LaMAGIC: Language-Model-based Topology Generation for Analog Integrated Circuits

Chen-Chia Chang [1 2] **Yikang Shen** [3] **Shaoze Fan** [4] **Jing Li** [4] **Shun Zhang** [3] **Ningyuan Cao** [5]
Yiran Chen [2] **Xin Zhang** [1 3]

Abstract

In the realm of electronic and electrical engineering, automation of analog circuit is increasingly vital given the complexity and customized requirements of modern applications. However, existing methods only develop search-based algorithms that require many simulation iterations to design a custom circuit topology, which is usually a time-consuming process. To this end, we introduce LaMAGIC, a pioneering language model-based topology generation model that leverages supervised finetuning for automated analog circuit design. LaMAGIC can efficiently generate an optimized circuit design from the custom specification in a single pass. Our approach involves a meticulous development and analysis of various input and output formulations for circuit. These formulations can ensure canonical representations of circuits and align with the autoregressive nature of LMs to effectively addressing the challenges of representing analog circuits as graphs. The experimental results show that LaMAGIC achieves a success rate of up to 96% under a strict tolerance of 0.01. We also examine the scalability and adaptability of LaMAGIC, specifically testing its performance on more complex circuits. Our findings reveal the enhanced effectiveness of our adjacency matrix-based circuit formulation with floating-point input, suggesting its suitability for handling intricate circuit designs. This research not only demonstrates the potential of language models in graph generation, but also builds a foundational framework for future explorations in automated analog circuit design.

[1]IBM T. J. Watson Research Center [2]Duke University [3]MIT-IBM Watson AI Lab [4]New Jersey Institute of Technology [5]University of Notre Dame. Correspondence to: Chen-Chia Chang <chenchia.chang@duke.edu>, Xin Zhang <xzhang@us.ibm.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

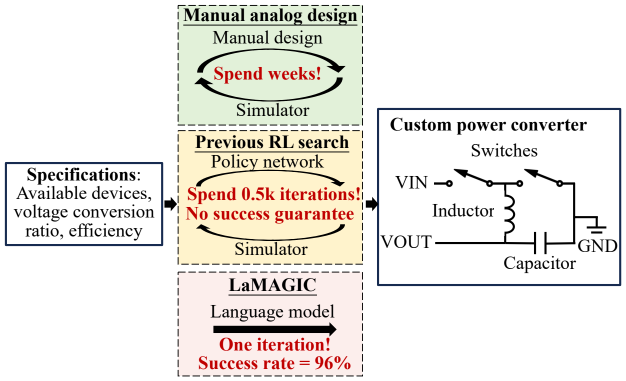


Figure 1. The scenario comparison between manual analog circuit design, previous RL search method (Fan et al., 2021), and our language model LaMAGIC.

1. Introduction

Analog circuit design, encompassing a broad range of applications, poses significant challenges due to its inherent complexity of the schematic and the precision of performance required in its execution. This complexity is particularly pronounced in the realm of power converters, which have become ubiquitous in an array of electronic and electrical devices. With the advent of diverse and customized electrical systems, such as electric vehicles, self-powered Internet of Things devices, and wearable or implantable biosensors, the demand for custom-designed power converters to meet specific supply power standards has surged. These converters, each with unique design specifications including voltage conversion ratio and power efficiency, exemplify the intricate and varied nature of analog circuit design. Traditional design methodologies, largely depending on pre-existing circuit topologies and extensive manual optimization, are increasingly inadequate in addressing the increasing needs of these applications. This reliance on conventional approaches not only prolongs the design process but also limits the potential for novel solutions in rapidly evolving domains. This gap highlights a critical need for an automated circuit design framework, capable of efficiently generating and optimizing high-quality power converter topologies based on specific design specifications.

While there have been notable efforts to address the challenges of automated analog circuit design, as highlighted in recent studies (Fan et al., 2021; Zhao & Zhang, 2022; Lu et al., 2023), these initiatives are yet to fully overcome the inherent complexities of the field. The approach (Fan et al., 2021) leverages a reinforcement learning (RL)-based tree sampling process for automating power converter design. However, this method exhibits limitations in terms of scalability and practical applicability, especially when generating circuits with varying performance specifications. It requires approximately 500 simulation queries for convergence each time a new circuit design is initiated, underscoring a significant challenge in efficiently adapting to different specifications. The other methods (Zhao & Zhang, 2022; Lu et al., 2023) also develop search-based algorithm to sample promising circuit, which requires lots of simulations when querying a new specification. This bottleneck highlights the necessity for a more direct and efficient generation method that can achieve desired performance criteria in a single iteration. Such a method can not only accelerate the automated design process but also enhance the practicality and applicability of automated analog circuit design, especially in rapidly evolving and diverse application areas.

The advent of large language models (LLMs) (Radford et al., 2019; Raffel et al., 2020; Chung et al., 2022; Nijkamp et al., 2023) has opened new frontiers in numerous fields, demonstrating remarkable capabilities in understanding and generating complex patterns and structures. This makes LLMs particularly promising for tackling the intricate challenges of automated analog circuit design. The core strength of LLMs lies in their ability to process and synthesize vast amounts of data, learning underlying patterns and relationships. In the context of analog circuit design, this translates to the potential for LLMs to understand and generate the nuanced and often non-linear relationships between different circuit components and their performance characteristics.

