EVA: An Efficient and Versatile Generative Engine for Targeted Discovery of Novel Analog Circuits

Jian Gao[1], Weimin Fu[2], Xiaolong Guo[2], Weidong Cao[3], Xuan Zhang[1]
[1]Northeastern University; [2]Kansas State University; [3]The George Washington University

***Abstract*—Analog circuit design has traditionally depended on manual expertise, slowing the discovery of novel topologies essential for advanced technologies like AI, 5G/6G, and quantum computing. While AI-driven methods have accelerated hardware design workflows, most of them focus on topology synthesis, often reusing known structures to achieve specific goals. The challenge of discovering entirely new, high-performance topologies remains largely underexplored due to its abstract nature. In this work, we introduce EVA, an efficient and versatile generative engine for discovering novel analog circuit topologies. EVA employs a bottom-up generation framework, using a decoder-only transformer to sequentially predict device pin connections and create diverse circuits from scratch. Pretraining on unlabeled circuit topologies builds foundational knowledge about circuit connectivity, achieving baseline discovery efficiency by generating valid circuits and reducing performance-labeled samples needed in fine-tuning. For targeted discovery of high-performance designs, EVA leverages two fine-tuning strategies—proximal policy optimization (PPO) and direct preference optimization (DPO)—to further enhance discovery efficiency for relevant, high-performing topologies. Experimental results across various circuit types highlight EVA's strengths in validity, novelty, versatility, and both training sample and discovery efficiency.**

I. Introduction

Analog circuit topologies' design has historically relied heavily on manual effort and domain expertise. This approach not only results in a lengthy design cycle but also restricts the discovery of novel analog circuit topologies. Emerging technologies, such as AI, 5G/6G, and quantum computing, impose stringent performance demands, creating a need for innovative analog circuit topologies beyond the capabilities of well-established designs. To support the advancement of these technologies, a critical question arises: how can we efficiently explore and invent circuit topologies that might achieve unprecedented performance, surpassing what human expertise alone has accomplished? Recently, AI-driven methodologies for analog hardware design have shown promise in accelerating the design cycle across various stages of the design flow [1]–[10]. However, existing research [11]–[13] mainly addresses the analog circuit topology synthesis problem, focusing on meeting specific design objectives by selecting or reusing known topologies or sub-blocks from existing designs. In contrast, the discovery problem of novel analog circuit topologies remains underexplored due to its abstract and complex nature.

In this work, we tackle the novel circuit topology discovery problem due to its scientific significance and potential for breakthrough innovations. Existing approaches have achieved great progress in topology discovery for certain types of analog circuits. CktGNN [1], a pioneering effort, employs a variational autoencoder (VAE) with a two-level graph neural network (GNN) to learn and generate diverse operational amplifier (Op-Amp) topologies. Similarly, Karahan et al. [14] propose an inverse design approach for the rapid synthesis of complex RF-to-terahertz (THz) matching networks by generating various electromagnetic (EM) structures. However, these approaches are limited in versatility and discovery efficiency without the ability to discover diverse types of analog circuit topologies while targeting certain performance preferences.

To address these challenges, we introduce EVA, an efficient and versatile generative engine for discovering novel analog circuit topologies. To achieve great versatility, EVA proposed a bottom-up generation framework that uses a decoder-only transformer to generate diverse analog designs from scratch, starting from a VSS pin and sequentially predicting the next device pin connection. Instead of training with performance-labeled topology from scratch, EVA is first pre-trained on a diverse set of unlabeled circuit topologies to learn the circuit connectivity. In this way, the model itself has already achieved a basic level of discovery efficiency by generating valid topology and setting a great foundation for finetuning. For targeted discovering specific circuit types with high performance within limited attempts, EVA proposed two fine-tuning strategies based on proximal policy optimization (PPO) [15] and direct preference optimization (DPO) [16] with a limited number of labeled topology thanks to the foundation laid by the pretrained model. In this way, EVA is able to achieve great training sample efficiency and discovery efficiency without wasting evaluation time on irrelevant or low-performing circuits. Our key contributions are as follows:

- EVA is a **versatile generative engine** that can discover diverse, novel analog circuit topologies by sequentially predicting device pin connections, starting from VSS.
- EVA pretrains a decoder-only transformer on an unlabeled dataset of circuit topologies, enabling efficient generation of valid circuits and significantly **improving training sample efficiency** by reducing the reliance on performance-labeled samples during fine-tuning.
- EVA **enhances discovery efficiency** by fine-tuning with PPO and DPO on a small set of performance-labeled topologies, focusing on high-performance circuits and minimizing irrelevant evaluations.

