AMSbench: A Comprehensive Benchmark for Evaluating MLLM Capabilities in AMS Circuits

Abstract

Multimodal large language models (MLLMs) are extensively applied in the field of Analog / Mixed-Signal (AMS) circuit design, demonstrating significant potential in circuit analysis and design. However, there is currently no comprehensive benchmark to assess the performance of existing models. To address this, we propose AMSbench, which includes tasks related to circuit schematic perception, circuit analysis, and circuit design. The AMSbench test set includes approximately 10,000 questions across varying difficulty levels for each task, and evaluates eight mainstream models, including both open-source and closed-source systems such as Qwen 2.5-VL-72B and Gemini 2.5 pro. The evaluation results reveal notable limitations of current MLLMs in addressing complex multimodal and circuit design tasks. These findings emphasize the need to enhance model understanding and application of circuit knowledge to narrow the gap between human expertise and model performance, ultimately aiming to enable fully automated AMS circuit design workflows.

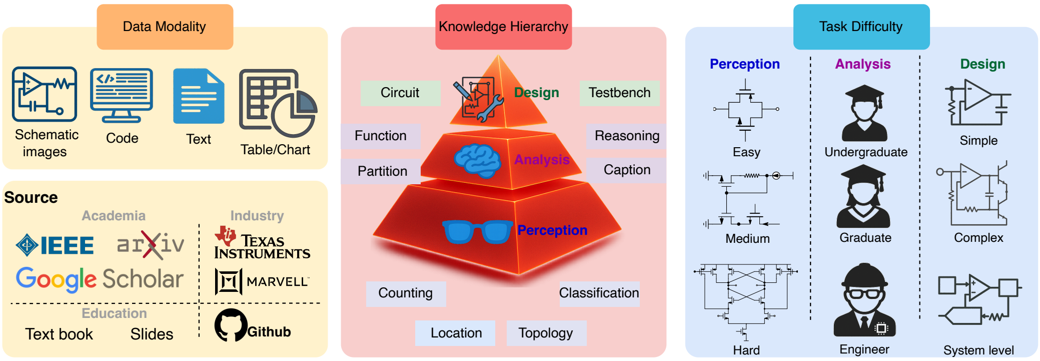


Figure 1: **Overview of AMSbench.** AMSbench includes multimodal question-answer pairs collected from both academia and industry. The tasks are divided into schematic perception, circuit analysis, and circuit design.

1 Introduction

The rapid advancement of large language models (LLMs) and multimodal large language models (MLLMs) has led to significant breakthroughs across diverse domains, including autonomous driving [1], scientific research [2, 3], mathematics [4, 5, 6], and programming [7]. In the domain of Electronic Design Automation (EDA), these models have shown promise, particularly in the automated design of digital circuits [8]. However, the design of analog/mixed-signal (AMS) circuits continues to pose considerable challenges due to the scarcity of high-quality data and the intrinsic

Preprint. Under review.

complexity of multimodal data. As a result, the exploration and application of LLMs in AMS circuit design remain limited and exhibit relatively poor performance [9, 10, 11]. Furthermore, current applications focus on verbal information, while AMS circuits rely on other modalities as well, such as schematics, plots, and charts.

A primary obstacle lies in the limited capability of existing MLLMs to accurately interpret circuit schematics. Unlike netlists, schematics convey richer and more nuanced structural information beyond abstract connectivity. Recent work [12, 8] has recognized this limitation and introduced tools capable of automatically converting schematics into netlists, thereby enabling the creation of large-scale, high-quality datasets suitable for training models. With the recent advancements in the visual capabilities of MLLMs—such as GPT-4o [13] and Qwen2.5 [14]—schematic recognition accuracy has improved significantly, laying a solid foundation for the automated analysis and design of AMS circuits. Despite these advancements, current applications often focus on isolated tasks—such as netlist generation [10, 15] and bug identification [16]—while lacking comprehensive evaluation frameworks.

