Artisan: Automated Operational Amplifier Design via Domain-specific Large Language Model

Zihao Chen[1,†], Jiangli Huang[1,†], Yiting Liu[2], Fan Yang[1,*], Li Shang[2], Dian Zhou[1,3,‡], Xuan Zeng[1,*]
[1]State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai, China
[2]China and Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
[3]Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA

ABSTRACT

This paper presents *Artisan*, an automated operational amplifier design framework using large language models (LLMs). We develop a bidirectional representation to align abstract circuit topologies with their structural and functional semantics. We further employ Tree-of-Thoughts and Chain-of-Thoughts approaches to model the design process as a hierarchical question-answer sequence, implemented by a mechanism of multi-agent interaction. A high-quality opamp dataset is developed to enhance the design proficiency of the Artisan-LLM. Experimental results demonstrate that Artisan outperforms state-of-the-art optimization-based methods and benchmark LLMs, in success rate, circuit performance metrics, and interpretability, while accelerating the design process by up to 50.1×. Artisan will be released for public access.

1 INTRODUCTION

The front-end design of analog circuits, focusing on topology design and parameter tuning, presents substantial challenges. Even for widely used analog circuits such as operational amplifiers (opamps), the design process remains time-consuming and labor-intensive.

Considerable research efforts have been devoted to automating the opamp design process. Parameter tuning, also known as *sizing*, is an optimization problem within continuous parametric spaces, where black-box optimization algorithms, such as Bayesian Optimization (BO), have shown mathematical efficacy [14–16]. In contrast, topology design is a high-dimensional discrete optimization problem, which is significantly more challenging. Recent studies have investigated a wide range of black-box optimization algorithms but with limited success, such as genetic algorithm (GA) [17, 21], BO [12, 13], and reinforcement Learning (RL) [3, 25]. This is because analog circuit design requires in-depth domain knowledge that spans physical understanding, mathematical reasoning, and practical engineering expertise. Without such domain knowledge, existing black-box optimization algorithms cannot efficiently explore the vast and discrete topological design space, and the produced opamp topological circuits are challenging to comprehend and hesitant to be adopted by experienced analog circuit designers.

The advent of LLMs presents fresh perspectives for EDA. LLMs excel at assimilating human knowledge conveyed in natural language, particularly benefitting knowledge-intensive tasks such as circuit design. Recent advancements have pioneered LLM-based digital circuit design. Instruction fine-tuning with open-source datasets has been used to generate Register Transfer Level (RTL) codes by LLMs [10, 19, 22]. Prompt engineering has been employed to enable GPT-4 as a design copilot [1, 2, 7]. Nevertheless, the field of LLM-assisted analog circuit design remains unexplored, possibly due to the following challenges:

Abstract circuit representation. Analog circuit structures, in the form of graphs, are represented as *netlists*. The gap between natural language semantics and the topological circuit representation, makes it challenging for LLM to connect topological structures with circuit functionality, let alone design or optimize circuits.

Knowledge-heavy design process. Opamp design is a domain-specific hierarchical process consisting of meticulous steps [9, 20], including topology selection, metric allocation, parameter calculation, and performance verification. Such a methodological and domain expertise-heavy process encompasses physical intuition, mathematical derivations, and engineering experience, which poses insurmountable barriers to off-the-shelf pre-trained LLMs.

Scarcity of Data. High-quality analog circuit designs and years of accumulated design expertise are valuable proprietary assets rarely made publicly available. Therefore, as shown in Section 4, it is unsurprising that the state-of-the-art GPT-4 (trained on public datasets) lacks sufficient knowledge in the analog circuit domain. Inevitably, constructing high-quality datasets is a prerequisite for LLM-based analog circuit design solutions.

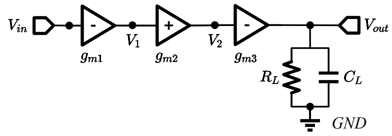
This paper presents *Artisan*, an automated opamp design framework using domain-specific LLM to tackle the aforementioned challenges. To the best of our knowledge, this is the first work applying LLMs to analog circuit design. Our contributions are as follows:

- Artisan offers a domain knowledge-based opamp front-end design workflow, from topology design to parameter tuning.
- We propose a bidirectional circuit representation for the Artisan-LLM to align the topological opamp netlist with the natural language semantics. Opamp structures are thereby automatically aligned with the functional descriptions in our training corpus.
- We model the design process as the Tree-of-Thoughts (ToT)-based decision-making [24] and the Chain-of-Thoughts (CoT)-based design flow [23]. A multi-agent question-answer (QA) framework is then proposed to decompose the hierarchical and methodological design process as learnable steps.

