. One strategy to make our problem solvable would be to reduce the number of topologies by reducing the component catalog size. This enables exact solution via enumeration, but the requires a less-accurate problem formulation. It would correspond to a move to the right away from the origin in Fig. 4.2 (A $\rightarrow$ B). Another option would be to keep the same design representation (component catalog), while selecting a simpler model for design evaluation (A $\rightarrow$ C). This may make complete evaluation possible, but would restrict available comparison metrics, and again results in a less-accurate problem formulation (moving away from the origin in both the predictive modeling and comparison metric dimensions). The active learning strategy allows us to remain at point A in Fig. 4.2 (close to the origin), at the cost of producing only approximately optimal solutions with no guarantees of the global optimality. In other words, a more accurate problem formulation is supported, but the solution is less accurate.

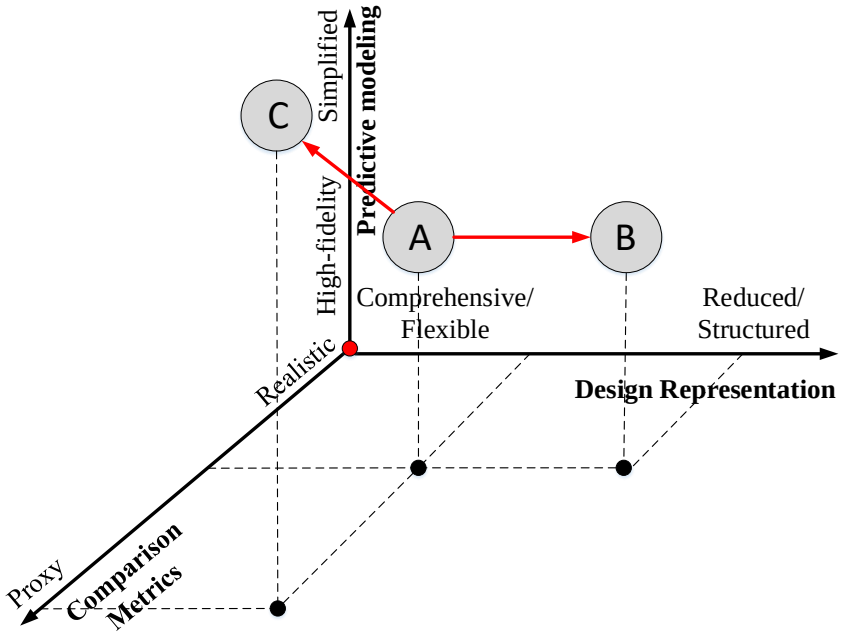



Figure 4.2: Conceptual design formulation space using different solution strategies

This chapter involves three primary contributions: 1) We introduce active learning as an effective tool for solving Case 1 synthesis problems. Using a circuit synthesis test problem, we compare several query strategies, and demonstrate the construction of a predictive model with fewer training samples that supports practical solution expense. 2) We generate new insights and identify query strategies that are effective for circuit synthesis via comparative analysis. 3) We discuss how the proposed framework could be extended to other topological design problems, such as active vehicle suspensions [109, 110] and fluid-based thermal management systems [201]. In addition, the relationship between active learning and adaptive surrogate modeling is clarified in this chapter.

The remainder of the chapter is organized as follows. In the next section, we detail the active learning method for circuit synthesis. Section 4.4 describes the frequency response circuit synthesis problem from which the dataset was obtained. Quantitative results are presented in Section 4.5. Discussion and conclusions are presented in Sections 4.6 and 4.7, respectively.

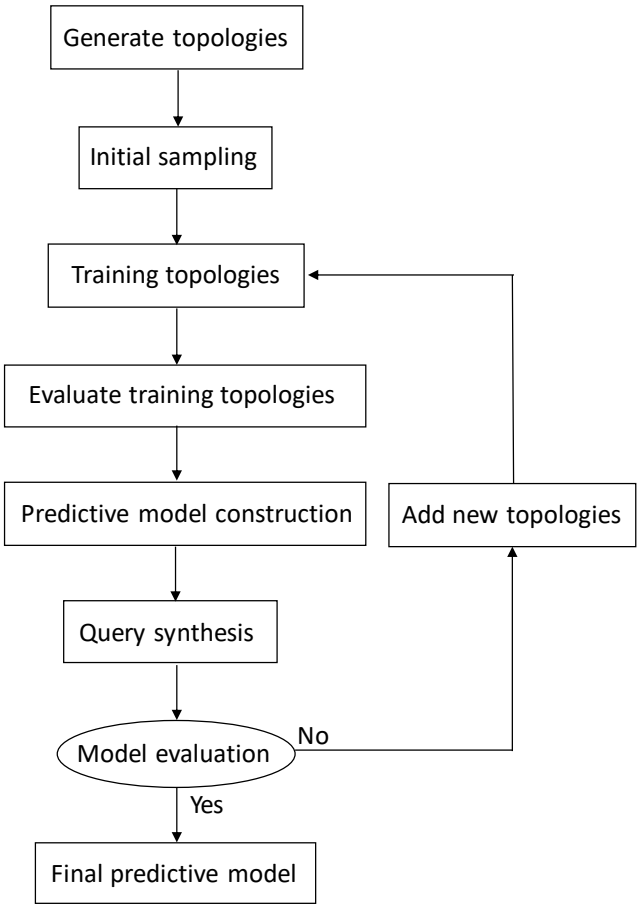



Figure 4.3: The active learning framework.

## 4.3 Methodology

In this section, the proposed active learning strategy for circuit synthesis is described. Figure 4.3 illustrates the overall active learning framework. The strategy assumes all possible $N$ circuit topologies $X = \{\mathbf{x}_i\}$ for $i = 1, 2, \ldots, N$ and $\mathbf{x}_i \in R^d$, are relatively easy to obtain, but evaluation of each circuit's performance metric $Y = \{y_i\}$ for $i = 1, 2, \cdots, N$, is relatively expensive. The mapping $f : \mathbf{x}_i \mapsto y_i$ is the true response function.

### 4.3.1 Initial Sampling

All unique and feasible circuit topologies are obtained using the efficient enumeration-based methodology introduced by Herber [109, 112]. 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 $N_l$ topologies are sampled randomly, denoted as an initial training set $X_l = \{\mathbf{x}_i\}$ for $i = 1, 2, \ldots, N_l$. The training set elements are then evaluated using the true response function to obtain the performance $Y_l = \{y_i\}$, where $y_i = f(\mathbf{x}_i)$ for $i = 1, 2, \ldots, N_l$. The unevaluated data set becomes $(X_u, Y_u)$, where $X_u = X \setminus X_l$ and $Y_u = Y \setminus Y_l$, respectively.

### 4.3.2 Predictive Model Construction

To approximate the performance $y_i$, a regression model is employed. Radial basis functions (RBFs) are often used with an artificial neural network to approximate the true unknown function [33]. Kriging is an approach for interpolation, where values are modeled using a Gaussian process [51, 216, 246]. Other methods may include response surface methodology (RSM) [26, 27], moving least-squares (MLS) [6, 144, 152], support vector regression (SVR) [244], adaptive regression splines [77], and inductive learning [146]. These approaches have been well-studied in the context of surrogate-based optimization [73, 248], an engineering design method that uses an approximate surrogate model constructed from true response samples as a strategy to reduce overall computational expense while maintaining solution accuracy. Optimization is performed using the computationally inexpensive surrogate model to make solution of an approximate problem tractable. Established surrogate modeling methods, however, are not applicable here because the design variables in the surrogate models are usually continuous, rather than the discrete topological variables needed for circuit synthesis (e.g., an adjacency matrix representation).

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 [31, 116, 117]. 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 [107]. The random forest uses bootstrap aggregating (also known as bagging) to improve the stability and accuracy of the learning algorithm [30]. For the given training set $T = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N_l\}$, bagging generates $B$ bootstrap sample sets $T^b = \{(\mathbf{x}_i^b, y_i^b), i = 1, \ldots, n^b, b = 1, \ldots, B\}$ 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 [30]:

$$\hat{y} = \hat{f}_B(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}). \tag{4.1}$$

The estimated standard deviation $s$ reflects uncertainty of the prediction:

$$s = \sqrt{\frac{\sum_{b=1}^{B} (\hat{f}_b(\mathbf{x}) - \hat{y})^2}{B - 1}}. \tag{4.2}$$

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 Bayesian optimization performed during the training phase [36, 87, 229].

### 4.3.3 Query Synthesis

Query synthesis is an influential element of active learning methods. To determine whether a new instance $\mathbf{x}$ should be queried (evaluated) 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) [73]:

$$LB(\mathbf{x}) = \hat{y}(\mathbf{x}) - As(\mathbf{x}), \tag{4.3}$$

where $\hat{y}(\mathbf{x})$ and $s(\mathbf{x})$ are the prediction and estimated standard deviation, respectively, given at $\mathbf{x}$ quantified by Eqs. (4.1) and (4.2). Here parameter $A$ controls the balance between exploitation and exploration; specifically, $A \to 0$ corresponds to pure exploitation, and $A \to \infty$ 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* [218]. Expected Improvement (EI) is a quantity that computes the improvement expected for a given mean $\hat{y}(\mathbf{x})$ and estimated standard deviation $s(\mathbf{x})$:

$$EI(\mathbf{x}) = (y_{min} - \hat{y}(\mathbf{x}))\,\Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right), \tag{4.4}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and PDF of the standard normal distribution, respectively, and $y_{\min}$ 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 [175]. Given a new instance $\mathbf{x}$ and training

sample $(\mathbf{x}_i, y_i) \in T$, the infill function is defined as [122, 131]:

$$v(\mathbf{x}) = D(\mathbf{x}) \cdot E(\mathbf{x}) \tag{4.5}$$

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 $E(\mathbf{x}) = \left| y_c - \hat{y}_c^{LOO} \right|$, 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 [122]. Similarly, density-weighted heuristics take into account both content information and input space region density for the new instance $\mathbf{x}$ [218]:

$$v(\mathbf{x}) = s(\mathbf{x}) \cdot \left( \frac{1}{|X_u|} \sum_{\mathbf{x}' \in X_u} d(\mathbf{x}, \mathbf{x}') \right)^{\beta} \tag{4.6}$$

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 $X_u$; $\beta$ is a hyper-parameter that balances the relative importance of both terms. The distance metric can be chosen as Euclidean distance or cosine similarity, depending on the problem.

### 4.3.4 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 [107, 124]. 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) [118]:

$$RMSE = \sqrt{\frac{\sum_{i=i}^{m}(y_i - \hat{y}_i)^2}{m}} \tag{4.7}$$

where $m$ is the number of validation points. Other metrics such as maximum absolute error and $R^2$ may also be considered [248]. 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.

### 4.3.5 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:

$$(X_l, Y_l) := (X_l, Y_l) \cup (X_q, Y_q) \tag{4.8}$$

$$(X_u, Y_u) := (X_u, Y_u) \setminus (X_q, Y_q) \tag{4.9}$$

The stopping condition is either 1) $RMSE$ falls below $\epsilon$, or 2) the active learning process reaches a predefined iteration number $n_{\text{iter}}$.

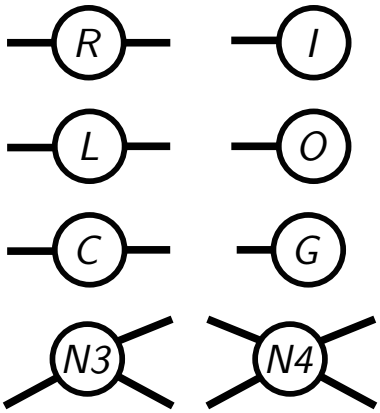



Figure 4.4: Some potential circuit component types [109]

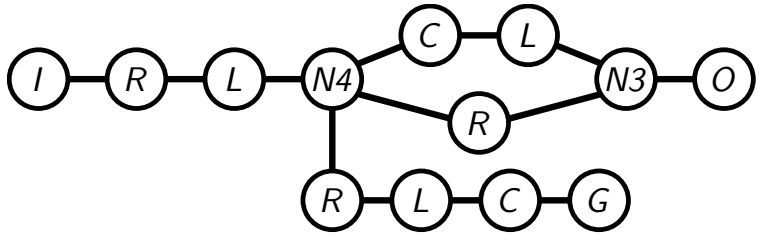
## 4.4 Design Problem

### 4.4.1 Problem Statement

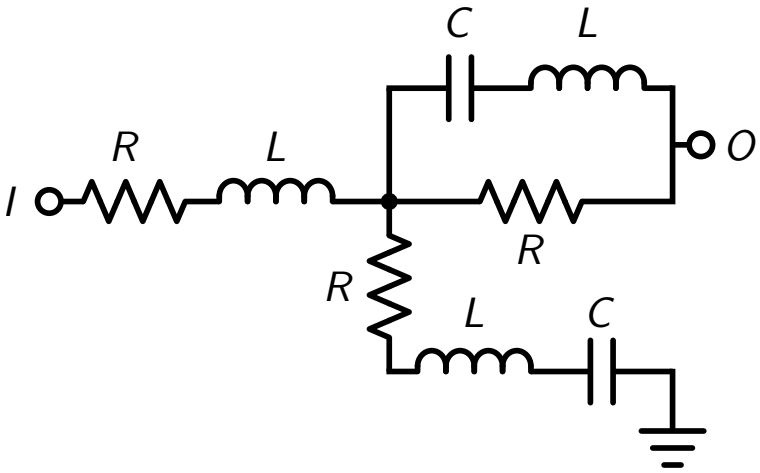
Here we present the case study for circuit design synthesis. A set of circuit elements shown in Fig. 4.4 with replicates can be combined construct a circuit, such as the one shown in Fig. 4.5. Resistors ($R$), capacitors ($C$), and inductors ($L$) are 2-port components, meaning that they must have connections 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 possible, 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. 4.5 and 4.6. Figure 4.6(a) 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 [94]:

$$|F(j\omega)| = \sqrt{\frac{2\pi}{10\omega}} \tag{4.10}$$

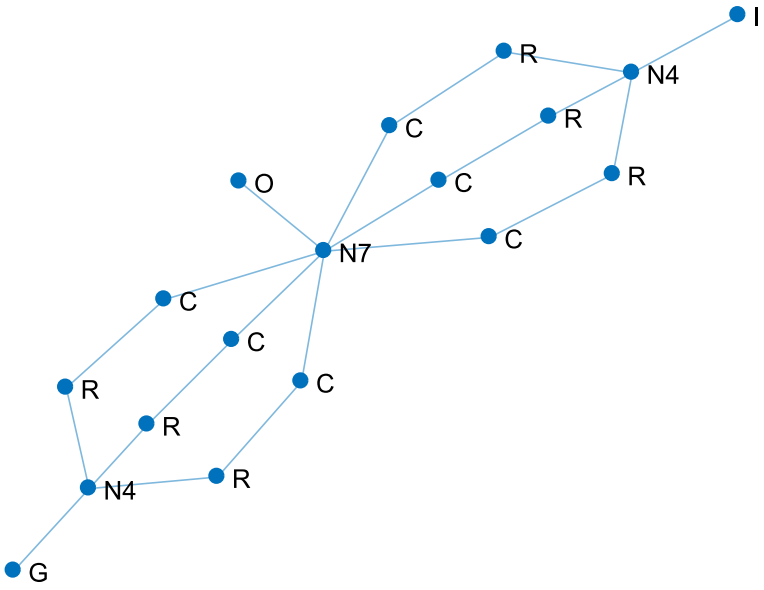



(a) Labeled graph

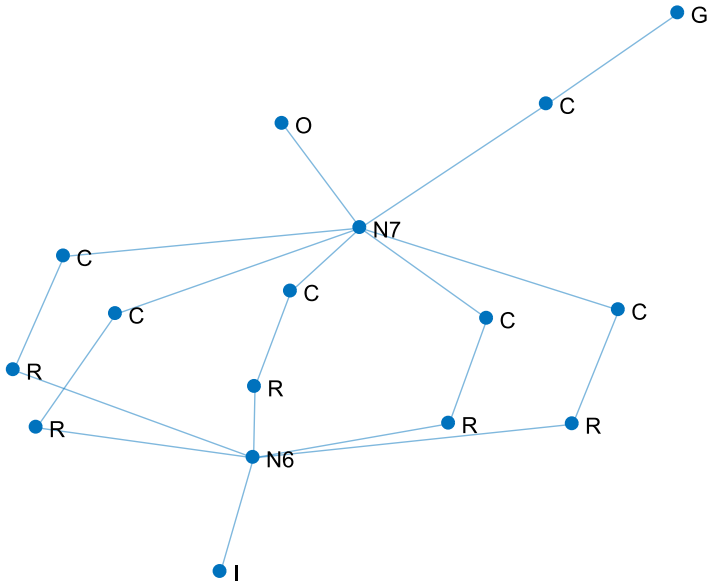



(b) Circuit schematic

Figure 4.5: Different representations for the same circuit [109]




(a) Example #1




(b) Example #2

Figure 4.6: Two different circuit topologies with twelve 2-port components [109]

where the frequency range of interest is:

$$0.2 \leq \frac{\omega}{2\pi} \leq 5, \tag{4.11}$$

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 [109]. All circuit topologies represented as labeled graphs were enumerated under a set of specifications using the enumeration algorithms in Ref. [112]. To evaluate the performance given a desired circuit topology, we consider the following minimization problem [109]:

$$\min_{\mathbf{z}} \quad E = \sum |r_k(\mathbf{z})|^2 \tag{4.12a}$$

$$\text{subject to:} \quad \mathbf{l} \leq \mathbf{z} \leq \mathbf{u} \tag{4.12b}$$

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. (4.10); ($\mathbf{l}$, $\mathbf{u}$) are the upper and lower bounds for $\mathbf{z}$.

### 4.4.2 Design 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 [109]:

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

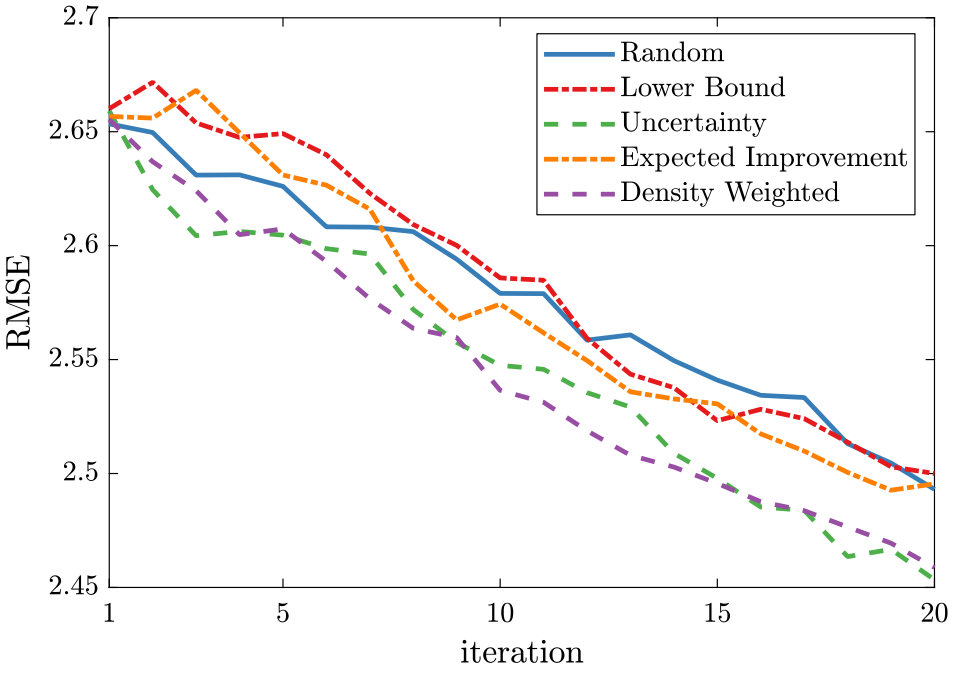
$$\text{(set 2)} \quad R \in [10^{-2}, 10^{5}]\ \Omega, \quad C \in [10^{-10}, 10^{0}]\ F$$

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

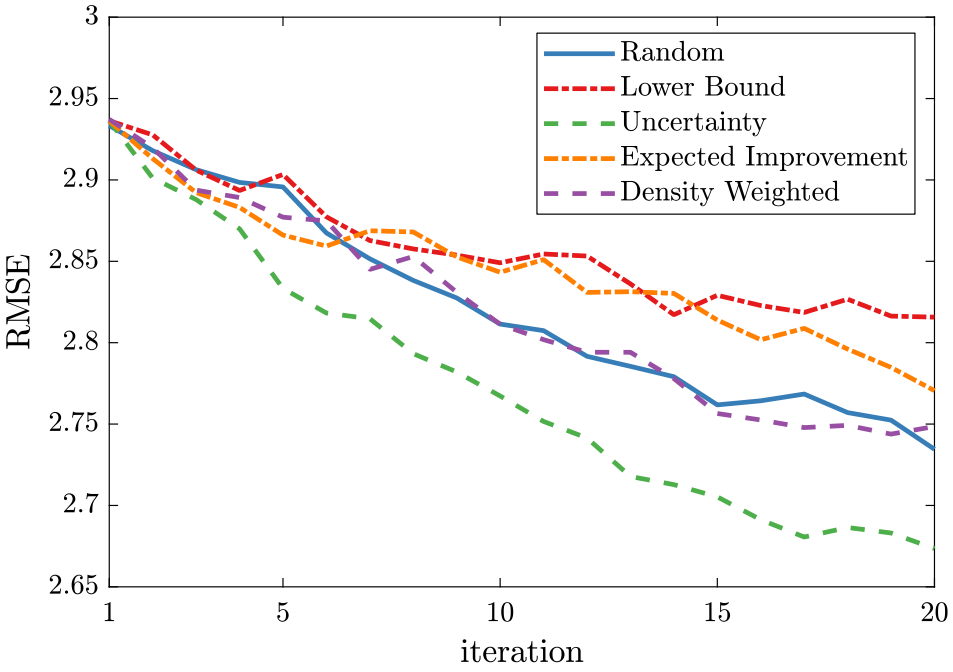
bounds of replicates for each component type. Network structure constraints (NSCs) were used to define the feasibility for the graphs. A collection of 43,249 circuit topologies (denoted as $X$) with unique transfer functions were enumerated, and thus the corresponding circuit performance could be evaluated. For more details see Refs. [109, 112].

### 4.4.3 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 4.6(a) and 4.6(b) 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 $18 \times 18$ and $16 \times 16$, 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 method for vectorizing features, is used to overcome this issue, turning the adjacency matrix feature space into a compact vector representation [254]. 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 dimensional vector into a lower dimensional feature space [252].




(a) Set 1




(b) Set 2

Figure 4.7: Comparisons between the different query methods

## 4.5 Results

### 4.5.1 Comparison of Query Strategies

We randomly selected an initial training set of 10,000 circuits that were then each evaluated to obtain corresponding true responses. 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 were queried from the training set in each iteration according to the criteria specified in Eqs. (4.3)–(4.6). The learning process is terminated after $N_{\text{iter}} = 20$ iterations. We then compare the different query strategies against a benchmark random sampling strategy. In random sampling, a set of 500 data points are obtained via random sampling 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 4.7(a) 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 4.7(b) 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. Later in this section additional parametric studies are performed to test the generality of these results.