In particular, there has been little systematic investigation into the following three fundamental questions:

1. How accurately can models recognize and interpret analog/mixed-signal circuit schematics?

2. What is the upper bound of domain-specific knowledge that models can attain in AMS circuit analysis and design?

3. To what degree are models capable of supporting the automation of AMS circuit design?

To address these questions and bridge the existing research gaps, we propose AMSbench, a comprehensive benchmark designed to evaluate the capabilities of advanced models in the context of AMS circuit design. AMSbench assesses model performance across three key dimensions: **perception**, **analysis**, and **design**.

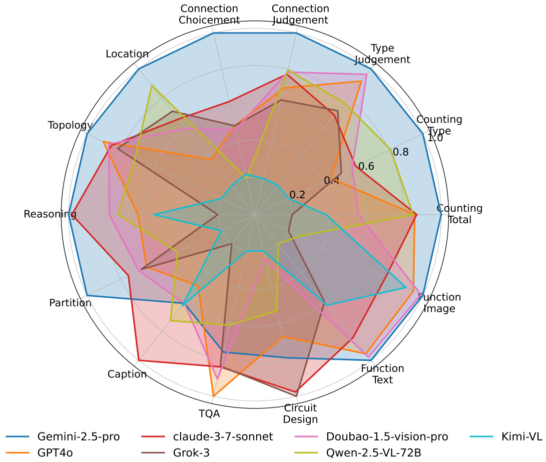


Figure 2: Comparison of 7 top MLLMs on 14 subtasks

In the perception task, the objective is to evaluate how accurately MLLMs can generate netlists directly from circuit schematics, reflecting their schematic recognition capabilities. This is a non-trivial challenge due to the large number of components and their intricate interconnections. We further decompose this task into sub-tasks such as component counting, type identification, and connection relationship recognition, culminating in the primary goal of accurate netlist generation. The analysis task examines the models' understanding of circuit functionality, their ability to identify critical building blocks, and their comprehension of trade-offs among performance metrics—key aspects in AMS circuit design and test. Finally, the design task investigates whether models can synthesize circuits that satisfy given specifications. We also evaluate their ability to generate appropriate testbenches to assess circuit performance across multiple criteria.

To the best of our knowledge, AMSbench is the first holistic benchmark that systematically evaluates the performance of advanced models in AMS circuits. The overall benchmarking results of state-of-the-art models using AMSbench are illustrated in Fig. 2. Our contributions are summarized as follows:

- We introduce **AMSbench**, a multimodal benchmark designed to rigorously evaluate the perception, analysis, and design capabilities of models in the AMS circuit domain. AMSbench consists of three major components: AMS-Perception (6k), AMS-Analysis (2k), and AMS-Design (68).
- We conduct a comprehensive evaluation of both open-source and proprietary models on AMSbench, providing detailed comparisons and performance insights across all tasks.
- We release the AMSbench dataset at the provided URL, fostering transparency and reproducibility in this emerging research area.

2 Related Work

2.1 LLM for circuit design

LLMs have demonstrated remarkable potential in the field of EDA, excelling in tasks related to system-level design [17], RTL [18, 19], synthesisand physical design of digital circuits. This success is primarily due to the modular nature of digital circuit descriptions, which resemble code. However, AMS circuit designs, with their transistor-level descriptions, pose a significantly greater challenge for LLMs in terms of accurate understanding and description. Despite this, some exploratory work has been undertaken in AMS circuit design [20, 21]. Artisian [11] develop an LLM that automatically generates operational amplifiers by combining advanced prompt engineering techniques like SFT and Tree of Thought. Analogcoder [10] propose using LLMs with predefined sub-circuit libraries for iterative design. AnalogGenie [9] converts circuit topologies into Eulerian circuit representations and uses SFT for synthesizing new circuits. Notably, to ensure the synthesis of circuits that meet specifications, AnalogGenie applies RLHF [22] (Reinforcement Learning with Human Feedback) as a post-training technique. ADO-LLM [23] combines LLMs with Bayesian optimization to generate higher-quality candidate points, enhancing efficiency in the sizing process. Layout Copilot uses multiple intelligent agents to improve the efficiency and performance of automated layout generation. AMSnet-kg [24] employs a knowledge graph-based RAG (Retrieval-Augmented Generation) approach, utilizing a large-scale, pre-constructed circuit database to select and generate circuit topologies that meet specifications. However, it is worth noting that these works mainly focus on purely language-based LLMs, while circuit design often relies heavily on schematic diagrams. Both CHAI [8] and AMSnet [12] point out that existing MLLMs still lack the capability to effectively recognize circuit schematics.