[†]Both authors contributed equally to this research.
[*]Corresponding authors: {yangfan, xzeng}@fudan.edu.cn.
[‡]Emeritus Professor, University of Texas at Dallas.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3655903



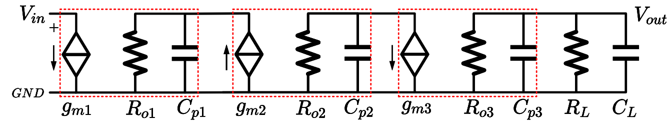


Figure 1: Behavior-level opamp modeling. Fig. 1(a) displays the basic three-stage opamp topology. Fig. 1(b) illustrates the associated small-signal model, with each red dotted box indicating an opamp stage.

- We produce a high-quality dataset with domain-specific corpus, encoded human expertise, and leveraged data augmentation techniques. The Artisan-LLM is trained on the dataset to drive the proposed Artisan framework.

Experimental results demonstrate that, targeting a wide range of design specifications (specs), Artisan outperforms state-of-the-art opamp optimization methods [3, 12], as well as the off-the-shelf GPT-4 [18] and Llama2 [4] in terms of circuit performance, the interpretability of design processes and results, overall efficiency, and the success rate. The model will be released for public access.

The rest of the paper is organized as follows. Section 2 introduces preliminaries. Section 3 describes the proposed approach. Section 4 presents experimental results. We conclude this paper in Section 5.

2 PRELIMINARIES

2.1 Problem Formulation

Opamp design can be mathematically formulated as an optimization problem, which is realized by adjusting the topology $g \in \mathcal{G}$ and a set of parameters $\mathbf{x} \in \mathcal{X}$ to meet design specs, as shown in Eq. (1).

$$\begin{aligned} &\max_{g\in\mathcal{G},\ \mathbf{x}\in\mathcal{X}} f\left(g,\mathbf{x}\right) \\ &s.t.\ c_i\left(g,\mathbf{x}\right) > c_{th}^i,\ \forall i \in \{1,...,N_c\}, \end{aligned} \tag{1}$$

where $c_i(\cdot)$ and c_{th}^i are the i-th metric and its corresponding spec, respectively. N_c denotes the number of metrics, which often include Gain, gain-bandwidth product (GBW), phase margin (PM), and Power. The objective function $f(\cdot)$ reflects the overall circuit performance, see Eq. (6) in Section 4.1.3 for details.

2.2 Opamp Topological Modeling

Three-stage opamps have garnered the most research interest over the years because they offer a reasonable trade-off between design complexity and performance versatility [9]. This paper focuses on the three-stage opamp design, which can be easily extended to support other opamp topologies.

Fig. 1(a) depicts the canonical three-stage cascode skeleton with five initial nodes. The i-th opamp stage is modeled by an ideal voltage-controlled current source (VCCS) g_{mi}, with a parallel pair of lumped output resistance R_{oi} and lumped parasitic capacitance C_{pi}, as shown in Fig. 1(b). Topological meta-modifications include adding feedforward (or feedback) transconductance stages, resistors, and capacitors at a set of legitimate positions.

The classical g_m/I_d method [8] can map opamps to the transistor level. We map the stage connected to the input node to a current mirror differential amplifier and the remaining stages to common source amplifiers, as illustrated in Fig. 6(c) and Fig. 6(d).

3 METHOD

This section presents Artisan. Section 3.1 summarizes the Artisan workflow. Section 3.2 introduces circuit representation in LLM. Section 3.3 details the LLM-based design process. Section 3.4 presents the dataset and the Artisan-LLM training method.

3.1 Overall Workflow

The workflow of Artisan is summarized in Fig. 2. Artisan is a multi-agent framework that decomposes the opamp design flow into a step-by-step domain-specific QA sequence using ToT [24] and CoT [23] methods. Since opamp design often involves tool-based formula calculations, Artisan is required to interact with third-party tools. To facilitate the multi-agent paradigm and tool invocation, Artisan builds upon the open-source Langchain framework [6], which provides APIs for customize multi-agent frameworks. Moreover, it enables LLM to invoke auxiliary tools via prompt instructions.

Given user-defined specs, Artisan recommends a suitable opamp architecture and then performs the detailed design flow, and produces a behavioral-level netlist as the design result. If this design fails to meet the specs, Artisan can also provide topological modification suggestions to explore the opamp design space using the parameter tuning tool [14]. The final opamp design is mapped to the transistor level using the open-source g_m/I_d scripts [11].

