Learning to Design Circuits

Hanrui Wang[*]
EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
hanrui@mit.edu

Jiacheng Yang[*]
EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
jcyoung@mit.edu

Hae-Seung Lee
EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
hslee@mtl.mit.edu

Song Han
EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
songhan@mit.edu

Abstract

Analog IC design relies on human experts to search for parameters that satisfy circuit specifications with their experience and intuitions, which is highly labor intensive, time consuming and suboptimal. Machine learning is a promising tool to automate this process. However, supervised learning is difficult for this task due to the low availability of training data: 1) Circuit simulation is slow, thus generating large-scale dataset is time-consuming; 2) Most circuit designs are propitiatory IPs within individual IC companies, making it expensive to collect large-scale datasets. We propose Learning to Design Circuits (L2DC) to leverage reinforcement learning that learns to efficiently generate new circuits data and to optimize circuits. We fix the schematic, and optimize the parameters of the transistors automatically by training an RL agent with no prior knowledge about optimizing circuits. After iteratively getting observations, generating a new set of transistor parameters, getting a reward, and adjusting the model, L2DC is able to optimize circuits. We evaluate L2DC on two transimpedance amplifiers. Trained for a day, our RL agent can achieve comparable or better performance than human experts trained for a quarter. It first learns to meet hard-constraints (eg. gain, bandwidth), and then learns to optimize good-to-have targets (eg. area, power). Compared with grid search-aided human design, L2DC can achieve $250\times$ higher sample efficiency with comparable performance. Under the same runtime constraint, the performance of L2DC is also better than Bayesian Optimization.

1 Introduction

Analog circuits process continuous signals, which exist in almost all electronics systems and provide important function of interfacing real-world signals with digital systems. Analog IC design has a large number of circuit parameters to tune, which is highly difficult for several reasons. First, the relationship between the parameters and performance is subtle and uncertain. Designers have few explicit patterns or deterministic rules to follow. Analog Circuits Octagon [1] characterizes strong coupled relations among performance metrics. Therefore, improving one aspect always incurs deterioration of another. A proper and intelligent trade-off between those metrics requires rich design experience and intuitions. Moreover, simulation of circuits is slow, especially for complex circuits such as ADCs, DACs, and PLLs. That makes the random search or exhaustive search impractical.
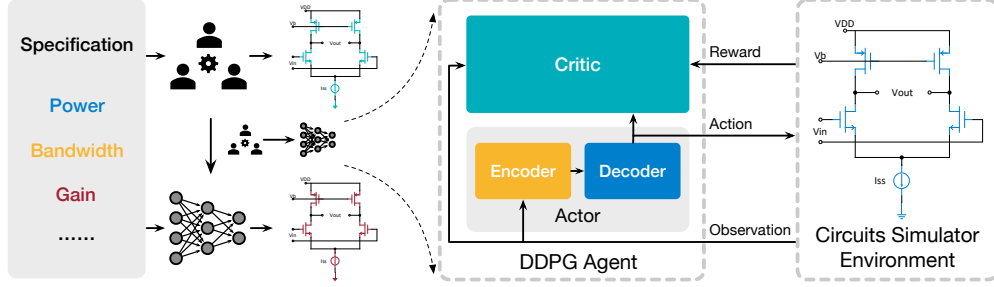
---

[*]Equal Contribution.

Figure 1: Learning to Design Circuits (L2DC) Method Overview.

There exist several methods to automate the circuit parameter search process. Particle swarm intelligence [2] is a popular approach. However, it is easy to fall into local optima in high-dimensional space and also suffers from a low convergence rate. Moreover, simulated annealing [3] is also utilized, but repeatedly annealing can be very slow and easy to fall into the local minimum. Although evolutionary algorithm [4] can be used to solve the optimization problem, the process is stochastic and lacks reproducibility. In [5], researchers also proposed model-based simulation-based hybrid method and utilized Bayesian Optimization [6] to search for parameter sets. However, the computational complexity of BO is prohibitive, making the runtime very long.

