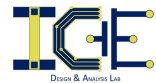


1

Emerging ML-AI Techniques for Analog EDA

Ioannis Savidis



1

Drexel University – Electrical & Computer Engineering

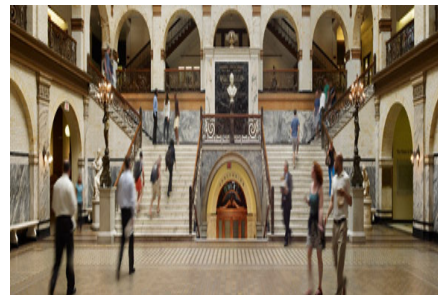
2



Drexel University:

- ✦ **Founded** in 1891 by financier and philanthropist Anthony J. Drexel
- ✦ **Location:** four campuses: 3 in Philadelphia, 1 in New Jersey (Mt. Laurel)
- ✦ **Student Enrollment:** 15,346 undergraduates 8,859 graduate and professional students
- ✦ **Student Geographic Distribution:** Students come from 50 U.S. states and 130 foreign countries. Nearly 8% are international students

Bossone building
(home to College of Engineering labs)



2

DREXEL UNIVERSITY - IOANNIS SAVIDIS GROUP

3



Degrees: B.S.E., Duke University
M.S., University of Rochester
Ph.D., University of Rochester (2013)

Research Interests

Analysis, modeling, and design methodologies for high performance digital and mixed-signal integrated circuits; Emerging integrated circuit technologies; Electrical and thermal modeling and characterization, signal and power integrity, and power and clock delivery for 3-D IC technologies; hardware security (obfuscation and side-channel analysis); algorithms and methodologies for design automation including ML/AI based optimization; On-chip power management; Low-power circuit techniques; Algorithms and methodologies for secure IC design

LABORATORY & TEAM

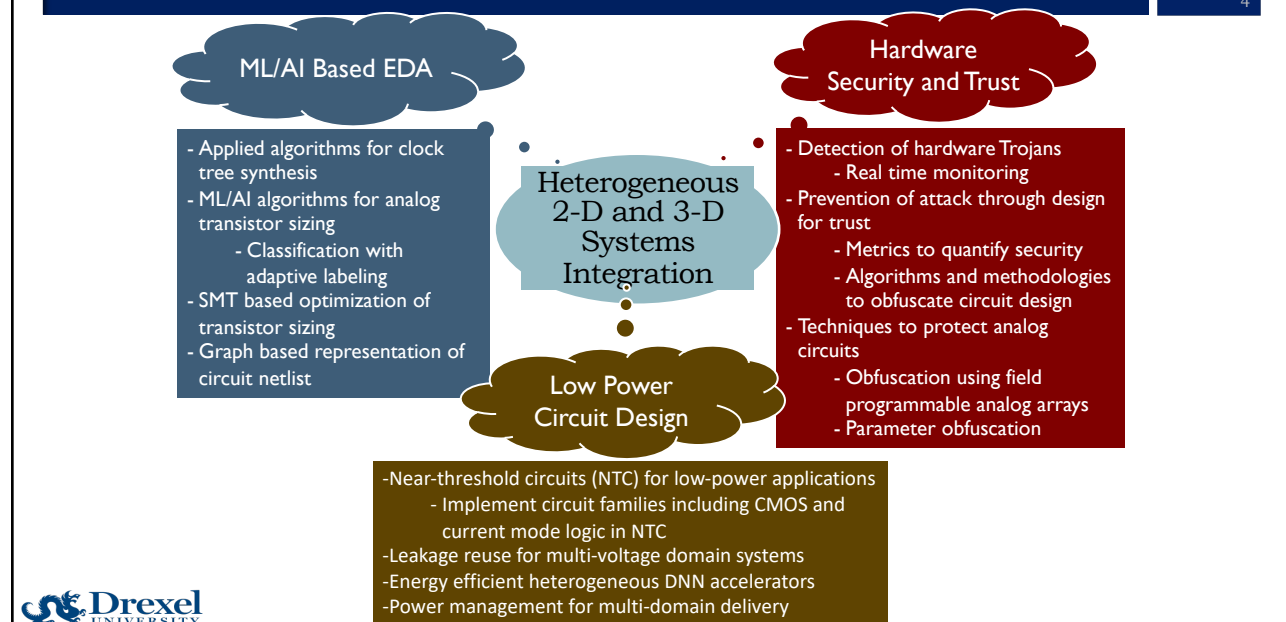
- **Seven Ph.D. students**
 - Alec Aversa – Sequential digital circuit obfuscation
 - Vaibhav Venugopal Rao – Analog IC IP protection
 - Saran Phatharodom – Digital obfuscation metrics
 - Jeff Wu – Application of ML/AI to analog IC design
 - Ziyi Chen – Analog IP protection
 - Ashish Sharma – Heterogeneous circuit integration
 - Pratik Shrestha – Digital security and application of ML/AI to digital IC design
- **One B.S. student**
 - Isabel Song (UPenn undergraduate) – ML/AI analog IC design
- **2,000 square feet of dedicated research space**
- **Access to leading CAD software packages:** Cadence (Virtuoso, Encounter, assure), Synopsys (Primetime, Hspice, Taurus), and Siemens Mentor Graphics (Calibre)



3

Drexel ICE Laboratory Research Overview

4



4

Outline of Presentation

5

- Background introduction
- Machine learning techniques for analog EDA
- Optimization techniques for analog EDA
- Case studies
- Conclusions



5

Outline of Presentation

6

- **Background introduction**
- Machine learning techniques for analog EDA
- Optimization techniques for analog EDA
- Case studies
- Conclusions

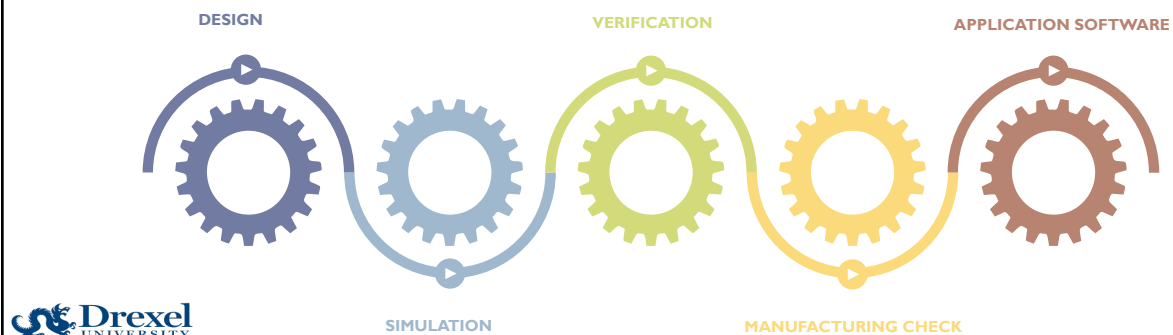


6

Background: Electronic Design Automation (EDA)

7

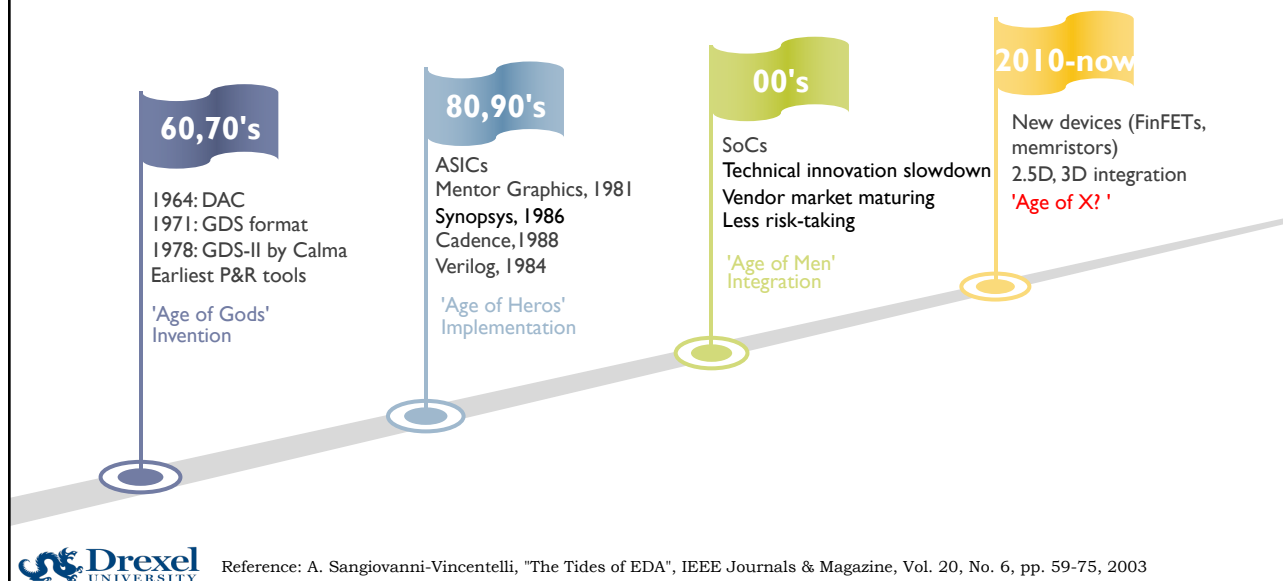
- EDA: a system of software solutions for the design of integrated circuits
- A wide range of applications:
 - High-performance Computing
 - Autonomous vehicle
 - IoT
 - AI
 - ...
- Primary Tools/Applications:



7

A Remembrance of the Past: Timeline of (Digital) EDA Development

8



8

Reference: A. Sangiovanni-Vincentelli, "The Tides of EDA", IEEE Journals & Magazine, Vol. 20, No. 6, pp. 59-75, 2003

Overview of Digital EDA

9

- ▶ Mature RTS-to-GDSII flow
 - ▶ Frontend: technology independent standardized design descriptions
 - ▶ Examples: VHDL, Verilog
 - ▶ Backend: physical implementation of circuits
 - ▶ Fabs provide libraries and simulation models for fab processes
- ▶ The high level of automation achieved in digital EDA results from **abstraction**
 - ▶ Programmability at high level
 - ▶ 'Divide and conquer' design strategy
 - ▶ Modular design for reusability

Functional Design

Logical Synthesis

Floorplanning

Placement

Clock Network
Synthesis

Routing

Signoff

- Parasitic Extraction
- Physical Verification
- Static Timing Analysis
- Power, IR-drop Analysis



9

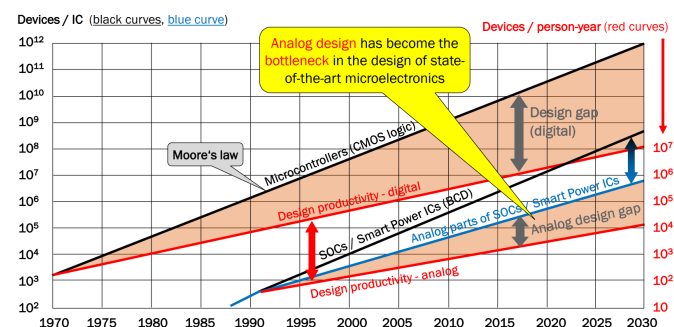
Background: Analog IC Design and Production

11

- ▶ Analog segment had the largest growth in sales for 2021 and 2022, even under different economic conditions
 - ▶ Big demand for analog chips
 - ▶ Less cyclical than logic and memory
- ▶ Analog design productivity lags behind digital design by orders of magnitude
 - ▶ Design productivity = number of devices integrated on a chip / required design effort in person-years
 - ▶ Results in long time to market
 - ▶ Significant design effort and experience required

Global Semiconductor Sales and Growth in 2021,2022

	Growth Rate	Sales (billion)
Analog	33.1%, 20.3%	74.0, 89.0
Logic	30.8%, 13.7%	154.8, 176.0
Memory	30.9%, -15.4%	153.8, 130.0



Reference: 1. Semiconductor Industry Association
 2. J. Scheible, "Optimized is Not Always Optimal,"
 Proceedings of the International Symposium on
 Physical Design, pp. 151–158, 2022

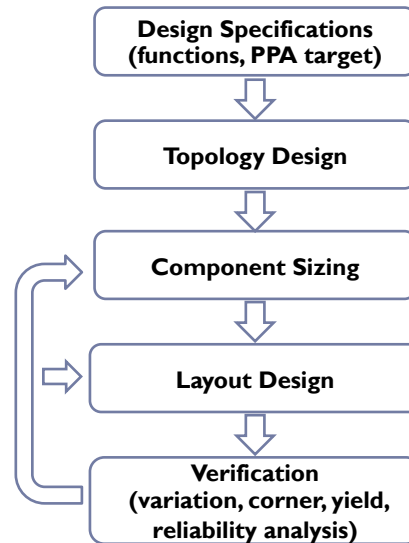


11

Typical Analog Synthesis Flow

12

- ▶ Top-down tasks
 - ▶ Specification formulation
 - ▶ Topology selection
 - ▶ Component sizing
 - ▶ Tuning the sizes of the devices in an analog circuit to meet design specifications
- ▶ Bottom-up tasks of layout design
 - ▶ Placement
 - ▶ Device grouping based on matching and symmetry
 - ▶ Determine coordinates of pins and devices in layout
 - ▶ Routing
 - ▶ Optimize routing under design constraints, e.g., area, design rules, matching
- ▶ Additional tasks:
 - ▶ Interconnect impedance prediction
 - ▶ Reduce the gap between schematic simulation results and post-layout results
 - ▶ Compensation for effects of PVT variations and reliability issues

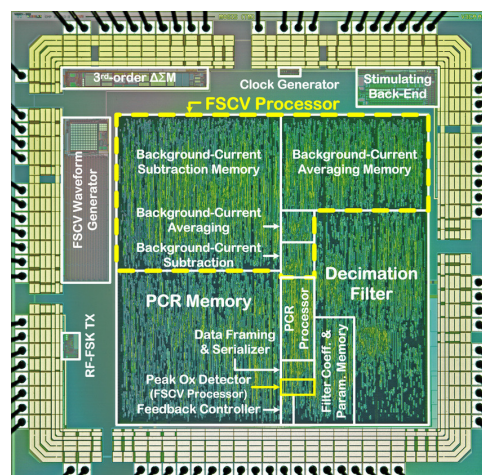


12

Summary and Motivation for Automated Analog Synthesis

13

- ▶ Significant opportunities for analog design
- ▶ Lack of productivity and design quality for analog EDA tools
- ▶ Increased circuit complexity with next-generation target application domains
 - ▶ Bio-inspired computing
 - ▶ 6G communications
 - ▶ Autonomous vehicles (cars, UAVs, etc.)
 - ▶ ...
- ▶ Challenges to automate analog design
 - ▶ Analog circuits are highly customized with various functionalities



13

Broad Categories of Analog ICs

14

Amplifiers

- Opamp
- OTA

Data Converters

- ADC
- DAC

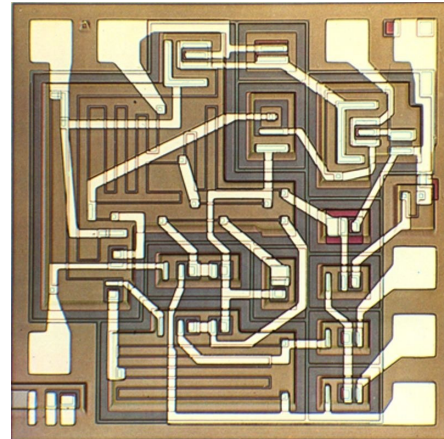
RF Transceivers

- LNA
- VCO
- Mixers
- PLL
- DLL
- PA

Filters

Reference Generators

- Crystal oscillator
- Bandgap references
- Biasing structures
- Clock generators and drivers



Layout of an $\mu A709$ operational amplifier, 1965
Credit: Fairchild Camera & Instrument Corporation

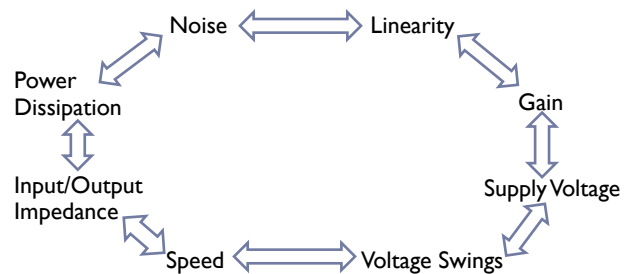


14

Summary and Motivation for Automated Analog Synthesis

15

- ▶ Immense opportunities for analog design
- ▶ Lack of productivity and design quality for analog EDA tools
- ▶ **Challenges to automate analog design**
 - ▶ Analog circuits are highly customized with various functionalities
 - ▶ **Additional design considerations beyond PPA**
 - **Example: typical considerations for the design of an op-amp:**



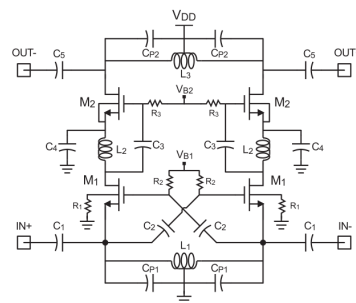
Reference: B. Razavi, "Design of Analog CMOS Integrated Circuits", McGraw-Hill, 2001

15

Summary and Motivation for Automated Analog Synthesis

16

- Immense opportunities for analog design
- Lack of productivity and design quality for analog EDA tools
- **Challenges to automate analog design**
 - Analog circuits are highly customized with various functionalities
 - Additional design considerations beyond PPA
 - **Complex and non-linear circuit behaviors resulting from physics**
 - **Design, performance, and process parameters all interrelated**
 - **Example: noise factor of a low-noise amplifier:**



$$F = 1 + \frac{R_l}{R_s} + \frac{R_g}{R_s} + \gamma X g_d R_s \left(\frac{\omega_0}{\omega_T} \right)^2$$



16

Does Digital Idea of Abstraction Apply to Analog EDA?

17

- Abstraction levels from low to high: device level $\Rightarrow$ sub-block level $\Rightarrow$ system level
- Past attempts
 - **Template-based 'standard' cells of analog blocks (amplifiers, comparators..)**
 - **Berkeley Analog Generator**



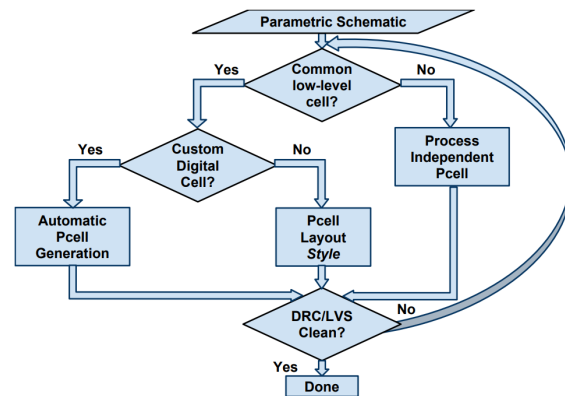
Reference: J. Crossley, et. al., "BAG: A Designer-oriented Integrated Framework For The Development Of AMS Circuit Generators," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 74–81, Nov. 2013

17

Template-based Analog Synthesis Tool: Berkeley Analog Generator (BAG)

18

- ▶ BAG follows pre-defined design procedures
 - ▶ Synopsys' Pycells as technology-independent parameterized layout cells
 - ▶ Functionality:
 - ▶ Technology characterization
 - ▶ Schematic and testbench translation
 - ▶ Simulator interfacing
 - ▶ Physical verification and extraction
 - ▶ Layout generation
 - ▶ Limitation:
 - ▶ Procedural design processes are inflexible
 - ▶ Require laborious setup and input by designers
 - ▶ Solutions are suboptimal



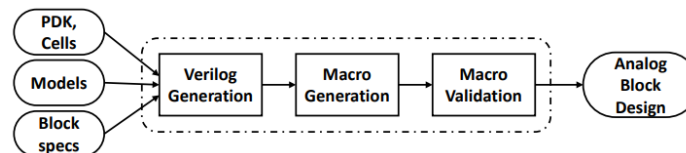
Reference: J. Crossley, et al., "BAG: A Designer-oriented Integrated Framework For The Development Of AMS Circuit Generators," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 74–81, Nov. 2013

18

Does Digital Idea of Abstraction Apply to Analog EDA?

19

- ▶ Abstract levels from low to high: device level $\Rightarrow$ sub-block level $\Rightarrow$ system level
- ▶ Past attempts
 - ▶ Template-based 'standard' cells of analog blocks (amplifiers, comparators..)
 - Berkeley Analog Generator
 - ▶ Verilog-AMS
 - FASoC: Fully-Autonomous SoC Synthesis using Customizable Cell-Based Synthesizable Analog Circuits
 - Applications: PLLs, power management, ADCs, and sensor interfaces
 - Limitation: only work for 'digital analog' circuits



Reference: Q. Zhang et al., "An Open-Source and Autonomous Temperature Sensor Generator Verified With 64 Instances in SkyWater 130 nm for Comprehensive Design Space Exploration," IEEE Solid-State Circuits Letters, Vol. 5, No. 1, pp. 174–177, 2022

19

Digitize AMS Circuits?