However, to fully harness this potential in the field of analog circuit design, a significant extension of LLM capabilities is required, specifically in the domain of graph generation. Analog circuits can be effectively represented as graphs, where components are nodes and connections are edges, encapsulating both the structural and functional aspects of the design. Extending LLMs to support graph generation would enable them to directly generate circuit topologies from specifications with high efficiency and accuracy.

This paper proposes LaMAGIC, an LM-based topology generation model for automated analog circuit design, especially for power converter applications. The application scenario is shown in Figure 1. To the best of our knowledge, we are the first to solve this problem through generative modeling and LM methodology. We develop a novel approach centered around the supervised finetuning (SFT) to customize the LM into the domain of intricate analog circuit design. The core of LaMAGIC is the meticulous examination of data formats and their influence on the model's capabilities. We propose and investigate various innovative input and output formulation for circuit generation to synchronize with the operational dynamics of LMs and effectively capture graph-based circuit representations using the autoregressive loss function of LMs. With the sequence-to-sequence modeling in LM, we can abstract the circuit generation into a process of sequential component or connection selection by utilizing the preceding subgraph information. In addition, we explore the effects of representing circuit specifications with different data types i.e., floating-point numbers versus characters. This aims to explore the effectiveness of feeding numerical input into LM to help the circuit learning.

Our contributions can be summerized as follows:

- We introduce LaMAGIC, a pioneering approach that adapts LMs to the domain of analog circuit design through SFT. This enables the efficient one-shot generation of custom circuit designs from user-defined specifications.

- We propose multiple novel circuit formulations designed to enhance circuit generation by (1) ensuring canonical representations, (2) enhancing compatibility with the autoregressive training methods and loss functions of the LM, and (3) utilizing float inputs to optimize LM's processing capabilities.

- Experimental results show that our model achieves superior 0.96 success rate under a strict tolerance of 0.01. We further conduct an extensive evaluation of the model's performance and its adaptability to more complex circuit designs. These evaluations provide critical insights into the effectiveness of different formulations, contributing significantly to future advancements in this field.

- By extending the functionalities of LMs to include graph generation, LaMAGIC marks a significant step forward in the generation of optimized circuit topologies directly from specifications. This expansion has the potential to inspire similar applications in other areas of graph generation within the LM domain.

These contributions collectively represent a significant advancement in the field of automated analog circuit design, particularly in improving the efficiency, accuracy, and applicability of LMs in generating custom and application-specific circuit designs.

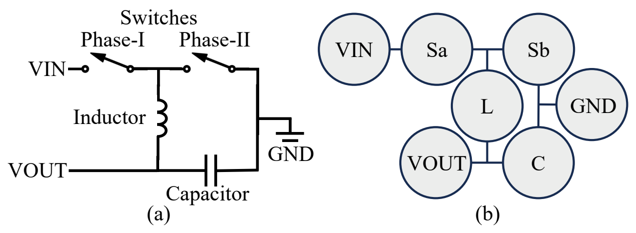


Figure 2. (a) An example power converter circuit and (b) its corresponding hypergraph representation.

2. Preliminaries

2.1. Analog Circuit Design

In the rapidly evolving domain of electrical and electronic engineering, automated analog circuit design, particularly for custom power converter applications, is very importance. This automation process aims to produce customized power converters without human interference, adhering to specific design specifications. Key among these specifications are the voltage conversion ratio and the power conversion efficiency. The voltage conversion ratio is defined as the ratio between input and output voltages, while the power conversion efficiency is the ratio of input power to output power. Another crucial aspect of power converter design is the duty cycle, which controls the duration of switches within the circuit, thereby influencing the output voltage and the overall performance of the circuit.

The circuit topology G is conceptualized as a hypergraph consisting of vertices V and hyperedges E. The vertices V include various analog devices and three external terminal ports. The device (or called *component* in later context), including capacitors C, inductors L, phase-I switches Sa, and phase-II switches Sb, is connected to other devices or ports via two outgoing edges. The terminal ports are the voltage input port VIN, the voltage output port $VOUT$, and the ground GND, each with a single outgoing edge. The hyperedges E symbolize the connections between these devices and ports. An example of the power converter and its hypergraph representation is shown in Figure 2.

2.2. Language Model

LMs (Radford et al., 2019; Raffel et al., 2020; Chung et al., 2022; Nijkamp et al., 2023), especially those using autoregressive training, are pivotal in natural language processing. Autoregressive LMs learn to predict the next token in a sequence based on previous tokens, utilizing an autoregressive loss function. This function calculates the loss as the negative sum of log probabilities for each predicted token, given the preceding ones. Formally, for a sequence $x_1, x_2, \ldots, x_n$, the loss ℓ is: $\ell = -\sum_{i=1}^{n} \log P(x_i|x_1, x_2, ..., x_{i-1})$. This method trains LMs to capture complex sequential patterns, essential for generating contextually and syntactically coherent sequences. In automated analog circuit design, this approach is particularly beneficial. During the training process, LM can learn how to base on the previous subgraph information to decide the next component and connection.

2.3. Problem Formulation

The objective of our model is to design circuit topologies and select appropriate duty cycles to achieve specified voltage conversion ratio and power conversion efficiency. Within our design framework, we consider a range of duty cycle options: $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. These options provide a framework for varying the ON times of switches to meet specific performance criteria. Based on these considerations, we define two distinct problem scenarios:

Problem 1 (Edge Generation). Given vertices V, a target voltage conversion ratio r, and an efficiency η, the task is to generate the connections E and determine the duty cycle $s \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ to construct a circuit that satisfies both r and η.