### 4.5.2 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, accurate 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 $(\tau_1, \tau_2)$ of size $n$ [132]. 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 4.8 and 4.9 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 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. Similar observations can be found for Set 2 in Fig. 4.9. The Kendal distances through the iterations

were also investigated in Figs. 4.10(a) and 4.10(b). 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 when considering the use of this synthesis method.

### 4.5.3 Parametric Studies

We conducted a parametric study on the number of new samples taken at each active learning iteration. The motivation for this study is to explore how these parameters affect the active learning performance, provide more fair assessments, and generate additional insights for practical application of the active learning synthesis strategy. A subset of 10,000 samples was obtained via random sampling of the initial training set. The number of new samples per iteration tested were: 250, 500, 750, and 1,000. Because the training set was enlarged at each iteration, we terminated the active learning after the training set neared 20,000 total circuits. For example, it required 11 iterations to reach a total of 20,000 circuits using 1,000 new samples per iteration, but only 15 iterations were needed to obtain 20,500 circuits and please note the total of circuits in the final training set may not be identical but are similar.). The remaining circuits were used as the test set. Again, we use the normalized Kendall tau distance $\tilde{K}$ as the metric. Table 4.1 summarizes the results. The number of samples taken per iteration does not significantly affect $\tilde{K}$. However, it was found that using a sample size of 750 slightly outperforms the other levels.

In addition to a parametric study on sample size, we investigated parameter $A$ used in the statistical lower bound approach defined in Eq. (4.3). The results, based on 750 samples per iteration, are shown in Table 4.2. Recall that $A$ controls the balance between exploitation and exploration. Values of 0.1, 2, and 10 for $A$ were examined, and compared with uncertainty sampling (the best-observed sampling strategy for both sets). For instance,

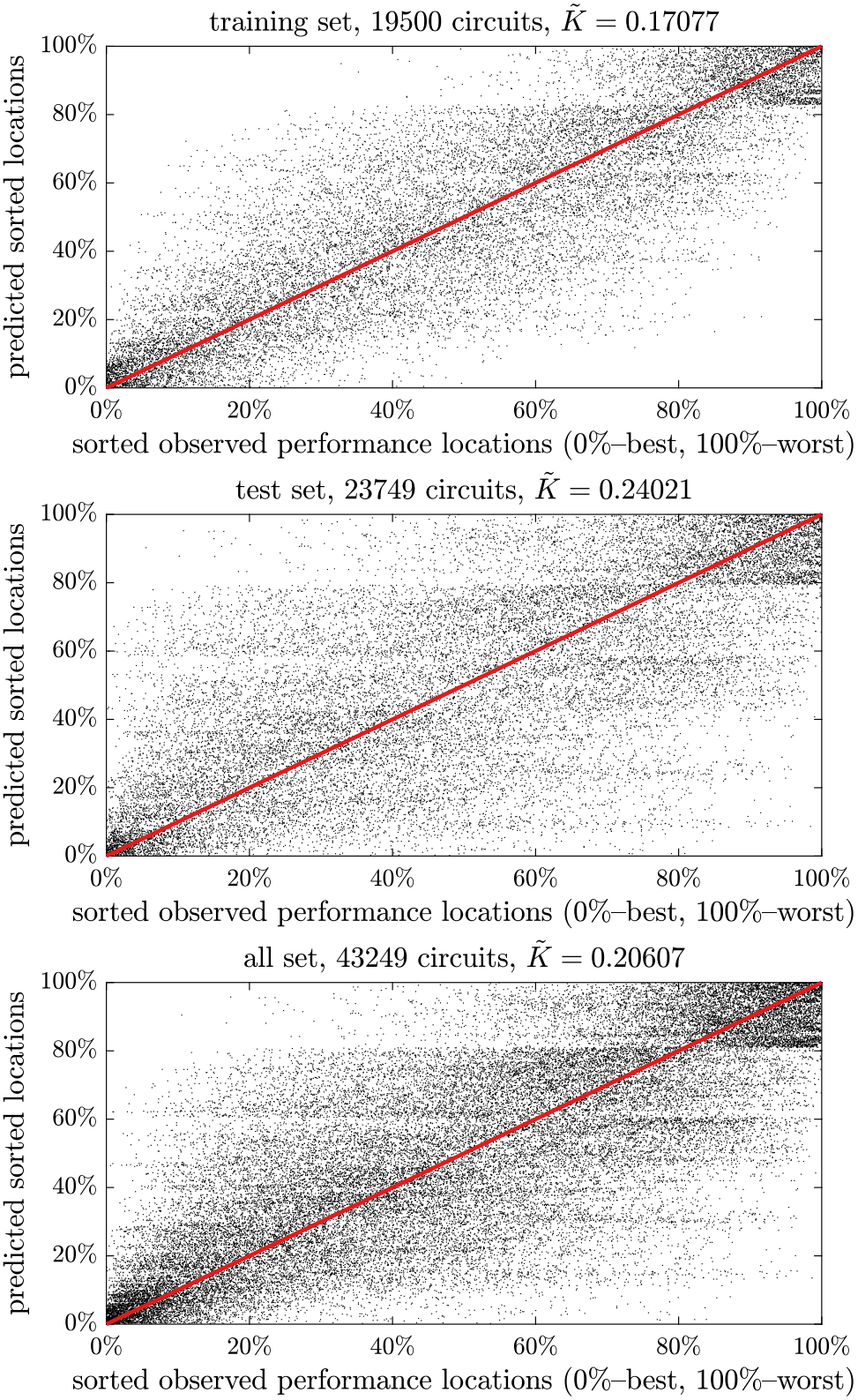



Figure 4.8: Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 1)

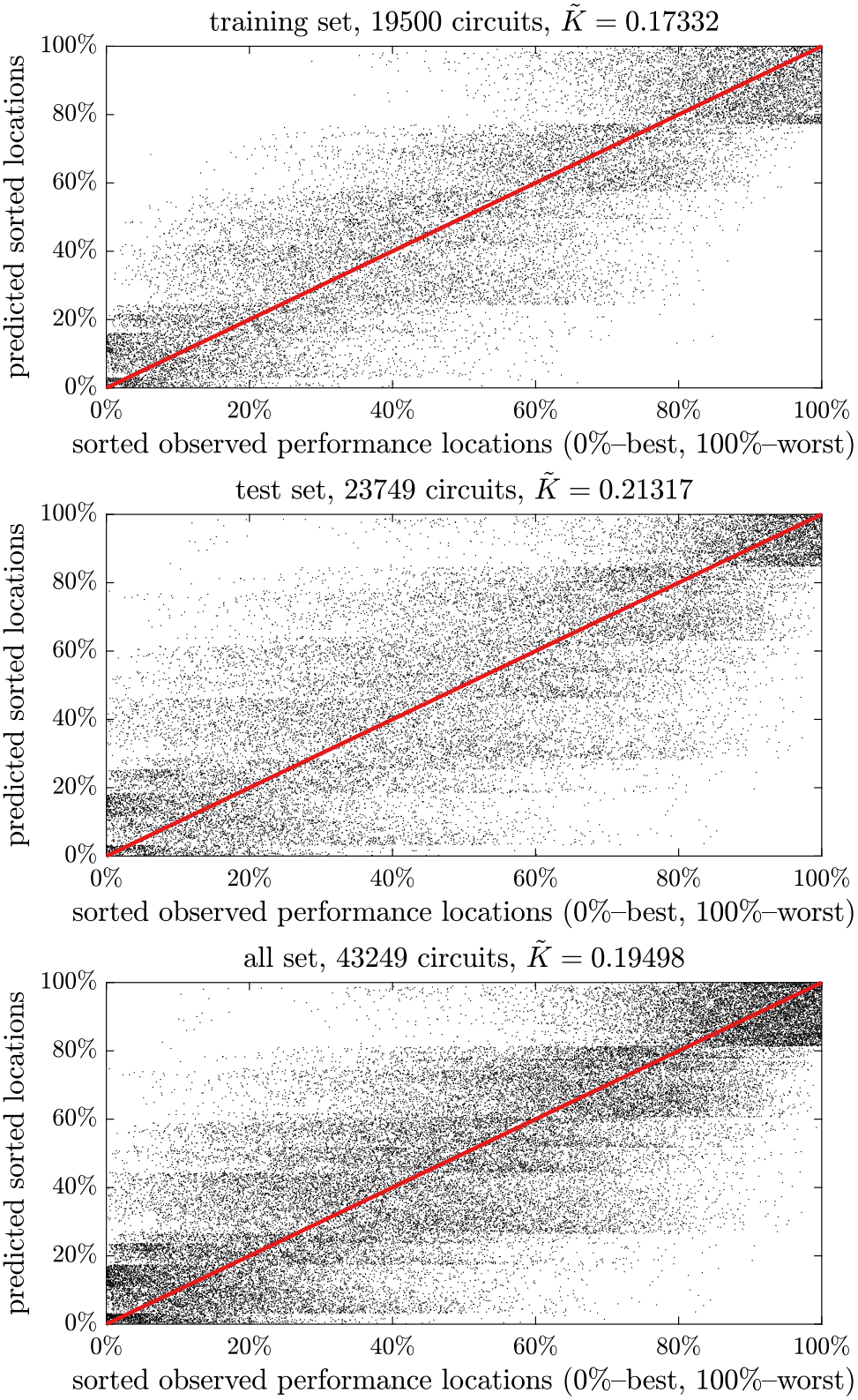
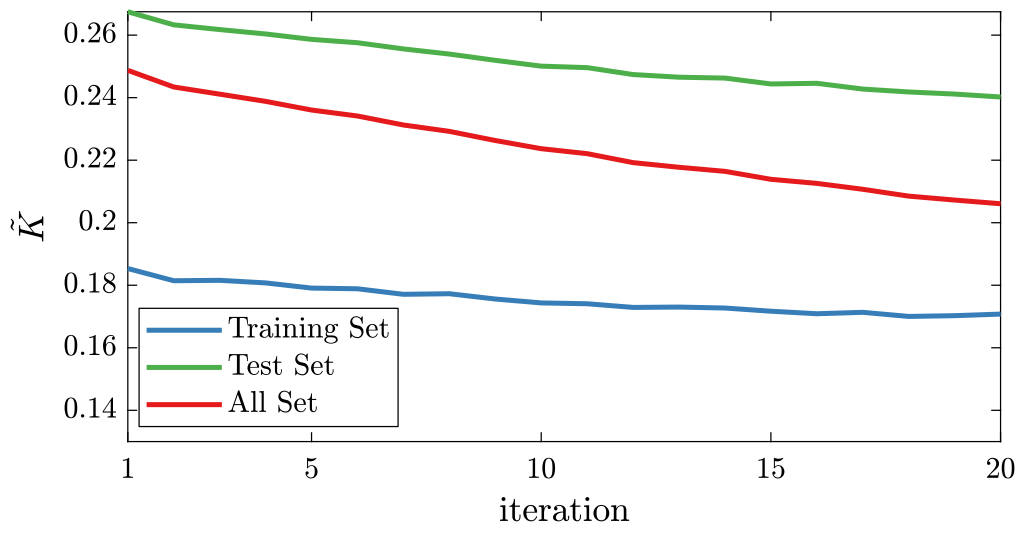
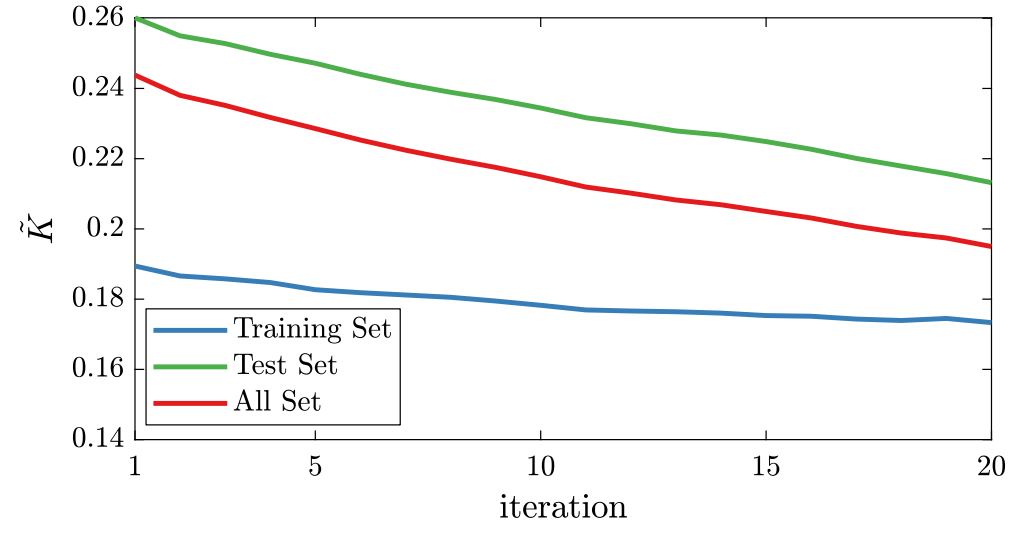



Figure 4.9: Predicted vs. observed sorted performance locations for different sets of circuits and their corresponding Kendall tau distances (Set 2)




(a) Set 1




(b) Set 2

Figure 4.10: The Kendall tau distances through the active learning iterations

Table 4.1: The parametric study on the number of new samples taken at each iteration

| | | $\tilde{K}$ | | |
|---|---|---|---|---|
| | Sample Size | Training set | Test set | All set |
| Set 1 | 250 | 0.1703 | 0.2382 | 0.2044 |
| | 500 | 0.1708 | 0.2402 | 0.2061 |
| | 750 | 0.1690 | 0.2378 | 0.2020 |
| | 1000 | 0.1703 | 0.2403 | 0.2049 |
| Set 2 | 250 | 0.1739 | 0.2131 | 0.1951 |
| | 500 | 0.1732 | 0.2132 | 0.1950 |
| | 750 | 0.1734 | 0.2117 | 0.1935 |
| | 1000 | 0.1731 | 0.2129 | 0.1944 |

Table 4.2: The parametric study on parameter $A$ in the statistical lower bound approach

| | | $\tilde{K}$ | | |
|---|---|---|---|---|
| | Method | Training set | Test set | All set |
| Set 1 | $A = 0.1$ | 0.1750 | 0.3208 | 0.2438 |
| | $A = 2$ | 0.1660 | 0.2912 | 0.1917 |
| | $A = 10$ | 0.1643 | 0.2204 | 0.1681 |
| | Uncertainty Sampling | 0.1690 | 0.2378 | 0.2020 |
| Set 2 | $A = 0.1$ | 0.1885 | 0.3118 | 0.2386 |
| | $A = 2$ | 0.1901 | 0.3094 | 0.1965 |
| | $A = 10$ | 0.1776 | 0.2461 | 0.1757 |
| | Uncertainty Sampling | 0.1734 | 0.2117 | 0.1935 |

the $\tilde{K}$ with $A = 10$ in the test and all sets are significantly less than those of values at 0.1 and 2. The uncertainty sampling strategy performs better in the test sets, consistent with the results presented in Figs. 4.7(a) and 4.7(b). Therefore, the result of $A = 10$ being preferred reveals that more weight should be placed on exploration to improve the active learning performance.

## 4.6 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 design evaluation cost reduction. For instance, query by disagreement (QBD) [49,106] and query by committee (QBC) [76,221] may be appropriate for the ensemble method. A clustering-based method could also be considered [130,190]. It 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 been used primarily for classification problems in text mining. It may be 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. (4.5) is computationally expensive to evaluate (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. Variance reduction, often known as *optimal experimental design*, is also limited [44,71]. 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 [218]. 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 possible, 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 [29, 78, 79]; other learners such as a radial basis function network may be applicable [33]. 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 [248]. 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. MBDO strategies have been extended to multi-objective surrogate modeling [1, 149, 209]. However, there are also some distinctions. First, the vector of design variables $\mathbf{x}$ in MBDO are often assumed to be composed of continuous real variables, 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 [199, 216], 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 Chapter 5 to address Case 2 synthesis problems discussed in Section 4.1, 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 generative model is a probabilistic model capable of producing both the observations and targets in the data set. Restricted Boltzmann machines [115], variational auto-encoders (VAEs) [137], and generative adversarial networks (GANs) [91] 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 [99]. For the circuit synthesis problem, GANs could be a promising option to generate circuit topologies.

## 4.7 Conclusion

In this chapter, 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 topol-

ogy; 2) we can make more accurate predictions using active learning than the conventional passive learning scheme (i.e., random sampling). The numerical experiments indicate that the uncertainty sampling query strategy is most effective among the tested methods for the tested circuit synthesis problem. 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.

There are some shortcomings and limitations in the results. For instance, we can only predict the rankings with moderate accuracy in Figs. 4.8 and 4.9. Figures 4.10(a) and 4.10(b) also indicate slow convergence behavior. There could be other methods that better predict the Kendall tau distance. Other metrics such as Sperman's rank correlation coefficient [182], could be used to characterize the ranking predictions. It is speculated that improving the effectiveness of the query strategy and learner may alleviate these issues.

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 (i.e., passive learning). Recent studies indicate that no query selection strategy is consistently or clearly the best across a general range of problems [218] because success 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 [126]. 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.

While only a single test problem was investigated, it is clear that the active learning strategy is promising, and it appears that it could be extended to other system topology design tasks. For instance, the active learning strategy is applicable to the low-pass filter problem given in Ref. [109] as it has the same problem structure as the test problem presented here. Active vehicle suspension design can also be posed as a similar problem using labeled

graphs, and this active learning strategy may be suited for evaluation cost reduction [110]. 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.

# Chapter 5

# HETEROGENEOUS SYSTEM TOPOLOGY DESIGN: CASE 2

**Collaborative Acknowledgement: Dr. Daniel Herber** provided circuit synthesis data, including both design specifications and evaluation results, using efficient enumeration for both the frequency response and low-pass filter design problems.

## 5.1 Introduction

In this chapter, we introduce a solution strategy for Case 2 synthesis problems and demonstrate this strategy using two different circuit synthesis problems. Recall that for Case 0 problems, it is possible to enumerate and evaluate all unique and feasible design topologies. In the previous chapter, active learning was used for Case 1 problems where we could enumerate all of the topologies of interest, but could only evaluate performance for a subset of them. Case 2 is more challenging; the topological design space is too large for enumeration, so different methods are needed.

The Case 2 synthesis solution strategy presented here is based on the concept of creating an indirect design representation that implicitly satisfies NSCs. Thus, infeasible topologies could be avoided when exploring candidate topologies. Existing efficient enumerative meth-

ods can solve Case 0 problems, but are limited by problem size, and cannot solve Case 2 problems (i.e., all topologies cannot be enumerated in practice). Evolutionary algorithms (EAs) are an alternative solution strategy that could be attempted, and do not require comparison metric properties such as differentiability, but when EAs with simple direct design encodings are tested for problems with a vast topological design space, they generally fail to produce meaningful results [48]. Direct encodings usually are high-dimension, and can lead to intractable problems if structure cannot be leveraged. Low-dimension abstract design representation strategies, including genotype-to-phenotype mappings, can help ease solution difficulties. Construction or identification of generative algorithms [134] or grammar rules [38] can aid creation of effective indirect encodings (genotype-to-phenotype mappings) to help improve EA solution effectiveness for large-scale topological design problems. So far, however, generative algorithms rely on human expertise to identify appropriate mappings, and grammar rules most often have been used in an exploratory way to support human creativity, as opposed to use for topology optimization.

More generalizable data-driven indirect representations have been proposed, but for restricted types of topology optimization problems. For instance, Cheney et al. successfully applied computational pattern-producing networks (CPPNs) to design of soft robots with materials [48]. While the combinatorial design space was large, the ground structure network that served as the basis of the design representation was sparsely connected compared to the types of dense ground structures considered in this chapter. More specifically, elements could only be connected to geometrically close neighbors, whereas in more general heterogeneous topology optimization problems elements can be connected to a wider range of other elements. In addition, the work by Cheney et al. did not consider NSCs other than geometric proximity. More general problems must manage several classes of NSCs.

An indirect representation may be chosen as a possible remedy for reducing the computational expensive. Recall that in the previous chapter, a reduced-dimension design representation could simplify the design space (see Fig. 4.1), but typically reduces the number of

design candidates that may be considered, resulting in a less accurate design formulation. If design space coverage is reduced, it would be ideal to reduce it in a targeted way that eliminates infeasible or other undesirable designs from consideration. For example, an indirect generative design representation was introduced by Khetan et al. where certain design constraints were guaranteed to be satisfied in an implicit manner [134]. This resulted in a more efficient design space search (i.e., higher-quality solutions and lower computational expense), but was developed only for a specific homogeneous topology optimization problem. All nodes represent the same material with the same type of functionality and allowed connectivity, and are parameterized by the same continuous variables for size optimization. Here, we would like to solve more general synthesis problems where nodes can represent fundamentally different components, and overcome the limitations of EAs with direct encodings (reviewed in the previous chapter), reliance on human expertise for creating generative encodings (e.g., [134]), and the limitation of existing methods based on data-driven encodings to simpler topology optimization problems (e.g., [48]).

As a strategy to remove reliance on human expertise, a data-driven strategy is introduced where targeted, indirect design representations are constructed in an automated way from design data derived from smaller representative component catalogs. Here we focus on techniques for generating circuit topologies in a manner that implicitly satisfies NSCs, improving the probability that generated circuits are feasible. This capability is particularly important for Case 2 synthesis problems because, without implicit constraint satisfaction, arbitrarily generated candidate topologies are highly unlikely to satisfy NSCs (see Ref. [109] for a discussion of the ratio of infeasible to feasible unique architecture candidates for a typical synthesis problem, which provides evidence for this unlikeliness). In other words, this targeted mapping narrows the search space and supports significantly more efficient search compared to a direct GA implementation or other available strategies for Case 2 synthesis problems.

It is proposed here to utilize a generative model to produce circuit topologies for use

in Case 2 circuit synthesis problems. A generative model, using language from the machine learning literature, is a probabilistic model that describes how data was generated. Variational autoencoders (VAEs) and generative adversarial networks (GANs) are two important types of generative models. VAEs have been applied to topology optimization and microstructure design. For instance, Guo et al. developed an indirect representation for heat spreader design using VAEs and style transfer in topology optimization [99]. Cang et al. predicted physical properties of heterogeneous materials via a deep VAE-based model [39]. However, it is well-recognized that VAEs tend to generate blurry samples due to the maximum likelihood training paradigm [66]. In contrast, GANs adopt a generator-discriminator mechanism in an adversarial manner and have exhibited a number of advantages over other generative models. For example, the generator requires few restrictions (e.g., no Markov chains or variational bounds are needed), and GANs are better at generating sharp samples than VAEs [90].

In a GAN, the generator attempts to produce "fake" samples generated from a low-dimensional latent vector that avoid being detected as fake by the discriminator. This adversarial mechanism is equivalent to a zero-sum game. GANs have been used to synthesize a number of realistic objects for design, such as interior design [207] and 3D objects [254]. However, GANs in engineering design, particularly circuit synthesis, have not been studied thoroughly. Here we utilize GANs as an indirect design representation for circuit synthesis problems. We design several numerical experiments to investigate the effectiveness of GANs in this context, including several GAN variants. As highlighted above, one objective here is to generate feasible topologies in a manner that implicitly satisfies circuit constraints to support more efficient Case 2 synthesis problem design search. The case studies presented here are frequency response and low-pass filter analog electric circuit design problems.