2.2 Benchmarking for EDA

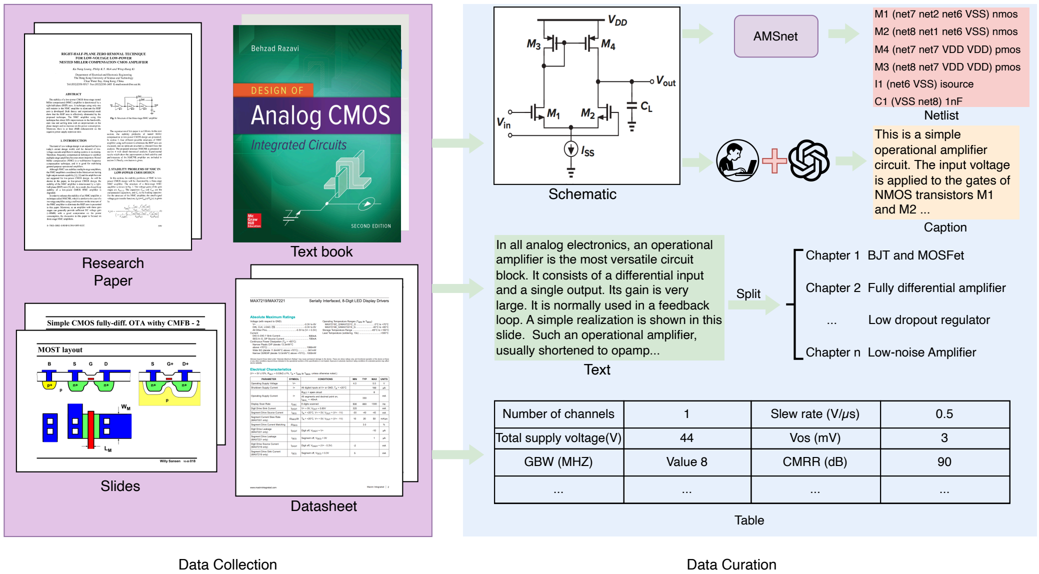
The academic infrastructure for LLM research in EDA has made significant progress, with many available benchmarks and datasets that facilitate more effective development of LLMs in the EDA field. VerilogEval [25] introduces a benchmark for evaluating Verilog code generation, while RTLLM [26] develops a benchmark for evaluating RTL code generation. However, these benchmarks focus primarily on digital circuits, and due to the complexity of analog circuits, benchmarks in the analog circuit domain are still lacking. Analogcoder [10] proposes a benchmark to evaluate LLMs in AMS circuit design, categorizing circuits into two levels: simple and complex. [27] presents a benchmark to assess LLMs' understanding of AMS circuits, including 510 simple questions. Currently, benchmarks in the AMS circuit and EDA domains are limited to LLMs and do not comprehensively evaluate MLLMs' ability to recognize, understand, and reason about circuit schematics or assess AMS circuit design. To address these gaps, we propose AMSbench.

3 AMSbench Construction

3.1 Data Collection and Curation

To cover a wide range of knowledge and typical question types in the AMS circuit domain, we gather a diverse collection of research papers, textbooks [28, 29, 30, 31], and commercial circuit datasheets. We convert all documents from PDF to Markdown format using MinerU[32], enabling efficient extraction of embedded visual elements such as circuit schematics. For schematic-to-netlist translation, we utilize AMSnet[12], which allows us to accurately recover component-level connectivity and circuit topology. To enrich the dataset with semantic information, we use a combination of manual annotations from field experts and outputs from state-of-the-art MLLMs [13, 14]. We then apply carefully crafted prompt engineering and filter strategies to generate detailed schematic captions. This process yields high-quality pairs of <circuit schematic, caption>.