Problem 2 (Topology Generation). In the scenario where device requirements are not predefined, the challenge expands to generating both vertices V and their corresponding connections E, along with deciding the duty cycle s. The goal remains to construct a circuit meeting r and η.

3. Power Converter Dataset Construction

The foundation of an effective automated analog circuit design system lies in the diversity of the dataset upon which the model is trained. In our main experiment (Section 5.2), we construct a dataset by randomly sampling topologies of 3, 4, and 5-component circuits. This range was chosen to encapsulate the varying degrees of complexity typical in power converter circuits, thereby ensuring that our model will be learned to handle a variety of design scenarios. Additionally, we ensure that each topology is not isomorphic, which not only prevents redundancy in our dataset but also reinforces the diversity of circuit designs. For each random-sampled topology, we generate five circuits using different duty cycles $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ according to our design space. Then, we simulate each circuit with NGSPICE (Nenzi P, 2011) to compute the corresponding voltage conversion ratio and efficiency. Next, we pruned out the invalid topologies reported by the simulator. The final dataset comprises input features including the simulated voltage conversion ratio and efficiency, and the output consisting of the circuit topology and the duty cycle. For 3, 4, and 5-component circuits, we have 1k, 14k, and 117k different data points respectively, since the design space will exponentially grow up along with the component numbers. In total, we randomly split around 120k data points for training and 12k for evaluation.

In the subsequent experiment (Section 5.3), we extend our dataset to include 6-component circuits, aiming to assess the transferability of our model previously trained on 3, 4, 5-component circuits. This step is crucial in evaluating the model's ability to generalize its learning and adapt to more complex circuit designs. In total, we sample 76k different 6-component circuits and allocate 9k data points for evaluation.

4. Language Model-based Topology Generation

4.1. Naïve Methods for Circuit Generation

Using LLMs to revolutionize the labor-intense tasks is a promising trend given the generation ability and the natural language formats for input and output of LLMs. Prompt engineering (Liu et al., 2023) and SFT (Nijkamp et al., 2022) are the most famous approach to adapt LLMs to specialized domains. However, the unique challenges of analog circuit design, which demand specialized expertise and interaction with simulators, present limitations for fundamental LLMs like GPT-4 (Lee et al., 2023) when used prompt engineering. Our experiment with GPT-4, involving few-shot prompt engineering with 100 samples in the analog circuit design domain, revealed that the model failed to produce circuits meeting our specific performance requirements. This outcome illustrates the limitations of using general-purpose LLMs and prompt engineering for highly specialized tasks.

As a result, in this work, we aim to explore SFT to build an LM for analog circuit generation. A naïve method is to formulate the problem into instruction-based context inspired by the recent success of instruction-based finetuning methods (Chung et al., 2022; Taori et al., 2023), as shown in the first formulation in Figure 3 for edge generation.

This format offers an user-friendly approach, designing the input as a natural language instruction with detailed specifications. However, this seemingly straightforward formulation might present some challenges for LMs:

1. Non-unique representations: The order of hyperedges can be permuted while still representing the same circuit structure, leading to multiple potential outputs representing for a single input. This non-uniqueness can complicate the learning process by causing ambiguity in the loss function calculation, potentially misguiding the model training.

2. Structured connection representation: A more structured format might be necessary for effectively representing connections, aligning better with the sequential nature of LM training and loss functions, to facilitate learning of connections and component selections.

3. Precision of technical specifications: Presenting technical specifications such as voltage conversion ratio and efficiency in textual format might reduce precision, requiring additional effort from the LM to interpret these values correctly.

4.2. Our Novel Circuit Formulations

In addressing the potential shortcomings of the initial circuit format, our study introduces several alternative representations to more effectively convey complex circuit design specifications to an LM. In the meantime, we can explore the potential of LM for graph generation.

The first of these formulations, which we refer to as the Canonical Formulation, is illustrated in Figure 3. By sorting the edges in connections based on a predefined vertex order, it organizes the circuit information to ensure a canonical, unambiguous representation for each design. Furthermore, it simplifies the language used in the instructions, removing redundant or non-essential elements. Thus, this method addresses the non-unique representation challenge.