We designed the numerical experiments to test the efficiency of generating circuit topologies for the GAN-based method. Figure 5.1 is a Venn diagram that helps illustrate the rationale behind the use of GANs for Case 2 indirect encodings. In practice, it would be

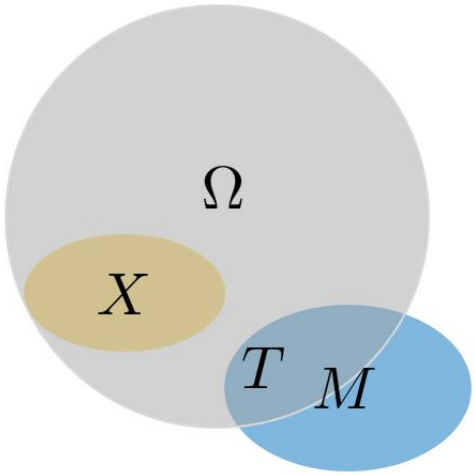



Figure 5.1: A Venn diagram for GAN-based methodology

impossible to obtain all possible unique and feasible circuit designs (i.e., set $\Omega$) via enumeration. Only a statistical sample $X$ that represents $\Omega$ can be obtained in a practical amount of time. Here we train different GAN models using $X$, and generate a new design set called $M$ through the generator. Suppose there are $T$ out of $M$ designs that are feasible. It is desirable to have the feasibility ratio $T/M$ as large as possible. The numerical experiments help quantify the effectiveness of various GAN models via this ratio, with the expectation that larger $T/M$ ratios will support more efficient topological design space exploration. Other factors will also have an impact, such as indirect encoding dimension, feasible design space ($\Omega$) properties, and search algorithm properties.

It should be noted here that GANs have been used successfully in a similar way using images of human faces [91,207]. For example, from a small set of human faces $X$, GANs have been trained to generate new faces not in $X$ that appear to be realistic. One could define realistic appearance to be a feasibility requirement. Generated faces that look realistic would then be the set $T$, and unrealistic faces would be $M\backslash\Omega$. The discriminator improves quality of generated faces by detecting fake ones. This established use of GANs in realistic synthetic human face generation from limited real data was observed to be parallel to the needs posed by Case 2 synthesis problems, and provides the rationale for choosing to investigate GANs for this purpose.

Using the GAN-based method (or other similar indirect encodings) for Case 2 synthesis problems involves a tradeoff between formulation and solution accuracy. As illustrated in the previous chapter, enumeration-based methods find exact solutions to Case 0 synthesis problems[1] (perfect solution accuracy), but only for simplified topological design spaces. As a result, enumerative strategies lie fairly far to the right along the design representation axis (see point $A$ in Fig. 5.2). Solution is exact, but formulation is inexact. Can we improve formulation accuracy by giving up some solution accuracy? Would this result in a better overall design solution?

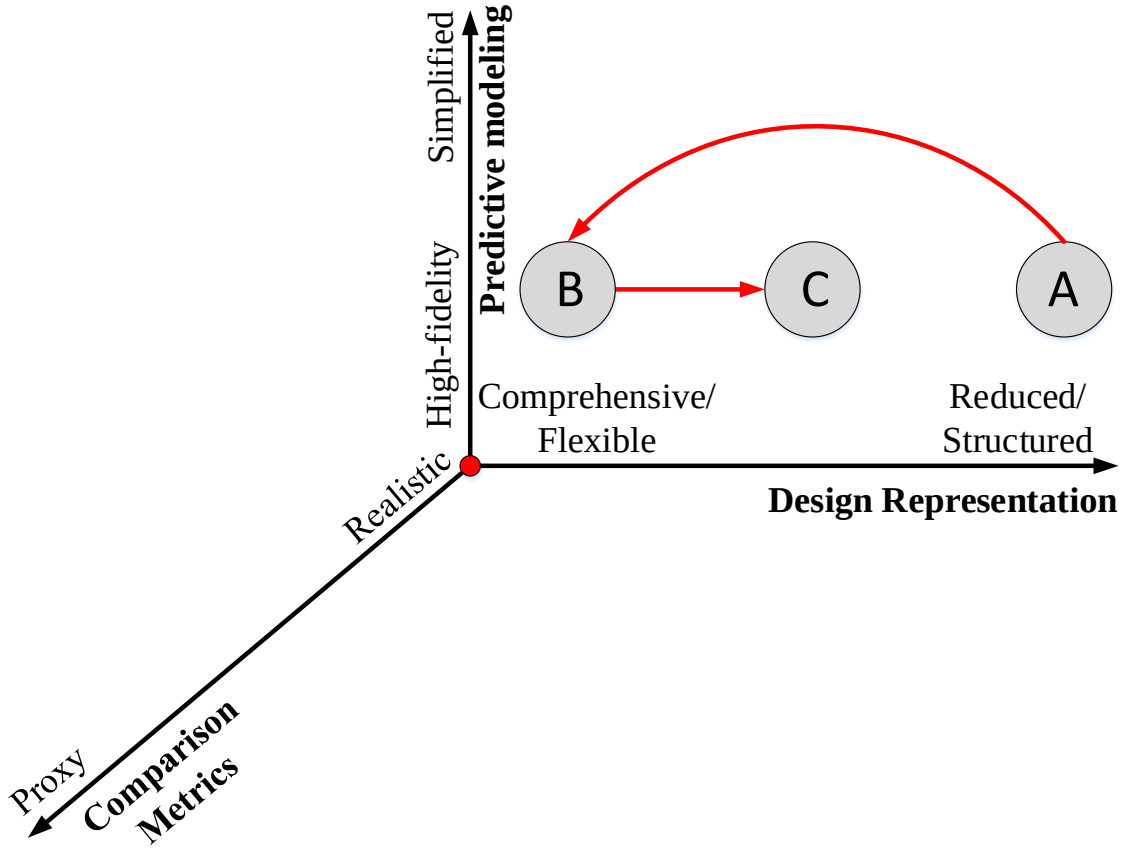



Figure 5.2: Design formulation analysis of the GAN-based methodology

[1]A Case 0 synthesis problem is one where it is possible to enumerate and evaluate all topologies defined by the problem formulation.

If we improve the comprehensiveness of the design representation to expand the set of topologies considered, we can move to the left toward the origin (point B), but this problem is no longer solvable via enumeration (Case 2 synthesis[2]). Please observe that reducing predictive model fidelity or comparison metric realism will not make a Case 2 problem solvable because the complete set of topologies cannot be enumerated. We could attempt to solve such a problem using an EA with a direct encoding, this is unlikely to be successful for such problems. Introducing a GAN as an indirect encoding reduces design representation dimension, and moves the formulation to point $C$, which is the right of point $B$, but not as far right as point $A$. We recognize that a GAN-based encoding will not cover the design space as comprehensively as point $B$ (some formulation inaccuracy/greater distance from that origin compared to $B$), but gives up some design coverage in return for practical solvability. In summary, by expanding the topological design space (moving from $A \rightarrow B$) and then representing this expanded design space approximately using a GAN (moving from $B \rightarrow C$), we create a net improvement in problem formulation, producing a more accurate formulation that may be solvable using indirect EA encodings. The formulation is more accurate than the Case 0 synthesis problem (point $A$), but the solution is approximate (not exact) due to the nature of EAs or other solution methods that would apply here.

Similar to the VAE-based design representation in Chapter 3 (see Fig. 3.1), the GAN-based method presented here, in essence, restructures the design space in terms of an abstract design representation. It may permit either low-or high-fidelity comparison metrics and predictive models, but more importantly directly tackles the challenge of not having enumerative data from the desired problem directly available for use in the solution strategy.

The contributions in this chapter include: 1) a new GAN-based synthesis strategy that entails a new perspective in designing electronic circuits that is fundamentally distinct from established circuit design methodologies. Specifically, artificial intelligence and design data

[2]Recall that Case 1 synthesis problems are not solvable via enumeration alone because of evaluation expense, not because the topologies cannot be enumerated. Here we focus on cases where the topological design space is too large to support enumeration; this property is independent of evaluation expense.

are leveraged to restructure topological design spaces to support efficient exploration. 2) Several GAN models are tested, and the improved WGAN is shown to be the most effective in identifying the feasible circuits using the reduced-dimension design representation. This indirect representation supports the execution of an active learning strategy by facilitating the generation of high-performance circuit topologies in an efficient manner. To our best knowledge, this is the first work where adversarial learning is applied to electronic circuit synthesis design.

The remainder of this chapter is organized as follows. The next section introduces the basics of the GANs. Section 5.3 describes the heterogeneous circuit synthesis design problems and associated design data. Section 5.4 outlines the numerical experiments. The numerical results are reported in Section 5.5. The discussion and conclusion are presented at end of this chapter.

## 5.2 Generative Adversarial Networks (GANs)

A generative adversarial network (GAN) is a class of unsupervised learning models introduced by Goodfellow et al. [91]. The GAN is composed of two artificial neural networks, a generator and discriminator. Figure 5.3 illustrates a basic GAN framework. Let $X_{\mathrm{r}} = \{\mathbf{x}\}_i^N$ for $i = 1, 2, \ldots, N$ and $\mathbf{x}_i \in R^d$ denote the real data samples drawn from a probability density distribution $P_r$. A latent vector $\mathbf{z} \in R^m$ is pre-defined with a prior density distribution $p_{\mathbf{z}}(\mathbf{z})$; here $p_z(\mathbf{z})$ is often chosen as a multivariate normal or uniform distribution. The role of the generator $G(\mathbf{z}; \theta_G)$ is to produce samples $\mathbf{x}_{\mathrm{g}}$ with a probability density $P_{\mathrm{g}}$ that approximates $P_{\mathrm{r}}$. Given a random sample $\mathbf{z}$, the generator maps the latent space $Z$ to the original data space $X$, i.e., $G : Z \to X$. The goal is to find parameter values $\theta_G$ for the generator $G(\mathbf{z}; \theta_G)$ such that $P_g$ is as close to $P_r$ as possible. Discriminator $D(\mathbf{x}; \theta_D)$ takes a sample $\mathbf{x} \in X$ as the input and outputs the probability of $\mathbf{x}$ being real. Both $D(\mathbf{x}; \theta_D)$ and $G(\mathbf{z}; \theta_G)$ can update iteratively in such a way that the generator produces "fake" samples capable of

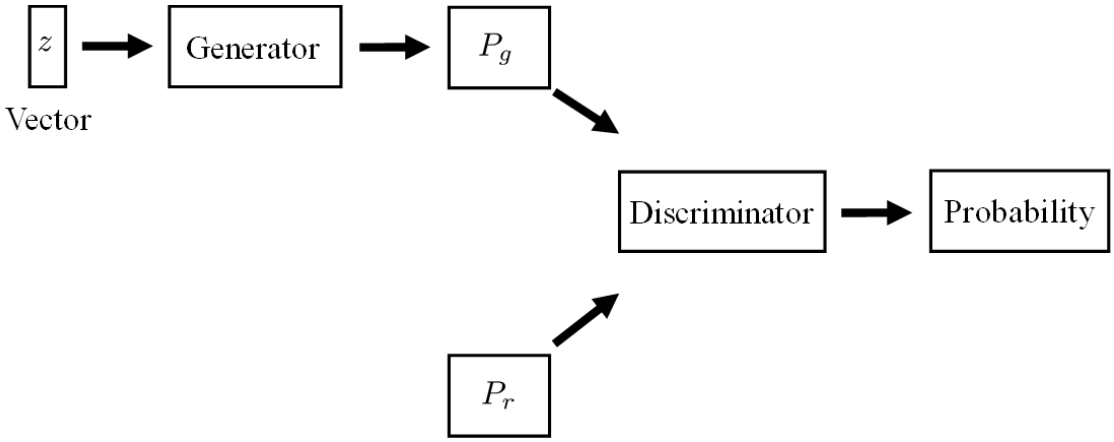



Figure 5.3: A basic GAN framework

"fooling" the discriminator, while the discriminator aims to distinguish the "fake" samples given by the generator from the real. The process is equivalent to a minimax two player game and eventually a Nash Equilibrium is reached [194]. The discriminator maximizes the probability of assigning the correct label to the samples from $P_g$ and $P_r$; the generator is trained to minimize $\log(1 - D(G(\mathbf{z}))$. The value function $V(G, D)$ is written as [91]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim P_r}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_z(\mathbf{z})}[1 - \log D(\mathbf{G}(\mathbf{z}))] \tag{5.1}$$

where $G(\mathbf{z})$ is the sample produced by the generator. Because $\mathbf{x}$ is real, $D(\mathbf{x})$ will be close to 1. $D(G(\mathbf{z}))$ outputs the probability for sample $G(\mathbf{z})$. The generator expects value of $D(G((z))$ to be large for a fixed discriminator $G$, and therefore a minimization over $G$ is used. As a powerful discriminator, $D(\mathbf{x})$ tends to be larger and $D(G(\mathbf{z}))$ should be smaller. Consequently, the value function $V(D, G)$ is maximized over $D$. In practice, $\min_G[1 - \log D(\mathbf{G}(\mathbf{z}))]$ often is replaced by $\max_G \log D(G(\mathbf{z}))$, because $1 - \log D(\mathbf{G}(\mathbf{z}))$ saturates when $G$ is poor, and $D$ can reject confidently the samples that are different from $\mathbf{x}$ used in the earlier training [91].

### 5.2.1 Standard GAN Shortcomings and Resolutions

While GANs have significant advantages, they are extremely difficult to train. One challenge is that the algorithm used to find the Nash equilibrium, usually a gradient-based method, may fail to converge [92, 217]. GANs also suffer mode collapse, i.e., cases where the generator maps several different latent vectors to the same output sample [90]. Because the loss function given in Eqn. (5.1) measures the Jensen–Shannon divergence between $P_r$ and $P_g$, this metric is insufficient when $P_r$ and $P_g$ do not overlap. Arjovsky and Bottou explained that supports of $P_r$ and $P_g$ lying on the lower dimensional manifolds contribute to training instability [12]. Several strategies have been proposed to improve stability [12, 178, 217]. For instance, Deep convolutional generative adversarial networks (DCGANs) apply convolutional layers to the discriminator and deconvolutional layers to the generator [121]. The convolutional filters extract features from the previous layer, and transforms $\mathbf{x}$ to the class label (probability). The generation of the DCGAN in essence flips the direction from $\mathbf{z}$ to $\mathbf{x}$ through multi-layer filters. The DCGAN is well-known for its stability and robustness properties, and are thus used widely. Other variants of GANs are also available, such as LAPGANs [59], AdaGANs [239], least square GANs [170], loss-sensitive GANs [205], and others.

### 5.2.2 Wasserstein GANs (WGANs)

Wasserstein GANs (WGANs) have attracted much attention recently [13, 95]. Rather than using JS divergence, The WGAN adopts Wasserstein Distance (also known as *Earth Mover's distance*) as the metric quantifying the similarity between $P_r$ and $P_g$:

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \tag{5.2}$$

where $\Pi(P_r, Pg)$ refers to the set of all possible joint distributions. The Wasserstein distance can give a meaningful and smooth representation of the distance between $P_r$ and $P_g$, even if

there exist no overlaps between the lower dimensional manifolds of both distributions. The WGANs makes the training process stable, and ensures diversity of the generated samples. Even multilayer perceptions (i.e., fully-connected layers) are sufficient to produce realistic samples. Arjovsky et al. proposed a transformation for the intractable term $\inf_{\gamma\sim\Pi(P_r,P_g)}$ in Eqn. (5.2) [13]:

$$W(P_r, P_g) = \frac{1}{K} \sup_{||f||_L \leq K} \mathbb{E}_{\mathbf{x}\sim P_r}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_g}[f(\mathbf{x})] \tag{5.3}$$

where $||f(\cdot)||_L \leq K$ is the $K$-Lipschitz continuous condition. If function $f(\cdot)$ is a family of $K$-Lipschitz continuous functions $\{f_w(\cdot)\}_{w\in W}$ parameterized by $w$, the loss function using Wasserstein distance can be re-written as [13]:

$$L(P_r, P_g) = W(P_r, P_g) = \max_{w\in W} \mathbb{E}_{\mathbf{x}\sim P_r}[f_w(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_G}[f_w(G(\mathbf{z}))]. \tag{5.4}$$

Here the discriminator no longer identifies the probability of a sample being real, but is instead trained to compute the Wasserstein distance by learning the $K$-Lipschitz continuous function $f_w(\cdot)$. The distance becomes smaller as the loss function decreases, implying that the generator is producing samples that are close to the real samples. To maintain the K-Lipschitz continuity of $f_w(\cdot)$, weight clipping, a simple but practical trick, is available: $w$ is restricted to a compact parameter space $[-c, c]$. WGANs do have shortcomings. Weight clipping may result in unstable training, slow convergence, and vanishing gradients. To overcome the issues, a gradient penalty has been introduced [95]. A soft version of a penalty on the gradient is included in the loss function to enforce the $K$-Lipschitz constraint [95]:

$$L(P_r, P_g) = \max_{w\in W} \mathbb{E}_{\mathbf{x}\sim P_r}[f_w(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_g}[f_w(G(\mathbf{z}))] + \lambda \mathbb{E}_{\mathbf{x}\sim P_{\hat{\mathbf{x}}}}[||\nabla_{\mathbf{x}} D(\mathbf{x})||_p - K]^2 \tag{5.5}$$

where $\hat{\mathbf{x}} = \epsilon \mathbf{x}_{P_r} + (1-\epsilon)\mathbf{x}_{P_g}$. This strategy is also called *improved training of WGANs* [95].

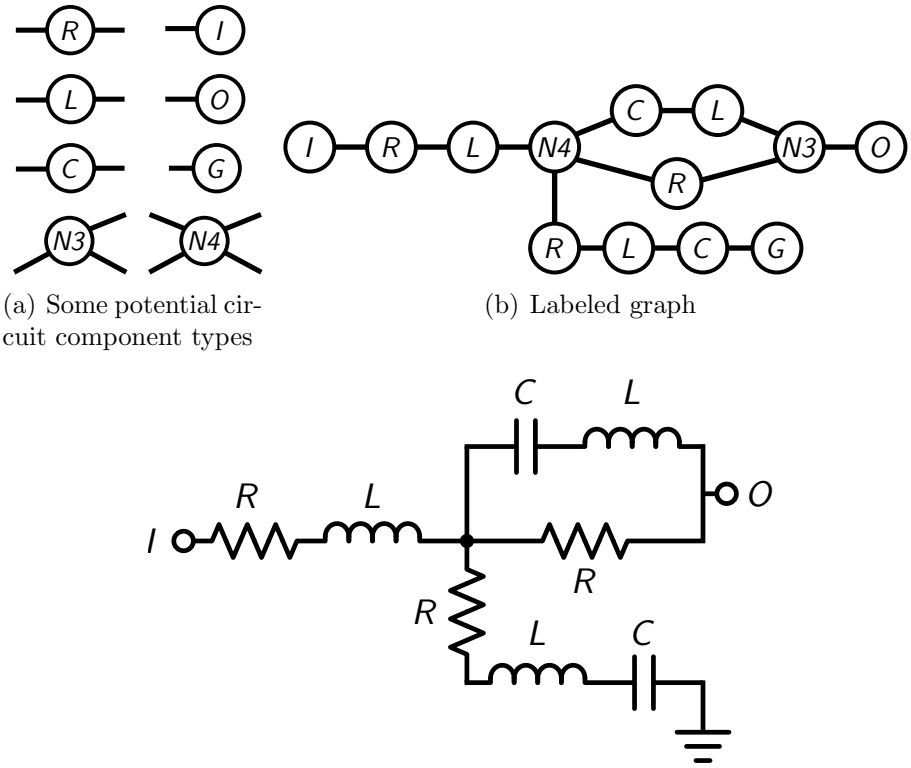
## 5.3 Circuit Synthesis

### 5.3.1 Problem Statement

In the circuit synthesis problem presented here, each electrical circuit component is represented as a node in an undirected graph. A representative set of components that could be used in a *RLC* analog electrical circuit is illustrated in Fig. 5.4(a). This illustration shows an example of an electrical circuit component catalog, including 1-port components (the input *I*, the output *O*, and the ground *G*), two-port components (resistors *R*, capacitors *C*, and inductors *L*), and multiports ($> 2$ ports for connecting to other components). Here the only multiports specified are common-voltage nodes (3- or 4-port common-voltage nodes *N3* and *N4*). *N5* and even higher multi-port *Nx* components may also be used in a circuit synthesis problem.

A circuit topology consists of components used and their connectivity. Circuit topologies can be represented as labeled graphs, where edges indicate connections between component ports. Figure 5.4(b) shows the labeled graph that corresponds to the circuit presented in Fig. 5.4(c). In this chapter, the adjacency matrix corresponding to labeled graphs is used as the circuit topology representation.

Here we consider two canonical circuit synthesis problems: 1) a frequency response matching problem, and 2) a low-pass filter realizability problem. In both of these cases, {*I*, *O*, *G*} are required. Figures 5.5(a) and 5.5(b) illustrate the relationship between these specified elements and portion of the overall system to be designed (i.e., the circuit synthesis domain). In both cases, candidate topologies must satisfy NSCs that govern what types of connections are allowed between components. Herber defined a set of vectors that specifies a desired circuit structure space [109], as well as an efficient enumeration strategy that successfully identified all possible circuit topologies for given component catalogs and NSCs. Although this enumeration-based strategy is feasible for larger catalogs than originally expected, it




(a) Some potential circuit component types

(b) Labeled graph

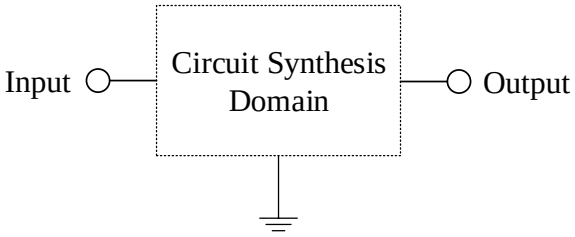
(c) Circuit schematic

Figure 5.4: Examples of the heterogeneous circuit [109]

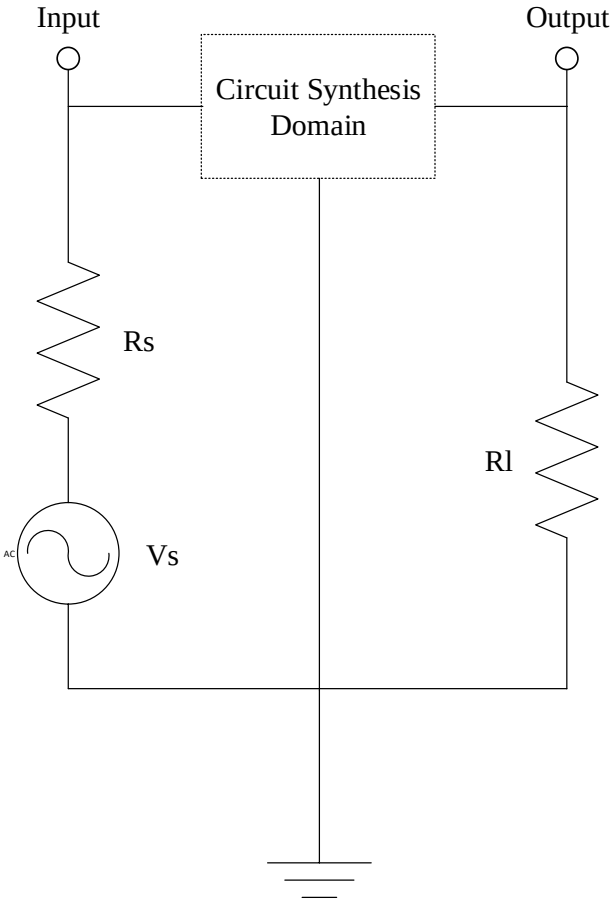
still is limited in practice to component catalogs within a certain size limit. Enumeration is important for obtaining the solution of Case 0 (all feasible and unique topologies can be enumerated and evaluated) and Case 1 synthesis problems. This has been performed in previous recent studies for both the frequency matching and filter design examples. Here we use this previously-developed enumeration and evaluation capability to generate design the data needed to execute the proposed GAN strategy for solving Case 2 problems.