This work is partially supported by NSF #2416375 and NSF #2526432.

- Experimental results across various analog circuit types demonstrate EVA's advantages in topology validity, novelty, versatility, sample efficiency, and discovery efficiency.

II. Related Work

A. ML methods for analog circuit topology synthesis

Analog circuit topology synthesis aims to develop circuits that meet design goals using established topologies or subblock topologies. Existing ML methods primarily address the issue of data scarcity when building models capable of automating analog circuit design. AnalogCoder [11] highlights that while large language models (LLMs) have been extensively trained on Python code, they lack exposure to the SPICE netlist, the industry-standard format for describing analog circuits. To bridge this gap, they employ domain-specific prompt engineering, enabling LLMs to generate PySpice code that can be seamlessly converted into SPICE netlists without additional training—thus maximizing training sample efficiency. Similarly, Artisan [12] translates SPICE netlists into natural language, allowing the development of domain-specific foundation models for analog circuit design, even with limited data. On the other hand, LaMAGIC [13], a fine-tuned masked language model (MLM), has been proposed to generate power converter circuit topologies by predicting the connections between predefined nodes. However, most current approaches to analog circuit topology design rely heavily on existing circuit topologies or subblock structures, which constrains their ability to discover novel topologies.

B. ML methods for novel analog circuit topology discovery

Novel analog circuit topology discovery aims to identify novel, non-existent circuit topologies that could potentially expand our understanding of analog circuits. Existing ML approaches have primarily focused on discovering designs at two distinct levels, each restricted to a specific type of analog circuit. CktGNN [1] addresses the schematic level by generating a variety of Op-Amp topologies. This method adopts a top-down graph generation approach but limits the design scope to sub-blocks without incorporating device-level details. While this constraint simplifies the generation process and ensures valid topologies (e.g., avoiding floating nodes), it also restricts the diversity of circuits that can be explored. Karahan et al. [14], on the other hand, focus on the physical level, employing an inverse-design approach using binary matrices to generate unintuitive EM structures for RF-to-THz matching networks. However, applying this method to schematic-level design is challenging, as the complexity of analog circuit topologies—due to the wide variety of device types and numbers—cannot be adequately captured by a binary matrix representation. Beyond versatility limitations, these methods are also constrained by inefficient discovery processes. Both approaches lack target discovery capabilities, relying heavily on trial and error, which leads to substantial evaluation overhead from real-world simulators due to frequent low-performance circuit assessments. To mitigate this, Karahan et al. proposed training a forward model to predict performance for specific EM structures, a promising direction but one that demands over 600k performance-labeled EM structures, making it extremely sample inefficient.

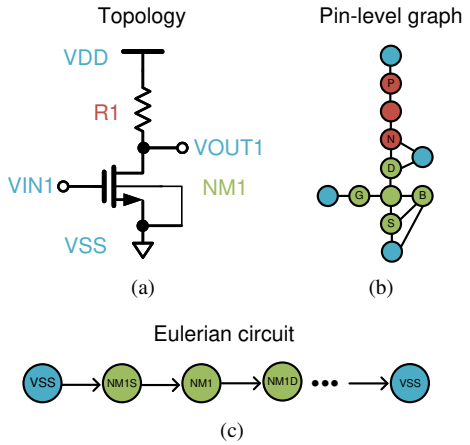

Fig. 1: An example of sequential graph presentation EVA used to represent the circuit topology [17]. (a) Original topology. (b) Pin-level graph representation. (c) Eulerian circuit.

III. EVA Framework

EVA in Figure 2 is an efficient and versatile generative engine designed to discover novel analog circuit topologies. It uses a device pin-level graph representation [17], sequentialized as an Eulerian circuit, to capture all possible connections in a compact format while retaining critical structural information. EVA leverages a decoder-only transformer architecture, pre-trained on diverse analog circuit topologies, to generate circuits from scratch by sequentially predicting device pin connections. Through fine-tuning strategies such as PPO and DPO, EVA efficiently generates high-performance circuit topologies with minimal labeled data, targeting specific circuit types and desired performance.