Machine learning is another promising method to address the above difficulties. However, supervised learning requires large scale dataset which consumes long time to generate. Besides, most of the analog IPs are proprietary, not available to the public. Therefore, we introduce L2DC method, which leverages reinforcement learning (RL) to generate circuits data by itself and learns from the data to search for best parameters. We train our RL agents from scratch without giving it any rules about circuits. In each iteration, the agent obtains observations from the environment, produces an action (a set of parameters) to the circuit simulator environment, and then receives a reward as a function of gain, bandwidth, power, area, *etc*. The observations consist of DC operating points, AC magnitude and phase responses and also transistors' states, obtained from simulation tools such as Hspice and Spectre. The reward is defined to optimize the desired Figures of Merits (FOM) composed of several performance metrics. By maximizing the reward, RL agent can optimize the circuit parameters.

Experimental results on two different circuits environments show that L2DC can achieve similar or better performance than human experts, Bayesian Optimization and random search. L2DC also has $250\times$ higher sample efficiency compared to grid search aided human expert design. The contributions of this paper are: 1) A reinforcement learning based analog circuit optimization method. It is a learning-based method that updates optimization strategy by itself with no need for empirical rules; 2) A sequence-to-sequence model to generate circuit parameters, which serves as the actor in the RL agent; 3) Our method achieves more than $250\times$ higher sample efficiency comparing to grid search based human design and gets comparable or better results. Under the same runtime constraint, our method can get better circuit performance than Bayesian Optimization.

2 Methodology

2.1 Problem Definition

The design specification of an analog IC contain hard constraints and optimization targets. For hard constraints, designers only need to meet them with no need for any over-optimization. For optimization targets, the rule is "the larger the better" or "the smaller the better". For optimization targets, there also exist thresholds specifying the minimum performance designers need to achieve.

Formally, we denote $\mathbf{x} \in \mathbb{R}^n$ as the parameters of H components, $\mathbf{y} \in \mathbb{R}^m$ as the specs to be satisfied, including hard constraints and thresholds for optimization targets. We assume the mapping $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is the circuit simulator which computes m performance metrics given n parameters. We define a ratio q for each spec c to measure to which extent the circuit satisfies the spec. If the metric should be larger than the spec c, $q_c(\mathbf{x}) = f_c(\mathbf{x})/y_c$. Otherwise $q_c(\mathbf{x}) = y_c/f_c(\mathbf{x})$. Then analog IC optimization can be formalized as a constrained optimization problem, that is, to maximize the sum of $q_c(\mathbf{x})$ of the optimization targets with all of the hard constraints being satisfied.

2.2 Multi-Step Simulation Environment

We present an overview of our L2DC method in Figure 1. L2DC is able to find the optimized parameters by several epochs of interactions with the simulation environment. Each epoch contains T
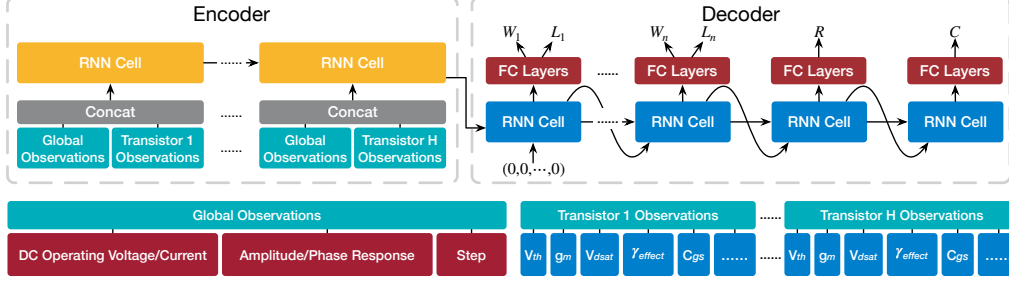
Figure 2: We use sequence to sequence model to generate circuit parameters (Top). Global and local observations for the RL agent (Bottom).

steps. For each step i, the RL agent takes an observation $\mathbf{o_i}$ from the environment (it can be regarded as state $\mathbf{s_i}$ in our environments), outputs an action $\mathbf{a_i}$ and then receives an reward r_i. By learning from history, the RL agent is able to optimize the expected cumulative reward.