### 5.3.2 Data

In the frequency response problem, a complete circuit is constructed using a transfer function between input *I* and output *O*. More desirable circuits will better satisfy the following target




(a) Template circuit 1: frequency response




(b) Template circuit 2: low pass filter

Figure 5.5: Two common circuit templates [109]

frequency response [94]:

$$|F(j\omega)| = \sqrt{\frac{2\pi}{10\omega}}, \tag{5.6}$$

where the frequency (Hz) range of interest is:

$$0.2 \leq \frac{\omega}{2\pi} \leq 5, \tag{5.7}$$

evaluated over 500 logarithmically-spaced points. All circuits containing $\{R,C\}$ with up to 6 impedance elements were explored previously via the efficient enumeration strategy developed by Herber [109]. A collection of 43,249 unique circuit topologies were identified with unique transfer functions.

The low-pass filter (LPF) problem is often used as the test example for the meta-heuristic methods discussed earlier. The design task is to generate LPFs using the circuit template (Fig. 5.5(b)) under different specifications and variable bounds. A total of 1,804,496 unique graphs were found, out of which 123,156 had up to 7 components and unique transfer functions. A smaller subset also satisfies NSCs [109].

Because the size of the adjacency matrices vary as circuit topologies change, a data-processing step was used to convert the adjacency matrix to a fixed-length vector that is more amenable for machine learning strategies. Two complete data sets with all the possible circuit topologies will be used for numerical experiments.

## 5.4 Numerical Experiment Design

### 5.4.1 Network Architecture Specifications

In this section, a comparative study is presented that quantifies how efficient various strategies are at generating circuit topologies. Different variants of GANs are trained, including vanilla GANs, WGANs, improved GANs, and DGGANs. The generator and discriminator

may be chosen either as a multilayer perceptron (i.e. fully connected layers), or a model using convolutional layers. The generator and discriminator have the same number of layers. Tables 5.1–5.6 summarize various GAN architectures considered for both design problems. Here $BS$ (batch size) is an input to the network. The pre-defined latent vector $n_z$ is drawn from a standard multivariate normal distribution.

In the vanilla GAN (Tables 5.1 and 5.2), two fully connected hidden layers are used. The upper adjacency matrix was flattened to a vector $465 \times 1$ and $630 \times 1$, respectively, as the input for the two design problems. The output has the same size as the input. To stabilize convergence of the vanilla GANs, the ratio of network updates between the discriminator and generator was set to 5:1. Leaky Rectified Linear Unit (LReLU) [167] and tanh are used to produce circuits in the last layer of the generator (additional transformation to the input is needed as tanh requires the input to be in [-1,1]).

There are four fully-connected layers in the WGANs and improved WGANs (Tables 5.3 and 5.4), where the loss function is replaced by the Wasserstein distance. The sigmoid activation in the last layer of the discriminator is eliminated; the logarithm in the generator and discriminator is no longer needed; the weights in both networks are truncated in [-0.01, 0.01]. Finally, RMSProp is used to train the networks [114]. The regularization term $\lambda$ is set to 100 for the improved WGANs. The hidden layer sizes are 128, 256, and 512, respectively.

For the DCGANs (Tables 5.5 and 5.6), the filters are assigned symmetrically. The filter with a size of $4 \times 4$ and stride $2 \times 2$ results in a (or an approximately) scalar of 2 in each layer. In the first three layers of the discriminator, convolution, batch-normalization, and LReLU activation are connected between the layers, and the last convolutional-sigmoid layer outputs the probability. The generator has the de-convolutional, batch normalization, and LReLU activation function appended to each of the first three layers sequentially, followed by the de-convolutional-tanh in the last layer. The Adaptive Moment Estimation (Adam) optimizer is used for the training [135].

Table 5.1: Frequency response: vanilla GAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS $\times$ $n_z$ |
| Generator Layer 1 | BS $\times$ 256 |
| Circuit Topology | BS $\times$ 465 |
| Discriminator Layer 1 | BS $\times$ 256 |
| Discriminator Layer 2 | BS $\times$ 1 |

Table 5.2: Low-pass filter: vanilla GAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS $\times$ $n_z$ |
| Generator Layer 1 | BS $\times$ 256 |
| Circuit Topology | BS $\times$ 630 |
| Discriminator Layer 1 | BS $\times$ 256 |
| Discriminator Layer 2 | BS $\times$ 1 |

Table 5.3: Frequency response: WGAN and improved WGAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS.$\times$ $n_z$ |
| Generator Layer 1 | BS $\times$ 128 |
| Generator Layer 2 | BS $\times$ 256 |
| Generator Layer 3 | BS $\times$ 512 |
| Circuit Topology | BS $\times$ 465 |
| Discriminator Layer 1 | BS $\times$ 512 |
| Discriminator Layer 2 | BS $\times$ 256 |
| Discriminator Layer 3 | BS $\times$ 128 |
| Discriminator Layer 4 | BS $\times$ 1 |

### 5.4.2 Evaluation of the GANs

Since complete data sets (denoted as $\Omega$) are available for both circuit synthesis problems, numerical experiments will be designed to test how likely a GAN is to produce a feasible circuit topology. The complete set $\Omega$ is not available for Case 2 synthesis problems; only

Table 5.4: Low-pass filter: WGAN and improved WGAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS.$\times$ $n_z$ |
| Generator Layer 1 | BS $\times$ 128 |
| Generator Layer 2 | BS $\times$ 256 |
| Generator Layer 3 | BS $\times$ 512 |
| Circuit Topology | BS $\times$ 630 |
| Discriminator Layer 1 | BS $\times$ 512 |
| Discriminator Layer 2 | BS $\times$ 256 |
| Discriminator Layer 3 | BS $\times$ 128 |
| Discriminator Layer 4 | BS $\times$ 1 |

Table 5.5: Frequency response: DCGAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS $\times$ 1 $\times$ 1 $\times$ $n_z$ |
| Generator De-Conv. Layer 1 | BS $\times$ 4 $\times$ 4 $\times$ 128 |
| Generator De-Conv. Layer 2 | BS $\times$ 8 $\times$ 8 $\times$ 64 |
| Generator De-Conv. Layer 3 | BS $\times$ 16 $\times$ 16 $\times$ 32 |
| Circuit Topology | BS $\times$ 32 $\times$ 32 $\times$ 1 |
| Discriminator Conv. Layer 4 | BS $\times$ 16 $\times$ 16 $\times$ 32 |
| Discriminator Conv. Layer 4 | BS $\times$ 8 $\times$ 8 $\times$ 64 |
| Discriminator Conv. Layer 4 | BS $\times$ 4 $\times$ 4 $\times$ 128 |
| Discriminator Conv. Layer 4 | BS $\times$ 1 $\times$ 1 $\times$ 1 |

Table 5.6: Low-pass filter: DCGAN

| Layer | Dimension |
|---|---|
| Random Tensor $\mathbf{z}$ | BS $\times$ 1 $\times$ 1 $\times$ $n_z$ |
| Generator De-Conv. Layer 1 | BS $\times$ 5 $\times$ 5 $\times$ 128 |
| Generator De-Conv. Layer 2 | BS $\times$ 9 $\times$ 9 $\times$ 64 |
| Generator De-Conv. Layer 3 | BS $\times$ 18 $\times$ 18 $\times$ 32 |
| Circuit Topology | BS $\times$ 36 $\times$ 36 $\times$ 1 |
| Discriminator Conv. Layer 1 | BS $\times$ 18 $\times$ 18 $\times$ 32 |
| Discriminator Conv. Layer 2 | BS $\times$ 9 $\times$ 9 $\times$ 64 |
| Discriminator Conv. Layer 3 | BS $\times$ 5 $\times$ 5 $\times$ 128 |
| Discriminator Conv. Layer 4 | BS $\times$ 1 $\times$ 1 $\times$ 1 |

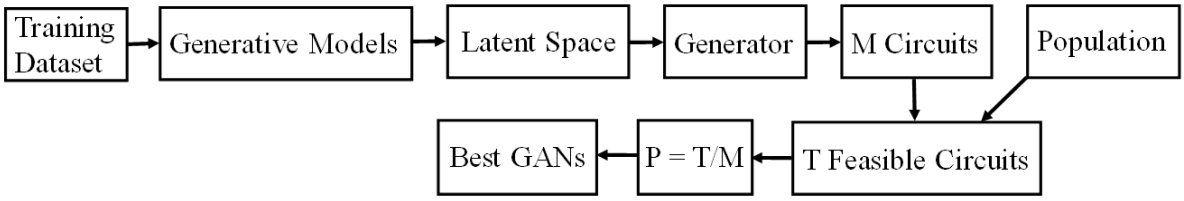



Figure 5.6: Procedure used to test GAN effectiveness at generating circuit topologies

a representative sample $X$ is available. In the experiments, the subset $X \subset \Omega$ is randomly selected as the training set. A new collection of samples, denoted as $M$, will be obtained using the generator. Suppose $T$ out of $M$ samples are in $\Omega$. A metric $P$, defined as:

$$P = |T|/|M| \tag{5.8}$$

is the ratio of the number of feasible generated topologies to the total number of generated topologies. Here we make the assumption that the larger the ratio of feasible topologies to generated topologies is, the better the performance of the GAN. A perfect GAN would generate the circuit topologies satisfying $T = M \subset \Omega$ (i.e., $P = 1$). We acknowledge that $P$ is a proxy objective function for GAN performance, but postulate that it is aligned with the objective of improve Case 2 synthesis problem solution performance. Figure 5.6 illustrates the procedure used to test GAN effectiveness at topology generation.

Predictive modeling can also be incorporated into the GAN framework to enable solution of the synthesis problem. While the GANs can to generate feasible topologies, the circuit synthesis problem also requires topology evaluation and identification of high-performance circuits. One strategy for solving the synthesis problem that combines active learning (using a predictive model) with GAN-based topology generation is depicted in Fig. 5.7. After circuit generation, NSCs are used to filter out any remaining infeasible topologies. The predictive model (e.g., random forest from the previous chapter) is used to approximate the performance of feasible circuits. This strategy may be particularly appropriate in cases where circuit evaluation is computationally expensive.

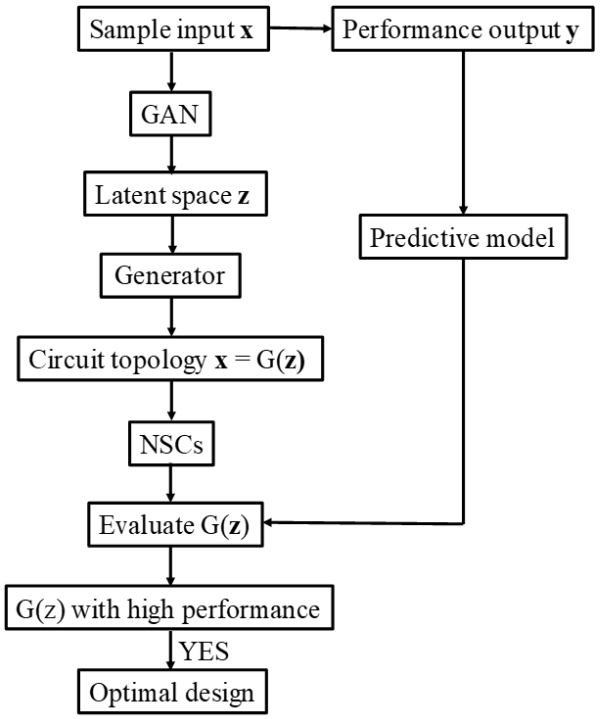


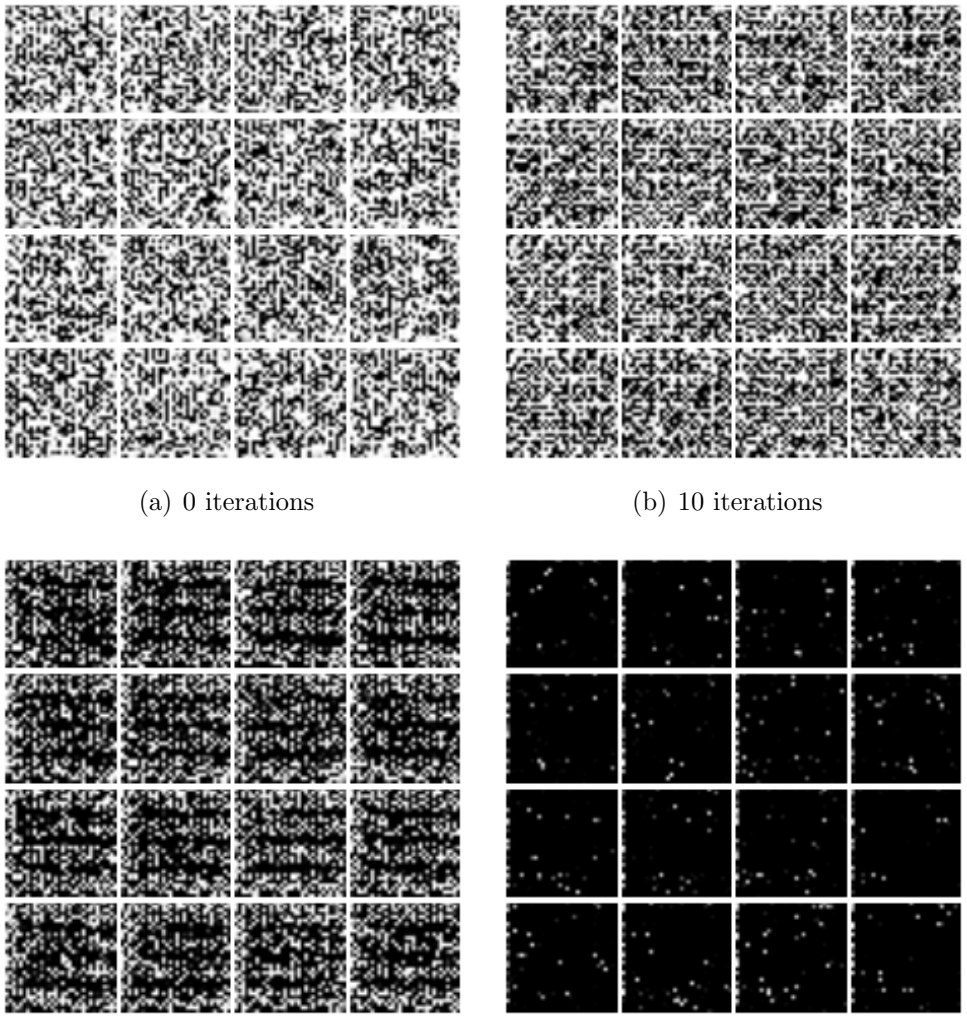

Figure 5.7: The GAN circuit synthesis framework using predictive modeling

## 5.5 Numerical Results

### 5.5.1 Comparison of GAN Models

An initial 20,000 samples drawn from $\Omega$ were used as the training sets for both design problems. The latent vector size was 64. Here we plot the adjacency matrices (symmetric) as a means to visualize circuit topologies. In the adjacency matrix, white entries indicate connected corresponding components (value 1), whereas black entries indicate no connection (value 0). Figure 5.8 shows a set of 16 randomly generated circuit topologies using the DCGAN for the frequency response problem. At the beginning (Fig. B.1(a)), because generator weights are assigned randomly, the generated topologies are essentially noise. However, as the training progresses, the generator has learned more about the training set, and starts to produce reasonable circuit topologies (Fig. B.1(d)). Furthermore, the generator has the

ability to produce sharper and symmetric circuit topologies by 1,500 iterations (Fig. B.2(d)). Similar visualization for the low-pass filter problem can be found in the Appendix (Figs. B.1 and B.2.)



(a) 0 iterations

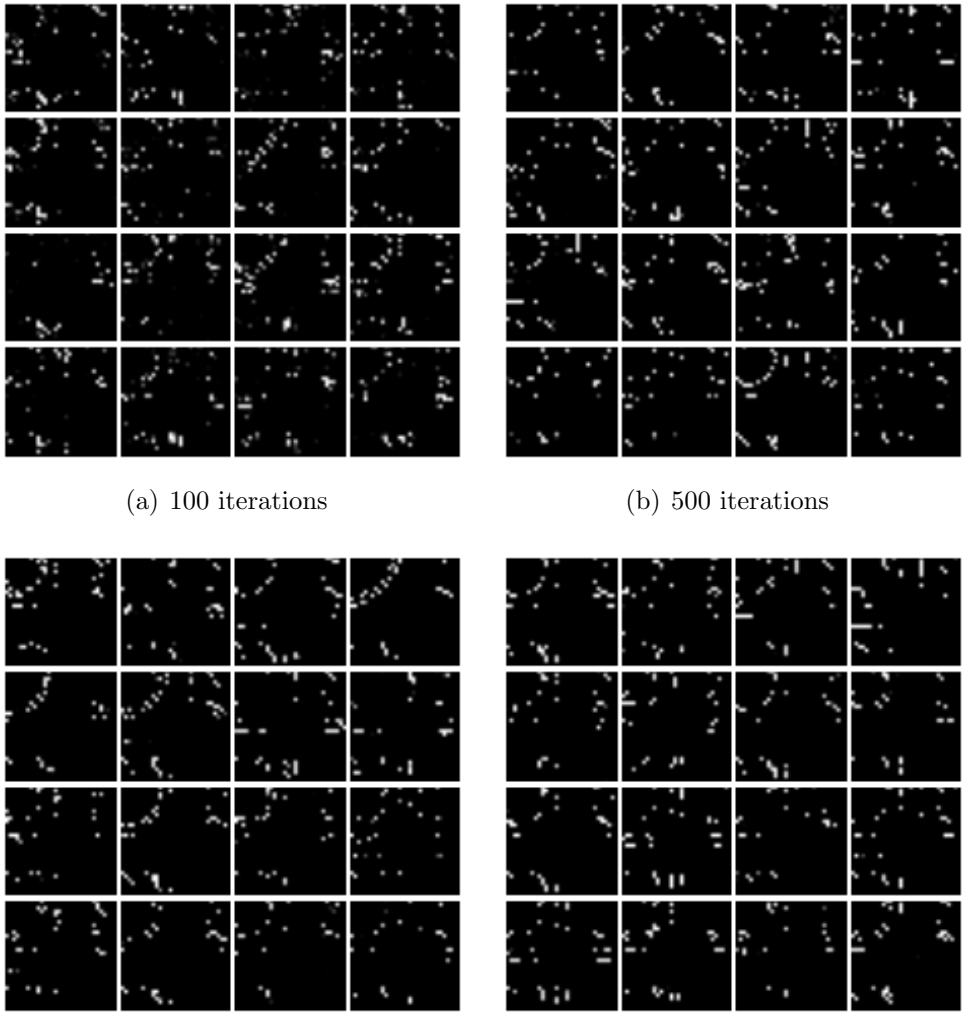
(b) 10 iterations

(c) 20 iterations

(d) 30 iterations

Figure 5.8: Frequency response: generated circuit topologies (0–30 iterations)

Figures 5.10 and 5.11 illustrate how the GAN iteratively works toward feasible generated circuit topologies. A randomly selected $M = 10,000$ new data samples were obtained using the generator. Each generated circuit topology was verified to be in the complete design data set (sizes 43,249 and 123,156 for the two problems, respectively). We recorded the number of feasible topologies $T$ for different GAN architectures and then computed the metric $P$.



(a) 100 iterations (b) 500 iterations

(c) 1000 iterations (d) 1500 iterations

Figure 5.9: Frequency response: generated circuit topologies (100–1500 iterations)

Figure 5.10 reports the result for the frequency response problem. It is observed that the improved WGAN performs the best with respect to $P$, with 1,990 out of 10,000 generated samples being feasible. The WGANs follows next, but the likelihood of the feasible topologies is decreased significantly, with only 3.44%. The vanilla GAN and DCGAN perform least well, with 1.51% and 0.36%, respectively. We specifically investigated how the training time affects the metric $P$. Figure 5.11 indicates that the likelihood of obtaining feasible samples using the improved WGAN is highly related to the number of iterations. In particular, 3,049 out of 10,000 are feasible (with iteration number 25,000). However, if we allow more iterations

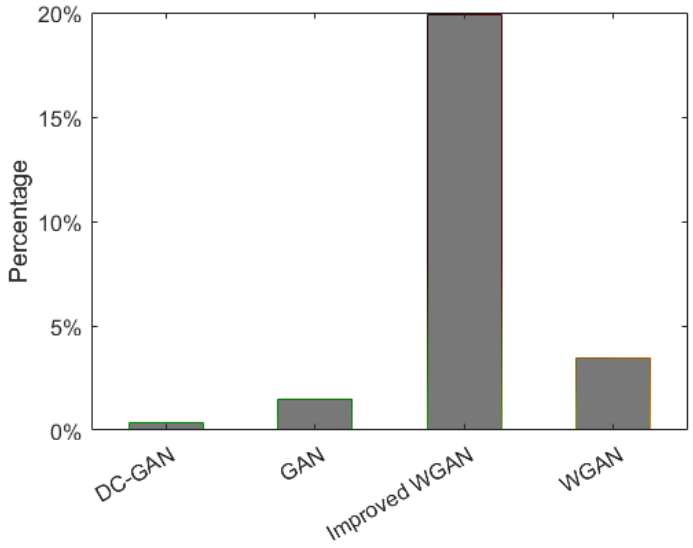



Figure 5.10: Frequency response: GAN generation performance

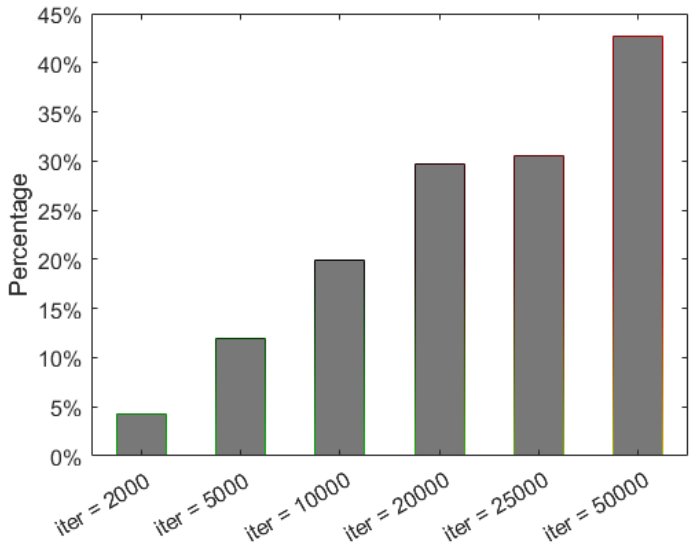



Figure 5.11: Frequency response: improved WGAN generation performance

(i.e. 50,000), the feasibility percentage is increased to 42.66%, implying a longer training time greatly improves generation efficiency. Figure 5.12 shows that both the generator and discriminator losses have become stable after a sufficient number of iterations. Similar results were obtained for the low-pass filter problem, and are presented in the Appendix (Figs. B.3–B.4).