A. Expressive and efficient sequential graph representation

Existing approaches to schematic-level analog circuit topology discovery generally formulate the problem as a graph generation task, where each circuit topology is represented as a device-level graph with nodes corresponding to individual devices (e.g., NMOS transistors) [1]. EVA, in contrast, adopts a device pin-level graph representation as shown in Figure 1 where each node corresponds to an individual device pin (e.g., NM1_G, NM1_D, NM1_S, and NM1_B) [17]. Since analog circuit topologies are typically sparse—most devices connect only to their immediate neighbors—traditional adjacency matrices are inefficient, as they waste space representing non-existent edges [1], [13]. To address this, EVA sequentializes the graph as an Eulerian circuit [17], a path that traverses each directed edge exactly once before returning to the starting node (e.g., VSS). Unlike prior approaches that limit graph structures to directed acyclic graphs (DAGs) [1]—which cannot universally represent all analog topologies—the Eulerian circuit

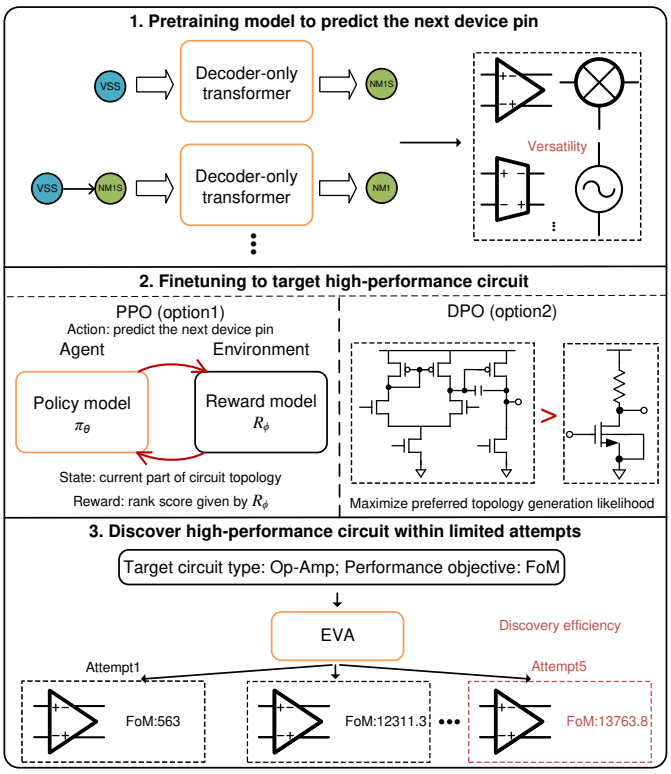

Fig. 2: Overview of EVA framework.

approach is more versatile, capable of representing any analog circuit that can be modeled as a finite connected undirected graph.

B. Pretraining model to generate diverse circuits

EVA employs a customized domain-specific tokenizer to encode and decode sequences for pretraining a decoder-only transformer to discover a wide range of analog circuit topologies. Each token in EVA's tokenizer corresponds either to a device pin (e.g., NM1_G, NM1_D, NM1_S, NM1_B) or to a circuit-level pin for the overall analog topology (e.g., VIN1, VOUT1, VDD, VSS). To ensure the model can generalize across circuit topologies with varying numbers and types of devices, EVA uses a data-driven approach to scan the entire dataset and determine device limits. A special token, "Truncate," is introduced to handle padding when sequences of varying lengths are processed.

With this domain-specific tokenizer, EVA pre-trains a decoder-only transformer to predict the next device pin within a circuit topology. Unlike typical language model pretraining, which crops sequences randomly from text, EVA ensures that each sequence corresponds to a single complete circuit topology. Given an corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$ representing one circuit topology, EVA maximizes the standard language modeling objective [18] to train the model.

$$\mathcal{L}_{\text{pretrained}}(\mathcal{U}) = \sum_i \log P\left(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta\right) \quad (1)$$

TABLE I: Rank score definitions for PPO finetuning

Reward	Definition
1.0	High performance relevant valid circuit
0.5	Low performance relevant valid circuit
-0.5	Irrelevant valid circuit
-1.0	Invalid circuit

Here, k represents the size of the context window, and the conditional probability P is modeled by a neural network with parameters Θ. During the generation phase, the model is initially provided with just one context token, "VSS," which serves as the starting node for all Eulerian circuits. From this single token, the model generates the rest of the sequence, completing it to represent an entire circuit topology.

C. Finetuning model to target high performance circuit

The pre-trained model in EVA has developed foundational knowledge of circuit connections across various circuit types, enabling it to achieve baseline discovery efficiency by generating valid circuits. To further enhance its ability to efficiently generate high-performance circuits within limited attempts, EVA fine-tunes this model to align with human preferences for both type and performance. Building on the foundation provided by pretraining, EVA can leverage a small set of performance-labeled topologies to guide generation towards high-performance designs. EVA introduces two fine-tuning strategies that balance training stability and overfitting.