For textbook-derived data, we organize content according to the logical structure and chapter alignment of each source. For datasheet content, we extract structured performance specifications associated with each circuit. Based on the extracted information, we manually construct a question–answer dataset focused on circuit principles, behavior, and performance metrics, as illustrated in Fig. 3.



```
M1 (net7 net2 net6 VSS) nmos
M2 (net8 net1 net6 VSS) nmos
M4 (net7 net7 VDD VDD) pmos
M3 (net8 net7 VDD VDD) pmos
I1 (net6 VSS) isource
C1 (VSS net8) 1nF
```

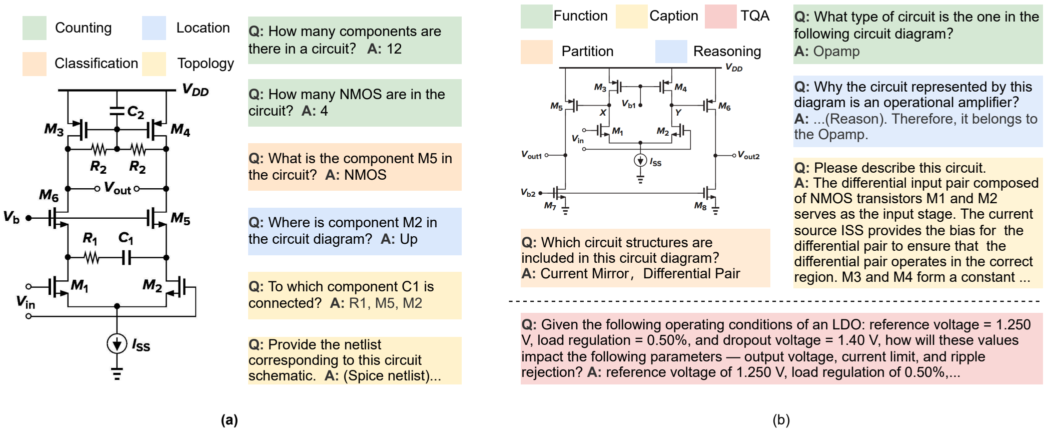
This is a simple operational amplifier circuit. The input voltage is applied to the gates of NMOS transistors M1 and M2 ...

In all analog electronics, an operational amplifier is the most versatile circuit block. It consists of a differential input and a single output. Its gain is very large. It is normally used in a feedback loop. A simple realization is shown in this slide. Such an operational amplifier, usually shortened to opamp...

Chapter 1	BJT and MOSFet
Chapter 2	Fully differential amplifier
...	Low dropout regulator
Chapter n	Low-noise Amplifier

Number of channels	1	Slew rate (V/μs)	0.5
Total supply voltage(V)	44	Vos (mV)	3
GBW (MHZ)	Value 8	CMRR (dB)	90
...	...	...	...

Figure 3: Data collection and curation for AMSBench



Q: How many components are there in a circuit?   **A:** 12

Q: How many NMOS are in the circuit? **A:** 4

Q: What is the component M5 in the circuit? **A:** NMOS

Q: Where is component M2 in the circuit diagram? **A:** Up

Q: To which component C1 is connected? **A:** R1, M5, M2

Q: Provide the netlist corresponding to this circuit schematic. **A:** (Spice netlist)...

Q: What type of circuit is the one in the following circuit diagram?
A: Opamp

Q: Why the circuit represented by this diagram is an operational amplifier?
A: ...(Reason). Therefore, it belongs to the Opamp.

Q: Which circuit structures are included in this circuit diagram?
A: Current Mirror，Differential Pair

Q: Please describe this circuit.
A: The differential input pair composed of NMOS transistors M1 and M2 serves as the input stage. The current source ISS provides the bias for the differential pair to ensure that the differential pair operates in the correct region. M3 and M4 form a constant ...