The simulation of a circuit is essentially an one-step process, meaning that the state information including voltage and current of the circuit's environment cannot be directly facilitated. To effectively feed the information to RL agent, we purpose the following multi-step environment.

Observations   As illustrated in Figure 2, at each step i, the observation $\mathbf{o_i}$ is separated into global observations and each transistor's own observations. The global observations contain high-level features of the simulation results, including the DC operating point voltage and current at each node, AC amplitude/phase responses and a one-hot vector encoding the current environment step. Local observations are the features of the i-th transistor, containing $V_{th}, g_m, V_{dsat}, \gamma_{effect}, I_D$, capacitance between transistor pins and so on. The initial values of all global observations and local transistor observations are set to zeros.

Action Space   Supposing there are n parameters to be searched, the reinforcement learning agent would output a normalized joint action $\overline{\mathbf{a_i}} \in [-1, 1]^n$ as the predicted parameters of each component at each step i. Then the actions $\overline{\mathbf{a_i}}$ are scale up to $\mathbf{a_i}$ according to the maximum/minimum constraints of each parameter $[\mathbf{pmin}, \mathbf{pmax}]$ where $\mathbf{a_i}$ contains the widths and lengths of transistors, capacitance of capacitors and resistance of resistors.

Reward Function   After the reinforcement learning agent outputs circuit parameters $\mathbf{a_i}$, the simulation environment $\mathbf{f}$ will benchmark on these parameters, generating simulation results of various performance metrics. We gather the metrics together as a scalar score d_i. Denote $K_1(\mathbf{x})$ as the sum of q_c of those hard constraints and $K_2(\mathbf{x})$ of those optimization target. Then d_i is defined as

$$d(\mathbf{x}) = \begin{cases} K_1(\mathbf{x}) + \alpha * K_2(\mathbf{x}) + e_0 & \text{if some hard constraints are not satisfied} \\ K_2(\mathbf{x}) + e_1 & \text{if all hard constraints are satisfied} \end{cases} \tag{1}$$

When the hard-constraints in the spec are not satisfied, DDPG will optimize hard-constraint requirements and optionally optimize optimization target requirements according to the coefficient α. When all the hard-constraints are satisfied, DDPG will only focus on optimization targets. e_0 and e_1 are two constants. They are used to make sure the scores after hard-constraints are satisfied are higher than those before hard-constraints are satisfied. To fit the reinforcement learning framework where the cumulative reward is maximized, the reward for the i-th step r_i is defined as the finite $d_i - d_{i-1}$.

2.3 DDPG Agent

As shown in Figure 2, the DDPG [7, 8] actor forms an encoder-decoder framework [9] which translates the observations to the actions. The order which we follow to feed the observations of transistors, is the order of signal propagation through a certain path from input ports to output ports, intuited by fact that the former components influence latter ones. The decoder generates transistor W and L in the same order as well. To explore the search space, we apply truncated uniform distribution as noise to each output. Namely, $\tilde{\mathbf{a}}_\mathbf{i} \sim \mathcal{U}(\max(\mathbf{a_i} - \boldsymbol{\sigma}, -1), \min(\mathbf{a_i} + \boldsymbol{\sigma}, 1))$, where $\boldsymbol{\sigma} \in [0, 1]$ denotes the noise volume. Besides, we also find parameter noise [10] improves the performance. For critic network, we simply use a multi-layer perceptron estimating the expected cumulative reward of the current policy.