1) Proximal policy optimization: EVA adopts PPO [15] as one of its reinforcement learning with human feedback (RLHF) strategies. PPO uses limited labeled samples to train a reward model as a labeler, enabling the pre-trained model to generate new samples, which are then labeled by the reward model for fine-tuning based on human preferences. Specifically, EVA utilizes a Plackett-Luce ranking model [19] for sequence reward, ranking multiple classes as shown in Table I. The reward model combines a rule-based checker with a multiclass classifier: it first checks if a generated circuit is valid (i.e., simulatable with default sizing) and then classifies circuit type and performance. To train this multiclass classifier, a limited set of circuit topologies is initially labeled for type and performance. For instance, if EVA aims to discover high-performance Op-Amps, only Op-Amps are relevant, while other types are irrelevant. The figure of merit (FoM) determines whether relevant circuits are high-performance, with Otsu's method [20] setting the FoM threshold. Once labeled, EVA adds a three-output linear layer on top of the final transformer layer, outputting a reward value after softmax.

With the trained reward model, PPO performs optimization as shown in Algorithm 1. The major components of PPO are defined as follows:

Agent: The agent is the policy model π_θ, which is the pre-trained model fine-tuned to align with human preferences. EVA adds a value head to it, which consists of a simple linear layer that takes hidden states as input and outputs a single scalar per token. The value function provides an estimate of the expected cumulative reward from any given state.

Algorithm 1 PPO RLHF Finetuning

Require: Policy model π_θ, Reference model $\pi_{\theta_{ref}}$, Reward model R_ϕ, Epochs N_{epochs}, PPO epochs N_{ppo}, Batch size D, Minibatch size B, Value loss coefficient vc

for epoch $= 1$ to N_{epochs} **do**
  Let π_θ generate a batch of sequences with size D and corresponding values and store them as x, y, V
  Compute PPO rewards $\mathbf{r}$ from Eq. (2)
  for ppo_epoch $= 1$ to N_{ppo} **do**
    Randomly sample minibatch $(x_b, y_b, V_b, \mathbf{r}_b)$
    Repredict $V_{b_{new}}$ for minibatch using π_θ
    Calculate $\mathcal{L}_{\text{policy}}$, $\mathcal{L}_{\text{value}}$ from Eq. (3) and Eq. (4)
    Calculate $\mathcal{L}_{\text{PPO}} = -\mathcal{L}_{\text{policy}} + vc \cdot \mathcal{L}_{\text{value}}$
    Backpropagate and update π_θ
  end for
end for

Environment: The environment is the reward model R_ϕ. This environment assigns rewards based on whether generated sequences meet criteria defined in Table I.

Action: The action is discrete by selecting a token y_t (i.e., device-pin) from a pre-defined tokenizer's look-up table based on the current context x_t (i.e., generated sequence).

State: The state is the current context x_t.

Reward: Starting from the initial state x_0 (i.e., "VSS"), the RL agent takes a discrete action by predicting the next token y_t given the current state x_t in each time step t, and the environment returns a reward r_t at that step. EVA defines the PPO reward function as follows:

$$r_t = R_\phi(x, y) - \beta \log \left(\frac{\pi_\theta(y_t \mid x_t)}{\pi_{\theta_{ref}}(y_t \mid x_t)} \right) \tag{2}$$

Here, $R_\phi(x, y)$, is the reward model's score that measures how well the entire sequence aligns with preference defined in Table I. The second term is a penalty that discourages π_θ from deviating too much from $\pi_{\theta_{ref}}$, with β controlling the strength of this penalty. The reference model $\pi_{\theta_{ref}}$ is a fixed version of the pre-trained language model that the policy model was initially based on.

In PPO, the entire sequence or circuit topology acts as a trajectory, with the objective of optimizing the policy model π_θ to generate sequences that maximize the expected accumulated reward via the advantage A and the policy loss $\mathcal{L}_{\text{policy}}$. Concurrently, PPO refines value prediction $V_{b_{\text{new}}}$ using the value loss $\mathcal{L}_{\text{value}}$. As shown in Algorithm 1, each epoch begins with PPO generating rollouts (sequences) using the policy model, and the reward model R_ϕ assigns rewards $\mathbf{r}$, where each r corresponds to a rollout. PPO then performs minibatch optimization. The policy loss $\mathcal{L}_{\text{policy}}$ is designed to maximize the expected advantage of the actions taken by the agent while preventing large, destabilizing updates to the policy.

$$\begin{aligned} \mathcal{L}_{\text{policy}} &= \mathbb{E}_t \left[\min \left(L \cdot A_t, \text{clip}(L, 1-\epsilon, 1+\epsilon) \cdot A_t \right) \right] \\ \text{where} \quad L &= \frac{\pi_\theta \left(y_{b_t} \mid x_{b_t} \right)}{\pi_{\theta_{\text{old}}} \left(y_{b_t} \mid x_{b_t} \right)} \end{aligned} \tag{3}$$