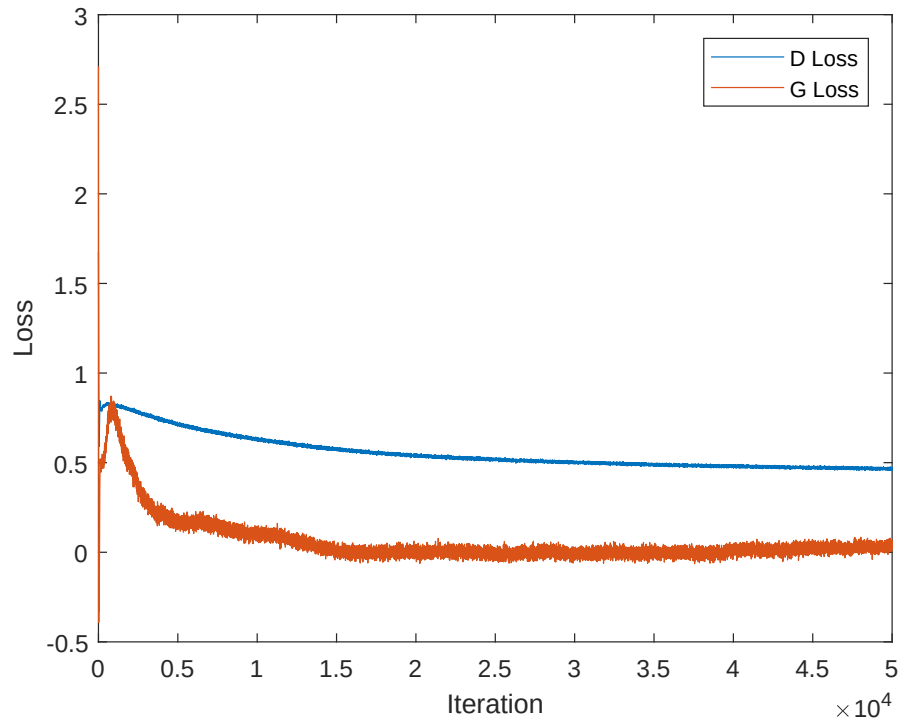



Figure 5.12: Frequency response: convergence of the improved WGAN

### 5.5.2 Parametric Studies of the Improved WGAN

Since the improved WGAN demonstrated efficient generation, we conducted a parametric study regarding two important parameters: the number of latent variables and $\lambda$. The purpose of the parametric study is to provide insights about the improved WGAN's capabilities for circuit synthesis. Here the frequency response problem is considered. The number of latent variables was varied across the following levels: 16, 32, 64, and 128 (with a fixed $\lambda = 10$). The value of $\lambda$ was varied across the levels: 0.1, 1, 10, and 100 (with a fixed number of latent variables: 64). For illustrative purposes, all the experiments were carried out using 5,000 iterations. Table 5.7 summarizes the results. We observe that using 64 latent variables and $\lambda = 10$ significantly outperforms the other parameter values (with a feasibility percentage of 19.90%). Too few latent variables (e.g. 16 and 32) result in inadequate learning of the improved WGAN, but too many latent variables (128) did not perform well, perhaps due to overfitting and more difficult training (i.e., more weights and bias must be learned). As

Table 5.7: Frequency response: Parametric study on improved WGAN parameters

| | Value | Percentage |
|---|---|---|
| Latent variable | 16 | 10.95% |
| | 32 | 12.13% |
| | 64 | 19.90% |
| | 128 | 10.7% |
| $\lambda$ | 0.1 | 9.16% |
| | 1 | 9.69% |
| | 10 | 19.90% |
| | 100 | 10.48% |

a penalty term, the values of $\lambda = 10$ is found to be an appropriate value for enhancing the capabilities of the improved WGAN for this problem. The parametric study for the low-pass filter problem can be found in Table B1.

### 5.5.3 The GAN Framework with Active Learning

Here we use active learning strategy in combination with GAN-based topology generation.The frequency response problem is tested here, where two data sets were used for the active learning tasks (see Section 4.4.2). Following the procedures in Fig. 5.7, the numerical experiment was implemented as follows: 1) solved Case 0 problem to obtain a statistical circuit topology sample $X$; 2) trained an improved WGAN using sample $X$; 3) applied the active learning strategy to sample $X$ to construct a predictive model; 4) used the generator to output a collection of 10,000 circuit topologies, and used NSCs to filter out remaining infeasible circuit topologies; 5) the final predictive model given by the active learning strategy predicts feasible circuit topology performance. In this study, 2,973 of the 10,000 circuit topologies were feasible.

Similar to the previous chapter, RMSE quantifies the average performance of the predictions, while the normalized Kendall tau distance $\tilde{K}$ measures the ranking order between the predicted values and true observations. Table 5.8 summarizes the RMSE and $\tilde{K}$ for

Table 5.8: Frequency response: performance of the GAN based method with active learning

| | RMSE | $\tilde{K}$ |
|---|---|---|
| Set 1 | 2.4942 | 0.2631 |
| Set 2 | 2.9266 | 0.2557 |

both data sets (with 2,973 feasible circuit topologies produced by the improved WGAN). It is interesting to observe that the RMSEs are significantly different. Different bounds on the component values may account for this sensitivity (see Section 4.4.2). The normalized Kendall tau distance $\tilde{K}$ shows that the majority of predictions is kept, meaning that design engineers using this technique could identify feasible circuit topologies with high-performance by simply sorting the prediction list. Figures 5.13 and 5.14 report the generated circuits with high predicted performance for both sets. This combined framework addresses Case 2 problems with high evaluation cost, facilitating identification of high-performance circuit topologies.

## 5.6 Discussion

The numerical results show that the improved WGAN can efficiently produce feasible circuit topologies in circuit synthesis. Here the Wasserstein distance in the loss function overcame the instability and mode collapse issues. One possible reason for the poor performance of the DCGAN could be insufficient training. Because all experiments were carried out using TensorFlow CPU implementations, DCGAN required a significant amount of time to complete the training. GPU computing could speed up the training process, enabling the generation of more robust results. However, it is still observed that even the improved WANs with fully connected layers (simpler network architecture) outperforms the DCGAN with convolutional layers (complicated network architecture). It might be helpful to implement a deep convolutional WGAN (DC-WGAN) architecture to gain more insights about the

generative design.

While this work demonstrates initial promising results for the use of GANs in synthesis, a number of improvements and further investigations should be performed. A number of factors affect GAN performance. For example, 4 hidden layers in the vanilla GAN were initially implemented. However, it turned out that the vanilla GAN experienced instability and mode collapse with this architecture. Replacing two hidden layers improved performance (see Tables 5.1 and 5.2).

Further investigations could be conducted to better understand the randomly generated sample produced by the GAN-based method. In the numerical experiment, because each generated circuit topologies used a fixed-size adjacency matrix, this could result in graph isomorphisms. In other words, the portion of the circuit identified as infeasible may be isomorphic to another feasible circuit in the complete data set. Recall that the improved WGAN can achieve as high as 42.66% feasibility percentage for the frequency response problem; the isomorphism check would potentially increase metric $P$ further. On the other hand, it is also worth exploring properties of the randomly generated circuit topologies, including uniqueness, component combination, frequency, and other aspects.

The GAN-based method could be extended to other research tasks in circuit synthesis. The numerical results imply that the improved WGAN could be a good option for generating high quality circuit topologies. Once the circuit topology $\mathbf{x}$ is available, the circuit performance must be assessed via sizing optimization. Figure 5.15 illustrates a possible workflow where the GAN framework can incorporated with sizing optimization. Additional detail regarding sizing optimization can be found in Ref. [109, 111].

The GAN-based methodology could be applied to other heterogeneous system topology design problems with similar properties. This assertion is based on recent work in enumeration and evaluation of various automotive suspension architectures, where different combinations of passive and active components are combined to improve comfort and handling metrics [109]. Figure 5.16 illustrates a simplified passive suspension model [10]. It represents

a single wheel suspension, and hence is referred to as a quarter-car model. The unsprung mass $m_{us}$ (or $U$) represents the inertia of the wheel and other components that move with the wheel. The stiffness and damping of the tire that connects $U$ to the road surface $z_0$ are modeled by the spring and damping coefficients, $k_t$ and $c_t$, respectively. As the vehicle moves forward with velocity $v$, the road surface elevation changes, inducing displacement and vibration of the unsprung mass. The sprung mass $m_s$ (or $S$) represents the vehicle body, i.e., all vehicle mass that is to be isolated from disturbances from road imperfections. In a conventional passive suspension system, the sprung and unsprung masses are connected by a spring and damper, modeled here using linear coefficients $k_s$ and $c_s$, respectively. Alternative architectures may be explored, where active force elements are included in between the sprung and unsprung masses, or additional passive components, such as vibration absorbers. Figure 5.16 illustrates a standard passive suspension model; active components and additional passive components are not pictured. This system architecture can be represented as a colored (labeled) graph. According to Ref. [109], one possible component catalog, based on terminology from bond-graph modeling, for this problem is defined by:

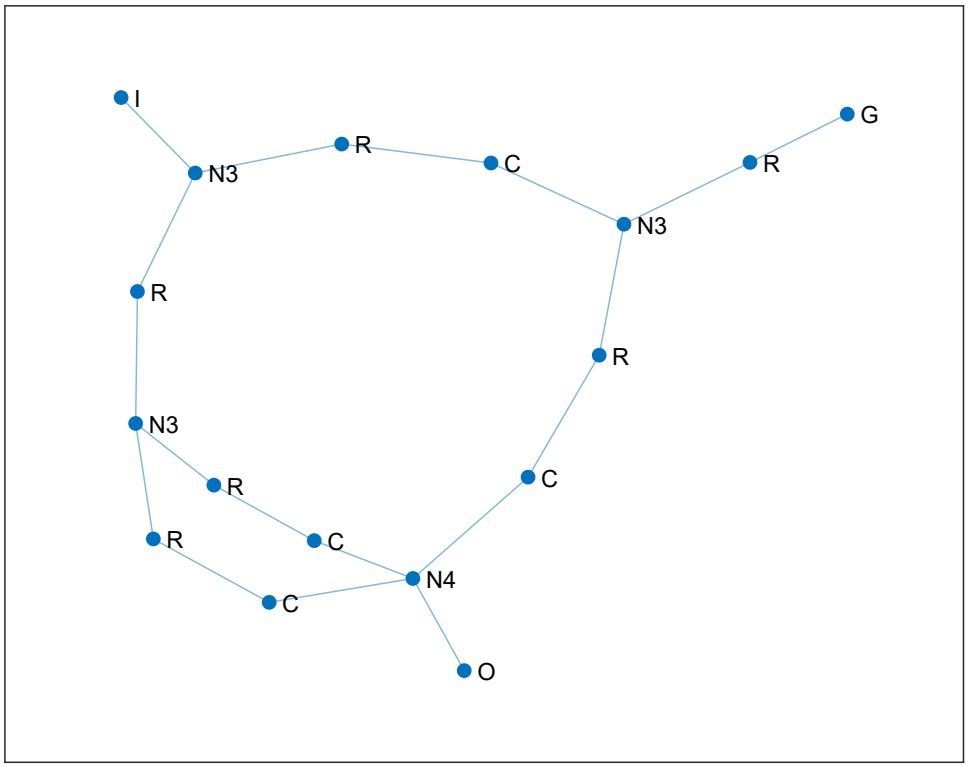
- $\{S, U, M\}$: Inertial ($I$) energy storage nodes (sprung, unsprung, and additional masses)
- $K$: Capacitance ($C$) energy storage node (springs)
- $B$: A subsystem that combines a spring and damper in parallel
- $F$: An effort source ($Se$), i.e., an active force actuator
- $P_x$: a node with $x$ ports

Similar to the circuit synthesis in this chapter, the each suspension component is regarded as a node, and the system topology can be represented in terms of an adjacency matrix. These properties would support GAN training, and subsequent generation of feasible suspension topologies.

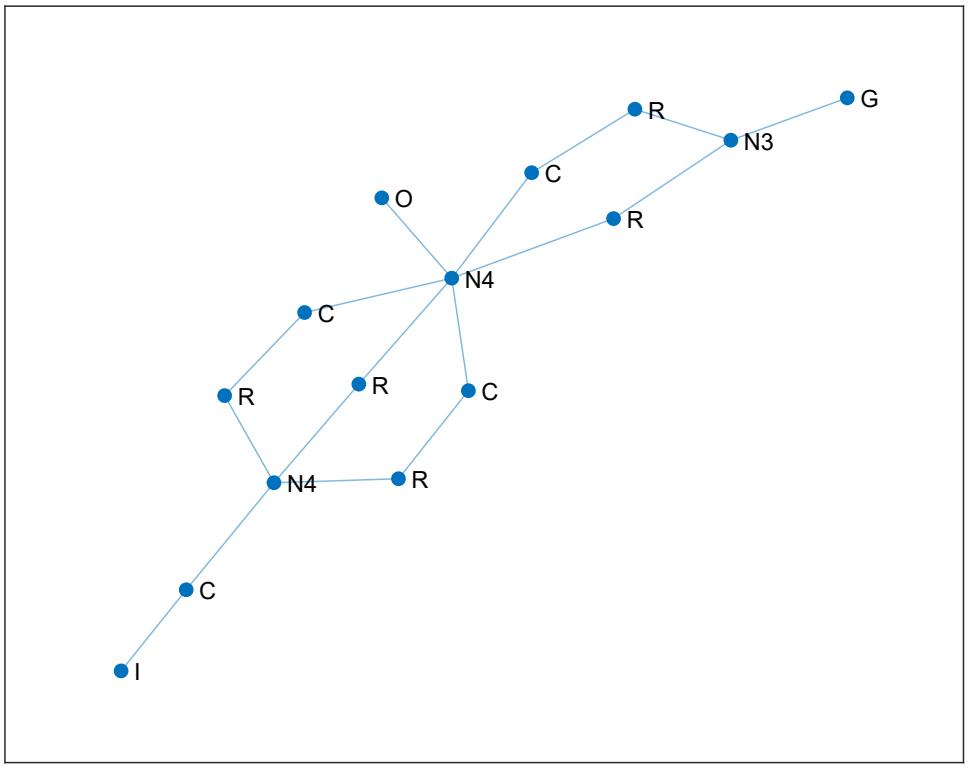
## 5.7 Conclusion

In this chapter, adversarial learning was presented as a new solution strategy for Case 2 heterogeneous circuit synthesis design. The generative adversarial network (GAN) is a generative model that learns to simulate data distribution. The GAN is trained iteratively such that the generator produces circuit topologies that are accepted by the discriminator. A lower-dimensional latent space can represent the circuit topology indirectly. A random sample drawn from a normal distribution is mapped to a circuit topology via the generator network. We studied the efficiency of generating feasible samples using different GAN models, including vanilla GAN, DCGAN, WGAN and improved WGAN. The frequency response and low-pass filter problems, containing complete topological design data, were used as the case studies. The numerical results indicate that the generator of the improved WGAN has a higher likelihood to generate feasible circuit topologies. A parametric study was carried out to identify the best parameter values for the improved WGAN. The GAN framework combined with active learning makes predictions for generated circuit performance, and the resulting normalized Kendall tau distance indicates that the performance ranking is similar. This is a promising approach for searching for high-performance circuit topologies, as well as other Case 2 synthesis problems. This work leverages design data and artificial intelligence, not only providing an indirect representation that generates the feasible circuit topologies in an automated manner, but also may be transferable, as the pre-trained discriminator may facilitate the development of the circuit property prediction model.

Future work for improving the GAN capabilities includes, but is not limited to, the use of both the DCGAN and improved WGAN, other generative models such as VAEs [136], PixelCNNs [242], and PixelRNNs [193]. One option to extend the functionality of the GAN framework would be to use it to support sizing optimization. An example of a distinct synthesis problem (active vehicle suspension design) was presented and shown to have the properties needed for solution via the methods presented here.

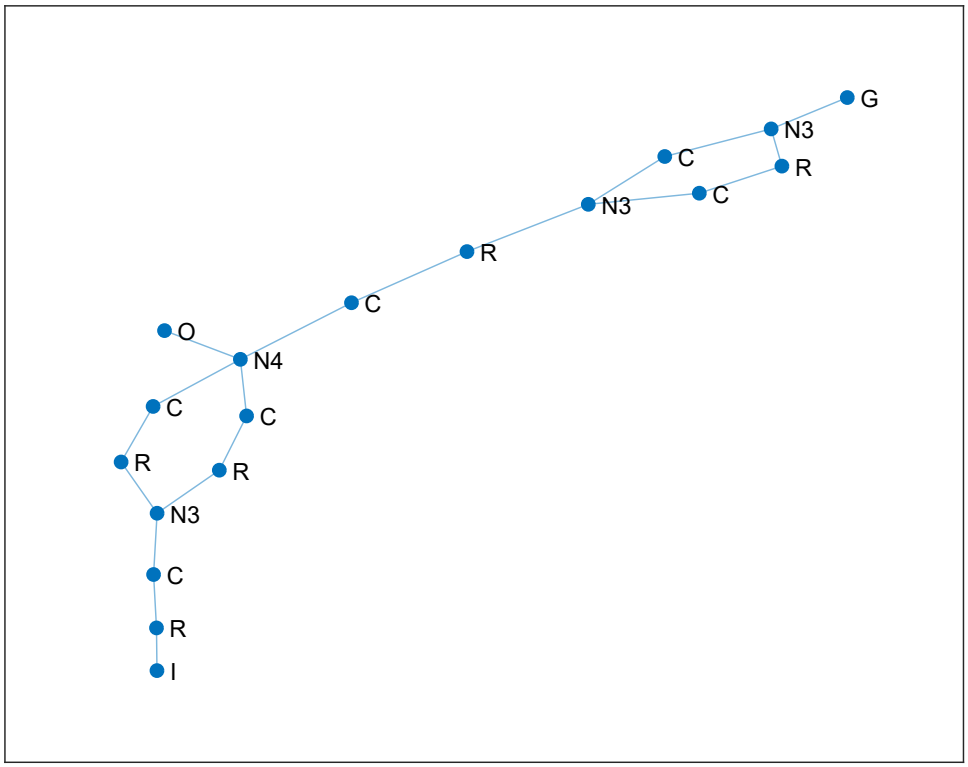



(a) Predicted circuit #1

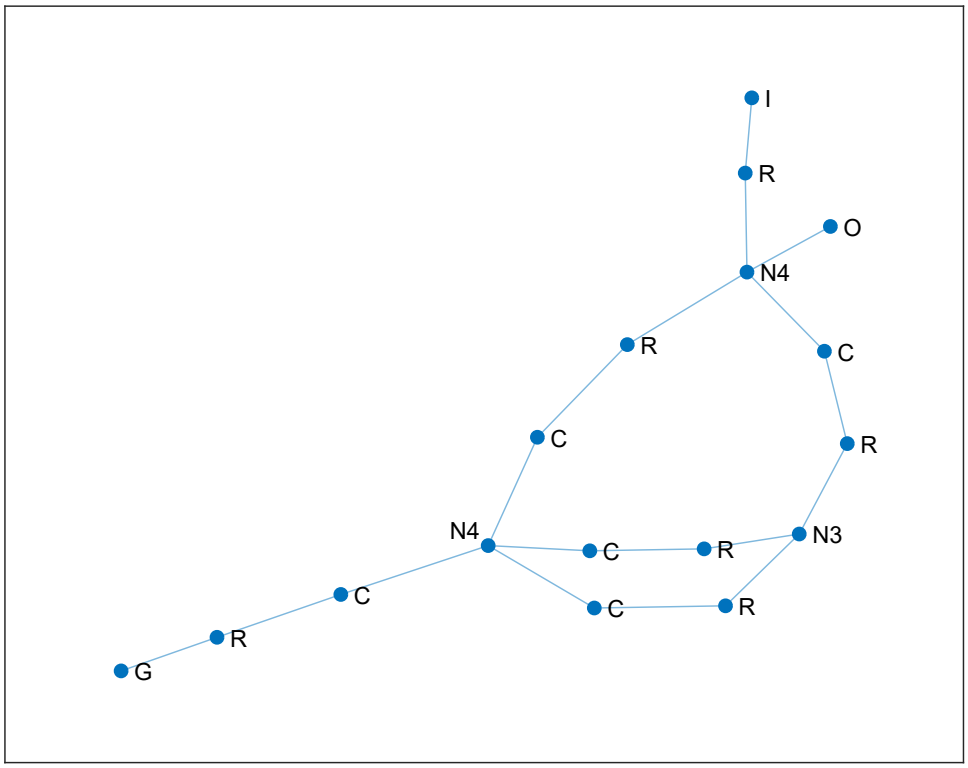



(b) Predicted circuit #2

Figure 5.13: Set 1: two high-performance circuit topologies given by the improved WGAN




(a) Predicted circuit #1




(b) Predicted circuit #2

Figure 5.14: Set 2: two high-performance circuit topologies given by the improved WGAN

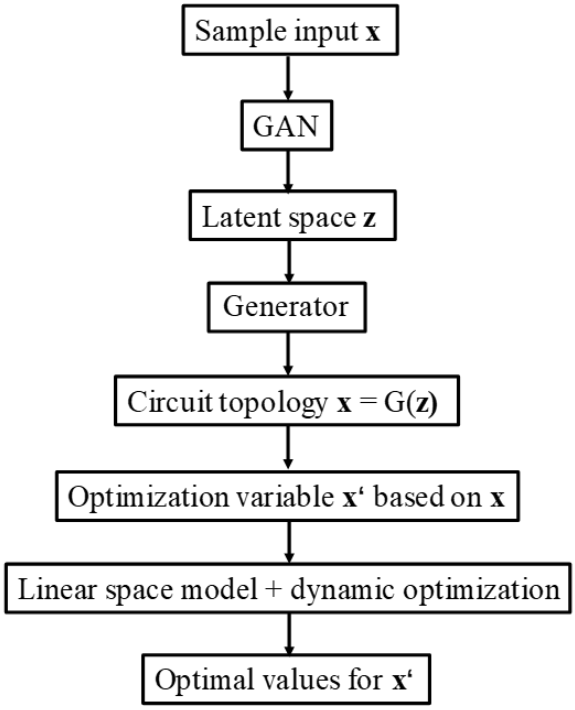



Figure 5.15: Proposed framework using the GAN

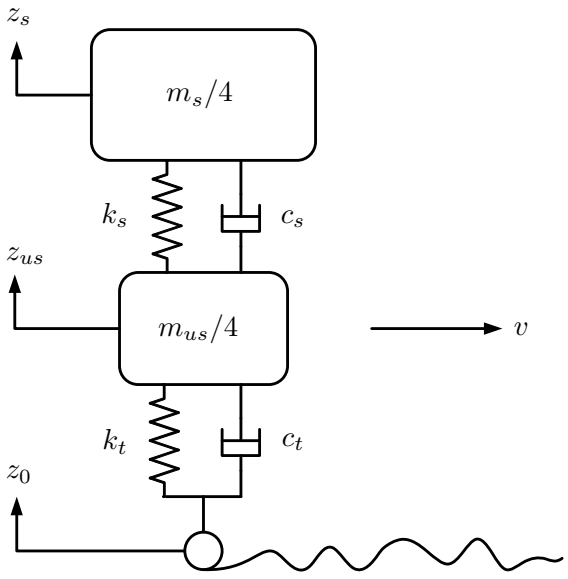



Figure 5.16: A linear quarter-car suspension model (standard passive system shown)

# Chapter 6

# CONCLUSION

In this chapter, a summary of the dissertation is presented. The core contributions and future research opportunities are be outlined.