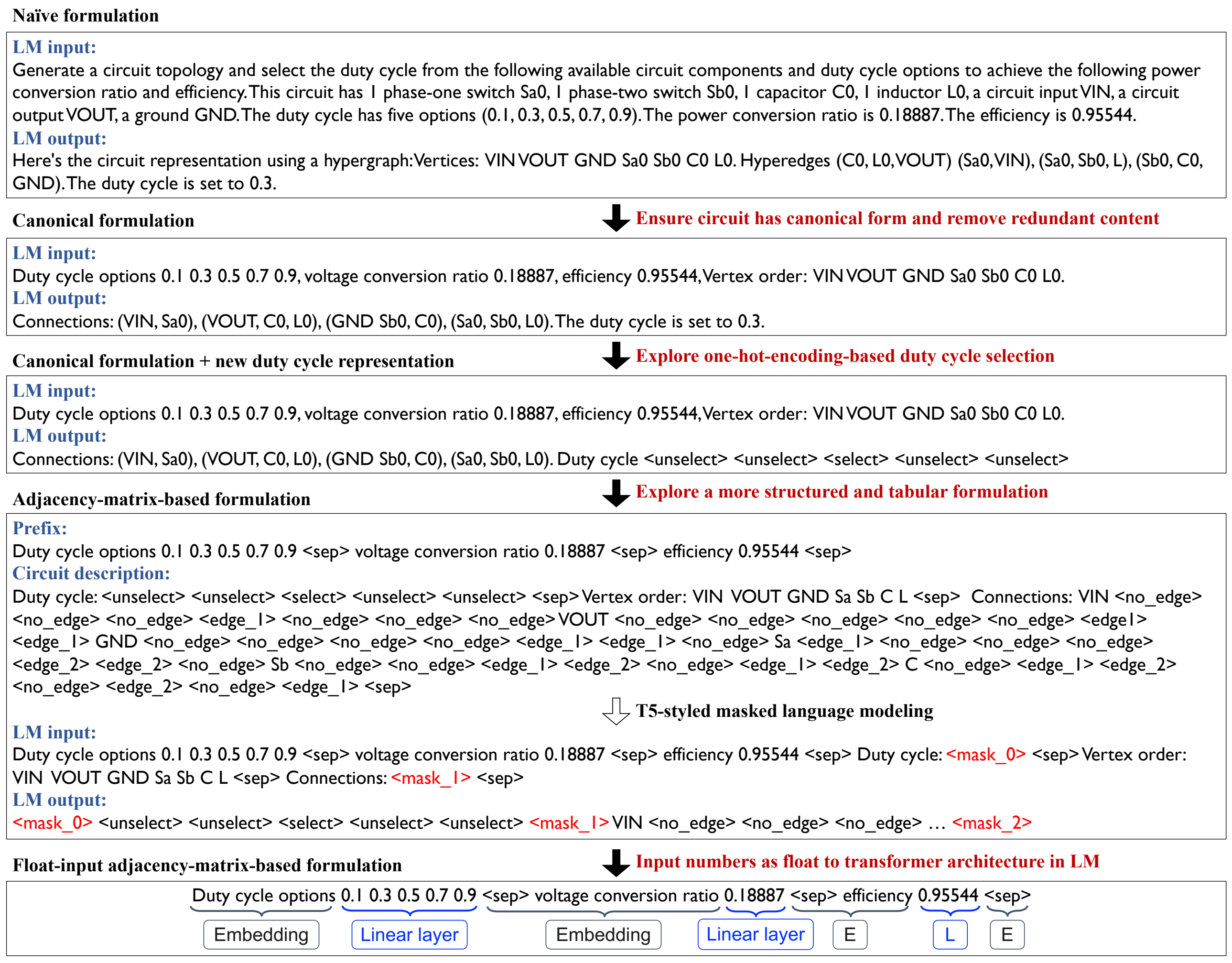
Building on the Canonical Form, we introduced a second formulation, named Canonical Formulation + New Duty Cycle Representation, as shown in Figure 3. This variant employs one-hot encoding for duty cycle selection, using 5 tokens of <select> or <unselect>. Only one token is marked as <select>, indicating the chosen duty cycle. This approach transforms a categorical choice into sequential selections.

Building upon our second formulation, we present a third formulation, the Adjacency-Matrix-based Formulation, which offers a structured and systematic approach to represent circuit topologies. This method, depicted in Figure 3, is designed to capture the graph structure effectively during training with an LM's loss function.

The formulation consists of two main parts: the prefix and the circuit description. The prefix includes duty cycle options, voltage conversion ratio, and efficiency. The circuit description entails the duty cycle selection, vertex order, and connections. Also, this formulation separates each requirement with a <sep> token to aid the model in differentiating between them.

For the circuit description, duty cycle selection is succinctly represented using one-hot encoding from the second formulation. Vertex order specifies the ports and devices in the circuit, while the connections are defined in an adjacency matrix format according to the vertex order. Distinct tokens <no_edge>, <edge_1>, <edge_2>, and <both_edges> represent the presence or absence of connections between vertices. Note that <edge_1> always precedes <edge_2> in each vertex's connection representation. Thus, this formulation ensures a canonical form for each circuit.

By translating the graph structure into a sequential format,



Naïve formulation

LM input:
Generate a circuit topology and select the duty cycle from the following available circuit components and duty cycle options to achieve the following power conversion ratio and efficiency. This circuit has 1 phase-one switch Sa0, 1 phase-two switch Sb0, 1 capacitor C0, 1 inductor L0, a circuit input VIN, a circuit output VOUT, a ground GND. The duty cycle has five options (0.1, 0.3, 0.5, 0.7, 0.9). The power conversion ratio is 0.18887. The efficiency is 0.95544.
LM output:
Here's the circuit representation using a hypergraph: Vertices: VIN VOUT GND Sa0 Sb0 C0 L0. Hyperedges (C0, L0, VOUT) (Sa0, VIN), (Sa0, Sb0, L), (Sb0, C0, GND). The duty cycle is set to 0.3.