Here, advantages $A_t = \delta_t + \gamma\lambda \cdot A_{t+1}$ where $\delta_t = r_{b_t} + \gamma V_b \left(x_{b_{t+1}} \right) - V_b \left(x_{b_t} \right)$ measure how much better (or worse) an action at time t is compared to the expected outcome calculated from value function $V_b(x_{b_{t+1}})$. L represents the probability ratio between the current and previous policies during minibatch optimization. With the clip objective, the policy loss in PPO strikes a balance between exploiting the current advantage—encouraging the model to increase the probability of actions that yield higher expected rewards—and maintaining stable, incremental updates to the policy. In PPO, the value loss $\mathcal{L}_{\text{value}}$ is calculated to enhance the accuracy of the model's value function.

$$\mathcal{L}_{\text{value}} = \frac{1}{2} \cdot \left(V_{b_{new}} \left(x_{b_t} \right) - G_t \right)^2 \tag{4}$$

The value loss $\mathcal{L}_{\text{value}}$ is the squared difference between the newly predicted value $V_{b_{\text{new}}} \left(x_{b_t} \right)$ and the target return G_t. The $G_t = A_t + V_b \left(x_{b_t} \right)$ represents the total expected reward for the trajectory from t onward, including the estimated value of future states. By incorporating both immediate and future rewards, G_t is a robust target for updating the value function.

2) Directed preference optimization: PPO is effective in preventing overfitting by continually generating new training data [21]. However, its complexity and stability issues arise from its sensitivity to hyperparameters and dependence on a reward model. To address these challenges in a simpler and more stable way, EVA introduces a finetuning strategy based on DPO [16]. Unlike PPO, which requires a separate reward model to label newly generated data, DPO finetunes the pre-trained model using a static dataset with expert-labeled data. This approach enhances stability, as the training labels are accurate and unaffected by the reward model's classification reliability. Moreover, DPO's objective function is simplified compared to PPO, with only a single hyperparameter, β, which controls the extent of π_θ deviation from the reference model.

$$\begin{aligned} \mathcal{L}_{\text{DPO}} \left(\pi_\theta; \pi_{\text{ref}} \right) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \\ \left[\log \sigma \left(\beta \log \frac{\pi_\theta \left(y_w \mid x \right)}{\pi_{\text{ref}} \left(y_w \mid x \right)} - \beta \log \frac{\pi_\theta \left(y_l \mid x \right)}{\pi_{\text{ref}} \left(y_l \mid x \right)} \right) \right] \end{aligned} \tag{5}$$

DPO utilizes the Bradley-Terry model [22] as its preference framework, focusing on pairwise comparisons (win or lose) rather than the multi-rank model employed in PPO. Specifically, optimizing $\mathcal{L}_{\text{DPO}}$ guides the pre-trained model to identify analog circuit topologies that maximize the expression $\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$, thus encouraging it to generate "winning" topologies while discouraging "losing" ones. Given the complexity of human preferences for analog circuit topologies, EVA still relies on human experts to label circuit topologies according to a multi-rank model, as defined in Table I. For any four data points where each belongs to a unique class, EVA transforms these into six unique win-lose pairs for DPO training (e.g., High performance relevant valid circuit $>$ Low performance relevant valid circuit, High performance relevant valid circuit $>$ Irrelevant valid circuit, etc.).

IV. Experiments

A. Experiments Setup

Datasets: The EVA circuit dataset comprises 3470 unique and real-world topologies across 11 circuit types: Op-Amps, LDOs, Bandgap references, Comparators, PLLs, LNAs, PAs, Mixers, VCOs, Power converters, and Switched Capacitor Samplers from public resources [23]–[28]. Each circuit's performance was assessed through circuit simulation, and a corresponding label was assigned based on its performance metrics and label definition defined in Table I.

EVA training setup: EVA splits the topology dataset into training and validation sets with a 9:1 ratio, ensuring the validation topologies remain unseen during training. EVA uses a depth-first search (DFS) to identify the Eulerian circuit for each topology. To prevent overfitting, it permutes DFS traversal order, creating multiple unique Eulerian circuits per topology, expanding the original 3470 topologies to 234393 sequences. EVA's generation model is a decoder-only transformer with 6 layers, 6 attention heads, and 11.825 million parameters, with a vocabulary size of 1029 and maximum sequence length of 1024. The reward model architecture is detailed in Section III-C1.