Q: Given the following operating conditions of an LDO: reference voltage = 1.250 V, load regulation = 0.50%, and dropout voltage = 1.40 V, how will these values impact the following parameters — output voltage, current limit, and ripple rejection? **A:** reference voltage of 1.250 V, load regulation of 0.50%,...

Figure 4: Example question generation for AMSbench

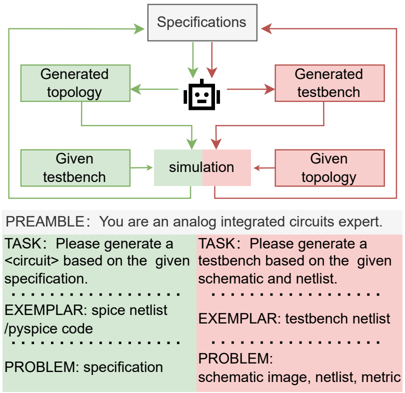
3.2 Evaluation

The goal of AMSbench is to thoroughly evaluate MLLMs on the potential applications and tasks in the AMS circuit domain, as shown in Fig. 1. For the design of specific problems, we develop a multi-dimensional evaluation framework that includes **perception**, **analysis**, and **design**. This framework addresses the potential uses of MLLMs in assisting users with interpreting and designing circuit schematics, both automatically and semi-automatically. Considering the complex data modalities and diverse tasks within the AMS circuit domain, our tasks encompass Visual Question Answering (VQA) and Textual Question Answering (TQA). These include multiple-choice questions, computational problems, and open-ended generative questions. We systematically construct questions for each task at multiple levels to accommodate various difficulties and circuit types.

Evaluation Dimensions For the **perception** tasks, we focus on recognizing *elements* in circuit schematics. We define an element as any component or device represented by a line in a netlist, such as transistors, resistors, subcircuit symbols, etc. MLLMs are evaluated on their accuracy of element counting, their precision in identifying the connectivity between elements, and their capability to recognize the entire circuit's netlist, as illustrated in Fig. 4(a). Accurate identification of elements, connectivity, and ports is fundamental to understanding and analyzing circuits. The complexity of element types and their connections in schematics makes this task particularly challenging, testing the MLLM's ability to perceive complex images more rigorously than traditional visual counting tasks.

For the **analysis** tasks, AMSbench primarily assesses the MLLMs' comprehension of circuit schematics. This includes the recognition and analysis of circuit functions, as well as the detection of functional building blocks within the circuits, as illustrated in Fig. 4(b). We also evaluate the LLMs' and MLLMs' understanding of the trade-offs between different circuit performances. Accurately analyzing a circuit and its corresponding performance metrics is crucial for ensuring the proper design of circuits, forming the basis for accurate circuit design.

For the **design** tasks, we consider both the design of circuits and testbenches, as shown in Fig. 5. Proper circuit design ensures that the functionality meets specifications, while the design of testbenches ensures that the circuit's performance can be accurately measured. These two tasks are central to the AI-driven automation of AMS circuit design. In setting up the circuit design task, we adopt and expand upon the benchmark defined by AnalogCoder [10].



PREAMBLE: You are an analog integrated circuits expert.

TASK: Please generate a <circuit> based on the given specification.

EXEMPLAR: spice netlist /pyspice code

PROBLEM: specification

TASK: Please generate a testbench based on the given schematic and netlist.

EXEMPLAR: testbench netlist

PROBLEM: schematic image, netlist, metric

Figure 5: The flowchart of design task

Difficulty Levels We classify the questions into three difficulty levels. Specifically, for the **perception** task, we categorize the difficulty based on the number of elements in the circuit: simple (num < 10), medium (10 < num < 20), and difficult (num > 20). For circuit functionality **analysis**, we classify the problems according to the circuit type and group them into two levels based on their appearance in educational stages: undergraduate and graduate levels. For testing the trade-offs between circuit performances, we assign a classification suitable for engineers. For the **design** task, we classify the circuits based on their complexity into three levels: simple, complex, and system-level circuits.