20

Digitally-assisted AMS design

- ▶ Examples
 - Calibration of nonlinearity of ADCs
 - Digital predistortion and noise shaping for DACs

Mostly-digital architectures

- ▶ Examples:
 - Data-converter-based transmitter and receiver
 - DPLL
 - DLDO

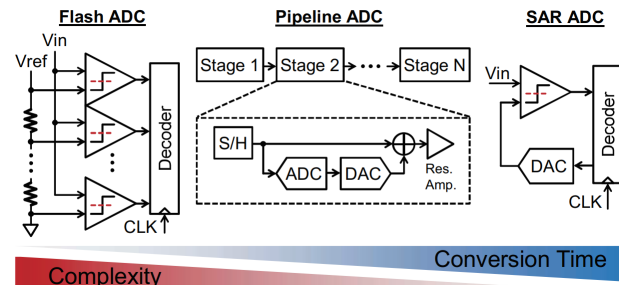
Digital-like AMS operations

- ▶ 5 GS/s time-to-digital (TDC) based on inverters and flip-flops only
- ▶ Filters, amplifiers implemented with digital standard cells

Advantage: auto synthesized with digital EDA flows

Drawback:

- ▶ Apply to mixed-signal circuits
- ▶ Does not apply to analog and RF circuits with high performance requirements
 - Customization needed



Reference: S. Su, et al., "Analog/Mixed-Signal Circuit Synthesis Enabled by the Advancements of Circuit Architectures and Machine Learning Algorithms," Proceedings of the Asia and South Pacific Design Automation Conference, pp.100–107, 2022

20

Does Digital Idea of Abstraction Apply to Analog EDA?

21

Abstract levels from low to high: device level $\Rightarrow$ sub-block level $\Rightarrow$ system level

Past attempts

- ▶ Template-based 'standard' cells of analog blocks (amplifiers, comparators, ...)
 - Berkeley Analog Generator
- ▶ Verilog-AMS
 - FASoC: Fully-Autonomous SoC Synthesis using Customizable Cell-Based Synthesizable Analog Circuits
 - PLLs, power management, ADCs, and sensor interfaces
 - Work for 'digital analog' circuits

Instead of attempting to standardize analog blocks or making analog circuits digital, focus on developing automated circuit design methodologies that allows customization



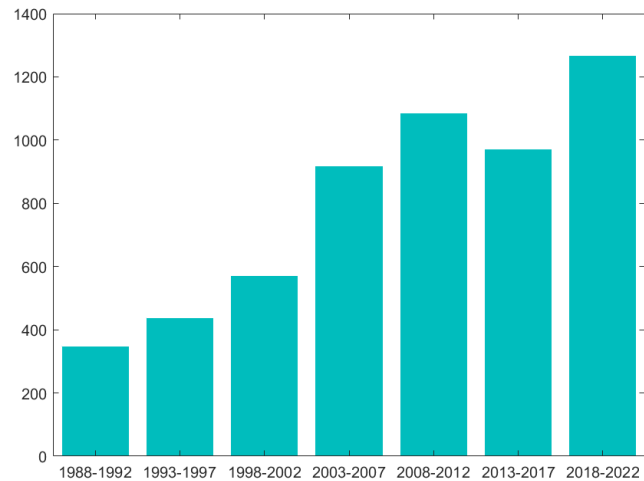
21

Interests in Analog EDA

22

- ▶ Papers with keyword of 'analog circuit design automation' increasing with time

Number of IEEE Papers with keyword 'analog circuit design automation'

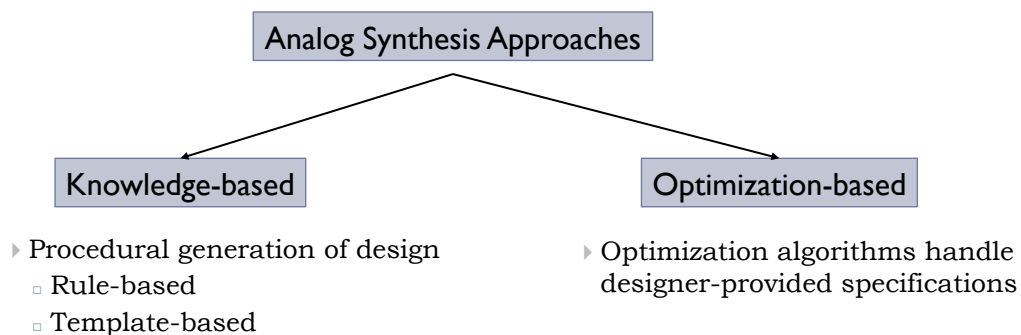


22

Overview of Heuristic Approaches for Analog Synthesis

23

- ▶ Design with analytical equations or rules set by human designers
- ▶ Top-down hierarchical design flow



23

Outline of Presentation

24

- ▶ Background introduction
- ▶ **Machine learning techniques for analog EDA**
 - ▶ Background on machine learning
 - ▶ Statistical learning algorithms
 - ▶ Neural-network-based learning algorithms
- ▶ Optimization techniques for analog EDA
- ▶ Case studies
- ▶ Conclusions



24

Outline of Presentation

25

- ▶ Background introduction
- ▶ **Machine learning techniques for analog EDA**
 - ▶ **Background on machine learning**
 - ▶ Statistical learning algorithms
 - ▶ Neural-network-based learning algorithms
- ▶ Optimization techniques for analog EDA
- ▶ Case studies
- ▶ Conclusions

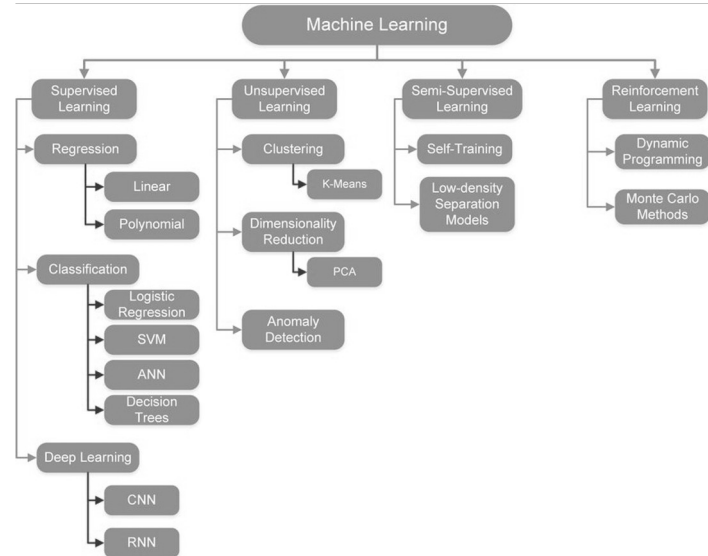


25

Potential Solution for Analog EDA: Machine Learning

26

- ▶ Machine learning: train models to learn from data
- ▶ Leverage information from data to improve performance on prediction and generation tasks
- ▶ Diverse algorithm choices



Reference: A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics," IEEE Access, Vol. 6, No.1, pp. 39117-39138, 2018

26

Machine Learning for Analog EDA

27

- ▶ ML and AI have been included in latest commercial EDA tools
 - ▶ Synopsys DSO.ai
 - ▶ Cadence Cerebrus
 - ▶ ...
- ▶ Additional capabilities provided by machine learning over heuristics
 - ▶ Design space exploration
 - ▶ Extract patterns of circuit parameters/characteristics from circuit data
 - ▶ Predict key metrics through the design stages
 - ▶ Guide the optimization/design process
 - ▶ Generate prototype designs
- ▶ ML for analog EDA in two key steps:

Learning

ML algorithms for circuit modeling

+

Optimization

Optimization algorithms for circuit design



27

Biggest Challenge of Applying ML to Analog EDA: Data

29

- ▶ Primary sources for circuit data:
 - ▶ Random sampling from the design space (design of experiments) with circuit simulators
 - Challenge:
 - Numerical circuit solvers are computationally costly
 - One execution run of system-level simulation may take hours or days
 - ▶ Expert designs generated by human designers or automation tools
 - Challenge:
 - Design IPs are often proprietary
 - Lack of benchmark circuits and standardized data format for analog EDA
- ▶ Vision/hope: open repository with production ready designs that are encrypted and secured but allow for ML design research by the community



29

The Biggest Challenge of Applying ML to Analog EDA: Data

30

- ▶ Primary sources for circuit data:
 - ▶ Random sampling from the design space (design of experiments) with circuit simulators
 - Challenge:
 - Numerical circuit solvers are computationally costly
 - One execution run of system-level simulation may take hours or days
 - ▶ Expert designs generated by human designers or automation tools
 - Challenge:
 - Design IPs are often proprietary
 - Lack of benchmark circuits and standardized data format for analog EDA

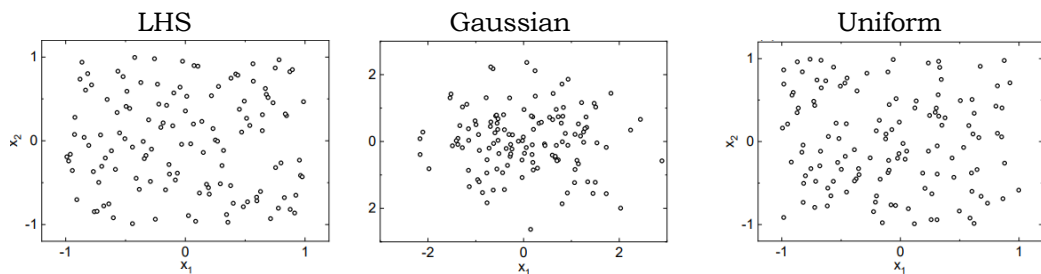


30

Sampling (Design of Experiments) for Circuit Data Generation

31

- ▶ Gaussian random sampling
- ▶ Uniform random sampling
- ▶ Latin hypercube sampling (LHS)
 - ▶ Divide design space into M equal intervals, sample from each interval
- ▶ Results show LHS performs best based on MSE and R-squared of models trained on moderate and large sample sizes



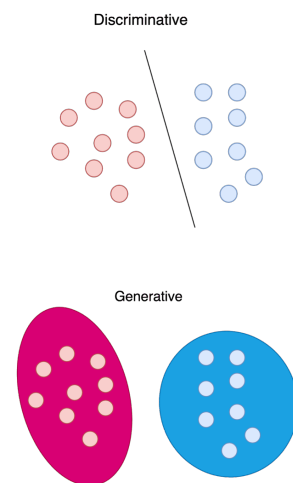
Reference: X. Shanguan, H. Ma, A. C. Cangellaris and X. Chen, "Effect of Sampling Method on the Regression Accuracy for a High-Speed Link Problem," Proceedings of the IEEE Conference on Electrical Performance of Electronic Packaging and Systems, pp.1-3, 2021

31

Data Generation for ML Model Training for Analog Circuits

32

- ▶ Two types of ML models for inputs x and label y
 - ▶ Discriminative models: learn mapping from x to y , i.e., $p(y|x)$
 - ▶ Generative: learn a distribution over data, i.e., $p(x, y)$
- ▶ Primary sources of circuit data
 - ▶ Random sampling from the design space (design of experiments)
 - ▶ Unbiased
 - ▶ Suitable for **discriminative** ML models
 - ▶ **Inappropriate for generative models because it is equivalent to learning from 'randomness'**
 - ▶ Past expert designs generated by human designers or prototype design automation tools
 - ▶ Biased
 - ▶ Suitable for **generative** ML models
 - ▶ **Inappropriate for discriminative models because data is 'biased' towards good design**



Source: theaisummer.com

32

Outline of Presentation

33

- Background introduction
- Machine learning techniques for analog EDA
 - Background on machine learning
 - Statistical learning algorithms
 - Neural-network-based learning algorithms
- Optimization techniques for analog EDA
- Case studies
- Conclusions



33

Statistical Learning Algorithms

35

- Statistical algorithms are predecessors of learning algorithms
- Algorithms widely utilized in EDA for decades:
 - Linear regression models
 - Gaussian process models
 - K-nearest neighbors
 - Support vector machines
 - Tree-based models



35

Statistical Learning Algorithms

36

- ▶ Statistical algorithms are predecessors of learning algorithms
- ▶ Algorithms widely utilized in EDA for decades:
 - ▶ Linear regression models
 - ▶ Gaussian process models
 - ▶ K-nearest neighbors
 - ▶ Support vector machines
 - ▶ Tree-based models



36

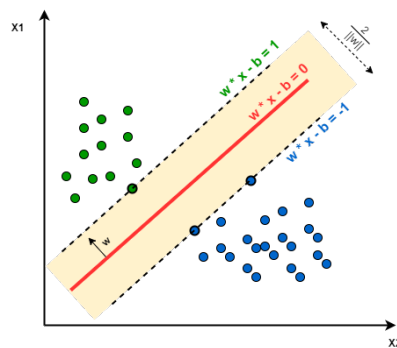
Linear Regression and Support Vector Machines

37

- ▶ For data $D = \{x, y\}$, a linear regression model is:

$$y_i = \sum \omega_j x_{ij} + N(0, \sigma^2)$$

- ▶ Support vector machine:
 - ▶ Decide a boundary with maximum distance from nearest points in each class



Source: baeldung.com

37

Statistical Learning Algorithms

38

- ▶ Statistical algorithms are predecessors of learning algorithms
- ▶ Algorithms widely utilized in EDA for decades:
 - ▶ Linear regression models
 - ▶ Gaussian process models
 - ▶ **K-nearest neighbors**
 - ▶ Support vector machines
 - ▶ Tree-based models



38

K-Nearest Neighbor Algorithm

39

- ▶ Classify based on distance between new data point and k nearest known data points
- ▶ Distance metrics:
 - ▶ Euclidean distance
 - ▶ Manhattan distance
 - ▶ Hamming distance
 - ▶ ...
- ▶ Requires storage of all data
 - ▶ 'Lazy learning'
- ▶ Advantages:
 - ▶ Easy to implement and adapt
 - ▶ Few hyperparameters
- ▶ Disadvantages:
 - ▶ Does not scale well to large dataset with high dimensionality
 - ▶ Bottleneck in memory



Source: IBM.com

39

Statistical Learning Algorithms

40

- ▶ Statistical algorithms are predecessors of learning algorithms
- ▶ Algorithms widely utilized in EDA for decades:
 - ▶ Linear regression models
 - ▶ Gaussian process models
 - ▶ K-nearest neighbors
 - ▶ Support vector machines
 - ▶ Tree-based models



40

Tree-based Models

41

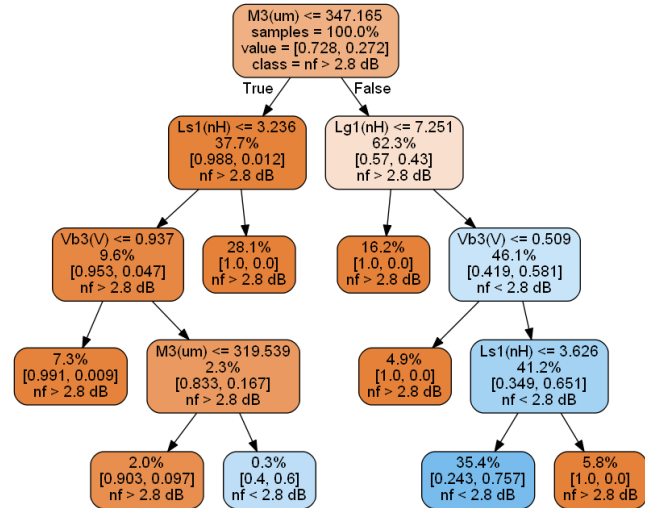
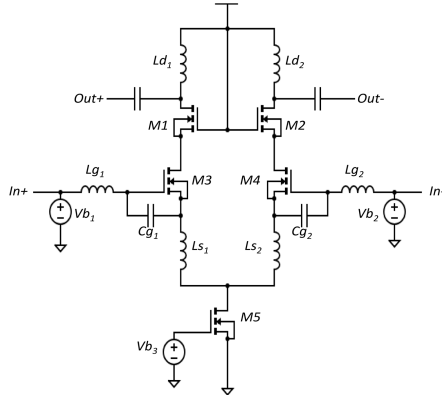
- ▶ Decision tree
 - ▶ Trained by maximizing information gain $\sum -p * \log_2 p_i$, where p is the probability of class i
- ▶ Ensemble of trees
 - ▶ Reduce overfitting resulting from a single tree
 - ▶ Random forest
 - ▶ For classification, return the class voted by most trees in ensemble
 - ▶ For regression, take the mean of predictions of all trees in ensemble
 - ▶ Gradient boost
 - ▶ Usually provides highest accuracy among tree-based models
- ▶ Advantages:
 - ▶ Data pre-processing not required
 - ▶ Fewer data required than neural networks
 - ▶ Interpretable tree structure provides additional design information
 - ▶ Allows ranking of feature importance



41

Application 1: Decision Tree to Predict Noise Figure Based on Sizing

- ▶ Device sizes are generated with LHS
- ▶ Performance (NF) is evaluated with SPICE
- ▶ Tree structure shows design space partitioning



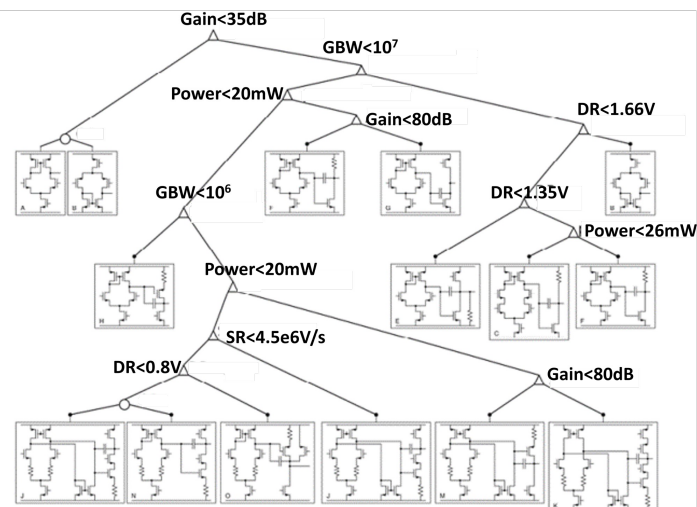
Reference: Z. Wu and I. Savidis, "CALT: Classification with Adaptive Labeling Thresholds for Analog Circuit Sizing," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 49–54, 2020

42

Application 2: Decision Tree to Map From Specifications To Topology

43

- ▶ Data is generated from MOJITO
 - ▶ MOJITO is an analog sizing platform that
 - ▶ Optimizes across thousands of analog circuit topologies
 - ▶ Returns a set of sized topologies with performance tradeoffs
- ▶ Trained tree model effectively provides topology selection rules for the target technology



Reference: I. T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Automated Extraction Of Expert Knowledge In Analog Topology Selection And Sizing," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 392-395, 2008

2. T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Simultaneous Multi-Topology Multi-Objective Sizing Across Thousands of Analog Circuit Topologies," Proceedings of the ACM/IEEE Design Automation Conference, pp. 944-947, 2007

43

Outline of Presentation

44

- Background introduction
- Machine learning techniques for analog EDA
 - Background on machine learning
 - Statistical learning algorithms
 - Neural-network-based learning algorithms
- Optimization techniques for analog EDA
- Case studies
- Conclusions



44

Artificial Neural Networks (ANNs)

46

- Fundamental reason for success of AI in the recent decade:
 - Boost in algorithmic power: novel algorithms based on generalized neural network architecture
 - Boost in computing power (which again benefits from the development of IC, EDA)
- Deep neural networks
 - Depth is determined based on problem complexity and dataset size
 - Layers represent different levels of abstraction

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}



Reference: T. Brown, et. al, 'Language Models are Few-Shot Learners', Advances in Neural Information Processing Systems, Vol. 33, No. 1, pp.1877-1901, 2022

46

Learning Scenarios Powered by Variants of ANNs

47

Differentiate by training scheme

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder



Differentiate by data format

- Multi-layer perceptrons (feedforward neural networks)
 - Tabular data
- Convolutional neural networks
 - Image data
- Graph neural networks
 - Graph-structured data
- Recurrent neural networks
 - Time-series data

Can combine any option from the left with any option from the right
 Example: train a recurrent neural network in an adversarial approach
 train a graph neural network in an unsupervised approach



47

Learning Scenarios Powered by Variants of ANNs

48

Differentiate by training scheme

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder



Differentiate by **data format**

- Multi-layer perceptrons (feedforward neural networks)
- Convolutional neural networks
- Graph neural networks
- Recurrent neural networks



Apply specialized filters for the input format



48

Learning Scenarios Powered by Variants of ANNs

49

Differentiate by training scheme

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder

X

Differentiate by **data format**

- **Multi-layer perceptrons (feedforward neural networks)**
- Convolutional neural networks
- Graph neural networks
- Recurrent neural networks



49

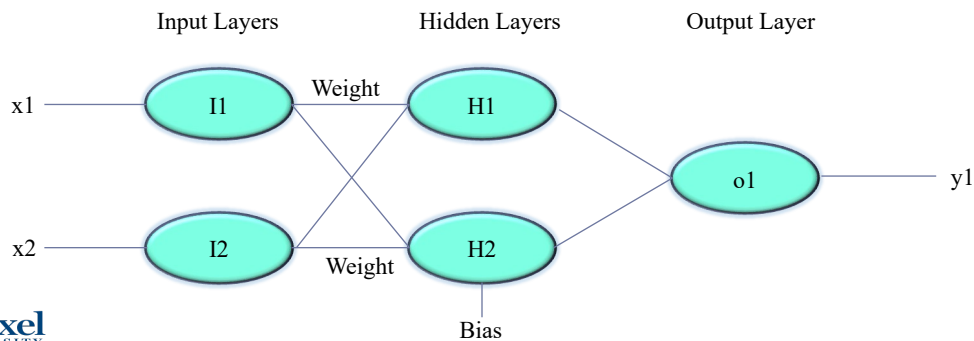
Multi-layer Perceptron (MLP) ANNs

50

- Each input neuron represents a piece of the data (image pixel, transistor feature...)
- Each neuron performs logistic regression:

$$h_{w,b}(x) = f(\omega^T x + b)$$

- A neural network maps from input neurons to output labels
 - Works well for tabular data

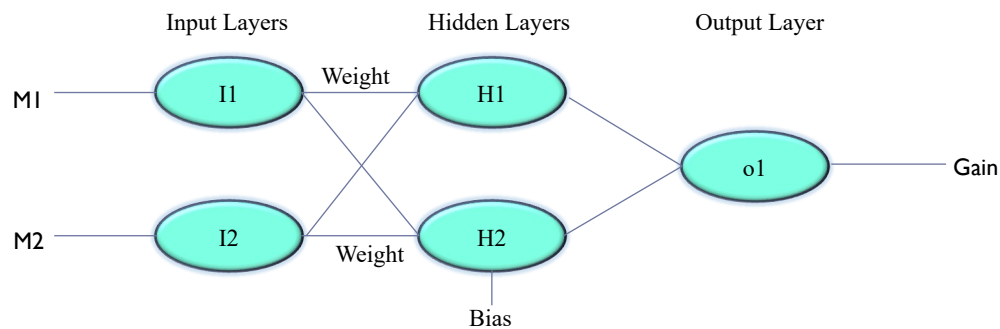


50

Example: Problem Formulation For A Circuit Sizing Task

51

- Objective: predict AC gain of an amplifier based on sizes of two transistors
- Intuitive problem formulation:
 - Input features: two transistor sizes, M1, M2
 - Output label: real-valued SPICE evaluation of gain
 - After data generation, train a simple MLP to map from device sizes to gain values



51

Learning Scenarios Powered by Variants of ANNs

52

Differentiate by training scheme

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder

X

Differentiate by **data format**

- Multi-layer perceptrons (feedforward neural networks)
- **Convolutional neural networks**
- Graph neural networks
- Recurrent neural networks



52

Image Data? Train Convolutional Neural Networks

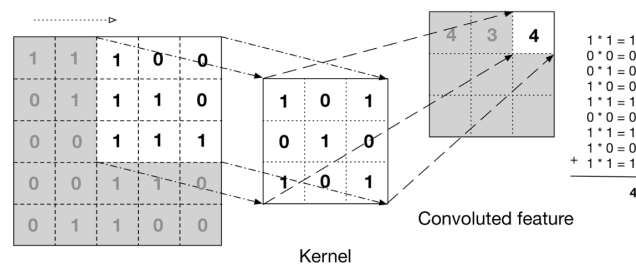
53

▶ Circuit data in image format:

- ▶ Physical design
 - ▶ Layout in GDSII
- ▶ View of a gate netlist
- ▶ Photos of fabricated PCBs
- ▶ ...