3 Experimental Results

3.1 Three-Stage Transimpedence Amplifier

The first environment is a three-stage transimpedence amplifier from the final project of Stanford EE214A [11]. The schematic of the circuit is shown in Figure 3. We compare L2DC with random
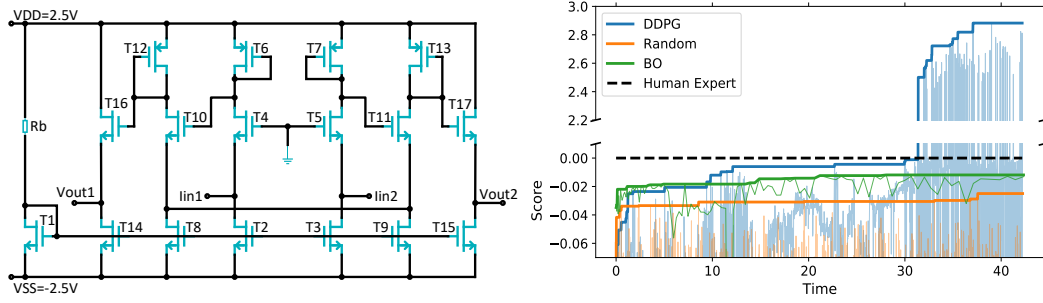
Figure 3: Left: Schematic of Three-stage transimpedence amplifier. Right: Learning curves of three-stage transimpedence amplifier.

Table 1: Results on three-stage transimpedence amplifier. Under the same runtime constraint, random search and MACE cannot meet the spec hard-constraints (as marked in red); DDPG is able to satisfy all the spec hard-constraints with smallest gate area, thus achieving highest score. The sample efficiency of DDPG is 250 × higher than human design.

	Number of Simulations (Same Runtime)	Sample Efficiency	Bandwidth (MHz)	Gain (kΩ)	Power (mW)	Gate Area (μm^2)	Score
Spec [11]	–	–	90.0	20.0	3.00	–	–
Human Expert[12]	10,000,000	1	90.1	20.2	**1.37**	211	0.00
Random	40,000	–	57.3	20.7	1.37	146	-0.02
DDPG	40,000	**250**	**92.5**	**20.7**	2.50	**90**	**2.88**
MACE [13]	1,160	–	72.5	21.1	4.25	130	-0.01

search, an grid search aided human expert design proposed by a PhD student in EE214A class as well as MACE [13], a Bayesian Optimization based algorithm for analog circuit optimization. The batch size of MACE is 10 and we use 50 samples for initialization.

We run DDPG, MACE and random search, each for about 40 hours. The comparison results are shown in Table 1. The learning curves are shown in Figure 3. Random search is not able to meet the bandwidth hard-constraint because there are seventeen transistors making the environment very complex and design space very large. MACE is also unable to meet the bandwidth and power hard-constraints. DDPG's design can meet all the hard-constraints and has slightly higher power but smaller gate area than the human expert design, so it achieves highest score. The power consumption of DDPG, though slightly higher than human design, can satisfy the course spec constraint. Moreover, the number of simulations of DDPG is 250× fewer than the grid search aided human design, demonstrating high sample efficiency of our L2DC method.
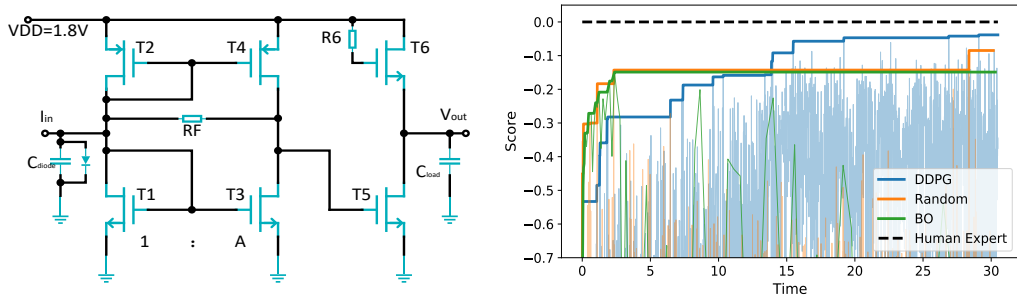

Figure 4: Left: Schematic of two-stage tranimpedence amplifier. Right: Learning curves of two-stage transimpedence amplifier.

4

Table 2: Results on two-stage transimpedence amplifier. Under the same runtime constraint, random and MACE are unbale to meet the noise hard-constraint (as marked in red); DDPG can satisfy all the spec hard-constraints and achieve 97.143% bandwidth of computer-aided human expert design. The sample efficiency of DDPG is 25 × higher than human design.