Baseline: To compare with EVA, we select AnalogCoder [11], Artisan [12], and LaMAGIC [13] as the representative ML baselines for analog circuit topology synthesis and select CktGNN [1] as the representative ML baselines for novel analog circuit topology discovery at the schematic level. The differences between these methods and EVA are discussed in Section II. We follow original work to produce their results.

Evaluation tasks and metrics: We evaluate generative quality across the following metrics: (1) Validity: An unsized circuit is valid if it can be simulated in SPICE without errors (e.g., floating or shorting nodes). Each method generates 1000 topologies, and we report the percentages that are valid. (2) Novelty: Each method generates 1000 topologies, and we measure the percentage of them that are different from the topologies in the dataset. To quantify the topology difference, We converted them into graphs and computed the maximum mean discrepancy (MMD) [29] between these and real-world graphs converted from our circuit dataset. (3) Versatility: The number of distinct analog circuit types generated indicates versatility. (4) Training sample efficiency: Efficiency is evaluated by comparing the number of labeled topologies (i.e., with known performance) required for training. (5) Discovery efficiency: Given the time cost of performance evaluation, each method only generates 10 topologies, and we report the maximum FoM (e.g., gain, bandwidth, power for Op-Amps) after sizing with a genetic algorithm and SPICE evaluation.

B. Necessity of pretraining and finetuning

We first perform an ablation study to evaluate the necessity of both pretraining and fine-tuning. As shown in Figure 3, PPO results reveal that the pretrain-only model establishes a solid foundation in generating irrelevant valid circuits (score = -0.5) and low-performance relevant valid circuits (score = 0.5). However, it lacks the ability to specifically target high-performance relevant valid circuits (score = 1), necessitating further fine-tuning. Similar observations are evident in DPO results, where the pretrain-only model shows no strong preference for winning topologies, as indicated by 0 reward accuracy. This suggests the model predominantly generates losing topologies during generation, i.e., $\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} < \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$. In contrast, the finetune-only model lacks the foundational understanding of circuit connectivity, making it challenging to effectively train on a limited number of labeled samples to generate valid topologies. For PPO, EVA's vast design space (1029^{1024}), dominated by invalid topologies, leads to an extremely sparse reward signal (i.e., most rewards are suboptimal). As a result, guiding the finetune-only model to discover high-performance relevant valid circuits from scratch becomes infeasible using a simple reward model. On the other hand, DPO results indicate that even an untrained model can achieve reward accuracy exceeding 50%, highlighting a limitation in this metric. Specifically, the metric only measures the preference for generating winning topologies, $\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} > \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$, without assessing the actual generation quality of winning topologies $\pi_\theta(y_w \mid x)$ or losing topologies $\pi_\theta(y_l \mid x)$. This limitation is further validated in Table II, where EVA (DPO only) fails to generate any valid topologies. In conclusion, EVA can effectively target high-performance relevant valid circuits only through a combination of pretraining and finetuning.

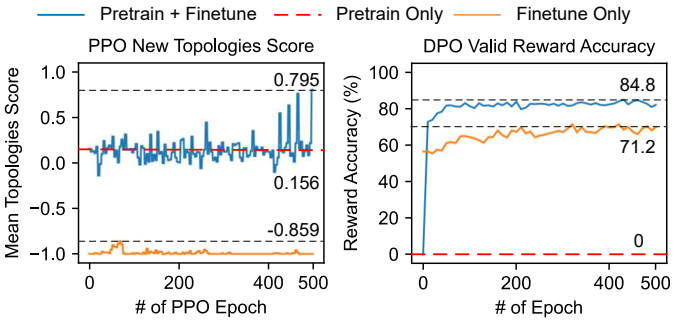

Fig. 3: PPO score and DPO validation reward accuracy comparison between Pretrain + Finetune, Pretrain only, and Finetune only while targeting Op-Amp design as an example.

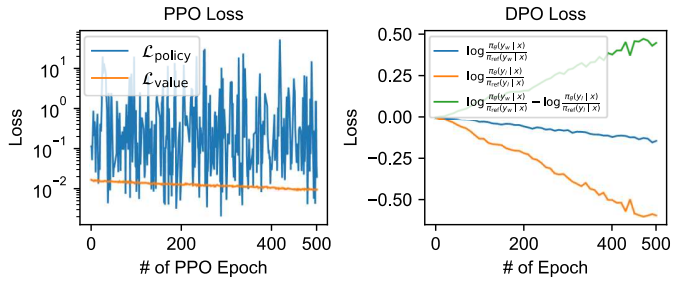

Fig. 4: EVA's PPO loss and DPO loss after pretraining while targeting Op-Amp design as an example.