▶ Convolutional neural networks:

- ▶ Each pixel is updated by the weighted sum of current pixel value and neighboring pixel values
- ▶ Mathematically, apply convolution between kernel and pixel values



Source: analyticsvidhya.com

53

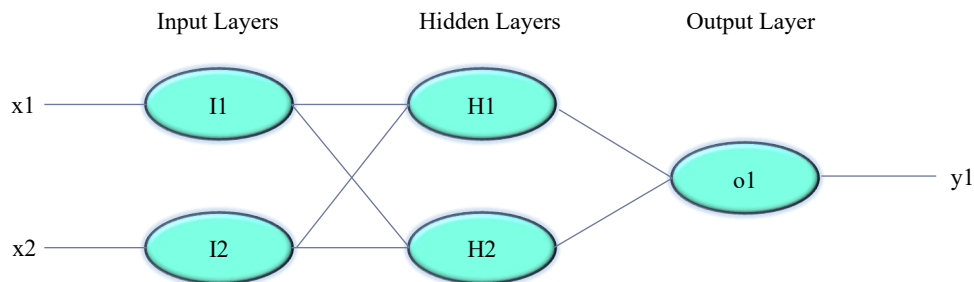
Multi-layer Perceptron (MLP) ANNs

54

- ▶ Each input neuron represents a piece of the data (image pixel, transistor feature...)
- ▶ Each neuron performs logistic regression:

$$h_{w,b}(x) = f(w^T x + b)$$

- ▶ A neural network maps from input neurons to output labels
 - ▶ Works well when data is tabular



Have We Missed Something With This Approach Of Circuit Modeling?

54

Learning Scenarios Powered by Variants of ANNs

55

Differentiate by training scheme

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder

X

Differentiate by **data format**

- Multi-layer perceptrons (feedforward neural networks)
- Convolutional neural networks
- **Graph neural networks**
- Recurrent neural networks



55

Circuit As a Graph: Graph Neural Network for Modeling of Analog Circuits

56

- Features engineering is critical to the performance of ML models
 - Diverse and arbitrary selection of features depending on designers' choices
- An analog circuit is modeled as a graph
 - Connectivity between devices (netlist information) is an important feature of a circuit
 - Often overlooked when utilizing MLPs and CNNs
 - Manually crafting circuit features based on connectivity is challenging
- **Solution: graph neural networks (GNNs)**
 - Automatically generate feature representations based on device connectivity
 - Additional topological information improves prediction accuracy

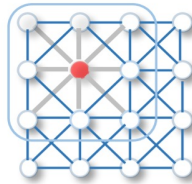


56

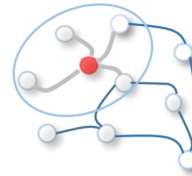
Graph Neural Networks

57

- ▶ Convolutional neural networks
 - ▶ Each pixel is a node
 - ▶ An image is a regular-shaped grid
 - ▶ Take a weighted average of node features (pixel values) along with neighboring node features



- ▶ Graph neural networks
 - ▶ Graph = (Vertices, Edges)
 - ▶ A graph is often irregular shaped
 - ▶ Objective is to combine features of the current node and neighboring node features
 - ▶ Similar to CNNs



GNNs are generalized versions of CNNs

Circuit applications: instead of training based on numerical features of a circuit, learn based on circuit graphs with numerical features associated with devices or entire circuits



Reference: Wu, Zonghan et al. "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 1, pp. 4-24, 2021

57

Vanilla Graph Convolutional Neural Networks (GCN)

58

Three primary steps:

- 1) Generate embeddings h_i for each graph node i
 - ▶ Map original feature vectors to vectors of a fixed dimension via linear neural network layers

Node embedding layers



58

Vanilla Graph Convolutional Neural Networks (GCN)

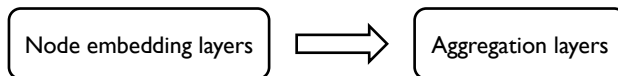
59

Three primary steps:

- 1) Generate embeddings h_i for each graph node i
 - Map original feature vectors to vectors of a fixed dimension via linear neural network layers
- 2) Aggregate embeddings of each node with embeddings of the neighboring nodes

$$h_i^{(l+1)} = \sigma(b^{(l)} + \sum_{j \in N(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)})$$

New embedding of node i Bias vector Index set of direct neighboring nodes of node i Nonlinear activation function Normalization coefficient Weight vector Current embedding of node j



59

Vanilla Graph Convolutional Neural Networks (GCN)

60

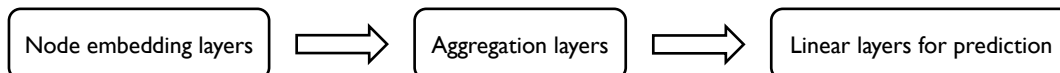
Three primary steps:

- 1) Generate embeddings h_i for each graph node i
 - Map original feature vectors to vectors of a fixed dimension via linear neural network layers
- 2) Aggregate embeddings of each node with embeddings of the neighboring nodes

$$h_i^{(l+1)} = \sigma(b^{(l)} + \sum_{j \in N(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)})$$

New embedding of node i Bias vector Index set of direct neighboring nodes of node i Nonlinear activation function Normalization coefficient Weight vector Current embedding of node j

- 3) Send final embeddings to linear neural network layers for target predictions



60

Variants of Graph Neural Networks

61

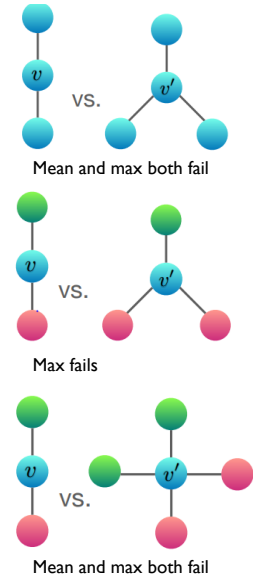
- ▶ GNN variants such as GCN and GraphSAGE fail to distinguish certain simple graph structures with mean and max pooling
 - ▶ Adopt sum-pooling

- ▶ Graph Isomorphism Networks (GIN):

$$h_v^{(l)} = \text{MLP}^*((1 + \epsilon^{(l)})h_v^{(l-1)}) + \sum_{u \in N(v)} h_u^{(l-1)}$$

importance of the center node relative to the neighboring nodes

- ▶ GIN generates distinguishable embeddings for different graph structures



Reference: J. K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?," Proceedings of the International Conference on Learning Representations, pp. 1–17, 2019

61

Challenges in Applying GNNs to Model Analog Circuits

62

- ▶ Lack of customized graph representation and learning algorithms to model analog circuits at the transistor level
 - ▶ Accounting for different edge types
 - ▶ Accounting for circuit hierarchies
- ▶ Hypergraphs are more suited to represent circuit nets than traditional graphs
 - ▶ In a traditional graph, an edge only connects a pair of nodes
 - ▶ In a hypergraph, a hyper-edge connects any number of nodes
- ▶ However, most GNN algorithms are designed for traditional graphs



62

Challenges in Applying GNNs to Model Analog Circuits

63

- ▶ Lack of customized graph representation and learning algorithms to model analog circuits at the transistor level
 - ▶ Accounting for different edge types
 - ▶ Accounting for circuit hierarchies
- ▶ Hypergraphs are more suited to represent circuit nets than traditional graphs
 - ▶ In a traditional graph, an edge only connects a pair of nodes
 - ▶ In a hypergraph, a hyper-edge connects any number of nodes
- ▶ However, most GNN algorithms are designed for traditional graphs



63

Edge-conditioned Convolution for Transistor-level Circuit Graph Modeling

64

- ▶ Each transistor as a graph node
- ▶ Six edge types defined to represent possible connections between transistors
 - ▶ Drain-drain
 - ▶ Gate-gate
 - ▶ Source-source
 - ▶ Drain-source
 - ▶ Gate-source
 - ▶ Drain-gate
- ▶ Edge-conditioned convolution:
 - ▶ Step1: aggregate node embeddings based on each edge type (one MLP for each edge type)
 - ▶ Step2: sum all individual edge-dependent embeddings

$$h_v^{(l+1)} = h_v^{(l)} W_0 + \sum_{u \in \mathcal{N}(v)} h_u^{(l)} \text{MLP}(e_{u \rightarrow v}) + b$$



Reference: M. Simonovsky and N. Komodakis, "Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs," Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 29–38, Jul. 2017

64

Challenges in Applying GNNs to Model Analog Circuits

65

- ▶ Lack of customized graph representation and learning algorithms to model analog circuits at the transistor level
 - ▶ Accounting for different edge types
 - ▶ Accounting for circuit hierarchies
- ▶ Hypergraphs are more suited to represent circuit nets than traditional graphs
 - ▶ In a traditional graph, an edge only connects a pair of nodes
 - ▶ In a hypergraph, a hyper-edge connects any number of nodes
- ▶ However, most GNN algorithms are designed for traditional graphs

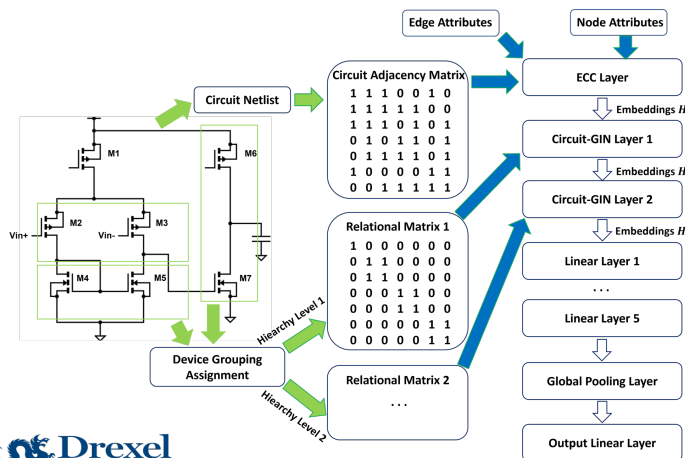


65

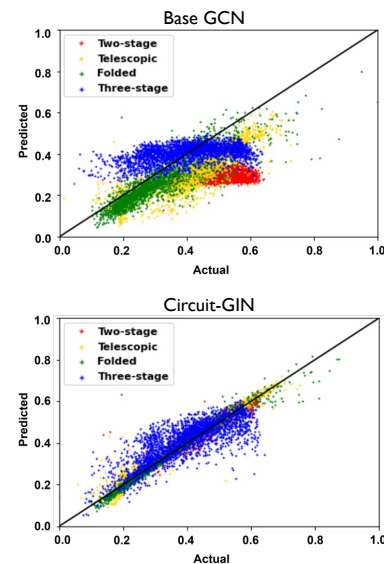
Graph Convolution Based on Circuit Hierarchies

66

- ▶ Model consists of two ECC layers and a Circuit-GIN layer:
 - ▶ Circuit-GIN: GIN on a relational graph based on circuit hierarchy
- ▶ Circuit-GIN model outperforms base GCN model by distinguishing between four op-amp topology graphs by up to 16.7% in R-squared



66



Choosing between CNN and GNN for Modeling of Analog Circuits

67

- Physical layout is usually better modeled with images
 - Geometric information included
 - Device coordinates
 - Pin coordinates
 - Instance orientations
 - ...
- Circuit topology is better represented with graphs
 - Graph isomorphism
 - Focus on topological connectivity information



67

Learning Scenarios Powered by Variants of ANNs

68

Differentiate by **training scheme**

- **Supervised**
- **Semi-supervised**
- **Unsupervised**
- Adversarial (GAN)
- Reinforcement
- Encoder-decoder

X

Differentiate by data format

- Multi-layer perceptrons (feedforward neural networks)
- Convolutional neural networks
- Graph neural networks
- Recurrent neural networks



68

Learning Scenarios Based On Availability of Training Labels

69

- ▶ Supervised learning
 - ▶ All training labels are available
 - ▶ Learning tasks
 - ▶ Classification
 - ▶ Regression
- ▶ Unsupervised learning
 - ▶ No labels are provided for the training set
 - ▶ Learn inherent distribution
 - ▶ Learning tasks
 - ▶ Clustering
 - ▶ Anomaly detection
- ▶ Semi-supervised learning
 - ▶ Labels are provided for a part of the training set



69

Learning Scenarios Powered by Variants of ANNs

70

Differentiate by **training scheme**

- ▶ Supervised
- ▶ Semi-supervised
- ▶ Unsupervised
- ▶ **Adversarial (GAN)**
- ▶ Reinforcement
- ▶ Encoder-decoder

X

Differentiate by data format

- ▶ Multi-layer perceptrons (feedforward neural networks)
- ▶ Convolutional neural networks
- ▶ Graph neural networks
- ▶ Recurrent neural networks

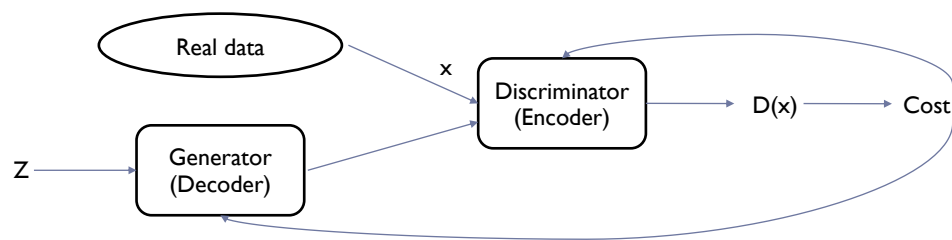


70

Generative Adversarial Networks

71

- Discriminator: $\max V(D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$
 - recognize real data
 - recognize generated data
- Generator objective function: $\min V(G) = \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$
 - optimize to fool discriminator
- Variants of GANs differ in objective function
- Suitable for prototype circuit design (layout) generation



Reference: I. Goodfellow, et al., "Generative Adversarial Nets", Proceedings of the International Conference on Neural Information Processing System, Vol. 2, No. 1, pp., 2672-2680, 2014

71

Practical Issues of Applying GAN

72

- Challenges in Training GANs
 - Mode collapse: limited varieties of samples are generated
 - Diminished gradient
 - Non-convergence
 - Overfitting **caused** by imbalance between the generator and discriminator
 - Highly sensitive to hyperparameters
- GAN for analog design generation
 - Difficult to be applied on tabular data
 - Suitable for image data
 - Placement images
 - Routing images



72

Learning Scenarios Powered by Variants of ANNs

73

Differentiate by **training scheme**

- Supervised
- Semi-supervised
- Unsupervised
- Adversarial (GAN)
- Reinforcement
- **Encoder-decoder**

X

Differentiate by data format

- Multi-layer perceptrons (feedforward neural networks)
- Convolutional neural networks
- Graph neural networks
- Recurrent neural networks

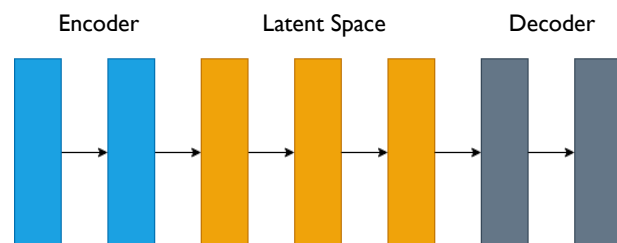


73

Variational Auto-Encoders (VAEs)

74

- Generative algorithm to encode distribution of training data, then generate new data with similar distribution
- Encoder: map input to a low-dimensional latent space
 - Effectively dimensionality reduction
- Decoder: convert signal in latent space back to input space
- Difference from GAN:
 - GAN generator takes noise as input
 - Higher-quality generation
 - Harder to train
 - VAE takes signal from the low-dimensional latent space as input
 - Lower-quality generation
 - Easier to train



Reference: D. Kingma and M. Welling, "Auto-Encoding Variational Bayes", Proceedings of the International Conference on Learning Representations, 2014

74

Outline of Presentation

75

- Background introduction
- Machine learning techniques for analog EDA
- **Optimization techniques for analog EDA**
 - Gradient-based algorithms
 - Heuristic algorithms
 - Learning-based algorithms
- Case studies
- Conclusions



75

Machine Learning for Analog EDA

76

- Additional capabilities provided by machine learning for analog EDA over heuristics
 - Design space exploration
 - Extract patterns of circuit parameters/characteristics from circuit data
 - Predict key metrics through the design stages
 - Guide the optimization/design process
 - Generate circuit designs
- ML for analog EDA in two key steps:

Learning

ML algorithms for circuit modeling

+

Optimization

Optimization algorithms for circuit design



76

Optimization Algorithms for EDA

77

▸ Gradient-based algorithms

▸ Heuristic algorithms

- Greedy algorithms
- Divide and conquer
- Dynamic programming
- Network flow algorithms
- Linear/integer programming
- Evolution-based algorithms
- Simulated annealing

▸ Learning-based algorithms

- Reinforcement learning
- Surrogate-assisted optimization algorithms

▸ General flow of optimization process

Require: Objective function f
 $x^{(0)} \leftarrow$ random point in the domain of f
for $i = 1, 2, \dots$ **do**
 $\Delta x \leftarrow \pi(f, \{x^{(0)}, \dots, x^{(i-1)}\})$
 if stopping condition is met **then**
 return $x^{(i-1)}$
 end if
 $x^{(i)} \leftarrow x^{(i-1)} + \Delta x$
end for



Reference: K. Li, J. Malik, "Learning to optimize," Proceedings of the International Conference on Learning Representations, 2017

77

Requirements on Optimization Algorithms for EDA

78

- Most optimization problems in physical design are NP-hard
- Optimization algorithms for EDA must have low time and space complexities, especially for physical design
 - When device count exceeds 100K, quadratic algorithms fail
 - Less of a problem for analog since device count is relatively small
- In physical design, key is to develop practical algorithms
 - Trade-off with optimality guarantee
- For an analog circuit, multi-objective optimization is required for multiple performance parameters
 - Trade-off between parameters (Pareto front) needed



Reference: N.A. Sherwani, "Algorithms for VLSI Physical Design Automation", Springer, 1995

78

Outline of Presentation

79

- ▶ Background introduction
- ▶ Machine learning techniques for analog EDA
- ▶ Optimization techniques for analog EDA
 - ▶ Gradient-based algorithms
 - ▶ Heuristic algorithms
 - ▶ Learning-based algorithms
- ▶ Case studies



79

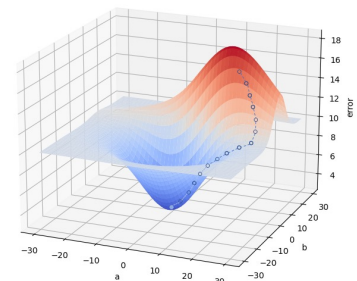
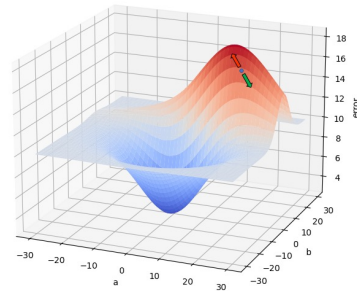
Gradient-based Optimization Algorithms

80

- ▶ At each step, search direction is defined by the gradient of the function evaluation
 - ▶ Intuitively, search along direction that reduces cost function at fastest rate
- ▶ Gradient descent
 - ▶ First-order search of local optimum

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

- ▶ Advantages:
 - ▶ Theoretical guarantee of optimality
 - ▶ Fast execution for each iteration of search
- ▶ Limitation:
 - ▶ Requires explicit differentiable functions
 - ▶ Slow convergence and local minima for non-convex search space



Source: Interactivechaos.com

80

Outline of Presentation

81

- Background introduction
- Machine learning techniques for analog EDA
- **Optimization techniques for analog EDA**
 - Gradient-based algorithms
 - **Heuristic algorithms**
 - Learning-based algorithms
- Case studies



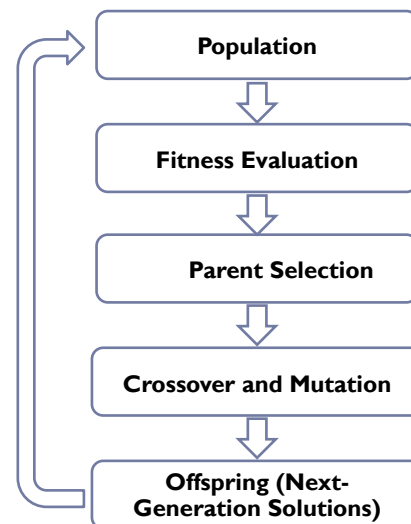
81

Evolutionary Algorithms

82

- Characteristics of evolutionary algorithms
 - Population-Based
 - Fitness-Oriented
 - A fitness parameter to represent the quality of a solution
 - Variation-Driven
 - Crossover and mutation generates variants randomly
- Broad categories
 - Genetic algorithm
 - Particle swarm
 - Differential evolution
 - ...