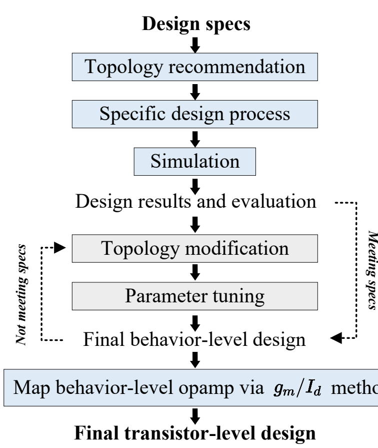


Figure 2: The overall workflow of Artisan.

3.2 Circuit Representation Alignment

This section presents circuit representation and alignment with LLM semantics. In summary, we construct a bidirectional circuit representation that aligns circuit topological structures with natural language descriptions. This allows Artisan's domain-specific LLM to establish semantic connections between topological structures and circuit functionality, enabling the opamp design and optimization based on domain knowledge.

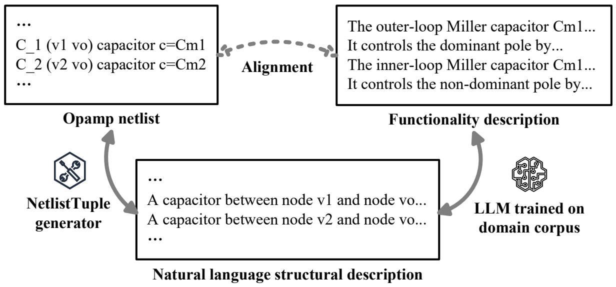


Figure 3: The bidirectional circuit representation.

3.2.1 Bidirectional Circuit representation. *Netlists* are the de facto representation of analog circuits, which describe the connectivities among devices in the form of graphs and sub-graphs (see Fig. 3). However, such graph representation cannot directly be used to Artisan due to the following reasons. 1) There is a clear gap between natural language semantics and circuit topological graph representation, impeding Artisan's generalization ability. 2) The netlist only depicts circuit structure, failing to capture the intrinsic circuit functional properties, let alone for circuit functional design and performance optimization.

To address the issues above, we propose a bidirectional circuit representation, NetlistTuple, to establish semantic alignment between circuit topological netlists and natural language representations. NetlistTuple is defined as follows.

$$\text{NetlistTuple}_i = (\text{netlist}_i, \text{description}_i), \tag{2}$$

where description_i serves as a textual description of the circuit netlist_i. As illustrated in Fig. 3, this natural language description aids Artisan in accurately identifying the circuit structural semantics within the netlist. Due to the prevalence of opamp functionalities described in natural language within Artisan's pre-training corpus, a semantic alignment exists between the opamp functionalities and the above *description*. With both the circuit structure and its functionalities aligned with natural language descriptions, the semantic gap between them is substantially eliminated. This alignment can enhance the generalization capabilities of Artisan.

3.2.2 NetlistTuple Generator. We develop a NetlistTuple generator for netlist sampling and automated annotation. We set all opamp topological connections except the basic skeleton in Fig. 1(a) tunable, each with 25 optional types, yielding up to one million opamp samples [11]. The generator randomly selects connection types for each tunable connection and assembles the netlists. It also produces the corresponding structural description of the netlist based on a rule-based connection type and position matching. We apply data augmentation techniques in Section 3.3 for data diversity.

Together, the produced NetlistTuple dataset serves as a bridge to achieve semantic alignment between the netlist and opamp functionalities, enabling Artisan to manipulate netlists adeptly based on domain knowledge.

3.3 Design Process Acquisition

This section introduces the methodology for driving Artisan to learn the domain-specific knowledge-heavy opamp design process. As shown in Fig. 4, we employ a hierarchical modeling approach, segmenting the design process in Fig. 2 into a top-level ToT-based decision-making process and a bottom-level CoT-based design flow. A multi-agent framework is utilized to organize the knowledge-intensive design flow as a cascading QA sequence, which are implemented through a CoT decomposition approach to guide the LLM in acquiring elaborate step-specific design skills.

3.3.1 ToT-based decision-making. As shown in Fig. 4, from a top-level perspective, the opamp design process is a decision tree. The first decision point is to choose an appropriate architecture based on the specs. The second decision point is to modify the architecture based on simulation feedback, especially in complex scenarios where a single design iteration might not suffice.

Acquisition of high-quality, annotated domain-specific data is crucial for ToT-based learning. To this end, we conduct a thorough analysis of opamp-related surveys [9, 20], drawing insights from the human expertise. Subsequently, we meticulously annotate details regarding to the performance preferences of mainstream architectures and the potential impacts of various architectural modification strategies. These data encode human expertise and equip Artisan with the capacity to make reasonable decisions.