In this dissertation, several data-driven approaches for system topology design have been studied. The motivation for these machine learning based strategies is to provide new solution methods that complement existing optimization, synthesis, enumeration, and other methods, and make possible the solution and understanding of new classes of system topology design problems.

Chapter 2 presented preliminary work that motivated deep investigation into new machine learning strategies for design. Literature network analysis has been used to analyze a number of research domains, but has not been applied before now in an extensive way to the engineering design research community. Here we studied a total of 1,668 articles published at the American Society of Mechanical Engineers (ASME) Design Automation Conference (DAC) during the years 2002–2015. A number of meaningful insights were identified via analysis of co-authorship and citation networks. Through analysis of the co-authorship network, we learned collaboration patterns existence of small-world-network properties. Other insights were obtained through two topic modeling approaches. The first topic modeling approach, the frequency-based method, was used to capture frequencies, evolution, correlation,

and association of research topics within the DAC community. The second topic modeling approach, PM, determined highly influential articles and performed cluster analysis using a propagation and mergence mechanism across the citation network. Several research recommendations and research opportunities were presented. One opportunity identified is more comprehensive use of machine learning and data science in design research, especially in ways that connect with existing design capabilities.

In Chapter 3, a data-driven design approach was developed for homogeneous system topology design. A deep learning strategy was developed and applied to heat conduction system design. An indirect low-dimension design representation was proposed based on a VAE and style transfer. As a generative model, the VAE makes a strong assumption regarding the distribution of latent variables, and uses a variational approach for latent representation learning. The VAE encodes 2D topologies into a low-dimensional latent space variable, and decodes samples from this space back to the original space. The first step of the proposed approach was to train a VAE, together with a deep style transfer network, using the design data given by solutions to a related topology optimization problem. The objective of the style transfer augmentation was to prevent occurrence of isolated material elements in topology generation. The second step involved multi-objective optimization for heat conduction with respect to the low-dimensional latent variables. Several different optimization solution strategies were investigated. The Pareto frontier (non-dominated solutions) was obtained using the reduced-dimension representation, and results showed improvement in both computational efficiency and solution quality. The core contribution of this chapter was a method that utilizes the capabilities of established topology optimization methods to generate design data based on a problem that is related to (but distinct from) the desired design problem, and then utilizes a machine learning strategy with this design data to solve the desired design problem directly. This is a promising and fundamentally new design approach.

While Chapters 4 and 5 concentrated on the heterogeneous circuit synthesis design, the re-

search objectives were distinct. Heterogeneous system topology design (synthesis) is different from homogeneous system topology design in that design candidate are be represented using a colored (labeled) graph. Recent developments in enumeration for design synthesis have improved design capabilities, but are limited to moderately-sized synthesis problems [109]. These new enumeration capabilities, however, are used as a basis for learning-based synthesis methods presented here. Two types of synthesis problems that go beyond the capabilities of enumeration were identified (Case 1 and Case 2) in Section 4.1, and these problem types are treated in Chapters 4 and 5, respectively. Electronic circuit synthesis case studies were used in both chapters.

Chapter 4 addressed Case 1 synthesis problems, where enumeration of all topologies is possible, but evaluation is impractical. An active learning strategy was introduced to reduce the number of topology evaluations required to obtain approximate solutions. Instead of evaluating the circuit performance directly for all candidates, we constructed a predictive model that approximates the mapping from topological inputs to real-valued outputs (performance metrics). This was used as a type of surrogate model to identify promising topologies, and to be strategic about what topologies to invest resources into evaluating. The active learning strategy was applied in an iterative manner, where new evaluations were added to the training set to improve prediction accuracy. *Random Forest* was chosen as the predictive model. Criteria for querying new topologies were examined, and the results indicate that uncertainty sampling is the most effective of the strategies tested. The objective of this work was to reduce the number of (computationally-expensive) evaluations required to obtain a good approximate solution. Active learning is distinct from the conventional machine learning in that normally a complete training set is chosen all at once (passive learning).

Chapter 5 addressed Case 2 synthesis problems. We aimed to create a general data-driven approach using generative adversarial networks (GANs) for the circuit synthesis that did not rely on intuition or domain knowledge. A reduced-dimension indirect design representation based on enumerated design data was constructed in an automated way. This

design representation could then support more rapid design space exploration using methods such as evolutionary algorithms in a reduced (and targeted) design space. GANs contain a discriminator and generator: the generator synthesizes data samples with the objective of "fooling" the discriminator, i.e., where the discriminator is tricked into accepting a generated sample as real instead of identifying it as fake. The role of the discriminator is to learn to more accurately to distinguish the real data from the fake. The generator and discriminator keep updating and competing with each other until an equilibrium is reached. To produce a circuit topology, the generator draws a random sample from the latent space, and maps it original circuit space. However, many topologies in this original circuit topological design space violate NSCs. It is inefficient to explore these infeasible designs. GANs are trained here to satisfy NSCs implicitly, resulting in a more targeted design space search. Two case studies are investigated here: frequency response and low-pass filter design problems for electric circuits. Several different GAN models (vanilla GANs, DCGANs, WGANs and improved WGANs) were compared using these case studies with respect to generation of feasible designs. The improved WGAN was demonstrated to be the most effective model. This finding may be important, because it allows for an automated and fast generation of feasible designs only by sampling from the latent space. The active learning strategy was also incorporated into the improved WGAN topology generation strategy to help select high performance circuit topologies. The GAN-based methodology is fundamentally distinct from existing circuit synthesis methods, including those based on direct EAs or comprehensive enumeration. This methodology may be extended to other design tasks, such as sizing optimization and performance prediction, other application domains (such as active automotive suspension architecture design), and has potential for being combined with other existing synthesis methods to enhance their capabilities.

In this dissertation, we also derived new insights from the numerical studies, especially regarding design method properties. For instance, through the comparative study, the MOGA and hybrid methods were able to identify non-dominated designs in the homogeneous sys-

tem topology design example. The improved WGAN turned out to be the most efficient for generating feasible circuit topologies. The parametric studies may provide useful insights to help design engineers who might consider using them for the design problems (e.g., the VAE-based methodology, active learning strategy, and GAN-based strategy). Iterative questioning and testing was performed to obtain both improved methods and deeper insights. In the early stages of the homogeneous system topology design study, PCA was tested first for dimension reduction, but it was determined to have too much information loss to be useful as an indirect design representation. As a nonlinear mapping, the VAE was then tested but suffered from scatters in the generated topologies. With the underlying reasoning behind the phenomena, it was found the style transfer networks would solve the issue. A quantitative hypothesis was supported empirically using a design space coverage metric that the reduced space representation approximately covers the true Pareto curve (see Figs. 3.19–3.21). In Chapter 2, knowledge was extracted using associate rule learning to discover the relationships between the core topics in the DAC research. In Chapter 3, we posit that additional systematic design optimization experiments should be performed where more than one heat sink is used, and possibly other problem conditions varied to obtain a more general data set for the machine learning strategies. Future work should address systematic approaches for designing design optimization experiments for generating data to be used in machine learning.

Each chapter identifies open questions and other opportunities for future work, such as enhancements to the DAC literature network analysis, extension to 3D problems for VAE-based topology optimization, design of design optimization experiments, more robust active learning methods for Case 1 synthesis problems, extension of the GAN method to other Case 2 synthesis problems, and deeper study of the complete solution process for Case 2 problems. In addition to items already identified, broader open questions related to the use of machine learning with data derived from design optimization include: 1) Given the inherent flexibility afforded to data generation via design optimization, what systematic

strategies could be used to generate rich, effective data sets with limited resources? 2) If the objectives in developing design optimization methods are no longer focused only on solving harder problems more efficiently, but now also include effectiveness as a tool for generating design data to learn from, how should design optimization tools be developed differently? 3) How else might machine learning be used with design optimization to support design synthesis besides reduced-dimension, targeted design representations? Could other more general or insightful forms of knowledge, such as design rules or procedures, be extracted from data obtained using design automation methods?

# Bibliography


[1] Simultaneous Design of Non-Newtonian Lubricant and Surface Texture Using Surrogate-Based Optimization.

[2] An Approach to Modeling & Representation of Heterogeneous Objects, author=Kumar, Vinod and Dutta, Debasish. *Journal of Mechanical Design 120*, 4 (1998), 659–667.

[3] Topology Optimization of Convection-dominated, Steady-state Heat Transfer Problems. *International Journal of Heat and Mass Transfer 50*, 15-16 (2007), 2859–2873.

[4] Agrawal, R., Imieliński, T., and Swami, A. Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Rec. 22*, 2 (June 1993), 207–216.

[5] Agrawal, R., and Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1994), VLDB '94, Morgan Kaufmann Publishers Inc., pp. 487–499.

[6] Aitken, A. C. On Least Squares and Linear Combination of Observations. *Proceedings of the Royal Society of Edinburgh 55* (1936), 42–48.

[7] Akundi, S. V., Simpson, T. W., and Reed, P. M. Multi-objective Design Optimization for Product Platform and Product Family Design Using Genetic Algorithms.

In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2005), American Society of Mechanical Engineers, pp. 999–1008.

[8] Albert, R., and Barabási, A.-L. Statistical Mechanics of Complex Networks. *Reviews of Modern Physics 74*, 1 (2002), 47.

[9] Alexandersen, J., Sigmund, O., and Aage, N. Large Scale Three-dimensional Topology Optimisation of Heat Sinks Cooled by Natural Convection. *International Journal of Heat and Mass Transfer 100* (2016), 876–891.

[10] Allison, J. T. Literature Network Analysis, 2014. https://www.youtube.com/watch?v=rFGD7YWqFYQ.

[11] Anyuru, A. *Professional Webgl Programming: Developing 3D Graphics for the Web*. John Wiley & Sons, 2012.

[12] Arjovsky, M., and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv preprint arXiv:1701.04862* (2017).

[13] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *arXiv preprint arXiv:1701.07875* (2017).

[14] ASME. International Design Engineering Technical Conferences & Computers & Information in Engineering Conference, 2017. https://www.asme.org/events/idetccie.

[15] Baesens, B. *Analytics in a Big Data World: The Essential Guide to Data Science and its Applications*. John Wiley & Sons, 2014.

[16] Barabási, A.-L., and Albert, R. Emergence of Scaling in Random Networks. *Science 286*, 5439 (1999), 509–512.

[17] Barnett, G. A., Huh, C., Kim, Y., and Park, H. W. Citations Among Comunication Journals and Other Disciplines: A Network Analysis. *Scientometrics 88*, 2 (2011), 449–469.

[18] Barton, J., Love, D., and Taylor, G. Design Determines 70% of Cost? A Review of Implications for Design Evaluation. *Journal of Engineering Design 12*, 1 (2001), 47–58.

[19] Bayrak, A. E., Ren, Y., and Papalambros, P. Y. Topology Generation for Hybrid Electric Vehicle Architecture Design. *Journal of Mechanical Design 138*, 8 (2016), 081401.

[20] Bendsøe, M. P. Optimal Shape Design as a Material Distribution Problem. *Structural and Multidisciplinary Optimization 1*, 4 (1989), 193–202.

[21] Bengio, Y., et al. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning 2*, 1 (2009), 1–127.

[22] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy Layer-wise Training of Deep Networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2006), NIPS'06, MIT Press, pp. 153–160.

[23] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*, Jan (2003), 993–1022.

[24] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment 2008*, 10 (2008), P10008.

[25] Börner, K., Maru, J. T., and Goldstone, R. L. The Simultaneous Evolution of Author and Paper networks. *Proceedings of the National Academy of Sciences 101*, suppl 1 (2004), 5266–5273.

[26] Box, G. E., and Draper, N. R. *Empirical Model-building and Response Surfaces*. John Wiley & Sons, 1987.

[27] Box, G. E., and Wilson, K. B. On the Experimental Attainment of Optimum Conditions. In *Breakthroughs in statistics*. Springer, 1992, pp. 270–310.

[28] Braha, D. *Data Mining for Design and Manufacturing: Methods and Applications*, vol. 3. Springer Science & Business Media, 2013.

[29] Breiman, L. Arcing the Edge. Tech. rep.

[30] Breiman, L. Bagging Predictors. *Machine learning 24*, 2 (1996), 123–140.

[31] Breiman, L. Random Forests. *Machine learning 45*, 1 (2001), 5–32.

[32] Brin, S., and Page, L. Reprint of: The Anatomy of A Large-scale Hypertextual Web Search Engine. *Computer Networks 56*, 18 (2012), 3825–3833.

[33] Broomhead, D. S., and Lowe, D. Radial Basis Functions, Multi-variable Functional Interpolation and Adaptive Networks. Tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[34] Bruccoleri, F., Klumperink, E. A., and Nauta, B. Generating All Two-MOS-transistor Amplifiers Leads to New wide-band LNAs. *IEEE Journal of Solid-State Circuits 36*, 7 (2001), 1032–1040.

[35] Buitelaar, P., and Eigner, T. Topic Extraction from Scientific Literature for Competency Management. In *the 7th International Semantic Web Conference* (2008).

[36] Bull, A. D. Convergence Rates of Efficient Global Optimization Algorithms. *Journal of Machine Learning Research 12*, Oct (2011), 2879–2904.

[37] Cambrosio, A., Keating, P., Mercier, S., Lewison, G., and Mogoutov, A. Mapping the Emergence and Development of Translational Cancer Research. *European Journal of Cancer 42*, 18 (2006), 3140–3148.

[38] Campbell, M. I., Rai, R., and Kurtoglu, T. A Stochastic Graph Grammar Algorithm for Interactive Search. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2009), American Society of Mechanical Engineers, pp. 829–840.

[39] Cang, R., Li, H., Yao, H., Jiao, Y., and Ren, Y. Improving Direct Physical Properties Prediction of Heterogeneous Materials from Imaging Data via Convolutional Neural Network and a Morphology-Aware Generative Model. *arXiv preprint arXiv:1712.03811* (2017).

[40] Cang, R., Xu, Y., Chen, S., Liu, Y., Jiao, Y., and Ren, M. Y. Microstructure Representation and Reconstruction of Heterogeneous Materials via Deep Belief Network for Computational Material Design. *Journal of Mechanical Design 139*, 7 (2017), 071404.

[41] Cantor, D., and Jones, B. *WebGL Beginner's Guide*. Packt Publishing Ltd, 2012.

[42] Casper, W. J., Eby, L. T., Bordeaux, C., Lockwood, A., and Lambert, D. A Review of Research Methods in IO/OB Work-family Research. *Journal of Applied Psychology 92*, 1 (2007), 28.

[43] Centola, D. The Social Origins of Networks and Diffusion. *American Journal of Sociology 120*, 5 (2015), 1295–1338.

[44] Chaloner, K., and Verdinelli, I. Bayesian Experimental Design: A Review. *Statistical Science* (1995), 273–304.

[45] Chan, E. DAC Citation Network, 2014. https://dac-network.herokuapp.com/#/home.

[46] Chen, P., and Redner, S. Community Structure of the Physical Review Citation Network. *Journal of Informetrics 4*, 3 (2010), 278–290.

[47] Chen, P., Xie, H., Maslov, S., and Redner, S. Finding Sientific Gems with Googles PageRank Algorithm. *Journal of Informetrics 1*, 1 (2007), 8–15.

[48] Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (2013), ACM, pp. 167–174.

[49] Cohn, D. A. Neural Network Exploration Using Optimal Experiment Design. In *Advances in Neural information Processing Systems* (1994), pp. 679–686.

[50] Collins, R., and Restivo, S. Development, Diversity, and Conflict in the Sociology of Science. *The Sociological Quarterly 24*, 2 (1983), 185–200.

[51] Cressie, N. Spatial Prediction and Ordinary Kriging. *Mathematical geology 20*, 4 (1988), 405–421.

[52] Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. Comparing Community Structure Identification. *Journal of Statistical Mechanics: Theory and Experiment 2005*, 09 (2005), P09008.

[53] Das, A., and Vemuri, R. An Automated Passive Analog Circuit Synthesis Framework Using Genetic Algorithms. In *VLSI, 2007. ISVLSI'07. IEEE Computer Society Annual Symposium on* (2007), IEEE, pp. 145–152.

[54] Das, A., and Vemuri, R. Topology Synthesis of Analog Circuits Based on Adaptively Generated Building Blocks. In *Proceedings of the 45th Annual Design Automation Conference* (2008), ACM, pp. 44–49.

[55] Dbouk, T. A Review about the Engineering Design of Optimal Heat Transfer Systems Using Topology Optimization. *Applied Thermal Engineering 112* (2017), 841–854.

[56] Deaton, J. D., and Grandhi, R. V. A Survey of Structural and Multidisciplinary Continuum Topology Optimization: Post 2000. *Structural and Multidisciplinary Optimization 49*, 1 (2014), 1–38.

[57] Deb, K. *Multi-objective Optimization using Evolutionary Algorithms*, vol. 16. John Wiley & Sons, 2001.

[58] Del Castillo, J. M. Enumeration of 1-DOF Planetary Gear Train Graphs Based on Functional Constraints. *Journal of Mechanical Design 124*, 4 (2002), 723–732.

[59] Denton, E. L., Chintala, S., Fergus, R., et al. Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks. In *Advances in neural information processing systems* (2015), pp. 1486–1494.

[60] Derrible, S. Network Centrality of Metro Systems. *PloS one 7*, 7 (2012), e40575.

[61] Dhar, V. Data Science and Prediction. *Commun. ACM 56*, 12 (Dec. 2013), 64–73.

[62] Ding, Y. Scientific Collaboration and Endorsement: Network Analysis of Coauthorship and Citation Networks. *Journal of Informetrics 5*, 1 (2011), 187–203.

[63] Ding, Y. Topic-based PageRank on Author Cocitation Networks. *Journal of the American Society for Information Science and Technology 62*, 3 (2011), 449–466.

[64] Ding, Y., Yan, E., Frazho, A., and Caverlee, J. PageRank for Ranking Authors in Co-citation Networks. *Journal of the American Society for Information Science and Technology 60*, 11 (2009), 2229–2243.

[65] Dosovitskiy, A., and Brox, T. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 658–666.

[66] Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially Learned Inference. *arXiv preprint arXiv:1606.00704* (2016).

[67] Efros, A. A., and Freeman, W. T. Image Quilting for Texture Synthesis and Transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 341–346.

[68] Efros, A. A., and Leung, T. K. Texture Synthesis by Non-parametric Sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 2, IEEE, pp. 1033–1038.

[69] El-Kishky, A., Song, Y., Wang, C., Voss, C. R., and Han, J. Scalable Topical Phrase Mining from Text Corpora. *Proceedings of the VLDB Endowment 8*, 3 (2014), 305–316.

[70] Fadel, G., Mocko, G., and Summers, J. NSF Sponsored Engineering Design and Systems Engineering Foundations Workshop (November 14-17, 2015), 2015. http://www.clemson.edu/centers-institutes/design/workshop/index.html.

[71] Fedorov, V. V. *Theory of Optimal Experiments*. Elsevier, 1972.

[72] Ferguson, S., Siddiqi, A., Lewis, K., and de Weck, O. L. Flexible and Reconfigurable Systems: Nomenclature and Review. In *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2007), American Society of Mechanical Engineers, pp. 249–263.

[73] Forrester, A. I., and Keane, A. J. Recent Advances in Surrogate-based Optimization. *Progress in Aerospace Sciences 45*, 1-3 (2009), 50–79.

[74] Fortunato, S. Community Detection in Graphs. *Physics Reports 486*, 3 (2010), 75–174.

[75] Foster, R. M. Geometrical Circuits of Electrical Networks. *Transactions of the American Institute of Electrical Engineers 51*, 2 (1932), 309–317.

[76] Freund, Y., and Schapire, R. E. A Dcision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of computer and system sciences 55*, 1 (1997), 119–139.

[77] Friedman, J. H. Multivariate Adaptive Regression Splines. *The annals of statistics* (1991), 1–67.

[78] Friedman, J. H. Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics* (2001), 1189–1232.

[79] Friedman, J. H. Stochastic Gradient Boosting. *Computational Statistics & Data Analysis 38*, 4 (2002), 367–378.

[80] Fu, K., Cagan, J., Kotovsky, K., and Wood, K. Discovering Structure in Design Databases Through Functional and Surface Based Mapping. *Journal of Mechanical Design 135*, 3 (2013), 031006.

[81] Fuge, M., Peters, B., and Agogino, A. Machine Learning Algorithms for Recommending Design Methods. *Journal of Mechanical Design 136*, 10 (2014), 101103.

[82] Fujii, A., Tokunaga, T., Inui, K., and Tanaka, H. Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics 24*, 4 (1998), 573–597.

[83] Gan, Z., Yang, Z., Shang, T., Yu, T., and Jiang, M. Automated Synthesis of Passive Analog Filters Using Graph Representation. *Expert Systems with Applications 37*, 3 (2010), 1887–1898.

[84] Gandomi, A., and Haider, M. Beyond the Hype: Big Data Concepts, Methods, and Analytics. *International Journal of Information Management 35*, 2 (2015), 137–144.

[85] Gao, Y., and Zhang, P. A Survey of Homogeneous and Heterogeneous System Architectures in High Performance Computing. In *Smart Cloud (SmartCloud), IEEE International Conference on* (2016), IEEE, pp. 170–175.

[86] Gatys, L. A., Ecker, A. S., and Bethge, M. A Neural Algorithm of Artistic Style. *arXiv preprint arXiv:1508.06576* (2015).

[87] Gelbart, M. A., Snoek, J., and Adams, R. P. Bayesian Optimization with Unknown Constraints. *arXiv preprint arXiv:1403.5607* (2014).

[88] Gersborg-Hansen, A., Bendsoe, M. P., and Sigmund, O. {Topology Optimization of Heat Conduction Problems Using the Finite Volume Method}. *Structural and Multidisciplinary Optimization 31* (mar 2006), 251–259.

[89] Goh, C., and Li, Y. GA Automated Design and Synthesis of Analog Circuits with Practical Constraints. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (2001), vol. 1, IEEE, pp. 170–177.

[90] Goodfellow, I. NIPS 2016 tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160* (2016).

[91] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (2014), pp. 2672–2680.

[92] Goodfellow, I. J. On Distinguishability Criteria for Estimating Generative Models. *arXiv preprint arXiv:1412.6515* (2014).

[93] Greenberg, S. A. How Citation Distortions Create Unfounded Authority: Analysis of a Citation Network. *BMJ: British Medical Journal* (2009), 210–213.