General flow of evolutionary algorithms



82

Simulated Annealing

83

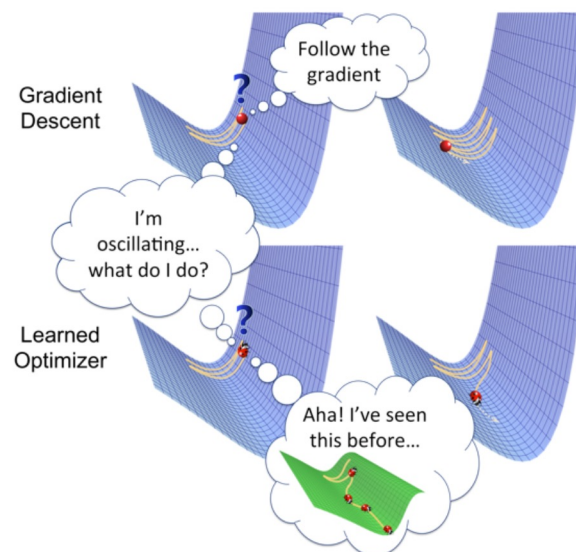
- A global optimization technique by approximation
- Preferred when search space is discrete
- Parameter: a temperature value that keeps decreasing
 - $\text{temperature} = \text{initial temperature} / (\text{iteration} + 1)$
- Steps:
 - Randomly initialize design variables and evaluate function value f_{old}
 - Sample again in the neighboring region and evaluate function value f_{new}
 - Action:
 - If function value improves, accept
 - If function value worsens, accept with probability $e^{-(f_{new} - f_{old}) / \text{temperature}}$
- Advantage:
 - Reduces chance of getting stuck at local optimum since worse candidates are accepted with certain probability



83

Reinforcement Learning for Optimization

84



Reference: K. Li, J. Malik, "Learning to optimize," Proceedings of the International Conference on Learning Representations, 2017

84

Outline of Presentation

85

- Background introduction
- Machine learning techniques for analog EDA
- Optimization techniques for analog EDA
 - Gradient-based algorithms
 - Heuristic algorithms
 - Learning-based algorithms
- Case studies



85

Reinforcement Learning (RL) for Optimization

86

- Define an agent to interact with the blackbox design space
 - Take action
 - Receive feedback
- Ingredients:
 - A state space: $\mathcal{S} \subseteq \mathbb{R}^D$
 - An action space: $\mathcal{A} \subseteq \mathbb{R}^d$
 - A cost function: $c : \mathcal{S} \rightarrow \mathbb{R}$
 - A time horizon: T
 - An initial state probability distribution: $p_i(s_0)$
 - A state transition probability distribution $p(s_{t+1} | s_t, a_t)$
- Objective: pick action (policy) $\pi(a_t | s_t, t)$ that maximizes expected cumulative rewards

Algorithm 1 General structure of optimization algorithms

Require: Objective function f

$x^{(0)} \leftarrow$ random point in the domain of f

for $i = 1, 2, \dots$ **do**

$\Delta x \leftarrow \phi(\{x^{(j)}, f(x^{(j)}), \nabla f(x^{(j)})\}_{j=0}^{i-1})$

if stopping condition is met **then**

return $x^{(i-1)}$

end if

$x^{(i)} \leftarrow x^{(i-1)} + \Delta x$

end for

Gradient Descent	$\phi(\cdot) = -\gamma \nabla f(x^{(i-1)})$
Momentum	$\phi(\cdot) = -\gamma \left(\sum_{j=0}^{i-1} \alpha^{i-1-j} \nabla f(x^{(j)}) \right)$
Learned Algorithm	$\phi(\cdot) = \text{Neural Net}$



Reference: K. Li, J. Malik, "Learning to optimize," Proceedings of the International Conference on Learning Representations, 2017

86

RL Formulation for Optimization on Analog Circuits

87

- *Time horizon* = number of controllable design parameters in the circuit
- Action: decide on the value of one design parameter at each step
- After T steps, a set of candidate solution is generated
- Evaluate the solutions from a circuit solver (e.g., SPICE), calculate reward
- Maximize expected cumulative rewards = optimize target
action space = feasible solution set for the design variables

Circuit applications:

- For placement optimization, move one instance each time
 - Instances: standard cells and macros for digital, devices or circuit blocks for analog
- For analog sizing, set figure of merit as optimization target
 - Action at each step is to determine the size of one transistor



Reference: A. F. Budak, Z. Jiang, K. Zhu, A. Mirhoseini, A. Goldie and D. Z. Pan, "Reinforcement Learning for Electronic Design Automation: Case Studies and Perspectives," Proceedings of the Asia and South Pacific Design Automation Conference pp.500-505, 2022

87

Surrogate-assisted Blackbox Optimization

88

- **Two primary steps:**
 - **Step 1: Surrogate modeling**
 - Gaussian process models
 - Neural networks
 - SVMs
 - ...
 - **Step 2: Active querying (adaptive sampling)**
 - Objective: sample points with maximum utility at minimum cost
 - Cost: number of points, simulation time
 - Methods:
 - Uncertainty sampling
 - Entropy
 - Information gain
 - Maximize KL divergence between posterior and prior
 - Query by committee
 - Response surface method



88

Bayesian Optimization

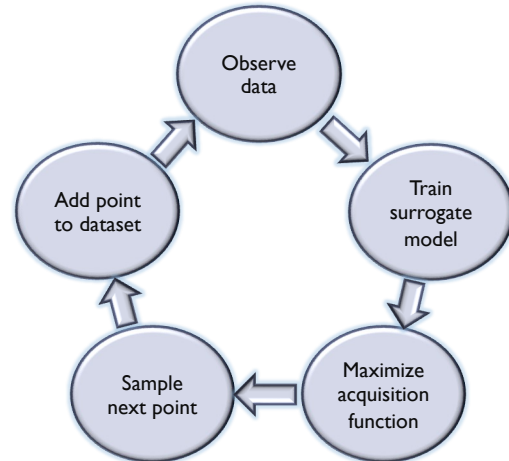
89

- ▶ A type of surrogate-assisted blackbox optimization technique
- ▶ Gaussian process function as surrogate model
- ▶ Acquisition function: determine next point to query while balancing exploitation and exploration of the search space
 - ▶ Expected improvement (EI)
 - ▶ Maximize expected improvement over current best value
- ▶ Upper confidence bound (UCB)
 - ▶ Search areas with either best function value or largest uncertainty

$$u(x) = \max(0, f' - f(x))$$

Current minimal value

$$\alpha(x; \lambda) = \underbrace{\mu(x)}_{\text{Exploitation}} + \underbrace{\lambda \sigma(x)}_{\text{Exploration}}$$



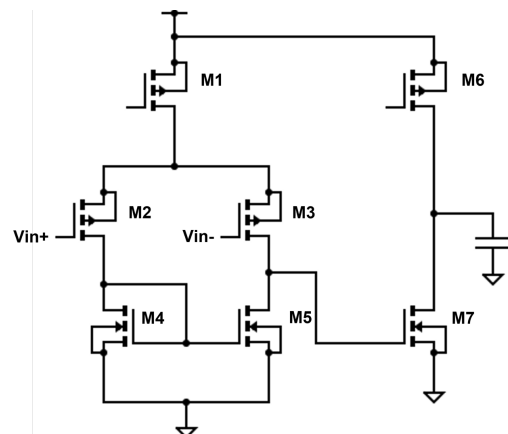
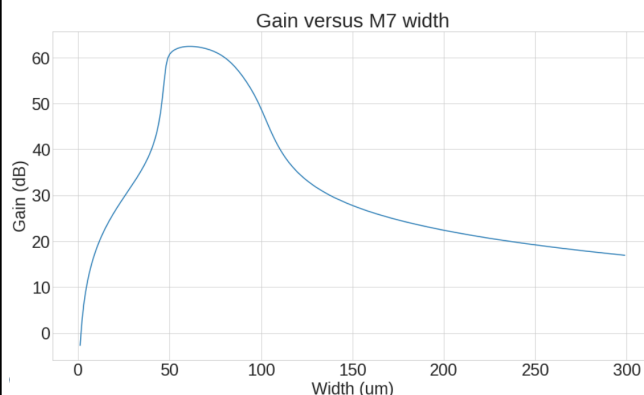
Reference: W. Lyu, et al., "An Efficient Bayesian Optimization Approach For Automated Optimization Of Analog Circuits," IEEE Transactions on Circuits and Systems, Vol. 65, No. 6, pp. 1954–1967, Jun. 2018

89

Example: Bayesian Optimization For the Sizing of An Amplifier

90

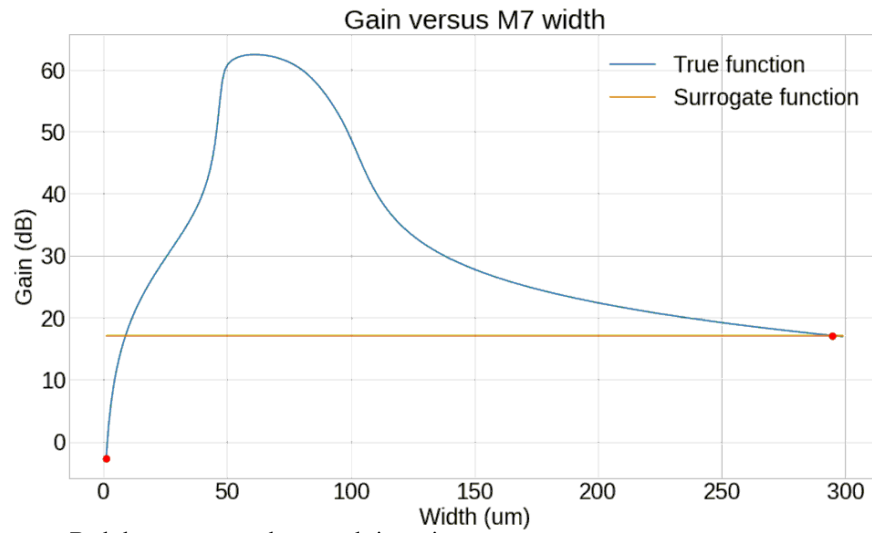
- ▶ Design parameter: W7 (transistor M7 width)
 - ▶ Assuming all other transistors are sized
- ▶ Performance parameter: AC gain
 - ▶ Specification: over 62dB
- ▶ As a reference, plot design space (gain vs W7) with a parametric sweep of W7:



90

Animation of Sampling and Surrogate Function Update with Bayesian Optimization

91



Red dot: new sample at each iteration

Grey region: uncertainty over the region between sampled points



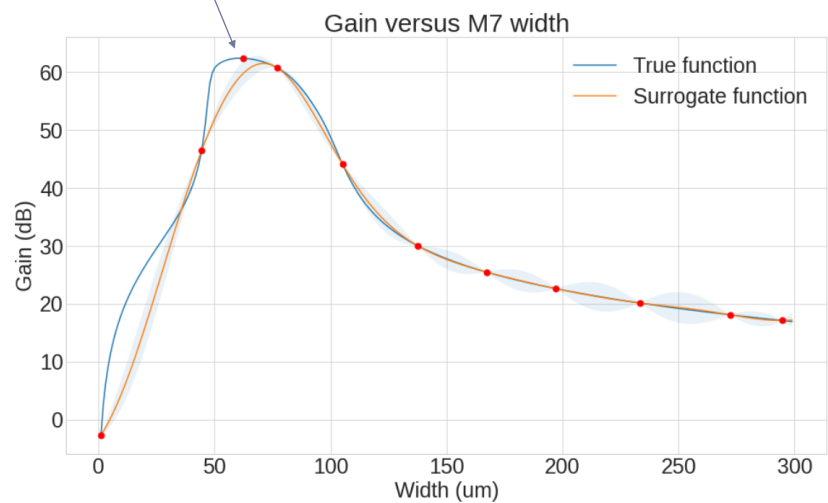
91

Animations of Sampling and Surrogate Function Update with Bayesian Optimization

92

Sampled point at 11th iteration, turn out to be qualified

- Qualified solution found to deliver gain of 62dB after 11 iterations



92

Comparison of Optimization Algorithms for Analog EDA

93

	Derivative-free?	Guarantee optimality?	Computational complexity	Allow prior knowledge (of distribution) ?	Tradeoff considerations
Gradient-based	No	Yes	Low	No	Yes
Evolution-based	Yes	No	Medium	No	Yes
Bayesian optimization	Yes	No	High	Yes	Yes
Reinforcement learning	Yes	No	High	Possibly	Not well formulated



93

Graph-based Optimization Algorithms for EDA

94

- Algorithms specifically for graph-structured design space
- Types of graph applications:
 - Graph partitioning
 - Graph traversal (including shortest path search)
 - Depth-first search
 - Search in order of increasing depth before returning to root node
 - Breadth-first search
 - Search in order of distance from the source node
 - Best-first search
 - Search guided by cost criteria



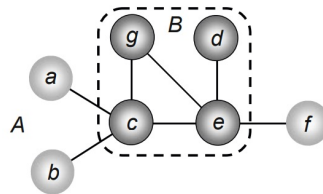
Reference: A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure," Springer, 2011

94

Graph Partitioning

95

- ▶ Objective of partitioning for digital physical design: minimize cut edges while balancing for partition sizes
 - ▶ Divide and conquer: place and route each partition separately before integration



- ▶ Objective of partitioning of analog circuits before layout: identify circuit hierarchies
 - ▶ Grouping utilized as constraints for placement and routing
 - ▶ Example: match for symmetry
 - ▶ Algorithms therefore differ from digital partitioning
 - ▶ Subgraph isomorphism



Reference: A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure," Springer, 2011

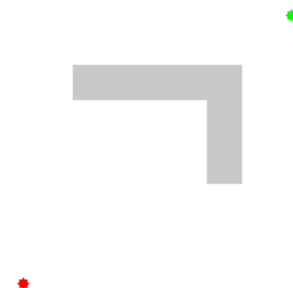
95

Shortest Path Search

96

- ▶ Routing regions represented as a graph
- ▶ Dijkstra's algorithm (a.k.a, maze routing):
 - ▶ Find shortest paths between a given node and all other nodes
- ▶ A* algorithm:
 - ▶ Find shortest path between a given node and a target node
 - ▶ Given an initial and final cell on a square grid
 - g : cost of moving from the initial cell to a certain cell on grid
 - h : estimated cost of moving from the current cell to the final cell
 - ▶ Euclidean distance
 - ▶ Manhattan distance
 - $f = g + h$
 - Procedure: select and move to the smallest f -valued cell
 - ▶ Limitation: high space complexity as storage of all nodes in paths is required

A* search between bottom-left red dot to upper-right green dot



Source: Wikipedia



Reference: A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure," Springer, 2011

96

Summary and Practical Considerations of Applying ML and Optimization for Analog EDA

97

- ▶ Training of a ML model itself requires optimization
 - ▶ Hyper-parameter tuning directly impacts model performance
- ▶ Total time cost = learning/training cost + optimization cost + simulation cost
 - ▶ Compared with ML model training, optimization (guided by ML models) is more computationally costly
 - ▶ Simulation acceleration is primarily job of EDA companies
 - ▶ Improving learning and optimization algorithms is target of research
- ▶ Curse of dimensionality
 - ▶ High feature dimensionality is problematic for all ML and optimization algorithms
- ▶ Effectiveness of EDA algorithms in practice relies on available computation power
 - ▶ Potential breakthrough with quantum computing
 - ▶ More realistically, parallelization, neuromorphic computing...



97

Outline of Presentation

98

- ▶ Background introduction
- ▶ Machine learning techniques for analog EDA
- ▶ Optimization techniques for analog EDA
- ▶ Case studies
 - ▶ Case study 1: component sizing of analog ICs
 - ▶ Case study 2: automated placement and routing
 - ▶ Case study 3: prediction of interconnect impedance
 - ▶ Case study 4: transfer learning for design migration
- ▶ Conclusions

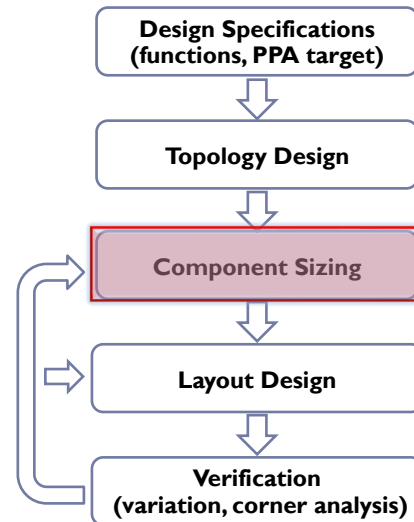


98

Case study 1: Component Sizing of Analog ICs

99

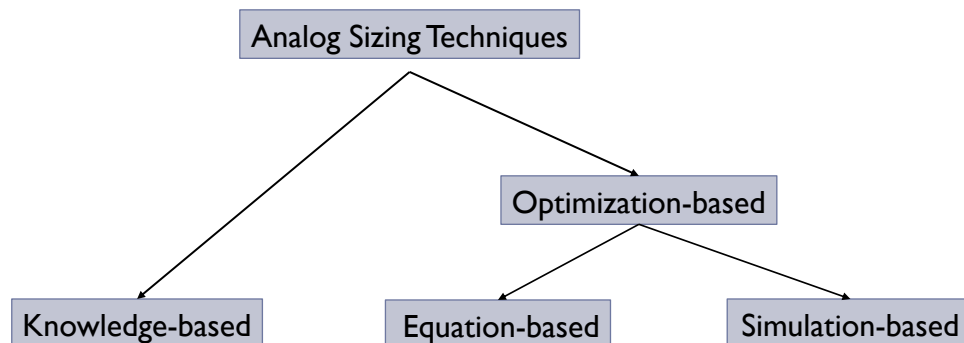
- ▶ Component sizing: tuning the sizes of the devices in an analog circuit to meet design specifications
 - ▶ Example: transistor width, length, number of fins and fingers; resistor, capacitor values...
 - ▶ A critical step in analog synthesis flow
- ▶ Utilize ML to:
 - ▶ Predict circuit performance based on design parameters
 - ▶ Predict layout-dependent effects and interconnect impedances to guide sizing optimization



99

Classification of Analog Sizing Techniques in Literature

100



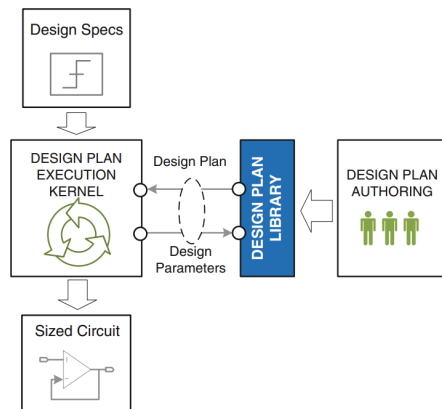
100

Classification of Analog Sizing Techniques in Literature

101

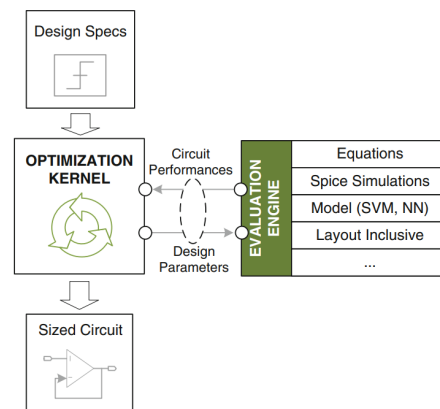
Knowledge-based sizing

- Pre-designed plans consisting of design equations and procedures



Optimization-based sizing

- Optimization problems formulated based on
 - Circuit equations
 - Simulation data

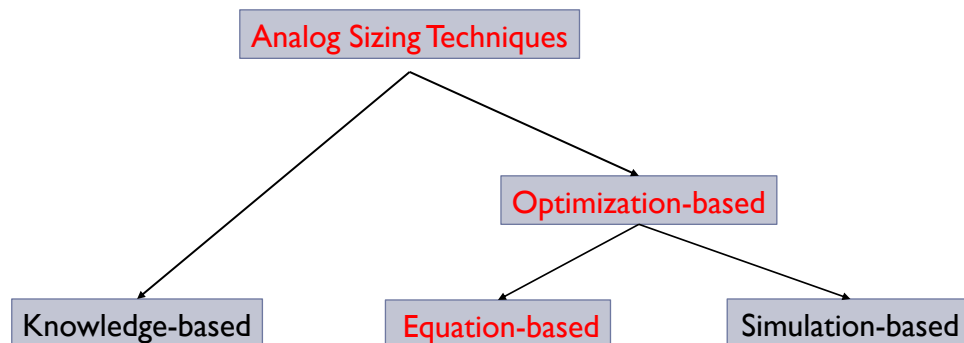


Reference: R. Lourenço, N. Lourenço, N. Horta, "AIDA-CMK: Multi-Algorithm Optimization Kernel Applied to Analog IC Sizing", Springer, 2015