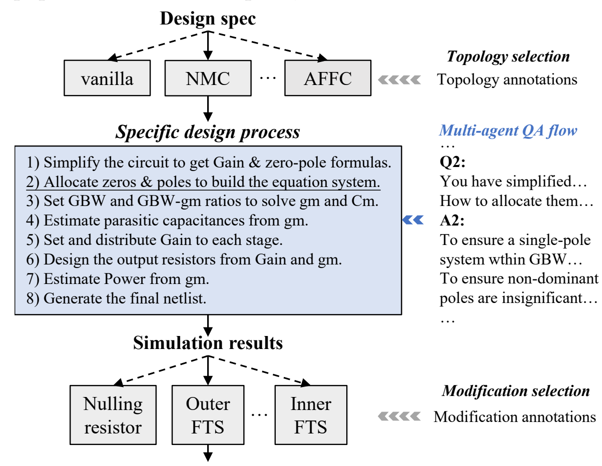


Figure 4: The hierarchical design process modeling.

3.3.2 CoT-based design flow. A detailed methodological design flow is employed for a specific opamp architecture. The design flow of a nested Miller compensation (NMC) opamp is shown in Fig. 4, where multiple stages like topology selection, zero-pole allocation, metric allocation, and performance verification, are distilled into eight distinct, logically connected, and rigorously executed steps.

Multi-agent modeling approach. We model the knowledge-heavy design process as a cascading QA sequence between a GPT-4-based question agent Artisan-Prompter and an domain-specific answering agent Artisan-LLM. During the opamp design process, Artisan-Prompter prompts Artisan-LLM to complete design tasks step by step. Fig. 5 illustrates the multi-agent interaction process. At the i-th design step, Artisan-LLM receives question Q_i to generate an answer A_i (see Eq. 3). Then, Artisan-Prompter receives the answer A_i to produce the next question Q_{i+1} (see Eq. 4). The first question Q_0 is human-defined design specs, and the last answer A_N is the final design proposal.

$$A_i = \text{Artisan-LLM}(Q_i), \tag{3}$$

$$Q_{i+1} = \text{Artisan-Prompter}(A_i). \tag{4}$$

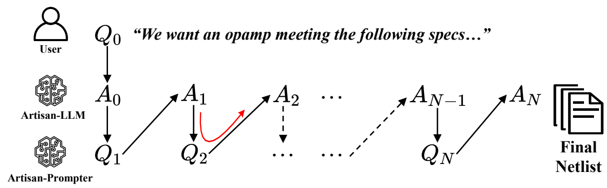


Figure 5: The CoT-based multi-agent modeling.

Take steps 2 and 3 in Fig. 4 as an example, Artisan-Prompter asks Artisan-LLM how to allocate zeros and poles, Artisan-LLM answers and asks Artisan-Prompter what to do next; Artisan-Prompter then prompts Artisan-LLM to solve the derived equations. Artisan-LLM thereby calculates the core parameters.

Decomposition-based learning method. Similar to the ToT-based decision-making process, we also engage human experts to annotate design documents for various types of opamp circuits in a QA format. With these design documents, we construct the DesignQA dataset to train Artisan-LLM in producing reliable answers to questions posed by Artisan-Prompter:

$$\text{DesignQA} = \{Q(i) \rightarrow A(i) | 0 \leq i \leq N\}. \quad (5)$$

As for the AQ process depicted in Eq. 4, it can be achieved using GPT-4 as the prompter through in-context learning. In this way, the entire design chain is decoupled into an interconnected QA sequence, significantly simplifying the task of guiding Artisan-LLM in learning the opamp design process. We employ the data augmentation technique described in Section 3.4 to scale and diversify the proposed DesignQA dataset, thereby enhancing the training effectiveness of Artisan-LLM.

3.4 Artisan-LLM and Opamp Dataset

Artisan-LLM adopts the open-source pre-trained Llama2-7b [4] as the foundational model, augmented with opamp design knowledge through a two-step training process. The first step is domain-adaptive pretraining (DAPT) on our domain-specific pre-training dataset. During this phase, Artisan is guided to acquire background knowledge of opamp design, encompassing theoretical knowledge and netlist interpretation. Subsequently, we proceed to execute supervised fine-tuning (SFT) on the fine-tuning dataset, where Artisan is directed to follow human instructions and proficiently executes the design process.

To address the scarcity of publicly available data, we construct a high-quality opamp dataset by collecting related corpus, annotating human expertise, and employing data augmentation techniques.

Analog circuit-related corpus. We collect a diverse range of analog circuit-related corpus, from websites encompassing forum posts, tutorial documents, and technical papers, as well as textbooks and research papers focusing on opamp design. A total of 142M tokens of data have been collected.