3.3 AMSBench Statistics

Fig. 6 illustrates the subtasks involved in the perception task along with the number of questions at varying difficulty levels. Fig. 7 presents statistical information for the analysis task and its various subtasks. The VQA tasks focus on evaluating the MLLM's ability to interpret circuit schematic images, while the TQA tasks assess the model's understanding of circuit knowledge and its awareness of performance trade-offs. Table 1 and Table 2 present an overview of the design tasks. For the circuit design section, we incorporated the benchmarks provided by AnalogCoder [10] and further extended them with additional circuit tasks, including system-level circuit design. The testbench design task fills a notable gap in the current community by introducing a previously underexplored category.

CKT_TYPE	NUM	CKT_TYPE	NUM	CKT_TYPE	NUM	CKT_TYPE	NUM
Amplifier	7 / 0 / 0	Oscillator	0 / 2 / 0	Subtractor	0 / 1 / 0	LDO	0 / 1 / 0
Inverter	2 / 0 / 0	Integrator	0 / 1 / 0	Schmitt trigger	0 / 1 / 0	Comparator	0 / 1 / 0
Current Mirror	2 / 0 / 0	Differentiator	0 / 1 / 0	VCO	0 / 1 / 0	Bandgap	0 / 1 / 0
Opamp	2 / 0 / 0	Adder	0 / 1 / 0	PLL	0 / 0 / 1	SAR-ADC	0 / 0 / 1

Table 1: Circuit Table with CKT_TYPE and NUM

ID	TestBench_TYPE	Num(Metrics)	ID	TestBench_TYPE	Num(Metrics)	ID	TestBench_TYPE	Num(Metrics)
1	OTA	7	5	MOS	1	9	LDO	7
2	Bootstrap	1	6	Telescope_Amplifier	7	10	Bandgap	4
3	Comparator	2	7	Folded Cascode Amp	5	11	Unit_capacitor	1
4	PLL	2	8	SAR-ADC	1	12	PLL-VCO	2

Table 2: Circuit Table with ID, TestBench_TYPE, and Number of Metrics

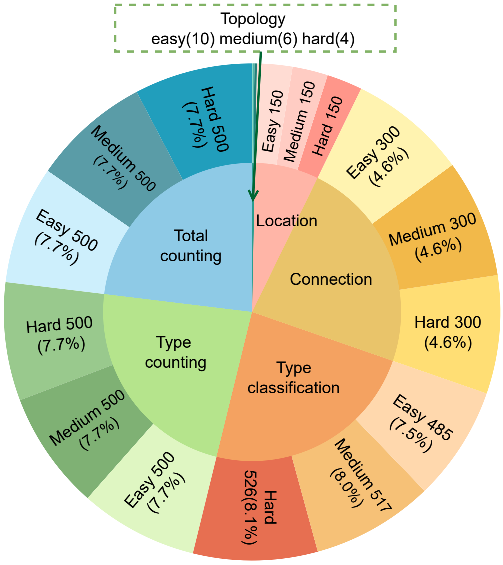


Figure 6: Data statistic of perception task

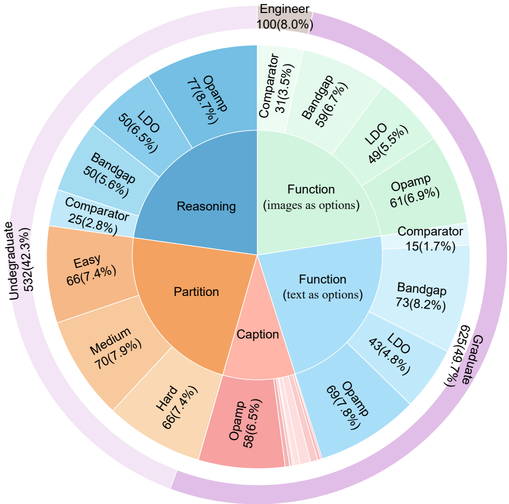


Figure 7: Data statistic of analysis task

4 Experiments

4.1 Models

We perform experiments on mainstream closed-source MLLMs: GPT4o [13], Grok-3 [33], Gemini-2.5-pro [34], Claude3.7 sonnet [35], Doubao-1.5-vision-pro-32k [36], and open-source models: Kimi-VL [37], Qwen2.5-VL 72B [14], DeepSeek-R1 [38]. We evaluate both TQA tasks on all models, and VQA tasks on all models except DeepSeek-R1. We use all open-source models with default parameters and deploy on up to 8 A100 GPUs.