101

Classification of Analog Sizing Techniques in Literature

102



102

Equation-based Sizing Approaches

103

- ▶ Circuit equations linking design parameters with performance parameters
 - ▶ Directly: device sizes, biasing voltages
 - ▶ Indirectly: g_m/I_d , inversion coefficient
- ▶ g_m/I_d : efficiency to translate current (power) into transconductance
 - ▶ g_m/I_d versus $I_d/(W/L)$ (e.g., normalized current) is a unique characteristic of a transistor
 - ▶ Directly related to circuit performance
 - ▶ Works for all transistor operating regions

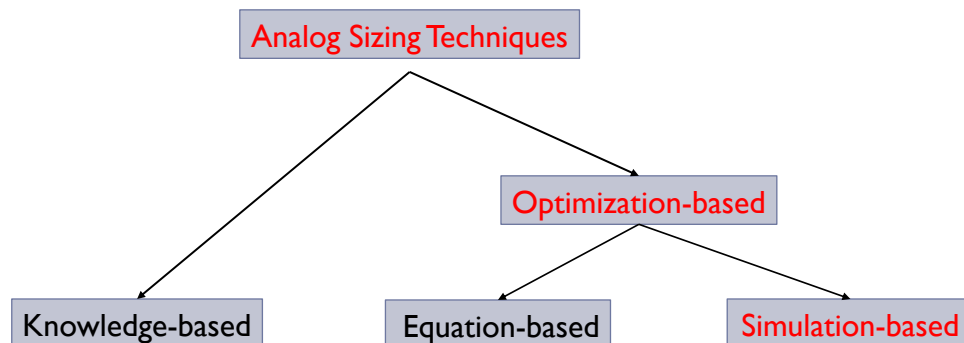


Reference: F. Silveira, D. Flandre and P. G. A. Jespers, "A g_m/I_d Based Methodology For The Design Of CMOS Analog Circuits And Its Application To The Synthesis Of A Silicon-on-insulator Micropower OTA," IEEE Journal of Solid-State Circuits, Vol. 31, No. 9, pp. 1314-1319, 1996

103

Classification of Analog Sizing Techniques in Literature

104



104

Simulation-based Optimization for Analog Circuit Sizing

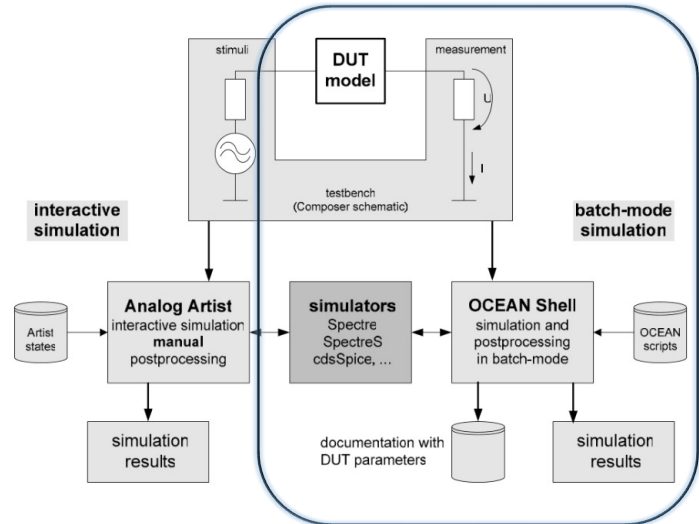
105

Treat the circuit as a blackbox

- ▶ Train surrogate models to map from the design space to the performance space
- ▶ Simulation-based sizing in two primary steps:
 - Performance modeling + Optimization
- ▶ No design equations needed
- ▶ No discrepancy between theoretical and simulated results

Challenges:

- ▶ Circuit simulations with numerical solvers are expensive
 - ▶ Need to improve sample efficiency
- ▶ Non-interpretable design processes



Reference: R. Frevert, et al., "Modeling and Simulation for RF System Design", Springer, pp.291, 2005

105

Practical Considerations of Simulation-based Sizing

106

- ▶ Two types of design loops:
 - ▶ Batch-mode 'flat' design loop
 - ❑ Sample many points at each iteration
 - ❑ Reduced overhead of read/write data files, but may not be the most sample-efficient (total number of simulated points)
 - ❑ Suitable for medium-cost circuit block simulation (e.g., op-amp)
 - ▶ Single-mode 'long sequential' design loop
 - ❑ Sample and query one point each iteration
 - ❑ Most efficient in total number of points simulated, but may not be the fastest
 - ❑ Suitable for high-cost system-level circuit simulation, (e.g., PLL)



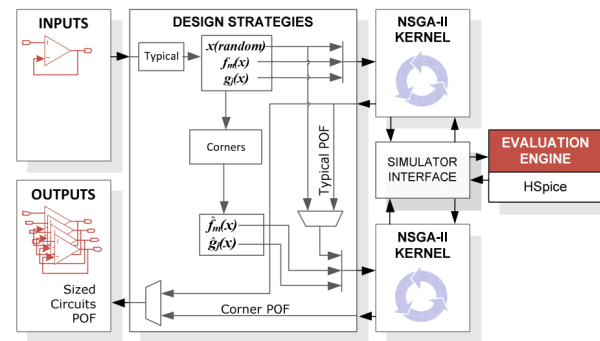
106

GENOM-POF: Multi-objective Evolutionary Synthesis of Analog ICs with Corners

107

- ▶ A multi-objective framework for the sizing of analog circuits based on evolutionary optimization algorithms
 - ▶ POF: Pareto optimal front
 - ▶ Optimization algorithm: genetic algorithm NSGA-II
- ▶ Inputs: Circuit netlist, testbench, design variables, design specifications
- ▶ Output: Sizing of devices
- ▶ Problem formulation
 - ▶ f : objective function
 - ▶ g : constraints
 - ▶ x : design variables

$$\begin{aligned} &\underset{x}{\text{minimize}} && f_m(x), \quad m = 1, 2, \dots, M, \\ &\text{subject to} && g_j(x) \geq 0, \quad j = 1, 2, \dots, N, \text{ and} \\ &&& x_{i_L} \leq x \leq x_{i_U}, \quad i = 1, 2, \dots, K, \end{aligned}$$



Reference: N. Lourenço and N. Horta, "GENOM-POF: Multi-objective Evolutionary Synthesis Of Analog ICs With Corners Validation," Proceedings of the International Conference on Genetic and Evolutionary Computation, pp. 1119– 1126, 2012

107

Compensation for Effects of Variations on Circuit Performance

108

- ▶ Simulation approaches
 - ▶ Monte Carlo simulation
 - ▶ Corner analysis
- ▶ Proposed methods for compensating for the effects of variation on circuit performance
 - ▶ Method 1: Evaluate the circuit performance at all corners of interest
 - ▶ Method 2: Optimize for typical corner first, then use typical-corner sizing as starting points to optimize for other corners



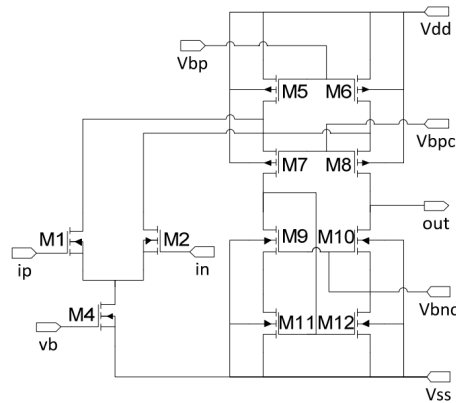
Reference: N. Lourenço and N. Horta, "GENOM-POF: Multi-objective Evolutionary Synthesis Of Analog ICs With Corners Validation," Proceedings of the International Conference on Genetic and Evolutionary Computation, pp. 1119– 1126, 2012

108

Simulation Results for Sizing an Op-amp in 180 nm with GENOM-POF

109

- ▶ 15 design variables
- ▶ Accounting for combinations of three process corners and three temperature corners
- ▶ Two target performance metrics: area, power



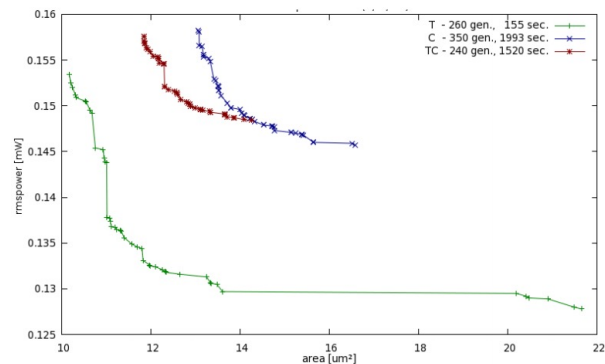
Reference: N. Lourenço and N. Horta, "GENOM-POF: Multi-objective Evolutionary Synthesis Of Analog ICs With Corners Validation," Proceedings of the International Conference on Genetic and Evolutionary Computation, pp. 1119–1126, 2012

109

Simulation Results for Sizing an Op-amp in 180 nm with GENOM-POF

110

- ▶ Compared with starting with all corners (method 1), starting with the typical corner, then optimizing for remaining corners (method 2) provides better solutions
 - ▶ More optimal performance
 - ▶ Faster execution



	Start with all corners	Start with typical, then remaining corners
Area (μm^2)	3.5	2.08
Power (mW)	2.06	1.76
Time (s)	572	227



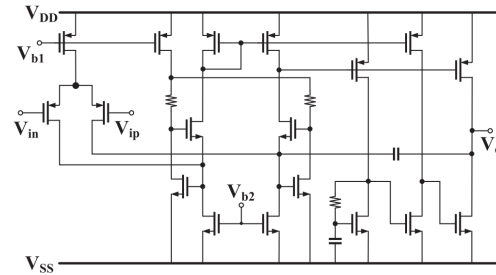
Reference: N. Lourenço and N. Horta, "GENOM-POF: Multi-objective Evolutionary Synthesis Of Analog ICs With Corners Validation," Proceedings of the International Conference on Genetic and Evolutionary Computation, pp. 1119–1126, 2012

110

An Efficient Bayesian Optimization Approach (WEIBO) for Sizing a Three-stage Amplifier

111

- ▶ Algorithm: Bayesian optimization
- ▶ Weighted Expected Improvement (WEI)
 - ▶ After initial sampling, a random weight vector w is generated to construct the WEI acquisition function for the surrogate-assisted optimization problem
- ▶ 24 design variables
- ▶ WEIBO delivers the optimal I_q
 - ▶ Second highest sample efficiency



minimize I_q ,
 subject to $GAIN > 100dB$,
 $UGF > 0.92MHz$,
 $PM > 52.5deg$,
 $GM > 19.5dB$,
 $SRR > 0.18V/\mu s$, and
 $SRF > 0.2V/\mu s$,

Algorithms	WEIBO	GASPAD	MSP	DE	PSO	SA
GAIN (dB)	100.57	101.6	100.81	102.73	102.39	102.49
UGF (MHz)	0.922	0.924	0.982	0.956	0.963	1.046
PM (deg)	52.52	52.60	53.22	54.62	54.25	56.70
GM (dB)	19.76	20.05	22.30	20.62	21.32	20.92
SR+ (V/ μs)	0.20	0.24	0.23	0.21	0.23	0.25
SR- (V/ μs)	0.46	0.55	0.51	0.54	0.51	0.49
$I_{qmean}(\mu A)$	27.87	31.55	49.60	41.26	44.22	59.78
Number of Samples	798	2744	2163	2400	2417	620



Reference: W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," IEEE Transactions on Circuits and Systems, Vol. 65, No. 6, pp. 1954–1967, 2018

111

VCAIT: Variation-aware Classification with Adaptive Labeling Thresholds for Analog Sizing

112

- ▶ Apply classification to predict whether a candidate solution satisfies the specification
- ▶ To address class imbalance: adaptively set the labeling thresholds
 - ▶ With an initial dataset,
 - ▶ If the design specification is a lower bound:
 - ▶ labeling threshold = min (design specification, ϵ percentile of the target metric in dataset)
 - ▶ If the design specification is an upper bound:
 - ▶ labeling threshold = max (design specification, (100- ϵ) percentile of the target metric in dataset)



Reference: Z. Wu and I. Savidis, "Variation-aware Analog Circuit Sizing with Classifier Chains," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 1–6, 2021

112

Design Flow of VCALT

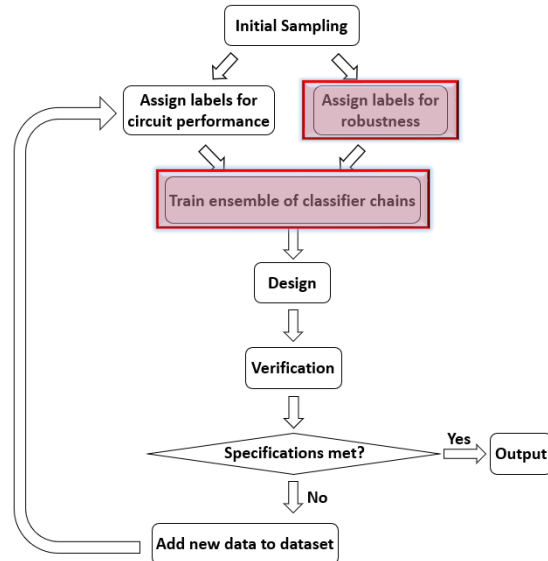
113

Optimization-based active querying:

- Adaptively set labeling thresholds
- Search for solutions that simultaneously maximize the output probability scores of classifiers for specification and robustness prediction:

$$x^* \in \arg \max(p_1(x), \dots, p_k(x), r_1(x), \dots, r_k(x))$$

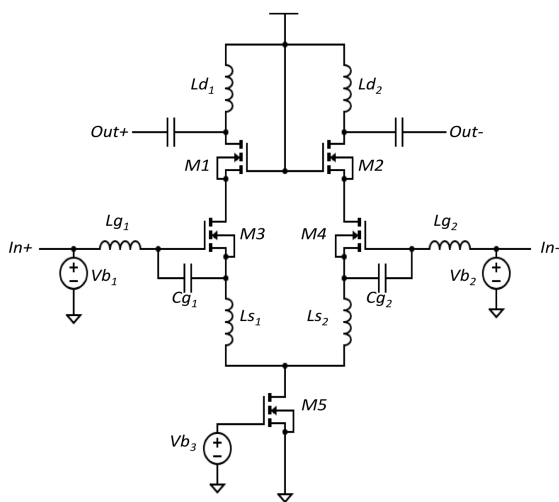
- Robustness parameter: performance fluctuations of the circuit across all corners**



113

Apply VCALT to the Design of a Low-noise Amplifier in 65 nm Technology

114



Design Constraints

$$\begin{cases} 60 \text{ nm} \leq \text{transistor widths} \leq 900 \text{ } \mu\text{m} \\ 0.01 \text{ nH} \leq \text{inductor sizes} \leq 12 \text{ nH} \\ 30 \text{ fF} \leq \text{capacitor sizes} \leq 20 \text{ pF} \\ 0 \text{ V} \leq \text{biasing voltages} \leq 1.2 \text{ V} \end{cases}$$

Include constraints for robustness

- 5 process corners and 3 temperature corners (TT, FF, SS, SF, FS) * (20°C, 80°C, 120°C)**

Design Specifications

Performance:

$$\begin{cases} \text{Gain} \geq 10 \text{ dB} \\ \text{NF} \leq 2.8 \text{ dB} \\ \text{IP3} \geq -5 \text{ dBm} \\ \text{Power} \leq 20 \text{ mW} \end{cases}$$

Robustness:

$$\begin{cases} \sigma_{\text{Gain}} \leq 1 \text{ dB} \\ \sigma_{\text{NF}} \leq 0.5 \text{ dB} \\ \sigma_{\text{IP3}} \leq 1 \text{ dBm} \\ \sigma_{\text{Power}} \leq 5 \text{ mW} \end{cases}$$



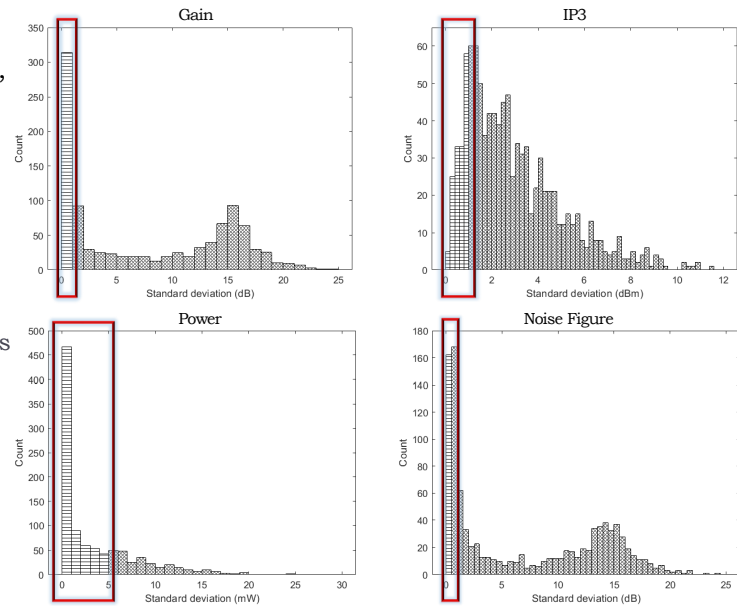
Reference: Z. Wu and I. Savidis, "Variation-aware Analog Circuit Sizing with Classifier Chains," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 1–6, 2021

114

Distribution of Standard Deviations of Performance Parameters across 15 Corners in the Initial Dataset

115

- ▶ Without accounting for robustness, design solutions have large fluctuations in circuit performance over different corners
- ▶ Classifiers to predict and select robust candidate solutions
 - ▶ Thresholds on the standard deviations are set by the designer



115

Comparison between Sizing Techniques

116

	Knowledge-based sizing	Optimization-based sizing
Execution	Fast	Slow
Manual effort required	High	Low
Technology scaling	Requires update	Don't care
Optimality of design solutions	Sub-optimal	Optimal



116

Outline of Presentation

117

- Background introduction
- Machine learning techniques for analog EDA
- Optimization techniques for analog EDA
- **Case studies**
 - Case study 1: component sizing of analog ICs
 - **Case study 2: automated placement and routing**
 - Case study 3: prediction of interconnect impedance
 - Case study 4: transfer learning for design migration
- Conclusions



117

Case Study 2: Automated Placement and Routing

118

Primary considerations for digital P&R:

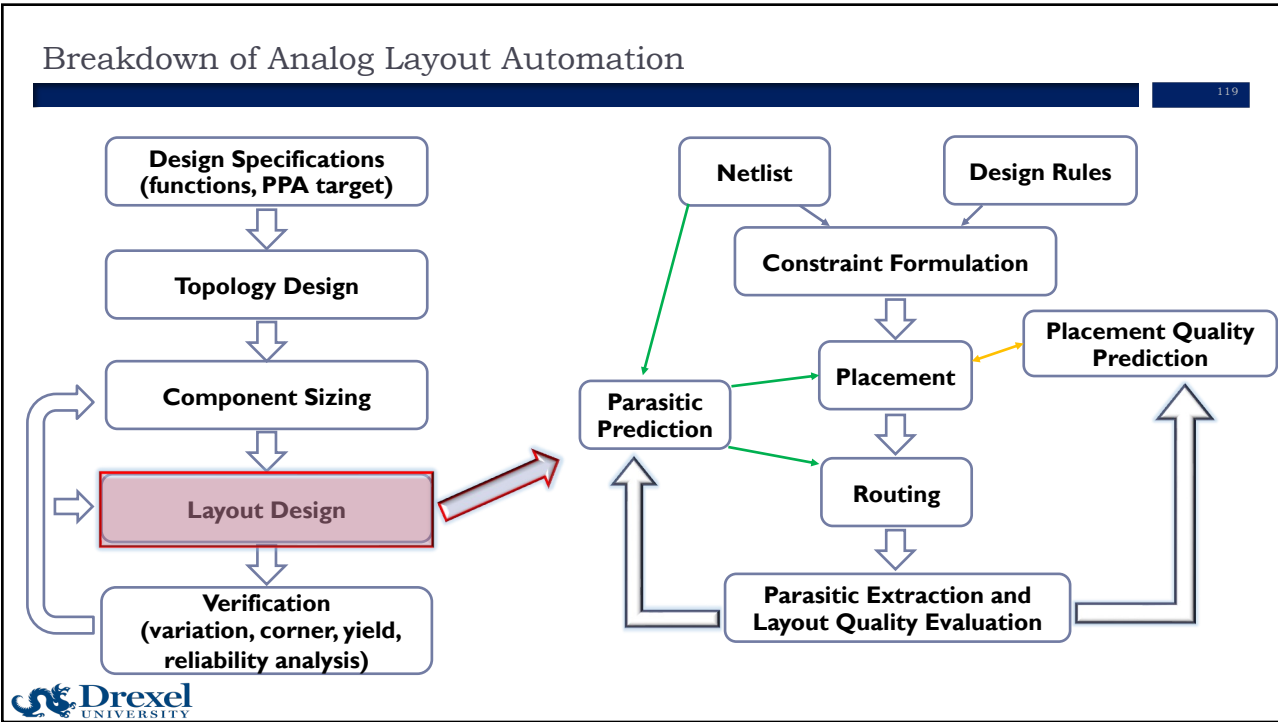
- Area minimization
 - Equivalent to wirelength minimization in most cases
- Design rule constraints

Additional considerations for analog P&R:

- Interconnect impedance
 - IR drop
 - Signal coupling
- Matching
 - Symmetry
 - To reduce device mismatch in differential pairs
 - Length-matching
 - Impedance-matching