Annotating human expertise. The above corpus does not elucidate the methodological opamp design process. Moreover, accurately assessing the performance limitations of specific circuit architectures requires extensive practical design experience, which is not publicly accessible. Therefore, human experts are engaged to annotate high-quality design documents in section 3.3.

Data argumentation. The effectiveness of datasets hinges on their data diversity and volume. NetlistTuple in Section 3.2 includes formatted netlists and monotonous description statements, thus lacking diversity. Similarly, DesignQA in Section 3.3 comprises annotated, templated design documents lacking diversity and quantity. Therefore, we employ the ChatGPT-API to enhance data diversity by rephrasing the texts in NetlistTuple and DesignQA.

Table 1: The Detailed Information of The Dataset

	Name	Samples (k)	Tokens (M)
Pre-training	Collected corpus	225	142
	NetlistTuple	13	23
	Total	**238**	**165**
Fine-tuning	Alpaca dataset	52	9
	DesignQA	14	16
	Total	**66**	**25**

Finally, we allocate the above datasets to different training stages. In addition, we also incorporate the 52K instruction-following data from Alpaca [5] project in the fine-tuning dataset. As the general-purpose chat instruction dataset, it endows Artisan with excellent general chat capabilities. The detailed information of the overall dataset is presented in Table 1. Practice shows the effectiveness of our small-scale yet high-quality dataset.

4 EVALUATION

4.1 Experiment Setting

4.1.1 Baselines. We evaluate the performance of Artisan by comparing it with the state-of-the-art black-box optimization approaches BOBO [12] and RLBO [3], and off-the-shelf LLMs including GPT-4 [18] and the open-source Llama2-7b-chat [4] (Llama2 for short).

4.1.2 Implementation details. Artisan is deployed on a server with 8 NVIDIA A100 GPUs for both training and inference processes. During the pertaining phase and the follow-up fine-tuning phase, we use the Adam optimizer with an initial learning rate of 2×10^{-6}, a batch size of 128, and a 4096-token context window.

4.1.3 Circuit settings. We consider four most important metrics of opamp: *Gain*, *GBW*, *PM*, and *Power*. The small-signal Figure of Merit (*FoM*) in Eq. (6) serves as the optimization object in Eq. (1):

$$FoM = \frac{GBW\ [\text{MHz}] \cdot C_L\ [\text{pF}]}{Power\ [\text{mW}]}. \quad (6)$$

The load resistance is set to 1 MΩ and the supply voltage is 1.8 V. All circuit simulations are performed using *Cadence Spectre* simulator.

Table 2: The Experimental Group Settings

Groups	*Gain*(dB)	*GBW*(MHz)	*PM*(°)	*Power*(μW)	C_L(pF)
G-1	>85	>0.7	>55	<250	10
G-2	>110	>0.7	>55	<250	10
G-3	>85	>5	>55	<250	10
G-4	>85	>0.7	>55	<50	10
G-5	>85	>0.7	>55	<250	1000

4.1.4 Experimental Groups. As shown in Table 2, our tests span a range of operating conditions to comprehensively evaluate each method's opamp design capabilities. First, we evaluate each method under the requirements specified in group G-1. Following this, groups G-2 to G-5 are established, each with higher requirements for one metric than G-1. Specifically, groups G-2 to G-5 emphasize high gain, high GBW, low power consumption, and ultra-large capacitive load driving capability, respectively.

Table 3: The Performance Comparison

	Exps	Succ. Rate	*Gain* (dB)	*GBW* (MHz)	*PM* (°)	*Power* (μW)	*FoM*	Time
BOBO	G-1	3/10	93.3	2.49	78.84	120.4	284.3	4.55h
	G-2	2/10	106.1	3.22	97.94	234.3	161.0	4.62h
	G-3	0	fail	fail	fail	fail	fail	6.09h
	G-4	0	fail	fail	fail	fail	fail	4.83h
	G-5	3/10	86.8	1.23	70.88	145.7	8578.2	4.93h
RLBO	G-1	3/10	102.7	5.32	68.76	221.4	247.4	5.28h
	G-2	2/10	106.5	4.73	84.97	226.0	220.8	5.38h
	G-3	2/10	96.8	5.28	83.78	102.4	468.6	5.44h
	G-4	0	fail	fail	fail	fail	fail	6.63h
	G-5	4/10	73.9	0.86	79.41	81.95	10921.8	6.29h
GPT-4	all	0	fail	fail	fail	fail	fail	-
Llama2	all	0	fail	fail	fail	fail	fail	-
Artisan	G-1	9/10	106.5	1.02	60.96	47.8	**289.2**	**7.68m**
	G-2	9/10	**112.0**	1.55	69.71	67.9	**255.5**	**8.54m**
	G-3	8/10	100.0	**6.79**	77.00	167.1	**504.7**	**15.58m**
	G-4	8/10	102.2	0.73	62.23	**30.4**	**258.3**	**7.94m**
	G-5	7/10	98.2	1.54	60.02	147.8	**12769.5**	**14.53m**