TABLE II: Performance comparison between EVA and existing analog circuit topology generation work.

Evaluation metric	Validity (%) ↑	Novelty		Versatility ↑	# of labeled topology ↓		FoM@10 [12], [13] ↑	
		Diff circuit (%) ↑	MMD ↓		Op-Amp	Power converter	Op-Amp	Power converter
AnalogCoder [11]	66.1	0	0	7	11	N/A	232.1	N/A
Artisan [12]	82	0	0	1	14000	N/A	12769.5	N/A
CktGNN [1]	68.5	93	0.313	1	10000	N/A	311.3	N/A
LaMAGIC [13]	75	3	0.001	1	N/A	132000	N/A	2.2
EVA (Pretrain)	**84**	**99**	**0.0518**	**11**	**0**	**0**	**274.1**	**2.5**
EVA (PPO only)	**0**	**N/A**	**N/A**	**11[a]**	**850**	**362**	**N/A**	**N/A**
EVA (DPO only)	**0**	**N/A**	**N/A**	**11[a]**	**850**	**362**	**N/A**	**N/A**
EVA (Pretrain+PPO)	**94**	**99**	**0.0509**	**11[a]**	**850**	**362**	**13647.2**	**3.3**
EVA (Pretrain+DPO)	**83**	**99**	**0.0491**	**11[a]**	**850**	**362**	**13763.8**	**3.4**

[a] Targeting another circuit type requires redo fine-tuning, which takes less than an hour on an A100 GPU.

C. Comparison between PPO and DPO for EVA finetuning

We compare PPO and DPO for EVA fine-tuning in terms of training stability and generation quality. PPO, an online RL method, dynamically generates new topologies and samples rewards from the environment, but its training instability stems from the stochastic reward signal r_{b_t}, leading to noisy advantage estimates A_t and oscillations in $\mathcal{L}_{\text{policy}}$. In contrast, DPO, an offline RL method, optimizes the relative log-likelihood between winning and losing topologies using static preference data, avoiding noisy rewards. For generation, PPO demonstrates superior validity by optimizing for both preference and reward-based quality through continuous exploration of new topologies. This enables it to overcome degeneration issues and produce valid, diverse outputs that outperform the pretrained model. DPO, however, shifts the model's preference to high-performance topologies in the dataset without learning from new data, leading to degeneration in validity. This degeneration is characterized by a reduction in both winning and losing topology generation likelihoods in Figure 4, with the latter declining faster. At low learning rates, this degeneration is manageable, and our results focus on these settings, as shown in the Figure 4 and Table II. Higher learning rates, while not depicted, can cause the model to collapse, generating repetitive tokens. Despite these limitations, DPO excels in targeted discovery, producing higher-FoM topologies within 10 attempts and achieving lower MMD values, which indicate a closer resemblance to real-world circuits.

D. Comparison between EVA and prior arts

Finally, we compare EVA with prior methods:

Validity: EVA demonstrates higher validity than previous methods. Top-down approaches like CktGNN and LaMAGIC rely on predefined representations, which limit their generalization to other circuit types and restrict the range of topologies they can be trained on. In contrast, EVA's universal circuit representation and bottom-up generation framework enable pretraining on diverse analog circuits with more training data.

Novelty: EVA excels in discovering novel circuits compared to AnalogCoder and Artisan, which primarily reuse existing topologies or subblocks. LaMAGIC is confined to a small design space (fewer than 4 devices), limiting its ability to discover new topologies. While CktGNN and EVA both support larger circuits (around 20 and 60 devices, respectively), CktGNN is trained on synthetic datasets, which lack critical real-world features. EVA, trained on real-world circuits, improves MMD by over $6\times$ and generates around 99% novel circuits.

Versatility: EVA outperforms Artisan, CktGNN, and LaMAGIC, which can only design one type of circuit. While AnalogCoder supports seven circuit types, it is limited to a synthesis library of just 20 topologies. EVA, trained on a dataset with 11 circuit types and over 3000 topologies (minimum 30 per type), demonstrates unparalleled versatility.

Sample efficiency: Unlike prior work that typically uses performance-labeled topology to train a model from scratch, EVA first pretrains on an unlabeled dataset to build a strong foundation in circuit connectivity. This enables efficient finetuning with only a small labeled dataset, achieving up to $364\times$ greater training sample efficiency compared to other methods.