[94] Grimbleby, J. B. Automatic Analogue Network Synthesis Using Genetic Algorithms. In *Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALESIA. First International Conference on (Conf. Publ. No. 414)* (1995), IET, pp. 53–58.

[95] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved Training of Wasserstein Gans. In *Advances in Neural Information Processing Systems* (2017), pp. 5769–5779.

[96] Guo, T. Design of Genetic Regulatory Networks. M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May 2014.

[97] Guo, T. Network Analysis of Design Automation Literature, 2017. https://github.com/TinghaoGuo/Network-Analysis-of-Design-Automation-Literature.

[98] Guo, T., and Allison, J. T. On the Use of Mathematical Programs with Complementary Constraints in Combined Topological and Parametric Design of Biochemical Enzyme Networks. *Engineering Optimization 49*, 2 (2017), 345–364.

[99] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T. An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum* (Jan. 2018), no. AIAA 2018-0804.

[100] Guo, T., Xu, J., Sun, Y., Dong, Y., Davis, N. E., and Allison, J. T. Network Analysis of Design Automation Literature. *Journal of Mechincial Design* (2017). Reference to be added by copyeditor after DOI assigned for the main paper hosted by ASME JMD.

[101] Guo, T., Xu, J., Sun, Y., Dong, Y., Davis, N. E., and Allison, J. T. Network Analysis of Design Automation Literature. In *Proceedings of the ASME 2017 International Design Engineering Techincal Conferences & Computers and Information in Engineering Conference* (Cleveland, Ohio USA, Aug 2017).

[102] Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring Network Structure, Dynamics, and Function Using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)* (Pasadena, CA USA, Aug. 2008), pp. 11–15.

[103] Hahsler, M. A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules, 2015. http://michael.hahsler.net/research/association_rules/measures.html.

[104] Hahsler, M., Grün, B., and Hornik, K. Arules - A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software 14*, 15 (2005), 1–25.

[105] Hahsler, M., and Karpienko, R. Visualizing Association Rules in Hierarchical Groups. *Journal of Business Economics* (2017), 1–19.

[106] Hanneke, S. *Theoretical Foundations of Active Learning.* PhD thesis, Carnegie Mellon University, 2009.

[107] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data mining, Inference and Prediction*, 2 ed. Springer, 2009.

[108] Hazen, B. T., Boone, C. A., Ezell, J. D., and Jones-Farmer, L. A. Data Quality for Data Science, Predictive Analytics, and Big Data in Supply Chain Management: An Introduction to the Problem and Suggestions for Research and Applications. *International Journal of Production Economics 154* (2014), 72–80.

[109] Herber, D. R. *Advances in Combined Architecture, Plant, and Control Design*. Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. 2017.

[110] Herber, D. R., and Allison, J. T. A Problem Class with Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions. In *(submitted to) ASME 2018 International Design Engineering Technical Conferences* (Quebec City, Canada, Aug. 2018).

[111] Herber, D. R., and Allison, J. T. A Problem Class with Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions. In *ASME 2018 International Design Engineering Technical Conferences (to appear)* (Quebec City, Canada, Aug. 2018), no. DETC2018-86213.

[112] Herber, D. R., Guo, T., and Allison, J. T. Enumeration of Architectures with Perfect Matchings. *Journal of Mechanical Design 139*, 5 (2017), 051403.

[113] Herrera, M., Roberts, D. C., and Gulbahce, N. Mapping the Evolution of Scientific Fields. *PloS one 5*, 5 (2010), e10355.

[114] Hinton, G., Srivastava, N., and Swersky, K. Neural Networks for Machine Learning Lecture 6a Overview of Mini-batch Gradient Descent.

[115] Hinton, G. E., and Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science 313*, 5786 (2006), 504–507.

[116] Ho, T. K. Random Decision Forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on* (1995), vol. 1, IEEE, pp. 278–282.

[117] Ho, T. K. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine intelligence 20*, 8 (1998), 832–844.

[118] Hyndman, R. J., and Koehler, A. B. Another Look at Measures of Forecast Accuracy. *International journal of forecasting 22*, 4 (2006), 679–688.

[119] Iga, A., Nishiwaki, S., Izui, K., and Yoshimura, M. Topology Optimization for Thermal Conductors Considering Design-Dependent Effects, Including Heat Conduction and Convection. *International Journal of Heat and Mass Transfer 52*, 11-12 (2009), 2721–2732.

[120] Ilies, H., Parkinson, M., Seepersad, C. C., Kokkolaras, M., Ragsdell, K., Papalambros, P. Y., Mistree, F., Williams, C., Rai, R., Panchal, J., Ferguson, S., DuPont, B., and Allison, J. T. New Perspectives on Design Automation: Celebrating the 40th Anniversary of the ASME Design Automation Conference. *ASME Journal of Mechanical Design 137*, 5 (May 2015), 050301.

[121] Ioffe, S., and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167* (2015).

[122] Iuliano, E. Adaptive Sampling Strategies for Surrogate-based Aerodynamic optimization. In *Application of Surrogate-based Global Optimization to Aerodynamic Design*. Springer, 2016, pp. 25–46.

[123] Jain, A. K., Murty, M. N., and Flynn, P. J. Data Clustering: A Review. *ACM Computing Surveys (CSUR) 31*, 3 (1999), 264–323.

[124] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning*, vol. 112. Springer, 2013.

[125] Jiao, R., and Choi, S.-K. NSF ESD/SYS Program Workshop and Grantees Meeting: Future Directions in Engineering Design and Systems Engineering, 2017. http://nsfworkshop.gatech.edu/.

[126] Jin, R., Chen, W., and Simpson, T. W. Comparative Studies of Metamodelling Techniques under Multiple Modelling Criteria. *Structural and multidisciplinary optimization 23*, 1 (2001), 1–13.

[127] Kaiser, M. Mean Clustering Coefficients: the Role of Isolated Nodes and Leafs on Clustering Measures for Small-world Networks. *New Journal of Physics 10*, 8 (2008), 083042.

[128] Kajikawa, Y., Ohno, J., Takeda, Y., Matsushima, K., and Komiyama, H. Creating an Academic Landscape of Sustainability Science: An Analysis of the Citation Network. *Sustainability Science 2*, 2 (2007), 221–231.

[129] Kajikawa, Y., and Takeda, Y. Citation Network Analysis of Organic LEDs. *Technological Forecasting and Social Change 76*, 8 (2009), 1115–1123.

[130] Kang, J., Ryu, K. R., and Kwon, H.-C. 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* (2004), Springer, pp. 384–388.

[131] Kato, H., and Funazaki, K.-i. POD-driven Adaptive Sampling for Efficient Surrogate Modeling and Its Application to Supersonic Turbine Optimization.

[132] Kendall, M. G. A New Measure of Rank Correlation. *Biometrika 30*, 1/2 (1938), 81–93.

[133] Khetan, A., and Allison, J. T. Large-Scale Topology Optimization Using Parametrized Boolean Networks. In *ASME 2014 International Design Engineering Technical Conferences* (Buffalo, NY, USA, Aug. 2014), no. DETC2014-34256, p. V02AT03A006.

[134] Khetan, A., Lohan, D. J., and Allison, J. T. Managing Variable-Dimension Structural Optimization Problems Using Generative Algorithms. *Structural and Multidisciplinary Optimization 50*, 4 (Oct. 2015), 695–715.

[135] Kingma, D. P., and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[136] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems* (2014), pp. 3581–3589.

[137] Kingma, D. P., and Welling, M. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).

[138] Kleinberg, J. M. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM) 46*, 5 (1999), 604–632.

[139] Klosik, D., and Bornholdt, S. The Citation Wake of Publications Detects Nobel Laureates' Papers. *PloS one 9*, 12 (2014), e113184.

[140] Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A., and Dunlap, F. Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming. *IEEE Transactions on evolutionary computation 1*, 2 (1997), 109–128.

[141] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[142] Kusiak, A. Big Data in Mechanical Engineering, 2015. https://www.asme.org/career-education/early-career-engineers/me-today/big-data-in-mechanical-engineering.

[143] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. In *ACM Transactions on Graphics (ToG)* (2003), vol. 22, ACM, pp. 277–286.

[144] Lancaster, P., and Salkauskas, K. Surfaces Generated by Moving Least Squares Methods. *Mathematics of computation 37*, 155 (1981), 141–158.

[145] Landhuis, E. Neuroscience: Big Brain, Big Data. *Nature 541*, 7638 (2017), 559–561.

[146] Langley, P., and Simon, H. A. Applications of Machine Learning and Rule Induction. *Communications of the ACM 38*, 11 (1995), 54–64.

[147] Le, Q., and Panchal, J. H. Building Smaller Sized Surrogate Models of Complex Bipartite Networks Based on Degree Distributions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 42*, 5 (2012), 1152–1166.

[148] Le, Q. V., et al. A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks.

[149] Lee, Y. H., Corman, R. E., Ewoldt, R. H., and Allison, J. T. A Multi-objective Adaptive Surrogate Modeling-Based Optimization (MO-ASMO) Framework Using Efficient Sampling Strategies. In *ASME 2017 International Design Engineering Technical Conferences* (Cleveland, OH, USA, Aug. 2017), no. DETC2017-67541, p. V02BT03A023.

[150] Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[151] Levenshtein, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet physics doklady 10* (1966), 707.

[152] Levin, D. The Approximation Power of Moving Least-squares. *Mathematics of Computation of the American Mathematical Society 67*, 224 (1998), 1517–1531.

[153] Lewis, D. D., and Gale, W. A. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (1994), Springer-Verlag New York, Inc., pp. 3–12.

[154] Li, X., Chen, H., Huang, Z., and Roco, M. C. Patent Citation Network in Nanotechnology (1976–2004). *Journal of Nanoparticle Research 9*, 3 (2007), 337–352.

[155] Li, X.-H., and Guan, Z.-H. Energy-aware Routing in Wireless Sensor Networks Using Local Betweenness Centrality. *International Journal of Distributed Sensor Networks 2013* (2013).

[156] Liben-Nowell, D., and Kleinberg, J. The Oink-prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology 58*, 7 (2007), 1019–1031.

[157] Liu, K., and Tovar, A. An Efficient 3D Topology Optimization Code Written in Matlab. *Structural and Multidisciplinary Optimization 50*, 6 (2014), 1175–1196.

[158] Liu, X., Bollen, J., Nelson, M. L., and Van de Sompel, H. Co-authorship Networks in the Digital Library Research Community. *Information Processing & Management 41*, 6 (2005), 1462–1480.

[159] Lohan, D. J., and Allison, J. T. Temperature Constraint Formulations for Heat Conduction Topology Optimization. In *ISSMO 12th World Congress of Structural and Multidisciplinary Optimisation, to appear* (Braunschweig, Germany, June 2017).

[160] Lohan, D. J., Dede, E., and Allison, J. T. Topology Optimization Formulations for Circuit Board Heat Spreader Design. In *Proceedings of 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (2016), p. 3669.

[161] Lohan, D. J., Dede, E. M., and Allison, J. T. Topology Optimization for Heat Conduction Using Generative Design Algorithms. *Structural and Multidisciplinary Optimization 55*, 3 (Mar. 2017), 1063–1077.

[162] Lohn, J. D., and Colombano, S. P. A Circuit Representation Technique for Automated Circuit Design. *IEEE Transactions on Evolutionary Computation 3*, 3 (1999), 205–219.

[163] Longino, H. The Social Dimensions of Scientific Knowledge. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., spring 2016 ed. Metaphysics Research Lab, Stanford University, 2016.

[164] Loper, E., and Bird, S. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics-Volume 1* (2002), Association for Computational Linguistics, pp. 63–70.

[165] Luan, S., Arora, M., Thurston, D. L., and Allison, J. T. Alternative design optimization formulations developing and comparing for a vibration damping example. In *ASME 2017 International Design Engineering Technical Conferences* (Cleveland, Ohio, USA, Aug. 2017), no. DETC2017-68337, p. V004T05A004.

[166] Ma, W., Trusina, A., El-Samad, H., Lim, W. A., and Tang, C. Defining Network Topologies That Can Achieve Biochemical Adaptation. *Cell 138*, 4 (2009), 760–773.

[167] Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models.

[168] Makhzani, A., and Frey, B. K-sparse Autoencoders. *arXiv preprint arXiv:1312.5663* (2013).

[169] Malliaros, F. D., and Vazirgiannis, M. Clustering and Community Detection in Directed Networks: A Survey. *Physics Reports 533*, 4 (2013), 95–142.

[170] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. Least Squares Generative Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), IEEE, pp. 2813–2821.

[171] MATLAB. Find Pareto Front of Multiple Fitness Functions Using Genetic Algorithm, 2017. https://www.mathworks.com/help/gads/gamultiobj.html.

[172] Matthews, P. C., Blessing, L. T., and Wallace, K. M. The Introduction of a Design Heuristics Extraction Method. *Advanced Engineering Informatics 16*, 1 (2002), 3–19.

[173] Mattson, C. A., and Wood, A. E. Nine Principles for Design for the Developing World as Derived from the Engineering Literature. *Journal of Mechanical Design 136*, 12 (2014), 121403.

[174] McCallumzy, A. K., and Nigamy, K. Employing EM and Pool-based Active Learning for Text Classification. In *Proceedings of International Conference on Machine Learning (ICML)*, Citeseer.

[175] Meckesheimer, M., Booker, A. J., Barton, R. R., and Simpson, T. W. Computationally Inexpensive Metamodel Assessment Strategies. *AIAA journal 40*, 10 (2002), 2053–2060.

[176] Merrian-Webster.com. "Heterogeneity.", 2017. https://www.merriam-webster.com/dictionary/heterogeneity.

[177] Merrian-Webster.com. "Homogeneity", 2017. https://www.merriam-webster.com/dictionary/homogeneity.

[178] Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. Unrolled Generative Adversarial Networks. *arXiv preprint arXiv:1611.02163* (2016).

[179] Miettinen, K. *Nonlinear Multiobjective Optimization*, vol. 12. Springer Science & Business Media, 2012.

[180] Mitea, O., Meissner, M., Hedrich, L., and Jores, P. Automated Constraint-driven Topology Synthesis for Analog Circuits. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011* (2011), IEEE, pp. 1–4.

[181] Moody, J. The Structure of a Social Science Collaboration Network: Disciplinary Cohesion from 1963 to 1999. *American Sociological Review 69*, 2 (2004), 213–238.

[182] Myers, J. L., Well, A. D., and Lorch Jr, R. F. *Research Design and Statistical Analysis*. Routledge, 2013.

[183] Narayanan, S. *The Betweenness Centrality of Biological Networks*. PhD thesis, Citeseer, 2005.

[184] Newman, M. E. The Structure of Scientific Collaboration Networks. *Proceedings of the National Academy of Sciences 98*, 2 (2001), 404–409.

[185] Newman, M. E. The Structure and Function of Complex Networks. *SIAM review 45*, 2 (2003), 167–256.

[186] Newman, M. E. Coauthorship Networks and Patterns of Scientific Collaboration. *Proceedings of the National Academy of Sciences 101*, suppl 1 (2004), 5200–5205.

[187] Newman, M. E. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E 69*, 6 (2004), 066133.

[188] Newman, M. E. Power Laws, Pareto Distributions and Zipf's Law. *Contemporary Physics 46*, 5 (2005), 323–351.

[189] Newman, M. E., and Girvan, M. Finding and Evaluating Community Structure in Networks. *Physical Review E 69*, 2 (2004), 026113.

[190] Nguyen, H. T., and Smeulders, A. Active Learning Using Pre-clustering. In *Proceedings of the Twenty-first International conference on Machine Learning* (2004), ACM, p. 79.

[191] Ni, C., Sugimoto, C., and Jiang, J. Degree, Closeness, and Betweenness: Application of Group Centrality Measurements to Explore Macro-disciplinary Evolution Diachronically. In *Proceedings of ISSI* (2011), pp. 1–13.

[192] Ochotta, E. S., Rutenbar, R. A., and Carley, L. R. Synthesis of High-performance Analog Circuits in ASTRX/OBLX. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 15*, 3 (1996), 273–294.

[193] Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. Pixel Recurrent Neural Networks. *arXiv preprint arXiv:1601.06759* (2016).

[194] Osborne, M. J., and Rubinstein, A. *A Course inGgame Theory*. 1994.

[195] Otte, E., and Rousseau, R. Social Network Analysis: A Powerful Strategy, Also for the Information Sciences. *Journal of Information Science 28*, 6 (2002), 441–453.

[196] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[197] Papalambros, P. Y. Design Science: Why, What and How. *Design Science 1* (2015).

[198] Parisi, T. *WebGL: Up and Running*. O'Reilly Media, Inc., 2012.

[199] Park, J.-S. Optimal Latin-hypercube Designs for Computer Experiments. *Journal of statistical planning and inference 39*, 1 (1994), 95–111.

[200] Pearson, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[201] Peddada, S. R. T., Herber, D. R., Pangborn, H., Alleyne, A. G., and Allison, J. T. Optimal Flow Control and Single Split Architecure Exploration for Fluid-Based Thermal Management. In *ASME 2018 International Design Engineering Technical Conferences* (2018), no. DETC2018-86148.

[202] Price, D. d. S. A General Theory of Bibliometric and Other Cumulative Advantage Processes. *Journal of the American Society for Information Science 27*, 5 (1976), 292–306.

[203] Price, D. J. d. S. Networks of Scientific Papers. *Science 149*, 3683 (1965), 510–515.

[204] Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of Population Structure Using Multilocus Genotype Data. *Genetics 155*, 2 (2000), 945–959.

[205] Qi, G.-J. Loss-sensitive Generative Adversarial Networks on Lipschitz Densities. *arXiv preprint arXiv:1701.06264* (2017).

[206] Qian, X., and Dutta, D. Physics-basedMmodeling for Heterogeneous Objects. *Ann Arbor 1001* (2003), 48105.

[207] Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434* (2015).

[208] Radhakrishnan, P., and Campbell, M. I. A Graph Grammar Based Scheme for Generating and Evaluating Planar Mechanisms. In *Design Computing and Cognition10*. Springer, 2011, pp. 663–679.

[209] Rao, L. G., Schuh, J. K., Ewoldt, R. H., and Allison, J. T. On Using Adaptive Surrogate Modeling in Design For Efficient Fluid Power. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Boston, MA, USA, Aug. 2015), no. DETC2015-46832, pp. V02BT03A024–V02BT03A024.

[210] Redner, S. Citation Statistics from More Than a Century of Physical Review. *arXiv preprint physics/0407137* (2004).

[211] Ren, Y., Bayrak, A. E., and Papalambros, P. Y. EcoRacer: Game-Based Optimal Electric Vehicle Design and Driver Control Using Human Players. *Journal of Mechanical Design 138*, 6 (2016), 061407.

[212] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive Auto-encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (2011), pp. 833–840.

[213] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive Auto-encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), pp. 833–840.

[214] Rose, S., Engel, D., Cramer, N., and Cowley, W. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory* (2010), 1–20.

[215] Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. The Author-topic Model for Authors and Documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (2004), AUAI Press, pp. 487–494.

[216] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. Design and Analysis of Computer Experiments. *Statistical Science* (1989), 409–423.

[217] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems* (2016), pp. 2234–2242.

[218] Settles, B. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 6*, 1 (2012), 1–114.

[219] Settles, B., and Craven, M. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the conference on empirical methods in natural language processing* (2008), Association for Computational Linguistics, pp. 1070–1079.

[220] Settles, B., Craven, M., and Friedland, L. Active Learning with Real Annotation Costs. In *Proceedings of the NIPS Workshop on Cost-sensitive Learning*.

[221] Seung, H. S., Opper, M., and Sompolinsky, H. Query by Committee. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), ACM, pp. 287–294.

[222] Sexton, T., and Ren, M. Y. Learning an Optimization Algorithm through Human Design Iterations. *arXiv preprint arXiv:1608.06984* (2016).

[223] Shibata, N., Kajikawa, Y., Takeda, Y., Sakata, I., and Matsushima, K. Detecting Emerging Research Fronts in Regenerative Medicine by the Citation Network Analysis of Scientific Publications. *Technological Forecasting and Social Change 78*, 2 (2011), 274–282.

[224] Sigmund, O. A 99 Line Topology Optimization Code Written in Matlab. *Structural and Multidisciplinary Optimization 21*, 2 (2001), 120–127.

[225] Sigmund, O., and Maute, K. Topology Optimization Approaches. *Structural and Multidisciplinary Optimization 48*, 6 (2013), 1031–1055.

[226] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature 529*, 7587 (2016), 484–489.

[227] Simonyan, K., and Zisserman, A. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[228] Smith, B. C., Settles, B., Hallows, W. C., Craven, M. W., and Denu, J. M. SIRT3 Substrate Specificity Determined by Peptide Arrays and Machine Learning. *ACS chemical biology 6*, 2 (2010), 146–157.

[229] Snoek, J., Larochelle, H., and Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in neural information processing systems* (2012), pp. 2951–2959.

[230] Su, D. An Interative Visulization of DAC Citation Network, 2016. https://github.com/sudongqi/Propagation_Mergence.

[231] Su, D. Propagation Mergence for Community Detection, 2017. Senior thesis, University of Illinois at Urbana-Champaign, https://www.ideals.illinois.edu/handle/2142/96023.

[232] Sugiyama, K., Ohsaki, H., and Imase, M. Structural Analysis of Paper Citation and Co-authorship Networks Using Network Analysis Techniques. *Graduate School of Information Science and Technology, Osaka University, Japan* (2006).

[233] Sun, Y., Barber, R., Gupta, M., Aggarwal, C. C., and Han, J. Co-author Relationship Prediction in Heterogeneous Bibliographic Networks. In *Advances in*

*Social Networks Analysis and Mining (ASONAM), 2011 International Conference on* (2011), IEEE, pp. 121–128.

[234] Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. Pathsim: Meta Path-based Top-k Similarity Search in Heterogeneous Information Networks. *Proceedings of the VLDB Endowment 4*, 11 (2011), 992–1003.

[235] Sussman, G., and Stallman, R. Heuristic Techniques in Computer-aided Circuit Analysis. *IEEE Transactions on Circuits and Systems 22*, 11 (1975), 857–865.

[236] Svanberg, K., and Svärd, H. Density Filters for Topology Optimization Based on the Pythagorean Means. *Structural and Multidisciplinary Optimization 48*, 5 (2013), 859–875.

[237] Symon, R., and Dangerfield, K. Application of Design to Cost in Engineering and Manufacturing. In *NATO AGARD Lecture Series, The Application of Design To Cost And Life Cycle Cost To Aircraft Engines* (1980), no. 107, pp. 15–16.

[238] Tang, J., Jin, R., and Zhang, J. A Topic Modeling Approach and Its Integration into the Random Walk Framework for Academic Search. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 1055–1060.

[239] Tolstikhin, I. O., Gelly, S., Bousquet, O., Simon-Gabriel, C.-J., and Schölkopf, B. Adagan: Boosting Generative Models. In *Advances in Neural Information Processing Systems* (2017), pp. 5430–5439.

[240] Turk-Browne, N. B. Functional Interactions As Big Data in the Human Brain. *Science 342*, 6158 (2013), 580–584.

[241] Uzzi, B., Amaral, L. A., and Reed-Tsochas, F. Small-world Networks and Management Science Research: A Review. *European Management Review 4*, 2 (2007), 77–91.