Canonical formulation **Ensure circuit has canonical form and remove redundant content**

LM input:
Duty cycle options 0.1 0.3 0.5 0.7 0.9, voltage conversion ratio 0.18887, efficiency 0.95544, Vertex order: VIN VOUT GND Sa0 Sb0 C0 L0.
LM output:
Connections: (VIN, Sa0), (VOUT, C0, L0), (GND Sb0, C0), (Sa0, Sb0, L0). The duty cycle is set to 0.3.

Canonical formulation + new duty cycle representation **Explore one-hot-encoding-based duty cycle selection**

LM input:
Duty cycle options 0.1 0.3 0.5 0.7 0.9, voltage conversion ratio 0.18887, efficiency 0.95544, Vertex order: VIN VOUT GND Sa0 Sb0 C0 L0.
LM output:
Connections: (VIN, Sa0), (VOUT, C0, L0), (GND Sb0, C0), (Sa0, Sb0, L0). Duty cycle <unselect> <unselect> <select> <unselect> <unselect>

Adjacency-matrix-based formulation **Explore a more structured and tabular formulation**

Prefix:
Duty cycle options 0.1 0.3 0.5 0.7 0.9 <sep> voltage conversion ratio 0.18887 <sep> efficiency 0.95544 <sep>
Circuit description:
Duty cycle: <unselect> <unselect> <select> <unselect> <unselect> <sep> Vertex order: VIN VOUT GND Sa Sb C L <sep> Connections: VIN <no_edge> <no_edge> <no_edge> <edge_1> <no_edge> <no_edge> <no_edge> VOUT <no_edge> <no_edge> <no_edge> <no_edge> <no_edge> <edge1> <edge_1> GND <no_edge> <no_edge> <no_edge> <no_edge> <edge_1> <edge_1> <no_edge> Sa <edge_1> <no_edge> <no_edge> <no_edge> <edge_2> <edge_2> <no_edge> Sb <no_edge> <no_edge> <edge_1> <edge_2> <no_edge> <edge_1> <edge_2> C <no_edge> <edge_1> <edge_2> <no_edge> <edge_2> <no_edge> <edge_1> <sep>

T5-styled masked language modeling

LM input:
Duty cycle options 0.1 0.3 0.5 0.7 0.9 <sep> voltage conversion ratio 0.18887 <sep> efficiency 0.95544 <sep> Duty cycle: <mask_0> <sep> Vertex order: VIN VOUT GND Sa Sb C L <sep> Connections: <mask_1> <sep>
LM output:
<mask_0> <unselect> <unselect> <select> <unselect> <unselect> <mask_1> VIN <no_edge> <no_edge> <no_edge> … <mask_2>

Float-input adjacency-matrix-based formulation **Input numbers as float to transformer architecture in LM**

Duty cycle options 0.1 0.3 0.5 0.7 0.9 <sep> voltage conversion ratio 0.18887 <sep> efficiency 0.95544 <sep>

Figure 3. A circuit example from Figure 2 representing by the naïve formulation, our canonical formulation, canonical formulation + new duty cycle representation, adjacency-matrix based formulation, and float-input adjacency based matrix formulation for edge generation task. In the top three formulation, we simply place the vertex-related description from LM input to output to achieve topology generation task. In adjacency-matrix based formulation, we employ T5-styled masked language modeling (Raffel et al., 2020) to perform circuit generation. For topology generation, we further mask the vertex order in LM input and place it to output.

LMs can leverage their loss function to accurately predict the next item in the sequence, which in this case corresponds to the next connection or component in the circuit. The model can utilize the information from the preceding subgraph to learn the likelihood of connections, thereby understanding the graph structure more effectively.

Then, we introduce a fourth approach that innovatively feeds numerical data, including duty cycle options, voltage conversion ratio, and efficiency, as floats directly into the LM, which is called Float-input Adjacency-matrix-based Formulation. This needs a modification to the traditional word embedding mechanism of the LM. We incorporate a shared linear layer to encode these numbers, subsequently integrating them with other word embeddings to be the transformer's input.

To apply matrix formulation for edge and topology generation, we employ T5-styled masked language modeling (Raffel et al., 2020). Each consecutive masked token will be replaced by one masking token. Masking tokens indicate LM the positions where predictions are required. For edge generation, two masking tokens are used to conceal the duty cycle selection and the connections, requiring the model to predict these aspects of the circuit based on the provided context. In topology generation, the scope of masking extends further to include vertex information, prompting the model to predict the entire circuit structure. This method is pivotal in transitioning from edge to topology generation, allowing the model to incrementally build circuit understanding and refine its generative capabilities to mirror the iterative problem-solving nature of analog circuit.

For all formulations, we take the Flan-T5 (Chung et al., 2022) as the pretrained model to perform SFT on our dataset. This is an encoder-decoder transformer structure with 248M parameters, which can facilitate our masked language modeling and handle the conditional generation for all formulations. We initially train models on edge generation and then extend the training for topology generation to gradually advance the capabilities of LMs. Furthermore, we integrate domain-specific tokens, such as VIN, Sa, and <edge_1>, into the tokenizer. The expanded tokenizer ensures that these unique elements are recognized as single entities, improving the model ability to parse and generate the specialized content for circuit designs. To enhance the robustness and generalization ability of our LM, we incorporate data augmentation techniques. Recognizing that the transformer backbone of the LM is not inherently permutation equivariant for graph, we introduce random vertex order permutations. This step exposes the model to diverse circuit configurations, enhancing its ability to generalize and preventing overfitting.