4.2 Experimental Result

We test BOBO, RLBO, GPT-4, Llama2, and Artisan on each experimental group 10 times to obtain average performance metrics. A comprehensive comparison of experimental outcomes, including the metrics, success rate, and execution time, is summarized in Table 3. GPT-4 and Llama2 consistently fail to design opamps in any instance, which is primarily attributed to the deficiency of their opamp-related knowledge and the absence of tailored training.

Artisan consistently outperforms BOBO and RLBO in terms of success rate across all experimental groups. Moreover, Artisan accelerates the design process by 20.4× to 50.1× while maintaining competitive FoM values. This enhanced performance is largely due to Artisan's integration of domain-specific knowledge and well-established design methodologies, enabling it to emulate human design expertise effectively. With this expertise, Artisan's design process mitigates the extensive trial-and-error simulations, thereby significantly increasing efficiency. On the other hand, both BOBO and RLBO are limited by the randomness in black-box search, leading to unstable success rates and long search time. Particularly, in all repeated tests in the G-4 group, they fail to achieve any success.

4.3 Design Example

To demonstrate the proficiency, especially the interpretability of Artisan, an illustrative design example is presented in Fig. 7. Artisan adeptly recommends the NMC architecture that aligns with the given specs (G-1). Then, it adopts the Butterworth design approach, starting with zero-pole analysis. Throughout the multi-agent-based design flow, Artisan accurately follows each step summarized in Fig. 4, and autonomously invokes the calculator if necessary (as in the $Q_3 \rightarrow A_3$ phase). Artisan also reasonably modifies the NMC design by adding the DFC block to drive the 1nF load. Note that benefiting from the circuit representation alignment method, Artisan can directly deliver the netlist of the final circuit. The final circuit is shown in Fig. 6(c) and Fig. 6(d).

On the other hand, BOBO and RLBO fail to earn such interpretability. Fig. 6(a) and Fig. 6(b) show the two opamps from [12] and [3], respectively. Among them, the series connection of feed-forward transconductance and capacitance or resistance is usually considered to be uninterpretable. In summary, Artisan exhibits interpretability, offering a significant advantage over black-box optimization methods BOBO and RLBO.

We also employ GPT-4 and Llama2 for opamp design. Fig. 7 displays snippets of their chat logs. Not surprisingly, off-the-shelf LLMs cannot execute a complete design process without specialized training. Therefore, we can only manually query them to test their single-step QA capabilities. GPT-4 appropriately recommends the NMC topology but provides an incorrect dominant-pole formula. Moreover, it suggests the MPMC architecture for topology modification, which cannot drive 1nF load.

Llama2 exhibits poorer performance with its considerably smaller model size. It either provides irrelevantly basic opamp principles or gives unprofessionally suggestions.

4.4 Experiment Summary

In summary, black-box optimization methods BOBO and RLBO exhibit low success rates, low efficiency, and a lack of interpretability. GPT-4 and Llama2 possess opamp-related knowledge but exhibit low proficiency due to insufficient training data. Artisan utilizes circuit representation alignment and employs the multi-agent QA framework, enabling a proficient opamp design process. Moreover, Artisan adopts well-established human design methodologies, which not only accelerates the design process by up to 50.1×, but also provides interpretability for the resulting circuit.

5 CONCLUSION

This paper introduces Artisan, a novel LLM-based automated opamp design framework. We propose a bidirectional circuit representation to align the circuit topologies with their structural and functional semantics. We further model the knowledge-intensive design process as ToT-based decision-making and CoT-based design flow, implemented by a multi-agent QA mechanism. We tailor a high-quality opamp dataset for Artisan to excel in opamp design. Experimental results show that, compared to state-of-the-art methods, Artisan not only significantly accelerates the design process with competitive circuit performance, but also offers design interpretability.

ACKNOWLEDGMENTS

This research is supported partly by National Key R&D Program of China 2020YFA0711900, 2020YFA0711901, partly by National Natural Science Foundation of China (NSFC) research projects 62141407, 62090025 and 92373207. The computations in this research were performed using the CFFF platform of Fudan University.