[242] van der Aalst, W. M. *Process Mining: Data Science in Action*. Springer, 2016.

[243] Van Horn, J. D., and Toga, A. W. Human Neuroimaging as a Big Data Science. *Brain Imaging and Behavior 8*, 2 (2014), 323–331.

[244] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

[245] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research 11*, Dec (2010), 3371–3408.

[246] Wahba, G. *Spline Models for Observational Data*, vol. 59. Siam, 1990.

[247] Waller, M. A., and Fawcett, S. E. Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *Journal of Business Logistics 34*, 2 (2013), 77–84.

[248] Wang, G. G., and Shan, S. Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design 129*, 4 (2007), 370–380.

[249] Wang, M. Y., Wang, X., and Guo, D. A Level Set Method for Structural Topology Optimization. *Computer Methods in Applied Mechanics and Engineering 192*, 1 (2003), 227–246.

[250] Watts, D. J., and Strogatz, S. H. Collective Dynamics of Ssmall-world Networks. *Nature 393*, 6684 (1998), 440–442.

[251] Wei, L.-Y., and Levoy, M. Fast Texture Synthesis Using Tree-structured Vector Quantization. In *Proceedings of the 27th annual conference on Computer graphics and*

*interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 479–488.

[252] Weinberger, K., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 1113–1120.

[253] Wellman, B., and Berkowitz, S. D. *Social Structures: A Network Approach*, vol. 2. CUP Archive, 1988.

[254] Wu, W., and Benesty, M. *FeatureHashing: Creates a Model Matrix via Feature Hashing with a Formula Interface*, 2016. R package version 0.9.1.1.

[255] Xing, C., Ma, L., and Yang, X. Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images. *Journal of Sensors 2016* (2015).

[256] Xu, H., Greene, M. S., Deng, H., Dikin, D., Brinson, C., Liu, W. K., Burkhart, C., Papakonstantopoulos, G., Poldneff, M., and Chen, W. Stochastic Reassembly Strategy for Managing Information Complexity in Heterogeneous Materials Analysis and Design. *Journal of Mechanical Design 135*, 10 (2013), 101010.

[257] Xu, H., Li, Y., Brinson, C., and Chen, W. A Descriptor-based Design Methodology for Developing Heterogeneous Microstructural Materials System. *Journal of Mechanical Design 136*, 5 (2014), 051007.

[258] Xu, H., Liu, R., Choudhary, A., and Chen, W. A Machine Learning-based Design Representation Method for Designing Heterogeneous Microstructures. *Journal of Mechanical Design 137*, 5 (2015), 051403.

[259] Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. A. Scan: A Structural Clustering Algorithm for Networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2007), ACM, pp. 824–833.

[260] Yang, Z., Hong, L., and Davison, B. D. Topic-driven Multi-type Citation Network Analysis. In *Adaptivity, Personalization and Fusion of Heterogeneous Information* (Paris, France, France, 2010), RIAO '10, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, pp. 24–31.

[261] Zhang, P., and Koppaka, L. Semantics-based Legal Citation Network. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law* (2007), ACM, pp. 123–130.

[262] Zhao, J., Mathieu, M., and LeCun, Y. Energy-based Generative Adversarial Network. *arXiv preprint arXiv:1609.03126* (2016).

[263] Zhao, S., Song, J., and Ermon, S. Towards Deeper Understanding of Variational Autoencoding Models. *arXiv preprint arXiv:1702.08658* (2017).

[264] Zhu, X., and Goldberg, A. B. Introduction to Semi-supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 3*, 1 (2009), 1–130.

[265] Zhuang, C., and Xiong, Z. Temperature-Constrained Topology Optimization of Transient Heat Conduction Problems. *Numerical Heat Transfer, Part B: Fundamentals 68*, 4 (2015), 366–385.

# Appendix A

# CHAPTER 3

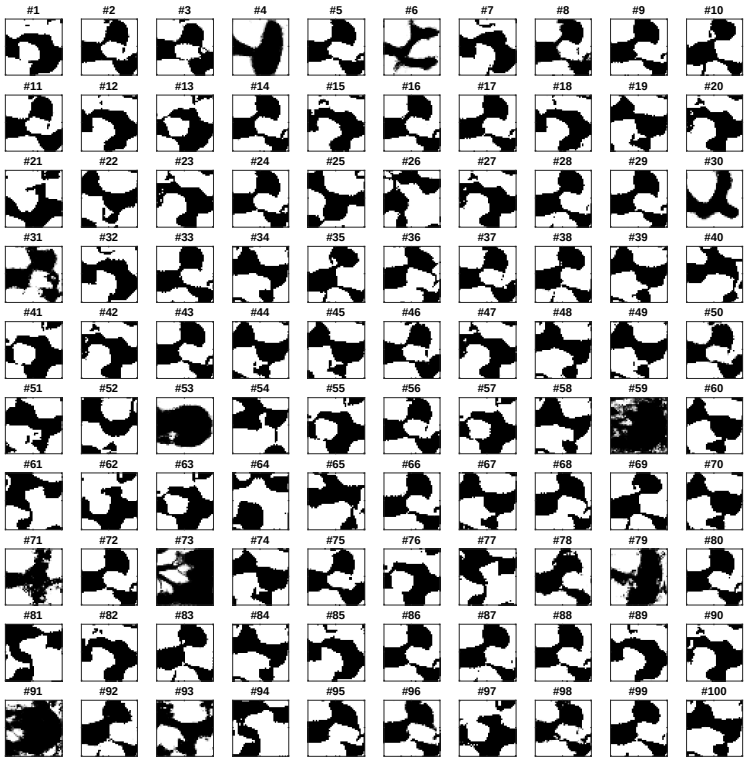


Figure A.1: A sample of dominated points for the weighted sum method

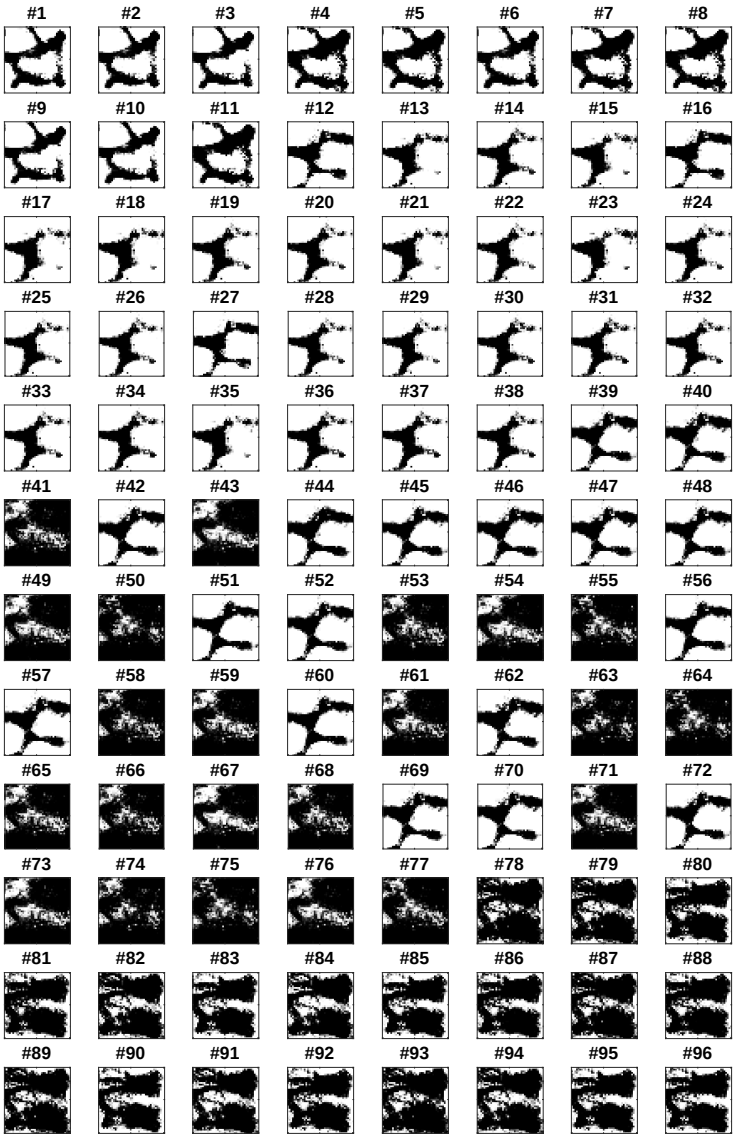



Figure A.2: Pareto-optimal solutions (#1 – #100) for the MOGA method (#latent variables = 20)

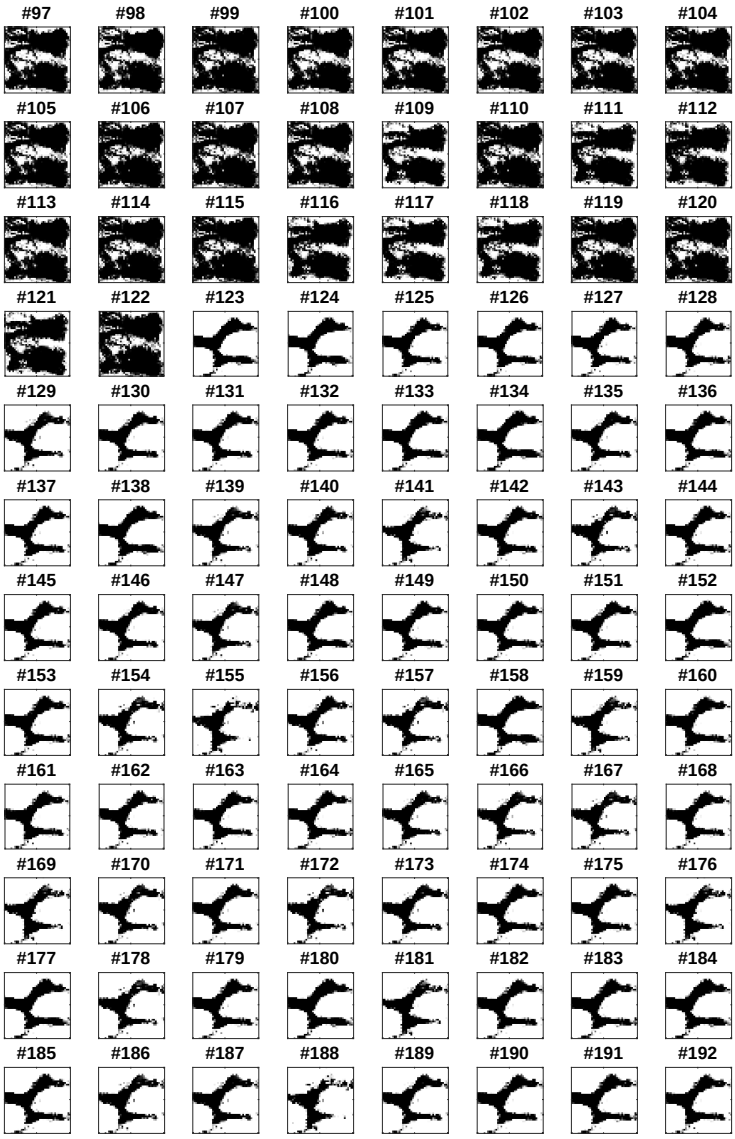



Figure A.3: Pareto-optimal solutions (#101 – #200) for the MOGA method (#latent variables = 20)

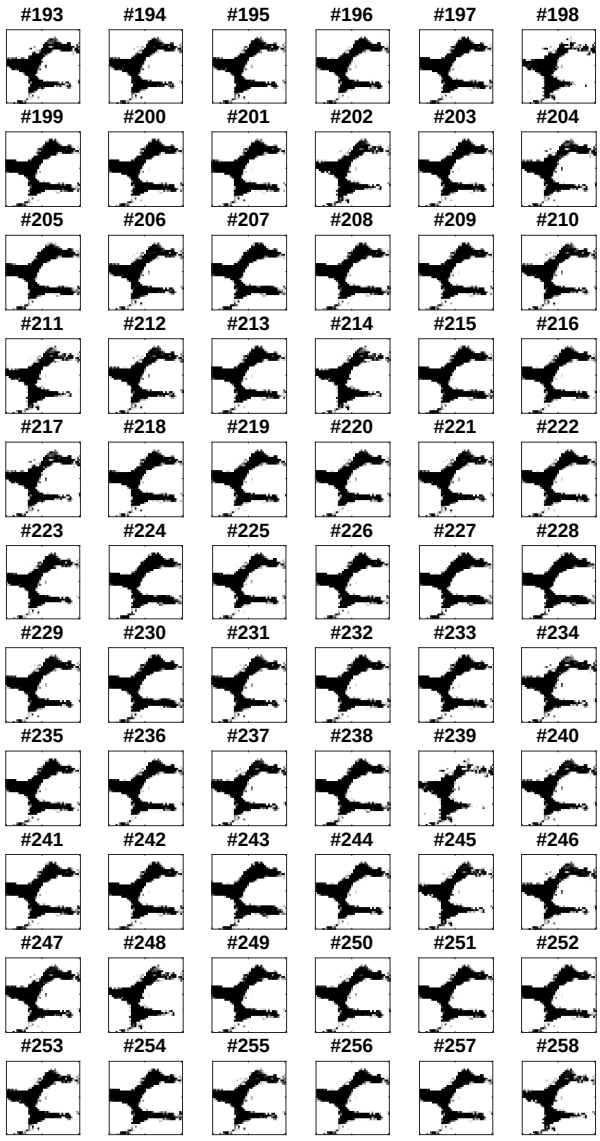


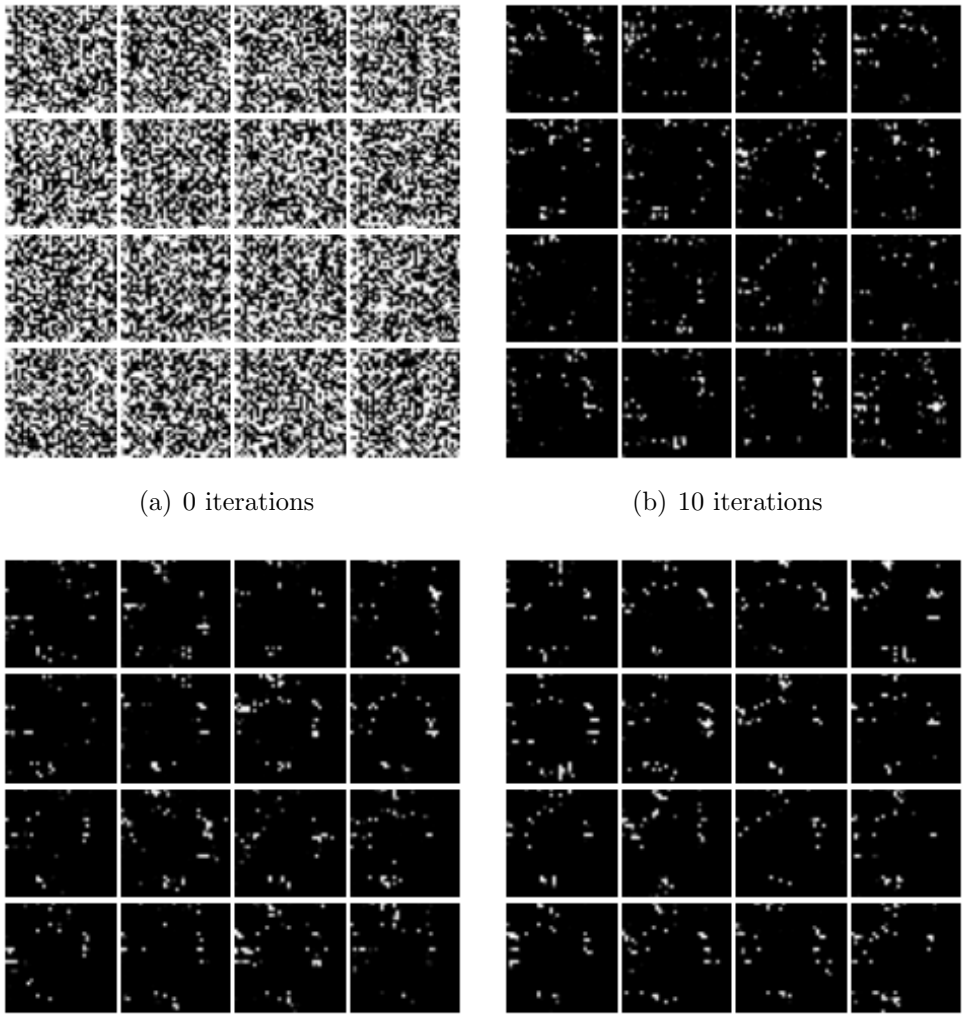

Figure A.4: Pareto-optimal solutions (#201 – #258) for the MOGA method (#latent variables = 20)

# Appendix B

# CHAPTER 5

Table B1: Low-pass filter: the parametric study on the parameters of the improved WGAN

| | Value | Percentage |
|---|---|---|
| Latent variable | 16 | 9.18% |
| | 32 | 9.06% |
| | 64 | 14.55% |
| | 128 | 10.26% |
| $\lambda$ | 0.1 | 7.03% |
| | 1 | 8.89% |
| | 10 | 14.55% |
| | 100 | 9.87% |



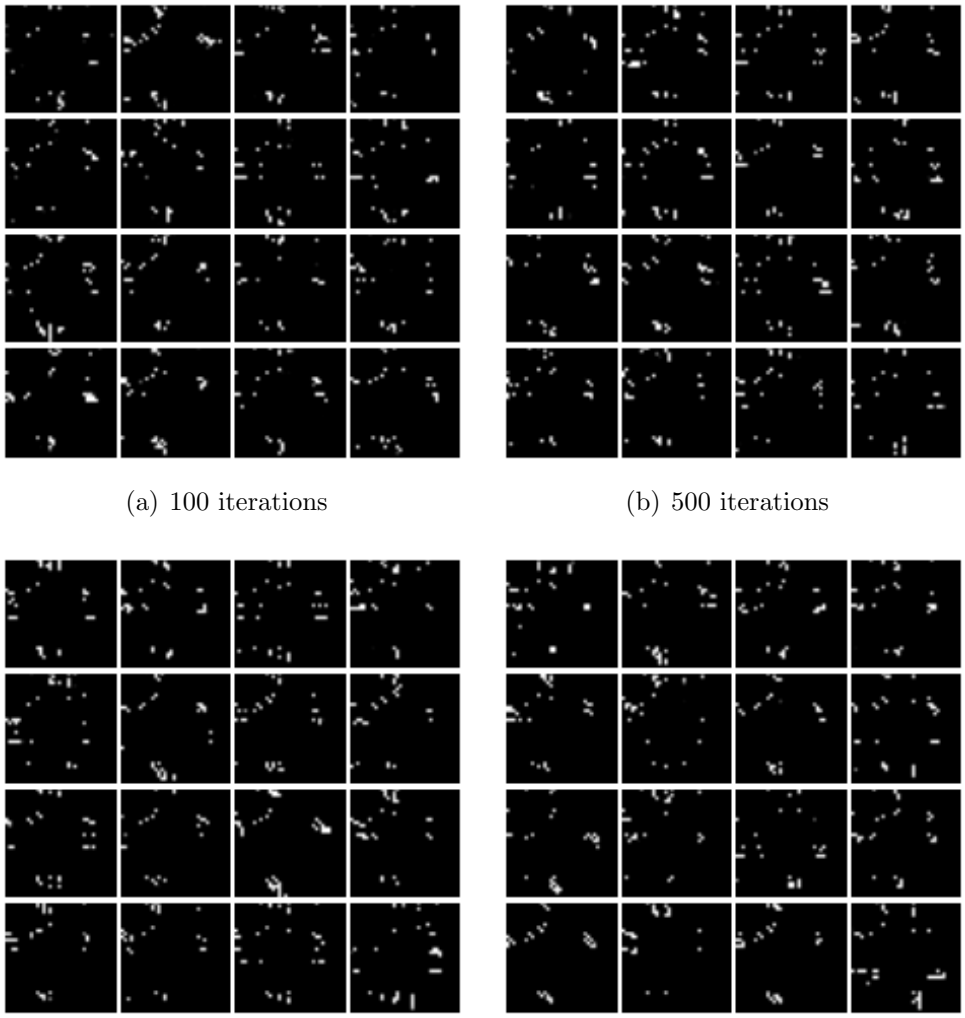
(a) 0 iterations

(b) 10 iterations

(c) 20 iterations

(d) 30 iterations

Figure B.1: Low-pass filter: generated circuit topologies (0 – 30 iterations)



(a) 100 iterations

(b) 500 iterations

(c) 1000 iterations

(d) 1500 iterations

Figure B.2: Low-pass filter: generated circuit topologies (100 – 1500 iterations)

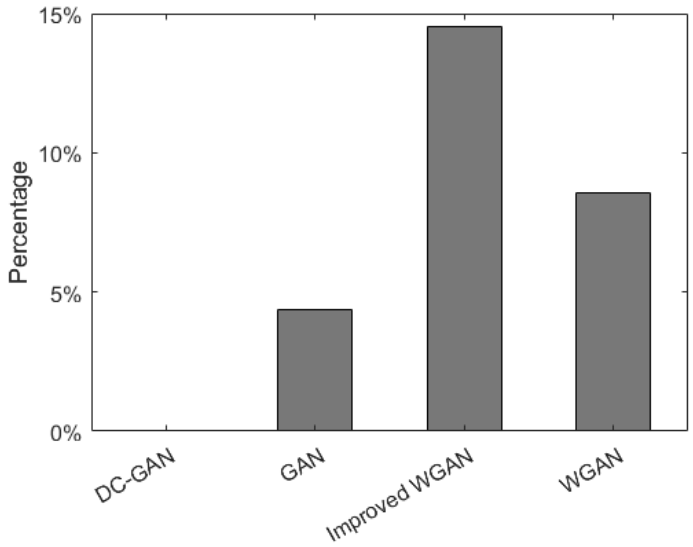



Figure B.3: Low-pass filter: GAN generation performance

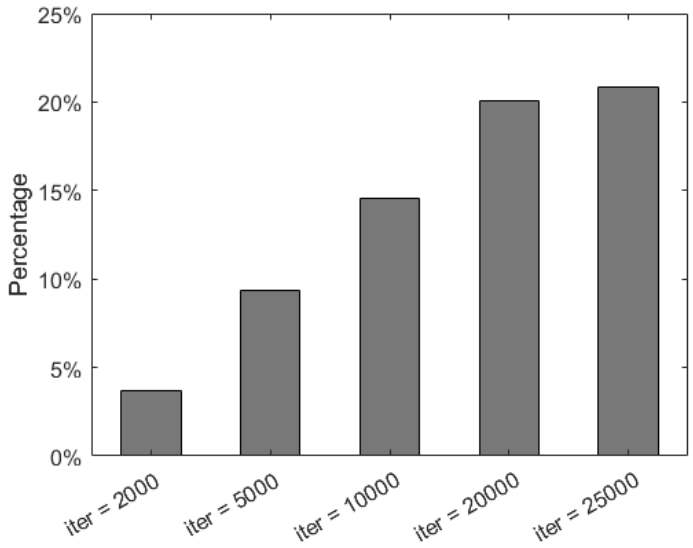



Figure B.4: Low-pass filter: improved WGAN generation performance