5. Experimental Results

5.1. Experiment Setup

Baselines. Since all related works (Fan et al., 2021; Zhao & Zhang, 2022; Lu et al., 2023) focus on search-based algorithms, we are the first to construct a generative model to bridge specifications and circuits in a one-time generation approach. Given the novelty of this application and the absence of prior work in building circuit generation models, our study aims to establish a baseline in the domain of automated analog circuit design, specifically focusing on the effectiveness of different input formulations. In addition, we comprehensively evaluate the model with 13K different input requirements from our testing set, while the other RL work (Fan et al., 2021) only perform five different specifications to evaluate its search engine.

We perform SFT on one baseline and four variants of circuit formulations for edge generation and topology generation in Figure 3: (1) naïve formulation (NF), (2) our first formulation with canonical form (CF), (3) our second formulation with canonical form and one-hot-encoding-based duty cycle selection (CFDC), (4) our adjacency-matrix-based formulation with pure text input to LM (PM), and (5) our adjacency-matrix-based formulation with float input (FM). For NF, CF, CFDC, and PM, we set voltage conversion ratio and efficiency with a six-decimal precision in the input. All models in the experiment were trained under identical hyperparameters, ensuring consistency across all other training variables.

Experimental platform and hyperparameters. Our experiment runs on a machine with one NVIDIA V100 GPU.

Table 1. MSE of voltage conversion ratio and efficiency evaluated on models trained with different circuit formulations for edge and topology generation.

	Edge generation task		Topology generation task	
MSE	Voltage	Efficiency	Voltage	Efficiency
NF	0.054	0.015	0.031	0.006
CF	0.016	0.004	0.008	0.003
CFDC	0.007	0.003	0.005	0.003
PM	0.007	**0.002**	0.009	0.003
FM	**0.006**	0.013	**0.002**	**0.001**

The hyperparameters of the LM training are detailed as follows: We perform training for 120 epochs using AdamW optimizer with a learning rate of 3×10^{-4} with a cosine scheduler using 300 warmup steps, a batch size of 128, and a L2 regularization strength of 10^{-5}.

Detailed model architecture. We use the encoder-decoder transformer structure with Flan-T5-base pretrained weights. It has 12 transformer layers in both encoder and decoder. Each layer has a key and value projection with dimension 64, a feed-forward layer with dimension 2048, and 12 heads.

Evaluation metrics. Our primary metrics for evaluation are the success rate of the generated circuits within varied tolerances and the Mean Squared Error (MSE) between the input specifications and the simulated performance of the generated circuits.

The success rate, inspired by the code generation task (Nijkamp et al., 2023), is defined as the proportion of generated circuits whose simulated performance fell within a tolerance t of the target input specifications. For instance, with a tolerance of $t = 0.1$, a target input voltage and efficiency of 0.9 and 0.8, respectively, require the generated circuit's performance to be within the ranges of 0.9±0.1 for voltage and 0.8±0.1 for efficiency to be considered successful. In the experiment, we consider the success rate under 10 different t ranging from 0.01 to 0.1. MSE is computed as the average squared difference between the input specifications and the corresponding simulated performance metrics, providing a quantitative measure of the prediction accuracy.

5.2. Generation Results on 3, 4, 5-Component Circuit

The results of applying different circuit formulations for edge and topology generation are shown in Figure 4 and Table 1. The proposed formulations, including CF, CFDC, PM, and FM, demonstrate a clear advantage over the baseline NF in both edge and topology generation tasks. This indicates the effectiveness of our formulations in guiding the model towards successful circuit generation.

For the edge generation task, while FM offered detailed numerical representation, it showed lower success rates at

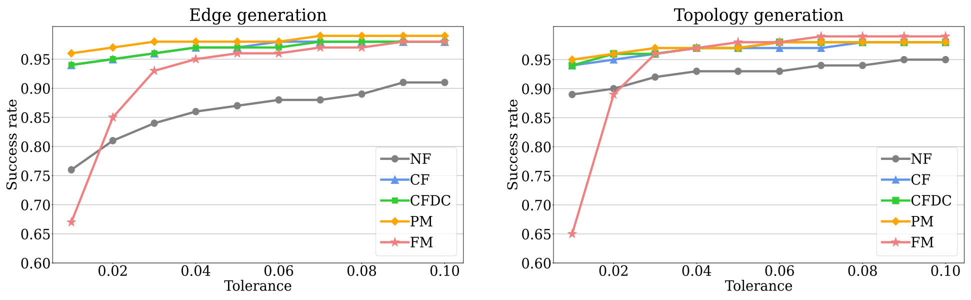


Figure 4. Success rates of models trained with different circuit formulations using 3, 4, 5-component circuits for edge generation and topology generation task. Circuit formulations include (1) the naïve formulation (NF), (2) the canonical form (CF), (3) the canonical form with one-hot-encoding-based duty cycle selection (CFDC), (4) the adjacency-matrix-based formulation with pure text input (PM), and (5) the adjacency-matrix-based formulation with float input (FM).