4.2 Metrics

For single-choice questions, we adopt accuracy (ACC) as the evaluation metric and F1 score for multi-choice questions. For netlist recognition tasks, we define a Netlist Edit Distance (NED) as the evaluation metric, with the calculation procedure illustrated in the Fig. 8. The NED for each schematic image is normalized as follows:

$$\mathrm{NED}_{norm} = \frac{|\,\mathrm{GT} \cup \mathrm{Pred}\,| - |\,\mathrm{GT} \cap \mathrm{Pred}\,|}{|\,\mathrm{GT}\,|} \tag{1}$$

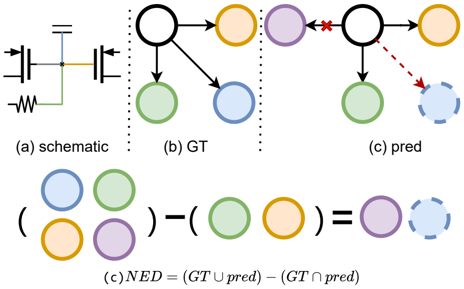


Figure 8: Edit distance computation between the GT netlist and the predicted netlist. The graph illustrates inter-device connections with each device abstracted as a node.

For evaluating the circuit design and testbench generation tasks, we use pass@k as the primary metric to measure the success rate of model-generated solutions.The pass@k metric is calculated as follows.For a given problem, the model generates k distinct answers. The pass@k value is determined by dividing the number of answers that pass the simulation check by k. For instance, if 5 answers are generated and 3 pass, then pass@5 = 3/5 = 0.6. Additionally, each task is evaluated through 5 repeated experiments, with the average of the pass@k values taken as the final result.

4.3 Main Results

perception Table 3 presents the models' performance on fundamental circuit schematic recognition tasks. Specifically, component counting and classification, both of which are essential for accurate netlist extraction. Gemini achieves the best overall results. However, due to the complexity and diversity of component types, the models show limitations in accurate counting. For component type classification, Gemini performs well, reaching 94% accuracy. Among open-source models, Qwen2.5-VL achieves 86%, suggesting that open-source models still have room for improvement in component type recognition.

Table 4 presents the accuracy of MLLMs in identifying inter-device connectivity. While the models can produce reasonably accurate predictions for local connections, they fall short in reconstructing the complete netlist. Even netlists produced by the best-performing model, Gemini 2.5 Pro, require substantial modifications to align with the ground truth. Closed-source models perform significantly better on this task. Some of the open-source models fails to produce outputs in the required format.

Models	**Total counting**		**Type counting**		**Device classification**	**Location judgement**
	ACC (↑)	**MSE** (↓)	**ACC** (↑)	**MSE** (↓)	**ACC** (↑)	**ACC** (↑)
Gemini 2.5 pro	**0.65**	**10.02**	**0.64**	**13.41**	**0.94**	**0.61**
GPT-4o	0.51	19.05	0.54	28.18	0.91	0.37
Claude-3-7-sonnet	0.36	18.38	0.55	24.18	0.83	0.48
Grok-3	0.22	60.71	0.50	26.48	0.84	0.50
Doubao-1.5-vision-pro	0.24	38.13	0.51	24.76	0.93	0.45
Kimi VL A3B	0.15	49.19	0.44	34.96	0.66	0.31
Qwen2.5 VL 72B	0.43	19.59	0.49	18.59	0.86	0.56

Table 3: Performance comparison of different models across various perception tasks