118



119

Analog Circuit Hierarchies

Four primary levels of an analog circuit

- Individual device level
- Sub-block level
 - Differential pairs
 - Current mirrors
 - ...
- Sub-circuit level
 - OTA
 - VCO
 - ...
- System level
 - RF transceiver
 - Data converter

Primitive	Schematic	Primitive	Schematic
Resistor		Current mirror 1	
Capacitor		Current mirror 2	
Capacitor array		Voltage reference	
Switch		Level shifter	
Diode-connected load		Current mirror load	
Differential pair		Current mirror bank	
Cross-coupled pair 1		Level shifter bank	
Cross-coupled pair 2		Dummy 1	
Differential load		Dummy 2	
Cascode pair		Decap	

Reference: T.Dhar, et al., "ALIGN: A System for Automating Analog Layout," IEEE Design & Test, Vol. 38, No. 2, pp. 8-18, 2021

120

120

Motivation to Perform Device Grouping and Circuit Hierarchy Recognition

- ▶ Human designers recognize analog subblocks
 - ▶ Account for device grouping and hierarchies when performing layout
 - ▶ Match devices for symmetry
- ▶ Heuristic techniques to recognize sub-circuits
 - ▶ Library-based: match with a library enumerating possible topologies
 - ▶ Rule-based: programmed rules to recognize sub-circuits
- ▶ Limitation of heuristic techniques:
 - ▶ Exhaustive enumeration of circuit blocks or recognition rules is often infeasible

```

graph TD
    Netlist --> CF[Constraint Formulation]
    DesignRules[Design Rules] --> CF
    CF --> Placement
    Placement --> Routing
    Routing --> PE[Parasitic Extraction and Layout Quality Evaluation]
    PE --> PP[Parasitic Prediction]
    PP --> Placement
    PE --> PQP[Placement Quality Prediction]
    PQP --> Placement
  
```

121

Knowledge-based Constraint Formulation: Sizing Rules Method

- ▶ A library of analog building blocks implemented based on a hierarchical set of circuit components
- ▶ Allows for subblock recognition and structural synthesis

Subblock recognition of a folded cascode amplifier

Function	Schematic NMOS (PMOS do.)	Hierarchical Level
Voltage-Controlled Resistor (vres) Volt.-Contr. Current Source (vccs)		0
Simple Current Mirror (cm)		1
Level Shifter (ls)		
Voltage Reference 1 (vr1)		
Current Mirror Load (cml)		
Differential Pair (dp)		
Voltage Reference 2 (vr2)		2
Flip-Flop (ff)		
Level Shifter Bank (LSB)		
Current Mirror Bank (CMB)		
Cascode Current Mirror (CCM)		
4-Transistor Current Mirror (4TCM)		3
Differential Stage (CM=cm/CCM/4TCM)		

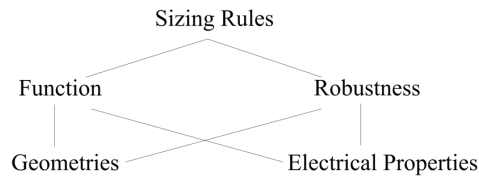
Reference: H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The Sizing Rules Method For Analog Integrated Circuit Design," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 343-349, 2001

122

Sizing Rules Method

123

- Constraints on four types of circuit properties



- 30 generic constraints listed for the circuit subblocks

- Examples of sizing rules:
 - Level 0: Constraints on transistor biasing
 - Level 2: Symmetry and matching constraints on current mirrors



Reference: H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The Sizing Rules Method For Analog Integrated Circuit Design," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 343–349, 2001

123

Analytical Approach for Subgraph Matching to Recognize Circuit Hierarchies

124

- Kolmogorov-Smirnov (K-S) test

- Measures the similarity between the eigenvalue distributions of two graphs
 - Higher score indicates higher similarity between the graphs:

$$D_n = \sup |F_{1,n}(x) - F_{2,m}(x)| \quad , \text{ where } F \text{ are cumulative distribution functions}$$

- Step 1: Compute eigenvalues of the two graph Laplacian matrices
- Step 2. Apply K-S test on the underlying distributions
- Step 3. The resulting k-value of the K-S test is used as the graph similarity score



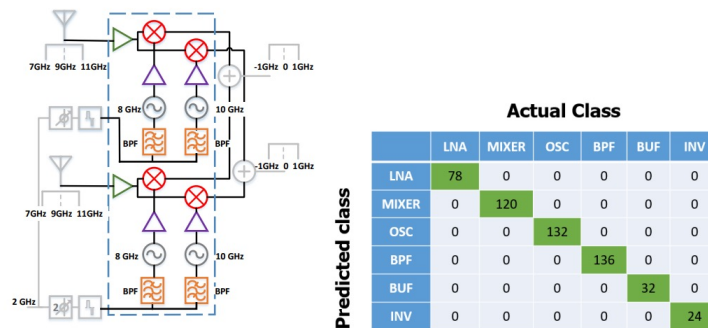
Reference: M. Liu, et al., "S3DET: Detecting System Symmetry Constraints for Analog Circuits with Graph Similarity," Proceedings of the Asia and South Pacific Design Automation Conference, pp. 193-198, 2020

124

GCN-based Approach for Symmetry Detection in Analog Circuits

125

- Algorithm: GCN trained on bipartite circuit graph
- Symmetry is identified at primitive cell level, block level and system level
- Limitation 1: hierarchical information of blocks are often directly available from the netlist
- Limitation 2: training data required for each circuit type



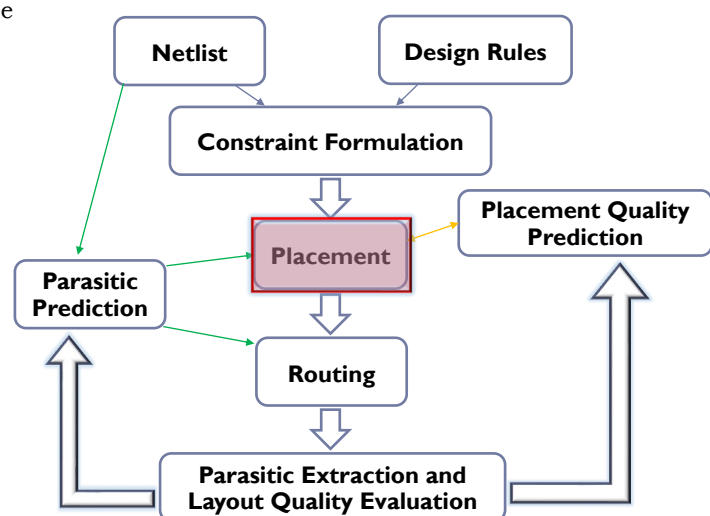
Reference: K. Kunal, et. al., "GANA: Graph Convolutional Network Based Automated Netlist Annotation For Analog Circuits," Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, pp. 55–60, 2020

125

Analog Placement

126

- Analog placement: before routing, determine the locations of each instance
- Objective: minimize area and expected wirelength, mitigate layout effects on performance
- Heuristic approach:
 - Step 1: Represent a placement solution with a data structure
 - Examples:
 - O-tree
 - Segment tree
 - Step 2: Apply optimization algorithms to perturb the data structure in search of optimal solutions
 - Mixed integer linear programming
 - Nonlinear programming



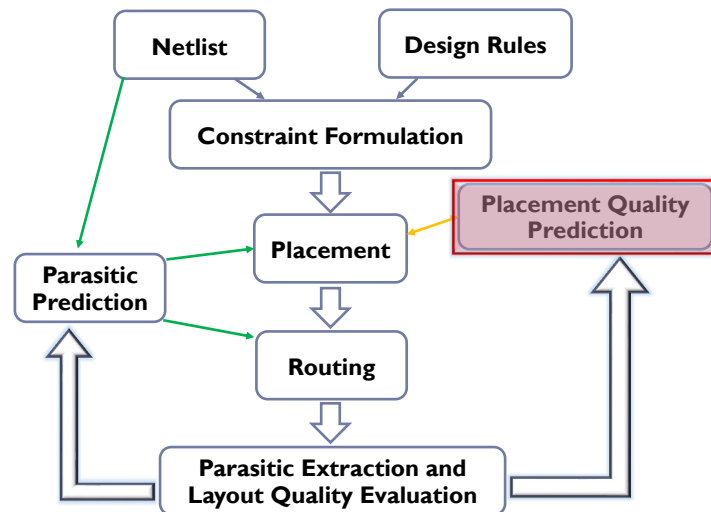
126

Motivation to Predict Placement Quality

127

- ▶ Example: difference between results of schematic simulation and post-layout simulation of an LNA design

Parameter	Schematic	Layout
Frequency	2.4 GHz	2.53 G
Gain (S_{21})	15.3 dB	13.2 dB
NF	2.8 dB	3.5 dB
IIP3	-5 dBm	-7.3 dBm

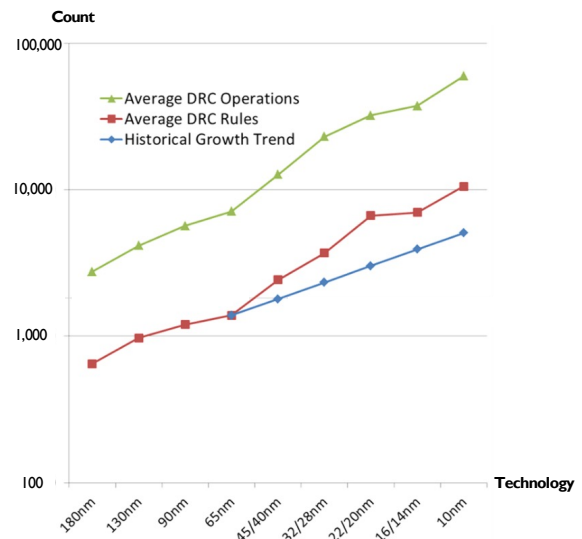


127

Analog Routing

128

- ▶ Objective: minimize wirelength and area while optimizing circuit performance
- ▶ Primary considerations (constraints for optimization)
 - ▶ Design rules
 - ▶ Size rules (e.g., minimum width)
 - ▶ Separation rules
 - ▶ Overlap rules
 - ▶ Matching for symmetry
 - ▶ Example: common-centroid design of differential pairs
 - ▶ Impedance-matching
 - ▶ Effects of interconnect impedance (R,L,C) on performance parameters
 - ▶ Often not translating to geometric constraints directly
 - ▶ Example: minimizing wirelength does not necessarily result in optimal performance



Source: Semiengineering.com

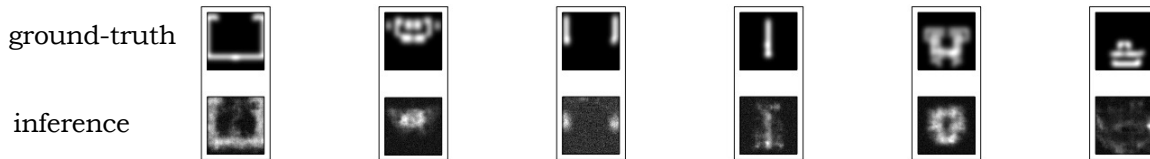


128

GeniousRoute: ML-guided Routing

129

- Algorithm: Variational auto-encoders (VAEs) trained on layout images
 - Encoder: map input image to low-dimensional space
 - Decoder: generate routing guidance
 - Label: routing region of nets
- Routing prediction: for a given placement, VAE predicts the probability map that a wire is placed in a region
- Routing algorithm: A* search algorithm guided by the trained VAE model
- Limitation: GeniusRoute is trained on a dataset consisting of comparators and amplifiers without generalizing to other analog circuit types



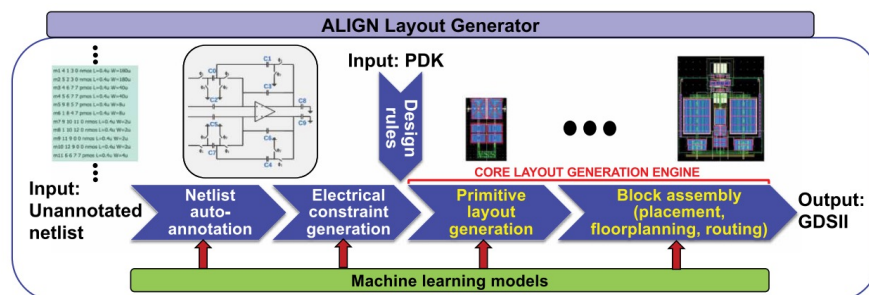
Reference: K. Zhu, et al., "GeniusRoute: A New Analog Routing Paradigm Using Generative Neural Network Guidance," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp.1-8, 2019

129

ALIGN: A System for Automating Analog Layout

130

- Translation of a SPICE-level netlist into a physical layout
 - Input: sized netlist of the topology, specifications, PDK description
 - Output: GDSII of layout
 - Bottom-up approach with a mix of algorithmic techniques, template-driven design, and ML
- Highlight:
 - 24-hour turnaround with no human in the loop
 - Accounts for circuit hierarchies
 - Accounts for analog layout techniques (e.g., common centroid layout)



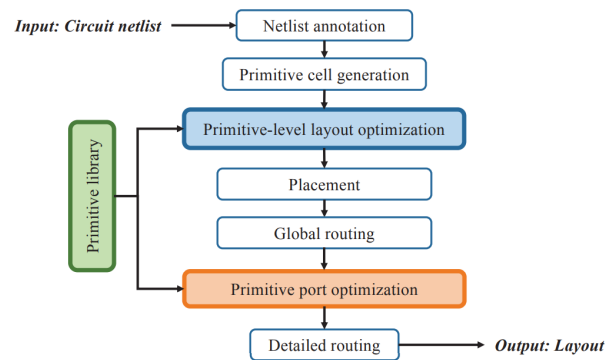
Reference: T. Dhar, et al., "ALIGN: A System for Automating Analog Layout," IEEE Design & Test, Vol. 38, No. 2, pp. 8-18, 2021

130

Synthesis Steps of ALIGN

131

- Design Rule abstraction
 - Constraints from PDK
- Netlist Auto-annotation
 - Groups transistors and passives in the input netlist into building blocks and identifies geometric constraints on the layout of each block
- Electrical Constraint Generation
 - Performance constraints turned into layout constraints, such as the maximum allowable route length
- Parameterized layout generation of primitives
 - Parameters: transistor size, capacitor size, resistor size
- Block assembly
 - Place and route all blocks based on design hierarchy



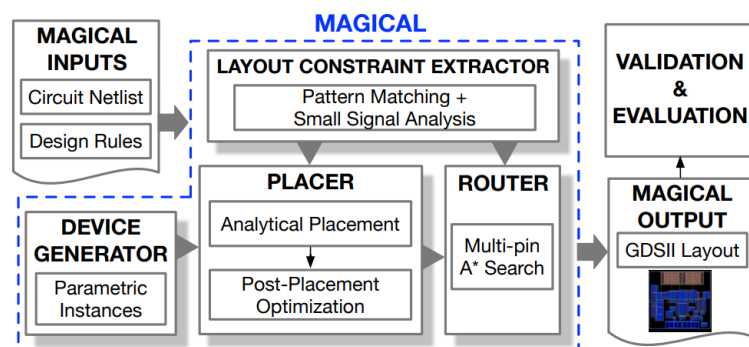
Reference: T. Dhar, et al., "ALIGN: A System for Automating Analog Layout," IEEE Design & Test, Vol. 38, No. 2, pp. 8-18, 2021

131

MAGICAL: Machine Generated Analog IC Layout

132

- Netlist to GDSII synthesis flow for analog circuits
- Core placement algorithm:
 - Non-linear programming-based global placer
 - Linear programming-based legalizer
- Core routing algorithm:
 - Obstacle-aware path-finding algorithm searching the feasible routing paths
- Highlight: silicon-proven with TSMC 40nm 1GS/s delta-sigma ADC



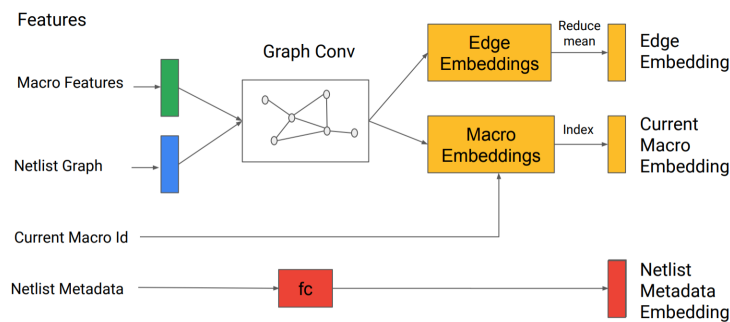
Reference: B. Xu, et al., "MAGICAL: Toward Fully Automated Analog IC Layout Leveraging Human and Machine Intelligence: Invited Paper," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp.1-8, 2019

132

Automated Cell Placement By Google

133

- ▶ Primarily for digital cell placement optimization with RL and GNN
 - ▶ Applications: design google accelerator chips (TPUs)
- ▶ RL for placing macros, heuristics to place standard cells
 - ▶ RL reward: expected wirelength (i.e., HPWL) and expected congestion
- ▶ Edge-based GNN operate on embeddings of placed partial graph and candidate node



Reference: A. Mirhoseini, et al., "A Graph Placement Methodology for Fast Chip Design", Nature, No. 594, pp. 207–212, 2021

133

Various Learning Scenarios Powered by Variants of ANNs

134

Differentiate by training scheme

- ▶ Supervised
- ▶ Semi-supervised
- ▶ Unsupervised
- ▶ Adversarial (GAN)
- ▶ Reinforcement
- ▶ Encoder-decoder

Differentiate by data format

- ▶ Multi-layer perceptrons (feedforward neural networks)
- ▶ Convolutional neural networks
- ▶ Graph neural networks
- ▶ Recurrent neural networks



Can combine any option from the left with any option from the right



134

Comparison of Analog Synthesis Platforms

135

- ▶ Even state-of-art analog synthesis platforms require pre-defined procedural rules

	Programming Language	Open-source?	Performs hierarchy recognition	Automated constraint generation	Technology dependency	Silicon-proven?
BAG	python	Yes	No	No	Independent	No
ALIGN	python, C	Yes	Yes	Yes	Requires compatible tech files	Yes
MAGICAL	Computation: C++, User interface and control: python	Yes	Yes	Yes	Requires compatible tech files	Yes



135

Outline of Presentation

136

- ▶ Background introduction
- ▶ Machine learning techniques for analog EDA
- ▶ Optimization techniques for analog EDA
- ▶ **Case studies**
 - ▶ Case study 1: component sizing of analog ICs
 - ▶ Case study 2: automated placement and routing
 - ▶ **Case study 3: prediction of interconnect impedance**
 - ▶ Case study 4: transfer learning for design migration
- ▶ Conclusions

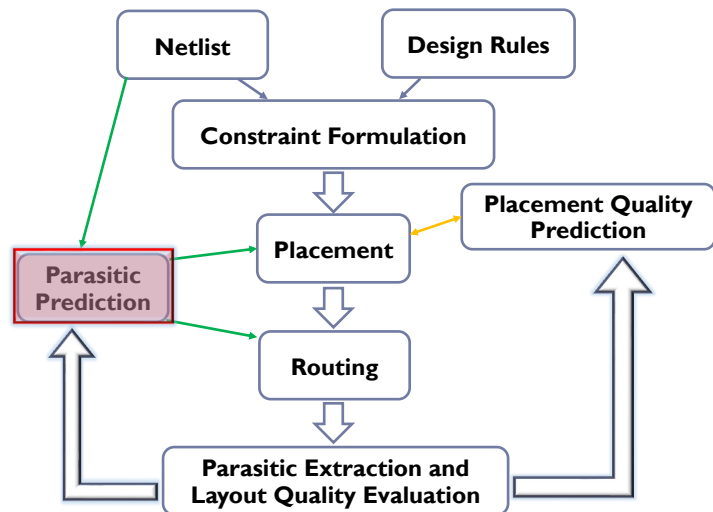


136

Case Study 3: Prediction of Interconnect Impedance

137

- Interconnect impedances have big impact on circuit performance
 - Results in gap between schematic and layout simulation results
- Most interconnect models are analytical
 - Few ML-based techniques
- Apply ML to predict parasitic impedances
 - Reduce error between pre-layout and post-layout simulation
 - Guide placement and routing
 - Reduce parasitic extraction simulations required
- Predict post-routing interconnect values at different stages of analog synthesis
 - At schematic level
 - At placement level



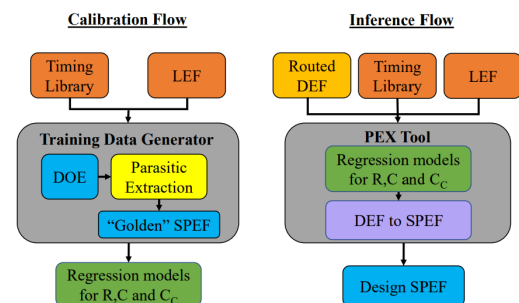
137

A Machine Learning Based Parasitic Extraction Tool

138

- Algorithm: regression
- Target circuit parameters for prediction: resistance, capacitance to ground, coupling, crossover, and crossunder capacitance of a net
- Training data is generated from Cadence Innovus with design of experiment (DOE)
 - Not exclusively for analog but provides physical modeling of interconnect capacitances
- Regression function is fixed
 - Only fitting regression parameters on data
 - Example: coupling capacitance expression

$$C_c = c/s + d \cdot l_{overlap} + e \cdot l_{overlap}/s + f$$
- Inflexible to model interconnects at advanced technologies
- No results reported