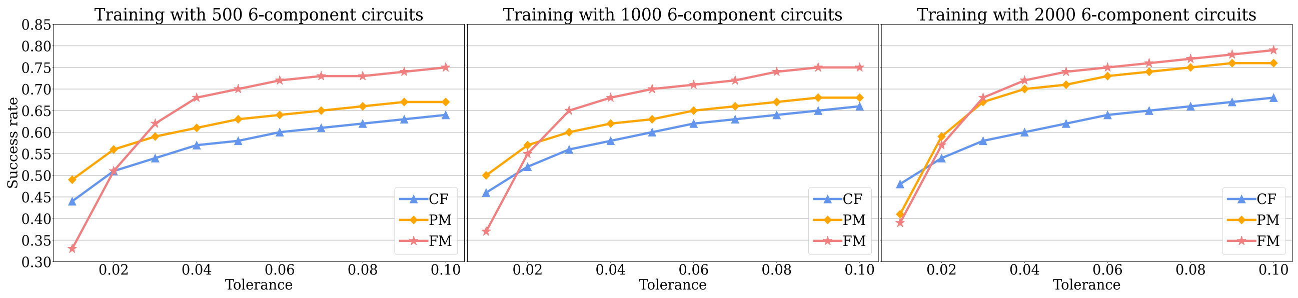


Figure 5. Success rates of models finetuned with different circuit formulations using 500, 1000, and 2000 6-component circuits.

smaller tolerances compared to PM, CF, and CFDC. This can be attributed to FM's precise numerical approach lacking the inherent flexibility of text-based inputs, making exact value predictions more challenging within tight tolerance ranges. In contrast, PM, CF, and CFDC provide textual context, aiding the model in more quickly and accurately interpreting simpler circuit connections. In the topology generation task, FM's detailed numerical input shows advantageous at broader tolerances with success rates of 0.99, aligning with the complexity required in these scenarios.

The discrepancy in FM's performance between edge and topology generation tasks suggests that numerical precision, while beneficial for complex tasks, may introduce complexities in tasks that require understanding simpler connections. The PM, CF, and CFDC, with their balance of structure and descriptive ease, lead to quicker and more accurate convergence in the edge generation task.

In addition, because encoder-only models theoretically offer the advantage of generating an entire graph in a single step, we experiment with using only the encoder component of T5, paired with PM, for circuit generation. Our result revealed that the encoder-only LM failed to converge and consistently produced invalid circuits, highlighting the importance of sequential reasoning and contextual understanding in autoregressive models for circuit design tasks.

Table 2. MSE of voltage conversion ratio and efficiency evaluated on models finetuned with different circuit formulations using 500, 1000, and 2000 6-component circuits.

	500		1000		2000	
MSE	Voltage	Efficency	Voltage	Efficency	Voltage	Efficency
CF	0.109	0.123	0.097	0.114	0.082	0.104
PM	0.092	0.087	0.096	0.089	0.051	**0.053**
FM	**0.050**	**0.068**	**0.048**	**0.049**	**0.038**	**0.053**

In summary, our analysis not only affirms the necessity of task-specific circuit formulations but also emphasizes the suitability of autoregressive LMs over encoder-only LMs for analog circuit design. These findings are valuable for guiding future research directions and model selection in automated circuit generation.

5.3. Transferability Evaluation on 6-Component Circuit

The scalability and adaptability of models in analog circuit design are crucial, particularly as the complexity of circuits increases. Our study initially focuses on models trained on circuits with 3, 4, and 5 components. We extend this to evaluate model performance on more complex 6-component circuits, providing essential insights into the circuit understanding capabilities of models in intricate scenarios.

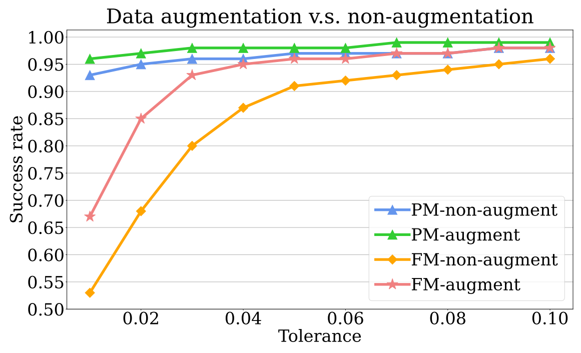


Figure 6. Success rates of models trained w/ and w/o vertex order permutation for edge generation.

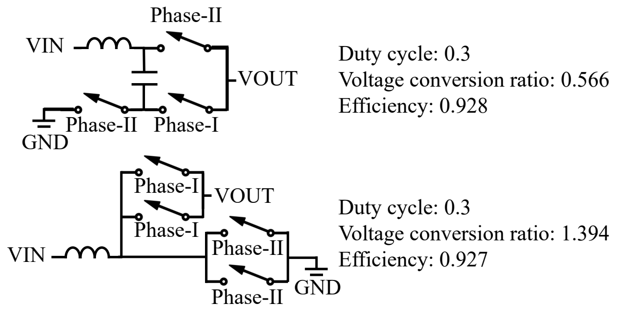


Figure 7. Two promising power converter topologies generated by our LM, demonstrating its ability to design functional circuit configurations that satisfy target design specifications.

Table 3. MSE of voltage conversion ratio and efficiency evaluated on models trained w/ and w/o vertex order permutation for edge generation.

	Edge generation task	
MSE	Voltage	Efficiency
PM-non-augment	0.010	0.005
PM-augment	**0.007**	**0.002**
FM-non-augment	0.040	**0.013**
FM-augment	**0.006**	**0.013**

Table 4. Training vs. testing loss evaluated on models trained with different formulation for edge generation.