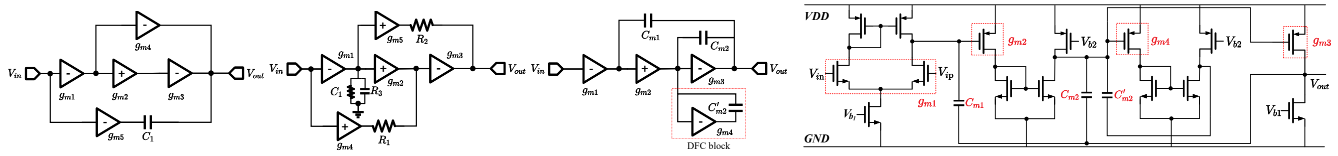
REFERENCES

[1] Jason Blocklove, Siddharth Garg, Ramesh Karri, and Hammond Pearce. 2023. Chip-Chat: Challenges and Opportunities in Conversational Hardware Design. *arXiv preprint arXiv:2305.13243* (2023).

[2] Kaiyan Chang, Ying Wang, Haimeng Ren, Mengdi Wang, Shengwen Liang, Yinhe Han, Huawei Li, and Xiaowei Li. 2023. ChipGPT: How far are we from natural language hardware design. *arXiv preprint arXiv:2305.14019* (2023).

[3] Zihao Chen, Songlei Meng, Fan Yang, Li Shang, and Xuan Zeng. 2023. TOTAL: Topology optimization of operational amplifier via reinforcement learning. In *2023 24th Int'l Symposium on Quality Electronic Design (ISQED)*. IEEE.

[4] Hugo Touvron et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]



(a) (b) (c) (d)

Figure 6: Opamp design examples. Fig. 6(a) and Fig. 6(b) show two typical opamp results in BOBO and RLBO literature [12] and [3]. Fig. 6(c) and Fig. 6(d) show behavioral-level and transistor-level schematics of the example circuit produced by Artisan.

A chat log example of Artisan.

Q0: Please design an opamp meeting the following specs: gain >85dB, PM >55dB, GBW >0.7MHz, and Power <250uW with capacitive load CL = 10pF.
A0: In this situation, you can use the classic nested Miller compensation (NMC) architecture because... In NMC architecture, there are two nested Miller capacitors, Cm1 and Cm2, to control the dominant and non-dominant poles, respectively. The design process is as follows:...

Q1: Based on the process, please analyze zero-pole distributions...
A1: Under the Miller effect of compensation capacitors Cm1 and Cm2, the transfer function is... Thus p1=1/[2*π*Cm1*gm2*gm3*Ro1*Ro2*(Ro3||RL)]...

Q2: How to allocate these poles in an NMC opamp?
A2: We set p1<GBW<p2 and p3 to build a single-pole system within the frequency ranging from 0 to GBW. In this condition, since Av= gm1*gm2*gm3*Ro1*Ro2*(Ro3||RL), we have GBW = Av*p1=gm1/Cm1. According to Butterworth methodology, we set GBW:p2:p3=1:2:4 to ensure... Finally, we get the following equations...

Q3: Please solve main design parameters from these equations...
A3: We empiracally set GBW=1MHz, thus gm3=...=8*π*GBW*CL=251.2u. Generally, the Cm1 and Cm2 are both in the pF level. We can take Cm1=4pF and Cm1=3pF, then we can get: gm1=gm3*Cm1/(4*CL)=...=25.12u and gm2=gm3*Cm2/(2*CL)=...=37.68u...
...

Q8: Design completed. Please give the final netlist...
A8: The final netlist with parameters instantiated is as follows...

Q9: When CL=1nF, the above NMC design suffers... How to...
A9: The NMC architecture fails to drive the large CL because... We can add a damping-factor-control (DFC) block with a gain stage gm4 and a feedback capacitor Cm3. The DFC block functions as a frequency-dependent capacitor because... Besides, the inner-loop Miller compensation capacitor Cm2 should be canceled because... The netlist is thus modified into:
...

A chat log example of GPT-4.

Q0: I want to design a behavioral level three-stage opamp meeting the following specs... Please recommend...
A0: NMC: Nested Miller Compensation is particularly effective for multi-stage amplifiers... 1) providing better PM and frequency compensation in three-stage cases. 2) allowing for trade-offs between..., thus adapting to your specs. 3) handling varying load conditions...

Q1: The NMC design flow includes... Please analyze the zero-pole distributions...
A1: ...1) The dominant pole is determined by the output stage and the load: p1=gm3/CL... 2) Non-dominant poles are higher due to compensation...
...
Q9: When CL=1nF, the NMC opamp suffers... How to modify...
A9: ... 1) Increase the compensation capacitance values to handle a larger load, which may impact bandwidth. 2) Consider the multi-path Miller compensation (MPMC) technique to add new path for the compensation...