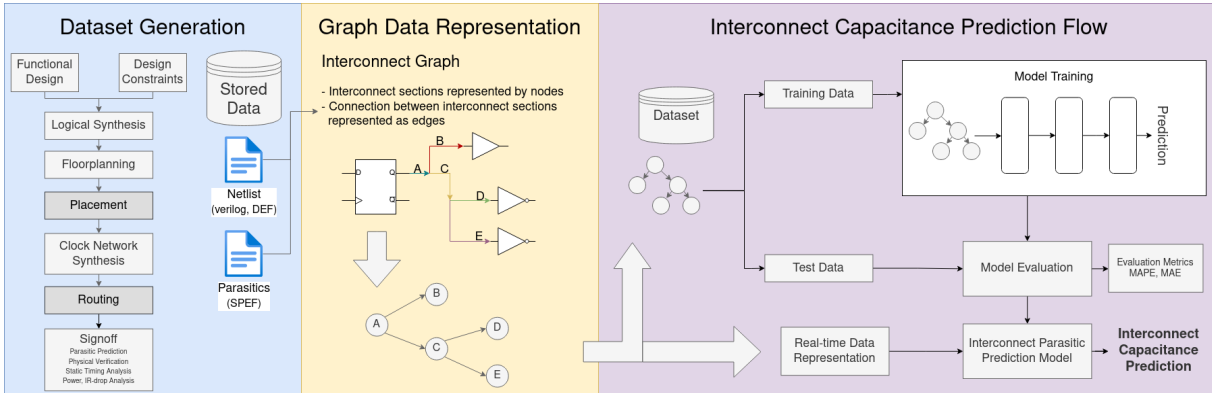


Reference: G. Pradipta, V. A. Chhabria, and S. S. Sapatnekar, "A Machine Learning Based Parasitic Extraction Tool", 2019

138

Parasitic Prediction Framework

139



- A digital EDA pipeline for
 - Parsing standard files of parasitic extraction
 - Training GNN models for estimation of parasitic values on interconnect segments

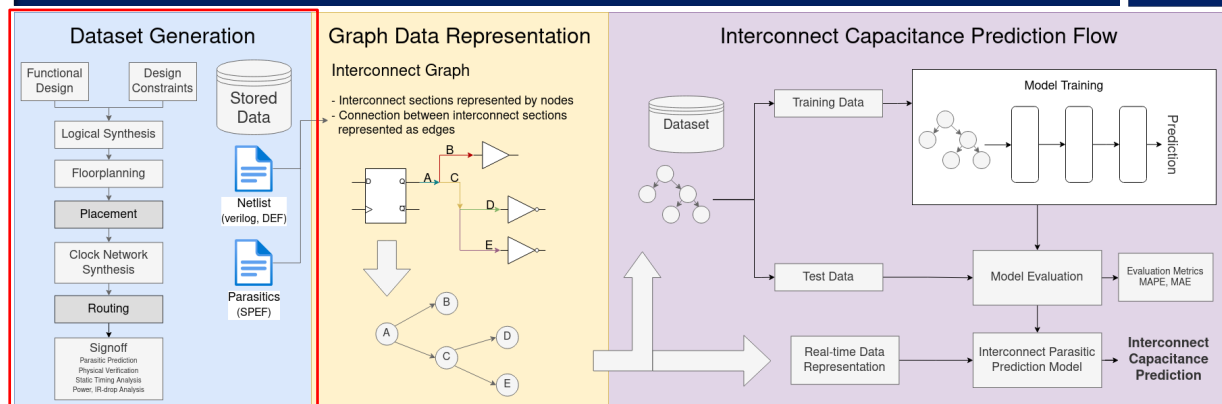


Reference: P. Shrestha and I. Savidis, "Graph Representation Learning for Parasitic Impedance Prediction of the Interconnect", Proceedings of the IEEE International Symposium on Circuits and Systems, 2023

139

Parasitic Prediction Framework: Dataset Generation

140



- Generate physical design data for each design stage
 - Multiple data files generated: Verilog file, DEF file, SPEF file

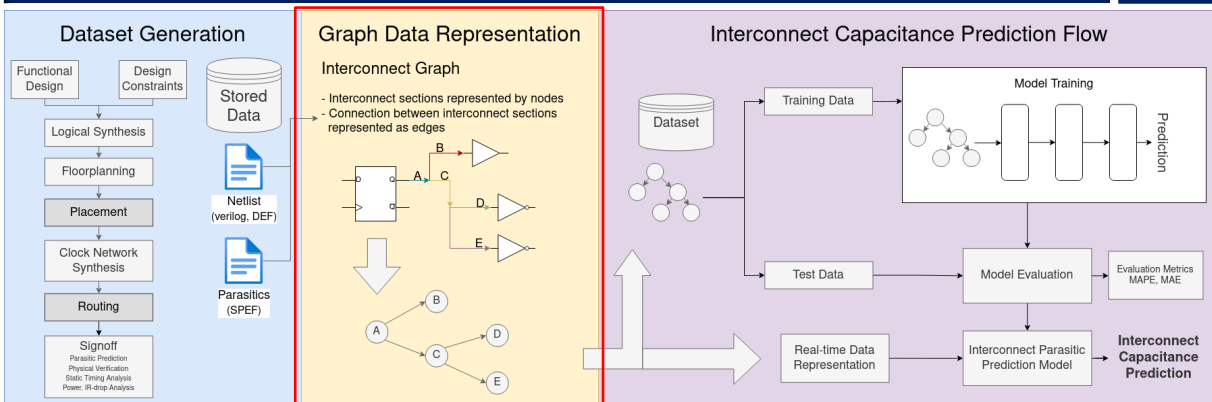


Reference: P. Shrestha and I. Savidis, "Graph Representation Learning for Parasitic Impedance Prediction of the Interconnect", Proceedings of the IEEE International Symposium on Circuits and Systems, 2023

140

Parasitic Prediction Framework: Graph Representation

141



- Utilize netlist and SPEF reports
 - Convert into interconnect spatial graphs
 - Populate with node features



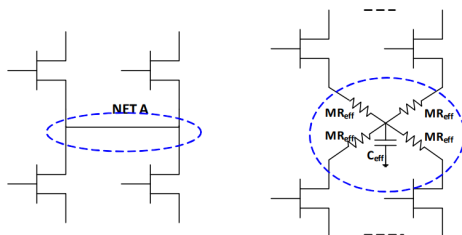
Reference: P. Shrestha and I. Savidis, "Graph Representation Learning for Parasitic Impedance Prediction of the Interconnect", Proceedings of the IEEE International Symposium on Circuits and Systems, 2023

141

MLParest: ML-based Parasitic Estimation for Custom Circuit Design

142

- Algorithm: random forest
- Apply fixed interconnect star model



- Error between pre-layout and post-layout circuit simulation is reduced from 37% to 8% on benchmark analog circuits
- Limitation: inflexibility of the interconnect model
 - Only effective for timing prediction



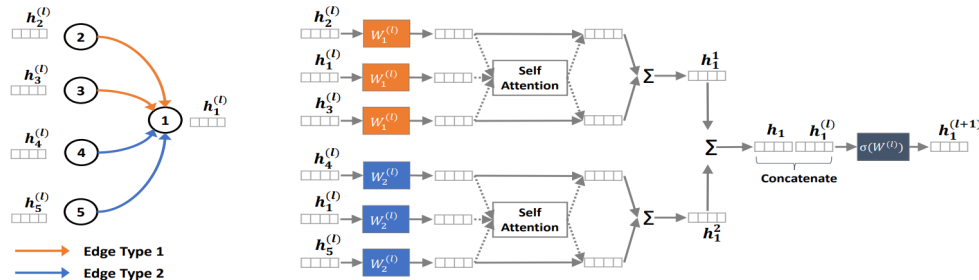
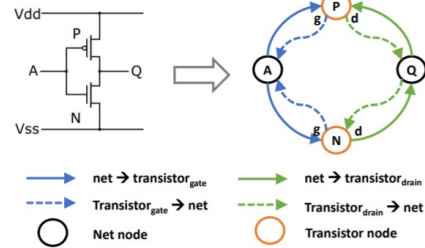
Reference: B.W. Shook. et.al, "MLParest: Machine Learning based Parasitic Estimation for Custom Circuit Design", Proceedings of the ACM/IEEE Design Automation Conference, pp. 1-6, 2020

142

ParaGraph: Layout Parasitics and Device Parameter Prediction with GNNs

143

- Graph representation of a circuit
 - Heterogeneous graph: devices and nets both as graph nodes
- Multiple sub-models for different capacitance ranges
- Transistor features:
 - gate poly length
 - number of fingers
 - number of fins
 - multiplier



Reference: H. Ren, G. F. Kokai, W. J. Turner and T. Ku, "ParaGraph: Layout Parasitics And Device Parameter Prediction Using Graph Neural Networks", Proceedings of the ACM/IEEE Design Automation Conference, pp. 1-6, 2020

143

ParaGraph: Layout Parasitics and Device Parameter Prediction with GNNs

144

- Simulation errors between pre-layout predictions and post-layout on 67 circuit metrics in the testing circuits:

Error Range	Layout w/o parasitics	Designer's Estimation	Prediction w/ XGB	Prediction w/ ParaGraph
< 10%	4	6	17	44
10%-20%	0	17	14	10
20%-30%	5	18	4	8
30%-40%	35	2	7	4
40%-50%	14	6	9	1
> 50%	9	18	16	0
Mean	37.75%	>100%	32.14%	9.60%
Geometric Mean	29.01%	43.57%	15.46%	4.00%

- GCN-based model achieves an average prediction R2 of 0.772 (110% better than XGBoost)
- Average simulation errors from over 100% with designer's estimation to less than 10%



Reference: H. Ren, G. F. Kokai, W. J. Turner and T. Ku, "ParaGraph: Layout Parasitics And Device Parameter Prediction Using Graph Neural Networks", Proceedings of the ACM/IEEE Design Automation Conference, pp. 1-6, 2020

144

Outline of Presentation

145

- Background introduction
- Machine learning techniques for analog EDA
- Optimization techniques for analog EDA
- **Case studies**
 - Case study 1: component sizing of analog ICs
 - Case study 2: automated placement and routing
 - Case study 3: prediction of interconnect impedance
 - **Case study 4: transfer learning for design migration**
- Conclusions



145

Case Study 4: Transfer Learning for IC Design Migration

146

- Challenge and opportunity: need of porting analog design between technology nodes
- Apply transfer learning for reuse of characterized circuit data and models
 - Assume data is available from a source domain to predict in a target domain
 - Benefits:
 - Faster training
 - Improved model performance
 - Requirement of less data
- Circuit Applications:
 - Migrate designs (sizing, placement, routing) across circuit topologies
 - Migrate designs across different technology nodes

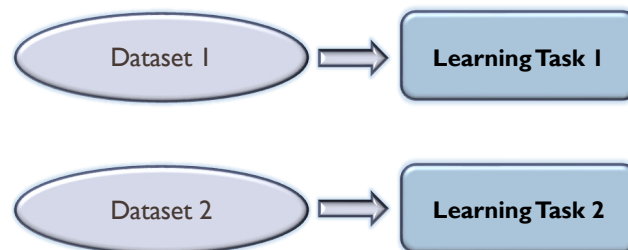


146

Traditional Supervised Learning

147

- Sample data in the search space of the given problem
 - Train independent machine learning models for each problem
- Challenge:
 - Circuit data is often proprietary
 - Acquiring sufficient data for each circuit task is costly or infeasible



147

Transfer Learning with Domain Adaptation

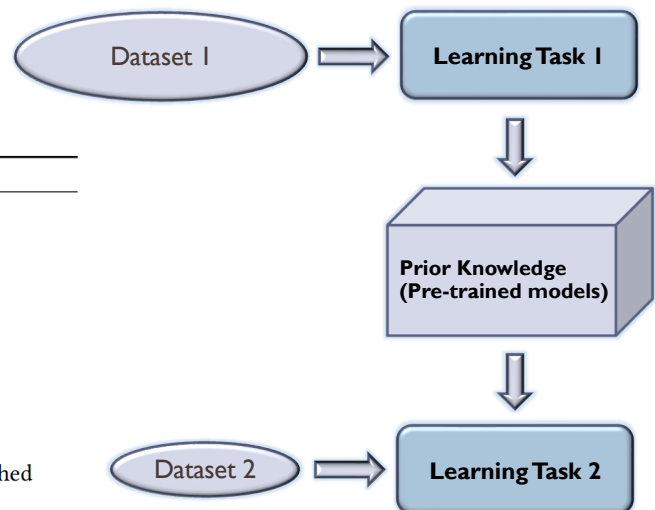
148

- Freeze a variable number of layers of the prior models
- Retrain with a smaller dataset in the target domain/node

Algorithm 1 Transfer Training with Frozen Layers.

```

1: procedure TRAIN
2:    $U_t \leftarrow$  training data from target domain
3:    $P \leftarrow$  pre-trained network
4:    $l \leftarrow$  number of intermediate layers
5:    $h \leftarrow$  number of frozen layers
6:   freeze first  $h$  intermediate layers of  $P$ 
7:   for  $i \in \{h+1, h+2, \dots, l\}$ 
8:     retrain layer  $i$  of  $P$  on  $U_t$ 
9:   end for
10:  repeat until the maximum number of epochs reached
11: end procedure
  
```

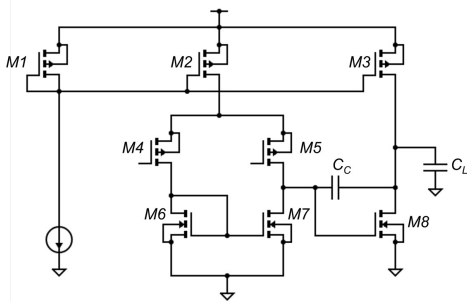


148

Transfer Performance Modeling across Technology Nodes for the Same Circuit

149

- ▶ Transfer learning is applied on models trained in 180nm for the performance modeling of an op-amp in 65nm
- ▶ Transfer learning significantly improves the sample efficiency for circuit performance modeling with simulation-based sizing data
- ▶ Up to 50% improvement in MAE on test data



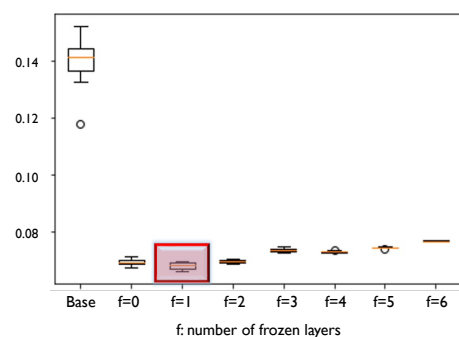
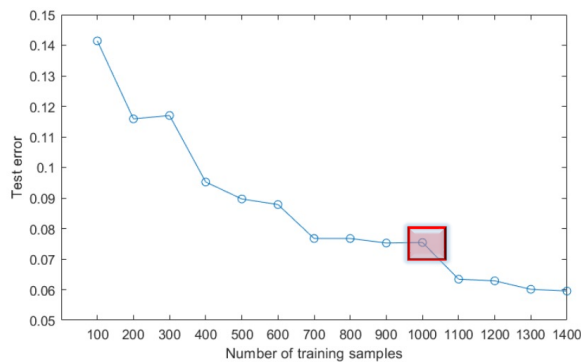
Reference: Z. Wu and I. Savidis, "Transfer Learning for Reuse of Analog Circuit Sizing Models Across Technology Nodes," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1-5, 2022.

149

Comparison of Sample Efficiency for Training of the Gain Predictor

150

- ▶ Standalone training requires 1000 training samples to achieve test error of 0.076
- ▶ Transfer learning requires 100 samples to achieve test error of 0.07



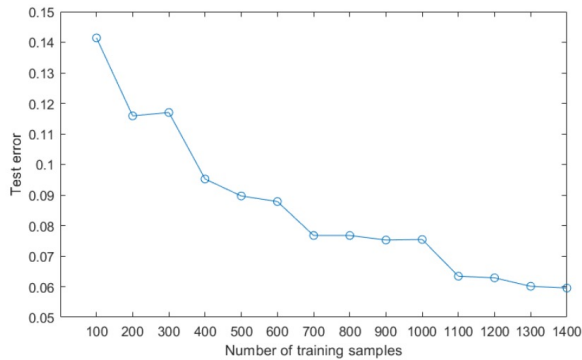
Reference: Z. Wu and I. Savidis, "Transfer Learning for Reuse of Analog Circuit Sizing Models Across Technology Nodes," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1-5, 2022.

150

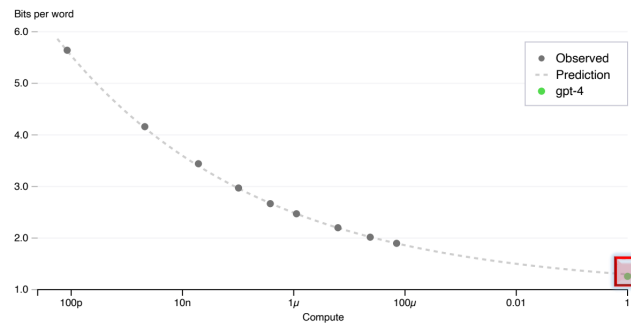
Predictive Scaling of ML models: Analogy with GPT 4

151

- ▶ Large training runs do not allow fine-grained model tuning
- ▶ Objective: predict performance of ML models as a function of dataset size
- ▶ Loss of language models well approximates power laws in terms of compute used to train the model
 - ▶ $Loss(C) = \alpha * Compute^b$



OpenAI codebase next word prediction



Reference: GPT4 Technical Report

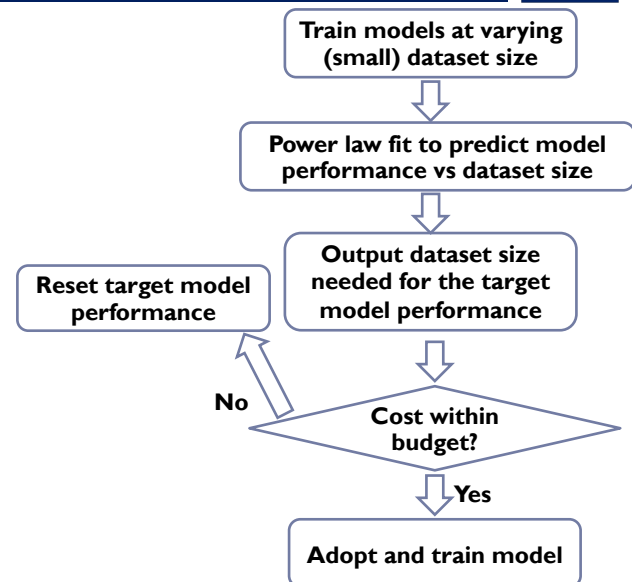


151

Predictive Scaling for ML-based Circuit Models

152

- ▶ Currently no standards on circuit data for analog EDA
 - ▶ Data is generated by each group with proprietary PDK for various design applications
- ▶ Predictive scaling is needed to **apply ML for analog EDA in scale**
 - ▶ Assurance and budget control of ML-based circuit models

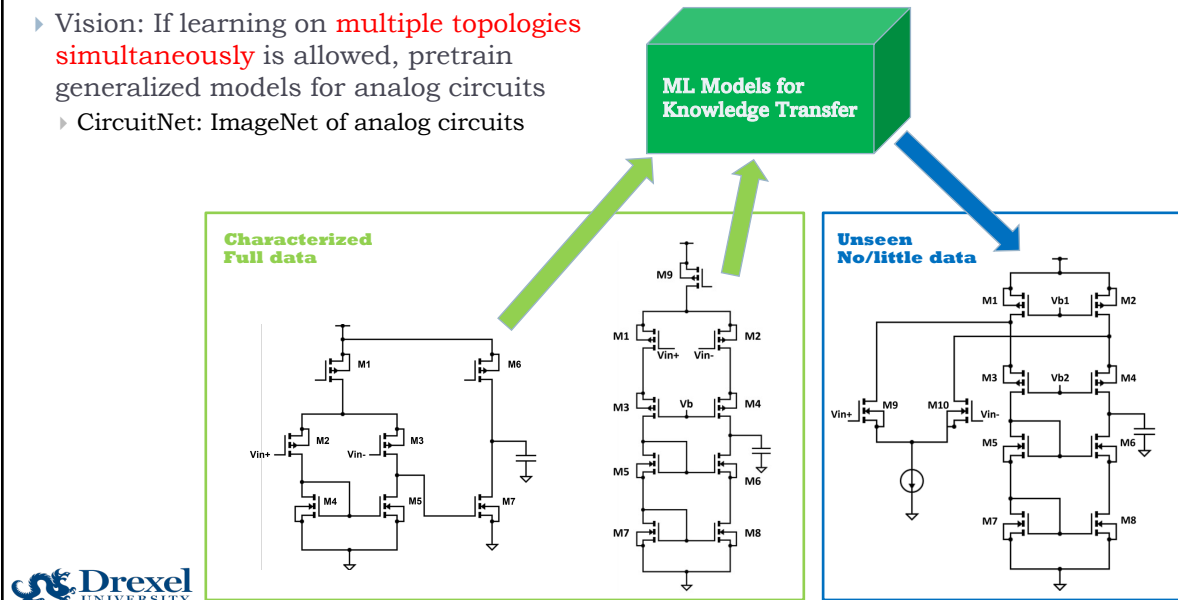


152

Transfer Performance Modeling across Analog Topologies

153

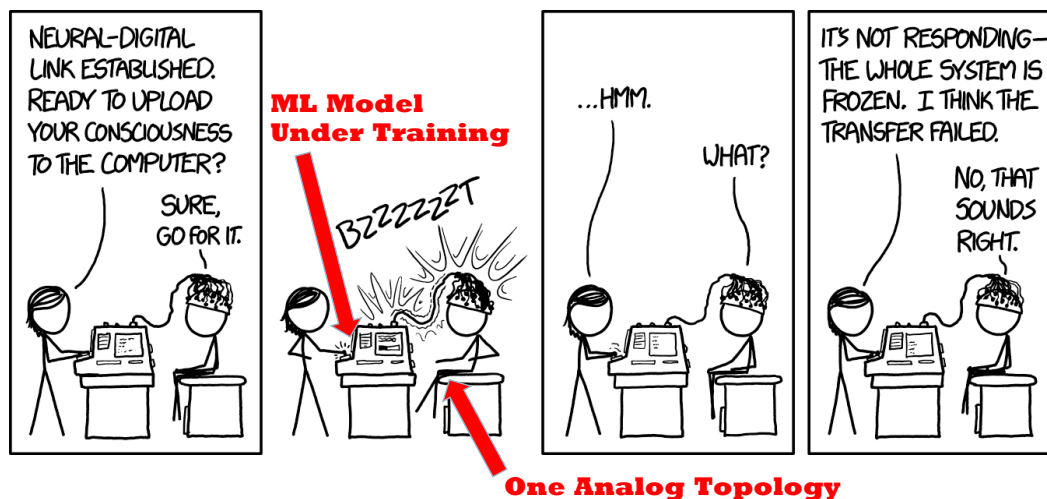
- Vision: If learning on **multiple topologies simultaneously** is allowed, pretrain generalized models for analog circuits
 - CircuitNet: ImageNet of analog circuits



153

Is Transferring Predictions Across Analog Topologies Even Possible?