Discovery efficiency: CktGNN and EVA (Pretrain) lack targeted discovery, often producing low-performance circuits within 10 attempts. AnalogCoder's synthesis library only contains simple circuits, which also limits the FoM of the circuit it synthesized. After finetuning, EVA is able to align with human preference and target high-performance circuits within limited attempts. Combining the foundation knowledge it learned from all sorts of analog circuit types during pretraining, EVA is able to discover unseen topologies with superior FoM compared to dedicated synthesizers like Artisan and LaMAGIC, excelling in both Op-Amp and power converter.

V. Conclusion

In this work, we propose EVA, a versatile and efficient generative engine for discovering novel analog circuit topologies. EVA employs a bottom-up generation framework using a decoder-only transformer to sequentially predict device pin connections, starting from a VSS pin. By pretraining on an unlabeled dataset of circuit topologies, EVA establishes a strong foundation for topology generation, enabling finetuning with minimal performance-labeled data. Through finetuning strategies based on PPO or DPO, EVA achieves high discovery efficiency, targeting high-performance circuits while minimizing wasteful evaluations. EVA paves the way for AI-driven discovery in analog circuit, unlocking unprecedented potential for innovation in next-generation technologies.

REFERENCES

[1] Z. Dong, W. Cao, M. Zhang, D. Tao, Y. Chen, and X. Zhang, "Cktgnn: Circuit graph neural network for electronic design automation," *arXiv preprint arXiv:2308.16406*, 2023.

[2] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[3] W. Cao, M. Benosman, X. Zhang, and R. Ma, "Domain knowledge-infused deep learning for automated analog/radio-frequency circuit parameter optimization," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 1015–1020.

[4] J. Gao, W. Cao, and X. Zhang, "Rose: Robust analog circuit parameter optimization with sampling-efficient reinforcement learning," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2023, pp. 1–6.

[5] W. Cao, J. Gao, T. Ma, R. Ma, M. Benosman, and X. Zhang, "Rose-opt: Robust and efficient analog circuit parameter optimization with knowledge-infused reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[6] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *International conference on machine learning*. PMLR, 2018, pp. 3306–3314.

[7] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137–148, 2009.

[8] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "Align: Open-source analog layout automation from the ground up," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–4.

[9] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "Magical: Toward fully automated analog ic layout leveraging human and machine intelligence," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[10] Z. Wu and I. Savidis, "Circuit-gnn: A graph neural network for transistor-level modeling of analog circuit hierarchies," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2023, pp. 1–5.

[11] Y. Lai, S. Lee, G. Chen, S. Poddar, M. Hu, D. Z. Pan, and P. Luo, "Analogcoder: Analog circuit design via training-free code generation," *arXiv preprint arXiv:2405.14918*, 2024.

[12] Z. Chen, J. Huang, Y. Liu, F. Yang, L. Shang, D. Zhou, and X. Zeng, "Artisan: Automated operational amplifier design via domain-specific large language model," in *2024 61th ACM/IEEE Design Automation Conference (DAC)*. ACM, 2024, pp. 1–6.

[13] C.-C. Chang, Y. Shan, S. Fan, J. Li, S. Zhang, N. Cao, Y. Chen, and X. Zhang, "Lamagic: Language-model-based topology generation for analog integrated circuits," *arXiv preprint arXiv:2407.18269*, 2024.

[14] E. A. Karahan, Z. Liu, and K. Sengupta, "Deep-learning-based inverse-designed millimeter-wave passives and power amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 11, pp. 3074–3088, 2023.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[16] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[17] J. Gao, W. Cao, J. Yang, and X. Zhang, "Analoggenie: A generative engine for automatic discovery of analog circuit topologies," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=jCPak79Kev

[18] A. Radford, "Improving language understanding by generative pre-training," 2018.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[20] N. Otsu *et al.*, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[21] S. Xu, W. Fu, J. Gao, W. Ye, W. Liu, Z. Mei, G. Wang, C. Yu, and Y. Wu, "Is dpo superior to ppo for llm alignment? a comprehensive study," *arXiv preprint arXiv:2404.10719*, 2024.

[22] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[23] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. USA: McGraw-Hill, Inc., 2000.

[24] B. Razavi and R. Behzad, *RF microelectronics*. Prentice hall New York, 2012, vol. 2.

[25] D. A. Johns and K. Martin, *Analog integrated circuit design*. John Wiley & Sons, 2008.

[26] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and design of analog integrated circuits*. John Wiley & Sons, 2009.

[27] P. E. Allen and D. R. Holberg, *CMOS analog circuit design*. Elsevier, 2011.

[28] H. Camenzind, *Designing analog chips*. Virtualbookworm Publishing, 2005.

[29] X. Guo and L. Zhao, "A systematic survey on deep generative models for graph generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5370–5390, 2022.