	Edge generation task	
Loss	Training	Testing
CF	0.05	0.07
CFDC	0.05	0.07
FM	0.07	0.09
PM	0.03	0.04

We finetune LMs previously trained on 3, 4, 5-component circuits using three different formulations (CF, PM, and FM). The finetuning is conducted on datasets of 500, 1000, and 2000 samples of 6-component circuits by given the model 6 component requirement for the edge generation task. Each larger dataset incorporates all circuits from the previous sets, allowing us to observe the performance trends as training data increased. A testing set comprising 9k 6-component circuits is used for evaluation.

As demonstrated in Figure 5 and Table 2, FM particularly demonstrates superior adaptability and performance, especially at larger tolerances as the training data volume increased. PM also shows strong adaptability, with success rates higher than CF. These performance gain can be attributed to the use of numerical inputs and the matrix formulation, which provide effectiveness for understanding complex circuit configurations.

We additionally train the model solely on 6-component circuits without pretraining on simpler circuits, and the model is unable to produce valid circuits. This finding underscores the importance of pretraining foundational models in analog circuit design, which is critical for understanding essential knowledge of basic circuit and effectively handling more advanced designs.

6. Discussion

6.1. Training with Vertex Order Permutation

In this section, we evaluate the impact of data augmentation using random vertex order permutation in training that aims at enhancing the models' generation ability. We train models using FM and PM with and without data augmentation for edge generation task, as shown in Figure 6 and Table 3. The results indicate that models training with data augmentation outperform those training without data augmentation. These findings indicate the crucial role of data augmentation in increasing the diversity and complexity of training data. Additionally, this technique is particularly beneficial for FM, which initially struggle with lower success rates, to enhance the model to handle numerical inputs.

6.2. Exploration of Unseen Circuit Topology

In the field of automated analog circuit design, the ability to discover and validate previously unseen circuit topologies is important for newly developed tools. The majority of existing designs is created by human experts, and conventional automation tools typically lack the capability to generate unseen or novel designs. According to experienced analog designers, a power converter with (1) a voltage conversion ratio ranging from 0.4 to 0.6 or greater than 1.2 and (2) an efficiency larger than 0.9 requires the most efforts to design manually. Thus, we further validate that our LM can successfully generate high-performance power converter circuits within this target specification. Two promising circuits with 5 and 6 components are shown in Figure 7. This achievement highlights the potential of our LM to serve as a valuable tool for designers to explore unseen design space for analog circuits, potentially pushing the boundaries of human designs in this area.

6.3. Analysis of Potential Overfitting

This section analyzes the potential for overfitting, a common challenge when employing sequential model architectures in graph generation tasks. The training and testing loss for our four different circuit formulations, CF, CFDC, PM, and FM, are presented in Table 4.

We observe more severe overfitting in CF and CFDC than FM and PM. The robustness of FM and PM is further validated by their performance in transferability experiments detailed in Section 5.3, where these formulations achieve higher success rates in complex 6-component circuits. This evidence supports that FM and PM are more effective in avoiding overfitting and can be more suitable for scalable and generalizable circuit design tasks.

7. Conclusion

In this paper, we propose LaMAGIC, an LM-based topology generation model for analog circuit design that can directly generate an optimized circuit design given the custom specification in a single pass. Our approach focuses on SFT and demonstrates the effectiveness of LMs in generating complex circuit topologies and deciding circuit parameters. We systematically develop and analyze various input and output formulations to ensure canonical representations and the compatibility with the autoregressive nature of LMs, addressing the specific challenges of representing analog circuits as graphs. Experimental results show that our novel circuit formulations can clearly outperform a naive formulation and achieve a success rate of up to 0.96 under a tolerance of 0.01. In addition, we examine the scalability and adaptability of our LMs on more complex six-component circuits. The results show that our proposed adjacency-matrix-based circuit formulation with float input to LM can have better effectiveness on complex circuit understanding, which can help the future research to focus on such formulations when dealing with complicated circuits. In the future, we will extend the capabilities of LaMAGIC to a wider range of analog components. Through this article, we hope to open new research pathways on automated analog circuit design or any other fields of automated design that could benefit from the potential of LMs in graph generation.

Acknowledgements

This work is partially supported by SRC 3104.001 and NSF 2106828. Special thanks to Wan-Hsuan Lin for contributing valuable ideas to this project.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Fan, S., Cao, N., Zhang, S., Li, J., Guo, X., and Zhang, X. From specification to topology: Automatic power converter design via reinforcement learning. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9. IEEE, 2021.

Lee, P., Bubeck, S., and Petro, J. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239, 2023.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Lu, J., Lei, L., Huang, J., Yang, F., Shang, L., and Zeng, X. Automatic op-amp generation from specification to layout. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

Nenzi P, V. H. Ngspice users manual version 23., 2011. URL `https://pkgs.fedoraproject.org/repo/extras/ngspice/ngspice23-manual.pdf/eb0d68eb463a41a0571757a00a5b9f9d/ngspice23-manual.pdf`.

Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S., and Xiong, C. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

Nijkamp, E., Hayashi, H., Xiong, C., Savarese, S., and Zhou, Y. Codegen2: Lessons for training llms on programming and natural languages. *arXiv preprint arXiv:2305.02309*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Zhao, Z. and Zhang, L. Analog integrated circuit topology synthesis with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(12):5138–5151, 2022.