	NO. of Simu. (Same Runtime)	Sample Efficiency	Noise (pA/$\sqrt{\text{Hz}}$)	Gain (dBΩ)	Peaking (dB)	Power (mW)	Gate Area (μm^2)	Band-width (GHz)	Score
Spec [14]	–	–	19.3	57.6	1.000	18.0	–	maximize	–
Human Expert[15]	1,289,618	1	**18.6**	57.7	0.927	8.11	6.17	**5.95**	**0.00**
Random	50,000	–	19.8	58.0	0.488	4.39	2.93	5.60	-0.08
DDPG	50,000	**25**	19.2	**58.1**	**0.963**	**3.18**	**2.61**	5.78	-0.03
MACE [13]	880	–	19.6	58.6	0.629	4.24	5.69	5.16	-0.15

3.2 Two-stage Transimpedence Amplifier

The second environment is a two-stage transimpedence amplifier. The schematic of the circuit is shown in Figure 4. The circuit is from Stanford EE214B design contest [14]. The contest specifies noise, gain, peaking, power as hard-constraints and bandwidth as optimization target.

We run DDPG, MACE and random search, each for about 30 hours. In Table 2, we compare DDPG result with random search, MACE, and human expert design which applies a g_m/I_D methodology to conduct design space search. The learning curves are shown in Figure 4. Human expert design achieves 6 GHz bandwidth with all hard constraints being satisfied therefore receives the "Most Innovative Design" award. Random cannot meet the noise hard constraints. MACE cannot meet the noise hard constraint either. DDPG meets all the hard constraints and achieves 5.78 GHz bandwidth which is already 97.143% of the human result, while the sample efficiency of L2DC is 25× better than human expert design.

4 Discussion

As shown in Figure 5, we plot the curves of performance metrics v.s. learning steps in the three-stage transimpedence amplifier. The vertical dash line indicates the step when hard-constraints are satisfied. We can observe that power goes up and then goes down; bandwidth goes up and then stays constant; gain goes up and then remains. Therefore, from the RL agent's point of view, it firstly sacrifice power to increase hard-constraints (bandwidth and gain). After those two metrics are met, RL agents tried to keep the bandwidth and gain constant and starts to optimize power which is a soft optimization target. From this phenomenon, we can infer that RL agent has learnt some strategies in analog circuit optimization.

5 Conclusion

We propose L2DC that leverages reinforcement learning to automatically optimize circuit parameters. Comparing to supervised learning, it does not need large scale training dataset which is difficult to obtain due to long simulation time and IP issues. We evaluate our methods on two different transimpedance amplifiers circuits. After iteratively getting observations, generating a new set of parameters by itself, getting a reward, and adjusting the model, L2DC is able to design circuits with better performance than both random search, Bayesian Optimization and human experts. L2DC works well on both two-stage transimpedence amplifier as well as complicated three-stage amplifier, demonstrating its generalization ability. Compared with grid search aided human design, L2DC can achieve comparable performance, with about **250×** higher sample efficiency. Under the same runtime constraint, L2DC can also get better circuit performance than Bayesian Optimization.