154



Drexel UNIVERSITY Source: xkcd.com

154

Challenges of Transferring Predictions Across Analog Topologies

155

- ▶ Each analog topology represents a unique mapping from the design space to performance space
 - ▶ Includes tradeoff considerations
- ▶ Traditional learning algorithms only work for a single topology if trained on device features
 - ▶ Different topologies $\Rightarrow$ different number of devices $\Rightarrow$ different feature dimensionality
- ▶ Therefore, models usually do not apply if circuit topology changes
 - ▶ Requires new data for a new circuit
- ▶ To transfer across topologies, first need models that learn topological information

Solution: Train Graph Neural Networks on Circuit Graphs



155

Transfer Performance Models Across Four Op-amp Topologies

156

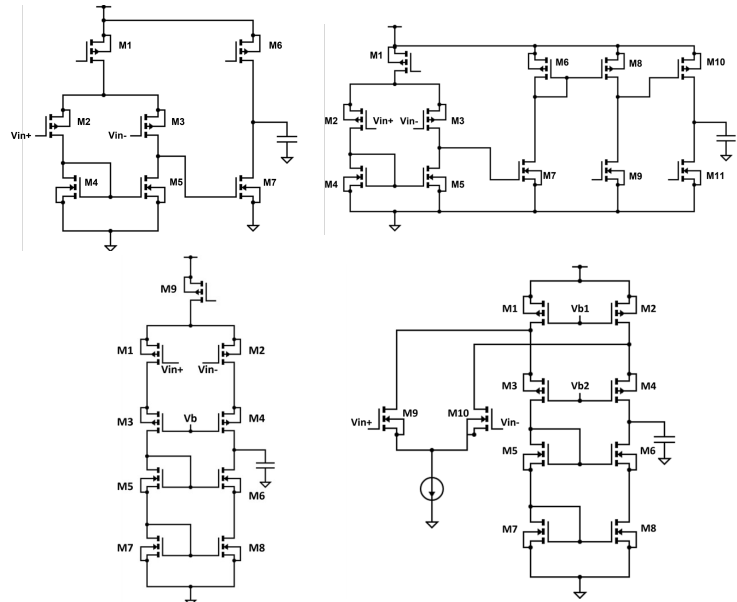
▶ Four op-amp topologies

- ▶ Two-stage
- ▶ Three-stage
- ▶ Telescopic cascode
- ▶ Folded cascode

▶ Device Features:

- ▶ Transistor sizing
- ▶ Transistor type

	Design Variables
Two-stage	W_1, W_2, W_4, W_6, W_7
Three-stage	$W_1, W_2, W_4, W_6, W_7, W_9, W_{10}, W_{11}$
Folded	W_1, W_3, W_5, W_7, W_9
Telescopic	W_1, W_3, W_5, W_7, W_9

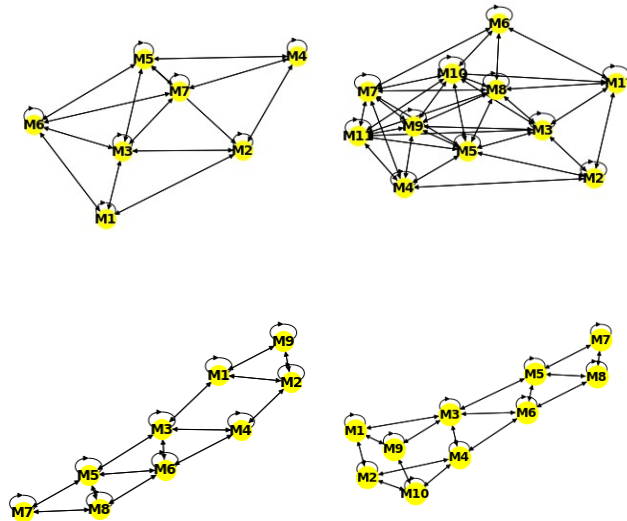
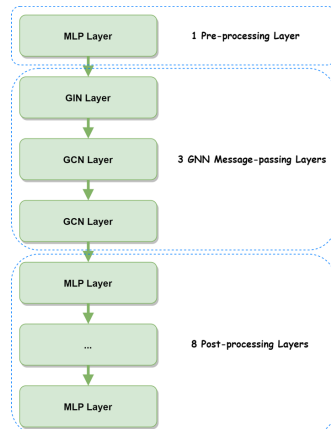


156

Transfer Learning in Two Design Scenarios

157

- ▶ Scenario 1: Zero-shot learning
 - ▶ No data provided for the target topology
 - ▶ Apply GNN
- ▶ Scenario 2: Few-shot learning
 - ▶ Small dataset of 100 points provided for the target topology
 - ▶ Apply GNN and transfer learning



157

Transfer Performance Modeling by Applying GNN and Transfer Learning

158

- ▶ Train GNN for three amplifier topologies, test on the fourth topology
- ▶ Zero-shot learning: GNNs provide coarse estimates of the circuit performance
 - ▶ GNNs result in less test errors than the baseline ANNs for 14 of the 20 cases
- ▶ Few-shot learning: fine-tuned GNNs with transfer learning provide an average reduction of 70.6% in test error (RMSE) as compared to ANN models

	Two-stage as Test	Three-stage as Test	Folded as Test	Telescopic as Test
Power	-83% / -88%	+300% / -75%	+47% / -10%	+449% / -94%
Gain	-48% / -77%	+29% / -61%	-21% / -66%	-19% / -78%
Slew Rate	-62% / -74%	-80% / -82%	-97% / -99%	-98% / -99%
CMRR	-50% / -70%	-55% / -61%	-43% / -60%	-78% / -81%
PSRR	-15% / -67%	+59% / -39%	+33% / -60%	-67% / -71%

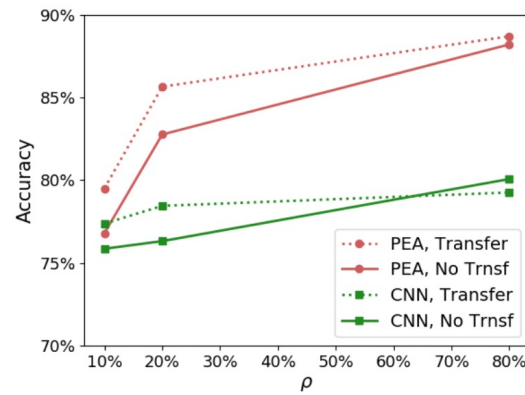
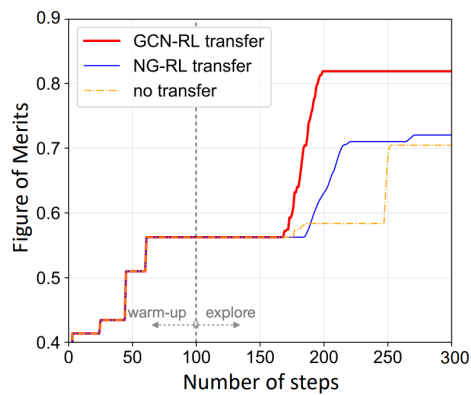
* Scenario 1 / Scenario 2

158

Benefits of Applying Transfer Learning for the Modeling of Analog Circuits in Past Works

159

- ▶ Transfer learning produces higher FOM solutions when reinforcement learning is applied for sizing
- ▶ Transferred knowledge of cascode OTA improves model performance in predicting placement quality of current mirror OTA



Reference: 1. H. Wang, et. al., "GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning," Proceedings of the IEEE/ACM Design Automation Conference, pp. 1-6, 2020
 2. Y. Li et al., "A Customized Graph Neural Network Model for Guiding Analog IC Placement," Proceedings of the IEEE/ACM International Conference On Computer Aided Design, pp. 1-9, 2020



159

Outline of Presentation

160

- ▶ Background introduction
- ▶ Machine learning techniques for analog EDA
- ▶ Optimization techniques for analog EDA
- ▶ Case studies
- ▶ Conclusions



160

Future Directions

161

- ▶ Improve reliability, robustness, and interpretability of ML models for analog EDA
 - ▶ More mature and standardized flow of analog synthesis
- ▶ Strong AI for analog synthesis
 - ▶ Current ML models identify correlation instead of causality among data
 - ▶ Learning causality requires expert knowledge
 - ▶ Ideal if AI understands human design thinking while discovering new rules for the synthesis of analog circuits
- ▶ Meta-learning
 - ▶ Learning what to learn
 - ▶ Learn parameter values for base (pre-trained) models for circuit tasks
 - ▶ Learning which model to learn
 - ▶ Auto select the ML and optimization algorithms best suited for a given circuit task
 - ▶ Learning how to learn
 - ▶ Auto hyperparameter tuning of ML models and generation of pipeline for analog EDA
 - Parsing of standard circuit files (SPICE, DSPF, LEF, DEF...)



161

Summary of AI-driven Analog EDA

162

- ▶ ML is applied to improve, not replace heuristics in analog synthesis flow
 - ▶ Heuristics with procedural synthesis still dominate latest analog EDA tools
- ▶ Benefits brought by ML for analog EDA:
 - ▶ Reduced simulations required, reduced turnaround time
 - ▶ Design space exploration
 - ▶ Prediction of parasitic impedances, reliability and variability
 - ▶ Guide optimization or direct generation of schematic and layout design
 - ▶ Migration and reuse of past designs
- ▶ Requirement on ML-based circuit models:
 - ▶ Sample efficiency
 - ▶ Generalization
 - ▶ Transferability
- ▶ Optimization is backbone of automated analog design flow
 - ▶ Need to handle higher device count and more restrictive design rules

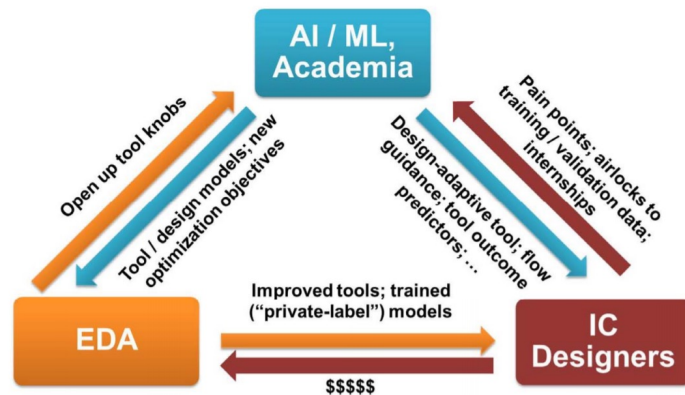


162

Conclusions

163

- ▶ EDA tools are like autonomous vehicles
 - ▶ Currently, driver control/attention is still required
- ▶ Analog EDA tools are like autonomous vehicles to be driven in more challenging road conditions
- ▶ Level of automation will keep rising
- ▶ More collaborations needed between circuit design, academia and industry

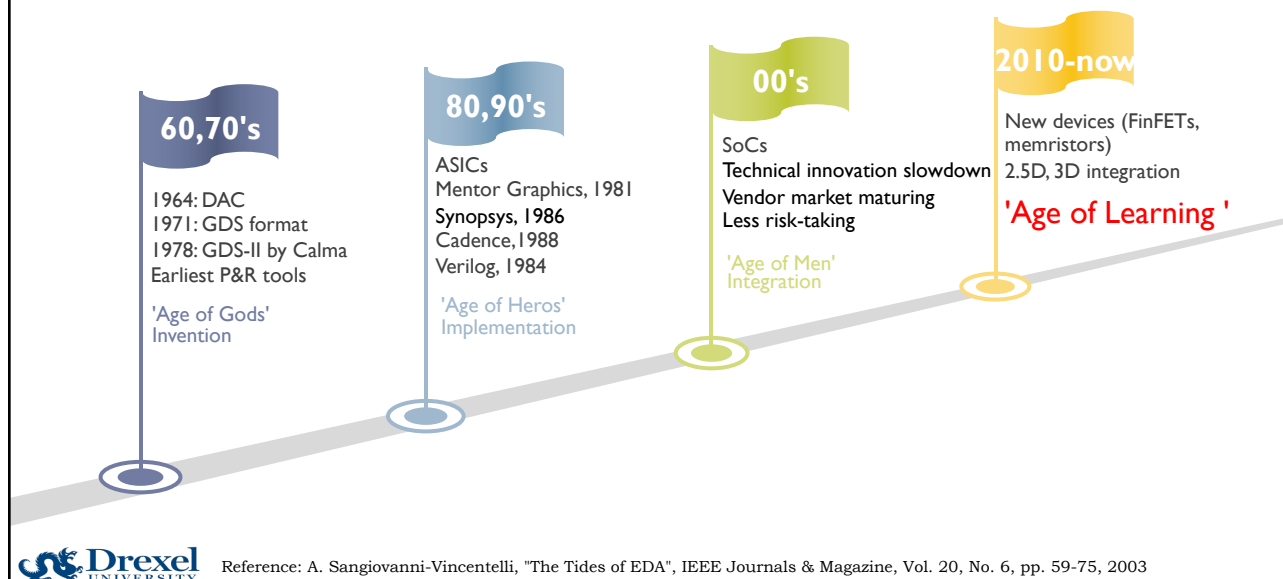


Reference: A. B. Kahng, "Machine Learning Applications in Physical Design: Recent Results and Directions", Proceedings of the International Symposium on Physical Design, pp. 68-73, 2018

163

A Remembrance of the Past: Timeline of (Digital) EDA Development

164



Reference: A. Sangiovanni-Vincentelli, "The Tides of EDA", IEEE Journals & Magazine, Vol. 20, No. 6, pp. 59-75, 2003

164



165



DESIGN & ANALYSIS LAB

<http://ice.ece.drexel.edu>


165

References

166

- ▶ A. Sangiovanni-Vincentelli, "The Tides of EDA," IEEE Journals & Magazine, Vol. 20, No. 6, pp. 59-75, 2003
- ▶ J. Scheible, "Optimized is Not Always Optimal," Proceedings of the International Symposium on Physical Design, pp. 151-158, 2022
- ▶ B. Razavi, "Design of Analog CMOS Integrated Circuits," McGraw-Hill, 2001
- ▶ J. Crossley, et al., "BAG: A Designer-oriented Integrated Framework For The Development Of AMS Circuit Generators," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 74-81, Nov. 2013
- ▶ Q. Zhang et al., "An Open-Source And Autonomous Temperature Sensor Generator Verified With 64 Instances In SkyWater 130 nm For Comprehensive Design Space Exploration," IEEE Solid-State Circuits Letters, Vol. 5, No.1, pp. 174-177, 2022
- ▶ S. Su, et al., "Analog/Mixed-Signal Circuit Synthesis Enabled by the Advancements of Circuit Architectures and Machine Learning Algorithms," Proceedings of the Asia and South Pacific Design Automation Conference, pp.100-107, 2022
- ▶ A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics," IEEE Access, Vol. 6, No.1, pp. 39117-39138, 2018
- ▶ X. Shangguan, H. Ma, A. C. Cangellaris and X. Chen, "Effect of Sampling Method on the Regression Accuracy for a High-Speed Link Problem," Proceedings of the IEEE Conference on Electrical Performance of Electronic Packaging and Systems, pp.1-3, 2021
- ▶ Z. Wu and I. Savidis, "CALT: Classification with Adaptive Labeling Thresholds for Analog Circuit Sizing," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 49-54, 2020
- ▶ T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Automated Extraction Of Expert Knowledge In Analog Topology Selection And Sizing," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 392-395, 2008
- ▶ T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Simultaneous Multi-Topology Multi-Objective Sizing Across Thousands of Analog Circuit Topologies," Proceedings of the ACM/IEEE Design Automation Conference, pp. 944-947, 2007
- ▶ T. Brown, et. al, "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems, Vol. 33, No. 1, pp.1877-1901, 2022
- ▶ J. K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?," Proceedings of the International Conference on Learning Representations, pp. 1-17, 2019
- ▶ K. Li, J. Malik, "Learning to optimize," Proceedings of the International Conference on Learning Representations, 2017



166

References

167

- ▶ A. F. Budak, Z. Jiang, K. Zhu, A. Mirhoseini, A. Goldie and D. Z. Pan, "Reinforcement Learning for Electronic Design Automation: Case Studies and Perspectives," Proceedings of the Asia and South Pacific Design Automation Conference pp.500-505, 2022
- ▶ W. Lyu, et al., "An Efficient Bayesian Optimization Approach For Automated Optimization Of Analog Circuits," IEEE Transactions on Circuits and Systems, Vol. 65, No. 6, pp. 1954–1967, 2018
- ▶ N. Lourenço and N. Horta, "GENOM-POF: Multi-objective Evolutionary Synthesis Of Analog ICs With Corners Validation," Proceedings of the International Conference on Genetic and Evolutionary Computation, pp. 1119– 1126, 2012
- ▶ R. Lourenço, N. Lourenço, N. Horta, "AIDA-CMK: Multi-Algorithm Optimization Kernel Applied To Analog IC Sizing," Springer, 2015
- ▶ F. Silveira, D. Flandre and P. G. A. Jespers, "A gm/Id Based Methodology For The Design Of CMOS Analog Circuits And Its Application To The Synthesis Of A Silicon-on-insulator Micropower OTA," IEEE Journal of Solid-State Circuits, Vol. 31, No. 9, pp. 1314–1319, 1996
- ▶ R. Frevert, et al., "Modeling and Simulation for RF System Design," Springer, pp.291, 2005
- ▶ Z. Wu and I. Savidis, "Variation-aware Analog Circuit Sizing with Classifier Chains," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 1–6, 2021
- ▶ T. Dhar, et al., "ALIGN: A System for Automating Analog Layout," IEEE Design & Test, Vol. 38, No. 2, pp. 8-18, 2021
- ▶ H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The Sizing Rules Method For Analog Integrated Circuit Design," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 343–349, 2001
- ▶ M. Liu, et al., "S3DET: Detecting System Symmetry Constraints for Analog Circuits with Graph Similarity," Proceedings of the Asia and South Pacific Design Automation Conference, pp. 193-198, 2020
- ▶ K. Kunal, et al., "GANA: Graph Convolutional Network Based Automated Netlist Annotation for Analog Circuits," Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, pp. 55–60, 2020
- ▶ K. Zhu, et al., "GeniusRoute: A New Analog Routing Paradigm Using Generative Neural Network Guidance," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp.1-8, 2019
- ▶ B. Xu, et al., "MAGICAL: Toward Fully Automated Analog IC Layout Leveraging Human and Machine Intelligence: Invited Paper," Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp.1-8, 2019
- ▶ A. Mirhoseini, et al., "A Graph Placement Methodology for Fast Chip Design," Nature, No. 594, pp. 207–212, 2021



167

References

168

- ▶ G. Pradipta, V. A. Chhabria, and S. S. Sapatnekar, "A Machine Learning Based Parasitic Extraction Tool," 2019
- ▶ B.W. Shook. et.al, "MLParest: Machine Learning based Parasitic Estimation for Custom Circuit Design," Proceedings of the ACM/IEEE Design Automation Conference, pp. 1–6, 2020
- ▶ H. Ren, G. F. Kokai, W. J. Turner and T. Ku. "ParaGraph: Layout Parasitics And Device Parameter Prediction Using Graph Neural Networks," Proceedings of the ACM/IEEE Design Automation Conference, pp. 1–6, 2020
- ▶ Z. Wu and I. Savidis, "Transfer Learning for Reuse of Analog Circuit Sizing Models Across Technology Nodes," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1–5, 2022
- ▶ Z. Wu and I. Savidis, "Transfer of Performance Models Across Analog Circuit Topologies with Graph Neural Networks," Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, pp. 159-165, 2022
- ▶ H. Wang, et. al., "GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning," Proceedings of the IEEE/ACM Design Automation Conference, pp. 1–6, 2020
- ▶ Y. Li et al., "A Customized Graph Neural Network Model for Guiding Analog IC Placement," Proceedings of the IEEE/ACM International Conference On Computer Aided Design, pp. 1-9, 2020
- ▶ A. B. Kahng, "Machine Learning Applications in Physical Design: Recent Results and Directions," Proceedings of the International Symposium on Physical Design, pp. 68-73, 2018
- ▶ M. Simonovsky and N. Komodakis, "Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs," Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 29–38, Jul. 2017
- ▶ D. Kingma and M. Welling, "Auto-Encoding Variational Bayes", Proceedings of the International Conference on Learning Representations, 2014
- ▶ I. Goodfellow, et al., "Generative Adversarial Nets", Proceedings of the International Conference on Neural Information Processing System, Vol. 2, No.1, pp.,2672-2680, 2014
- ▶ P. Shrestha and I. Savidis, "Graph Representation Learning for Parasitic Impedance Prediction of the Interconnect," Proceedings of the IEEE International Symposium on Circuits and Systems, 2023
- ▶ A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure," Springer, 2011



168