A chat log example of Llama2.

Q0: I want to design a behavioral level three-stage opamp meeting the following specs... Please recommend...
A0: ...You can use a multi-stage opamp architecture... Stage 1: current feedback opamp... Stage 2: voltage follower... Stage 3: voltage follower...

Q1: The NMC design flow includes... Please analyze the zero-pole distributions...
A1: ...z=(R1+R2)/(2*R3) and p=(R1+R2)/(2*R3), where R1 and R2 are feedback resistors, and R3 is the input impedance...
...
Q9: When CL=1nF, the NMC opamp suffers... How to modify...
A9: ... 1) Increase the Miller capacitance values... 2) Adjust the transconductance ratios of the three stages to reduce the load on each stage... 3) Increase the number of stages...

Figure 7: The chat logs of Artisan and LLM baselines. We present selected chat log excerpts to illustrate Artisan's design process and compare it with GPT-4 and Llama2. The conversation is summarized, with details like specific formula calculations omitted.

[5] Rohan Taori et al. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[6] Asbjørn Følstad and Marita Skjuve. 2019. Chatbots for customer service: user experience and motivation. In *Proceedings of the 1st international conference on conversational user interfaces*. 1–9.

[7] Yonggan Fu, Yongan Zhang, Zhongzhi Yu, Sixu Li, Zhifan Ye, Chaojian Li, Cheng Wan, and Yingyan Lin. 2023. GPT4AIGChip: Towards Next-Generation AI Accelerator Design Automation via Large Language Models. *arXiv preprint arXiv:2309.10730* (2023).

[8] Paul Jespers. 2009. *The gm/ID Methodology, a sizing tool for low-voltage analog CMOS Circuits: The semi-empirical and compact model approaches.* Springer Science & Business Media.

[9] Ka Nang Leung and Philip KT Mok. 2001. Analysis of multistage amplifier-frequency compensation. *IEEE transactions on circuits and systems I: fundamental theory and applications* 48, 9 (2001), 1041–1056.

[10] Mingjie Liu, Nathaniel Pinckney, Brucek Khailany, and Haoxing Ren. 2023. VerilogEval: Evaluating Large Language Models for Verilog Code Generation. *arXiv preprint arXiv:2309.07544* (2023).

[11] Jialin Lu, Liangbo Lei, Jiangli Huang, Fan Yang, Li Shang, and Xuan Zeng. 2023. Automatic Op-Amp Generation from Specification to Layout. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2023).

[12] Jialin Lu, Liangbo Lei, Fan Yang, Li Shang, and Xuan Zeng. 2022. Topology optimization of operational amplifier in continuous space via graph embedding. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 142–147.

[13] Jialin Lu, Liangbo Lei, Fan Yang, Changhao Yan, and Xuan Zeng. 2021. Automated compensation scheme design for operational amplifier via bayesian optimization. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 517–522.

[14] Wenlong Lyu, Pan Xue, Fan Yang, Changhao Yan, Zhiliang Hong, Xuan Zeng, and Dian Zhou. 2017. An efficient bayesian optimization approach for automated optimization of analog circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 6 (2017), 1954–1967.

[15] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. 2018. Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*. PMLR, 3306–3314.

[16] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. 2018. Multi-objective bayesian optimization for analog/rf circuit synthesis. In *Proceedings of the 55th Annual Design Automation Conference*. 1–6.

[17] Claudio Mattiussi and Dario Floreano. 2007. Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on evolutionary computation* 11, 5 (2007), 596–607.

[18] OpenAI. 2023. GPT-4 Technical Report. (2023). arXiv:2303.08774 [cs.CL]

[19] Hammond Pearce, Benjamin Tan, and Ramesh Karri. 2020. Dave: Deriving automatically verilog from english. In *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*. 27–32.

[20] Joseph Riad, Johan J Estrada-López, and Edgar Sánchez-Sinencio. 2019. Classification and design space exploration of low-power three-stage operational transconductance amplifier architectures for wide load ranges. *Electronics* 8, 11 (2019), 1268.

[21] Žiga Rojec, Árpád Bűrmen, and Iztok Fajfar. 2019. Analog circuit topology synthesis by means of evolutionary computation. *Engineering Applications of Artificial Intelligence* 80 (2019), 48–65.

[22] Shailja et al. Thakur. 2023. Benchmarking Large Language Models for Automated Verilog RTL Code Generation. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–6.

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL]

[24] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601 [cs.CL]

[25] Zhenxin Zhao and Lihong Zhang. 2022. Analog integrated circuit topology aynthesis with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2022).