6 Acknowledgements
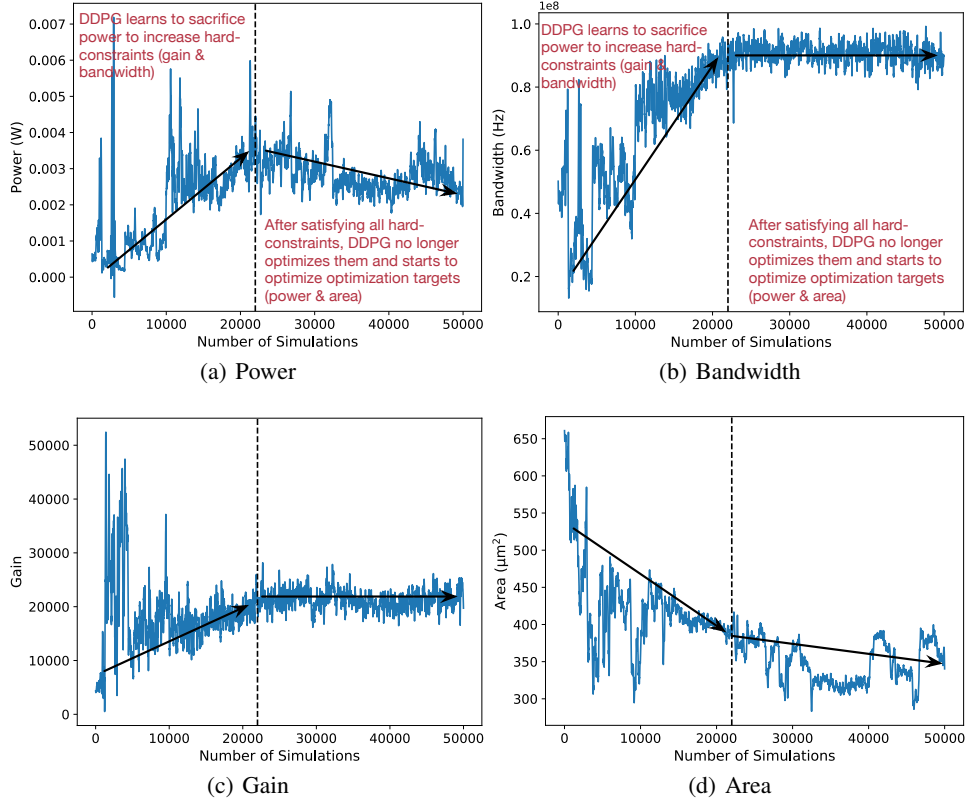
(a) Power

(b) Bandwidth

(c) Gain

(d) Area

Figure 5: The learning curve of the circuit RL agent. The vertical dashed line is the time when those hard-constraints (gain, bandwidth) are satisfied. RL agent learns that it should first optimize hard-constraints (for example, obtaining more gain and more bandwidth at the cost of sacrificing more power), then improve those soft optimization targets (Fig.(a) and (d): decrease the power and area) while keeping hard-constraints constant (Fig.(b) and (c): maintains gain and bandwidth).

References

[1] Razavi Behzad. Design of analog cmos integrated circuits. *International Edition*, 400, 2001. 1

[2] Prakash Kumar Rout, Debiprasad Priyabrata Acharya, and Ganapati Panda. A multiobjective optimization based fast and robust design methodology for low power and low phase noise current starved vco. *IEEE Transactions on Semiconductor Manufacturing*, 27(1):43–50, 2014. 2

[3] Rodney Phelps, Michael Krasnicki, Rob A Rutenbar, L Richard Carley, and James R Hellums. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(6):703–717, 2000. 2

[4] Bo Liu, Francisco V Fernández, Georges Gielen, R Castro-López, and Elisenda Roca. A memetic approach to the automatic design of high-performance analog integrated circuits. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 14(3):42, 2009. 2

[5] Wenlong Lyu, Pan Xue, Fan Yang, Changhao Yan, Zhiliang Hong, Xuan Zeng, and Dian Zhou. An efficient bayesian optimization approach for automated optimization of analog circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(6):1954–1967, 2018. 2

[6] Martin Pelikan, David E Goldberg, and Erick Cantú-Paz. Boa: The bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, pages 525–532. Morgan Kaufmann Publishers Inc., 1999. 2

[7] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A Ried-miller. Deterministic Policy Gradient Algorithms. *ICML*, 2014. 3

[8] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, cs.LG, 2015. 3

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. *arXiv.org*, September 2014. 3

[10] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter Space Noise for Exploration. June 2017. 3

[11] Robert Dutton and Boris Murmann. Stanford ee214a - fundamentals of analog integrated circuit design final project. 3, 4

[12] Danny Bankman. Stanford ee214a - fundamentals of analog integrated circuit design, design project report. 4

[13] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International Conference on Machine Learning*, pages 3312–3320, 2018. 4, 5

[14] Boris Murmann. Stanford ee214b - advanced analog integrated circuits design, design contest. 5

[15] Danny Bankman. Stanford ee214b - advanced analog integrated circuits design, design contest 'most innovative design award'. 5