Models	**Connection judgement**	**Connection choicement**	**Topology**
	ACC (↑)	**F1-score** (↑)	**NED** (↓)
Gemini 2.5 pro	**0.85**	**0.88**	**0.91**
GPT-4o	0.73	0.65	1.40
Claude-3-7-sonnet	0.76	0.71	1.65
Grok-3	0.70	0.65	1.84
Doubao-1.5-vision-pro	0.76	0.64	1.57
Kimi VL A3B	0.53	0.53	–
Qwen2.5 VL 72B	0.77	0.52	2.38

Table 4: Comparison of models on connection judgement, choicement, and topology generation tasks

Analysis Table 5 evaluates the models' AMS circuit analysis capabilities, including both schematic interpretation and analysis of trade-offs in circuit performance. In schematic understanding, different MLLMs exhibit distinct strengths: Gemini demonstrates the highest accuracy in identifying and analyzing functional building blocks, while Grok-3 provides more accurate overall descriptions of circuit behavior. Table 16 shows that current models can achieve relatively high accuracy in analyzing circuit knowledge designed by undergraduate and graduate students. However, they perform poorly in understanding the trade-offs between circuit performance metrics commonly encountered in industry. Even the best-performing model, GPT-4o, only achieves 58% accuracy, indicating that LLMs currently lack a clear understanding of the expected performance characteristics of each circuit in the design process.

Design Table 6 shows the performance of the models on circuit design and testbench design tasks. For the former task, Grok-3 and Claude-Sonnet achieve the best results; however, for the latter task, none of the current models are able to directly generate syntactically correct testbench circuits, with the occasionally exception of GPT-4o. One possible reason is that the current pretraining data lacks sufficient testbench-related knowledge, and the metrics that need to be measured vary across different circuits, making testbench generation highly challenging.

Models	Reasoning	Partition	Caption	Function text	Function image	TQA
	ACC (↑)	F1 (↑)	ACC (↑)	ACC (↑)	ACC (↑)	ACC (↑)
Gemini 2.5 pro	**0.92**	**0.80**	0.70	**0.95**	**0.94**	0.72
GPT-4o	0.77	0.57	0.61	0.93	0.89	**0.78**
Claude-3-7-sonnet	0.91	0.64	**0.98**	0.88	0.74	0.74
Grok-3	0.61	0.59	0.41	0.77	0.22	0.74
Doubao-1.5-vision-pro	0.83	0.60	0.70	0.94	0.93	0.76
Kimi VL A3B	0.74	0.25	0.71	0.59	0.28	0.59
Qwen2.5 VL 72B	0.82	0.45	0.78	0.78	0.85	0.69

Table 5: Comparison of models on reasoning, partition identification, caption generation, circuit type prediction, and TQA tasks

Model	CKT design			TB design	
	Pass@3	Pass@5	Pass@10	Syntax@5	Metric@5
Gemini 2.5 pro	0.57	0.54	0.43	0	0
GPT-4o	0.47	0.49	0.42	**0.084**	0
Claude-3-7-sonnet	0.63	**0.64**	0.50	0	0
Grok-3	**0.65**	0.54	**0.61**	0	0
Doubao-1.5-vision-pro	0.45	0.24	0.15	0	0
Qwen2.5 VL 72B	0.47	0.41	0.33	0	0
Kimi VL A3B	0.41	0.25	0.13	0	0
DeepSeek-R1	0.55	0.51	0.45	-	-

Table 6: Pass rates comparison across models at different levels (Pass@3, Pass@5, Pass@10) and a Testbench design metric. *Syntax*: generated testbench is syntactically correct to run simulation. *Metric*: generated testbench is topologically and parametrically correct and produces the correct performance metric. Averages between all circuit types in Table 1, full results are available in the appendix in Tables 18-21.

5 Observation and Findings

Based on the models' performance across various tasks, we summarize and analyze the current challenges faced by Multimodal Large Language Models (MLLMs) in the field of AMS circuit from the dimensions of perception, analysis, and design.

Perceptual capabilities: Existing MLLMs remain incapable of accurately interpreting circuit schematics. While certain models demonstrate promising performance in capturing localized connectivity patterns, their effectiveness significantly deteriorates when tasked with comprehensive netlist extraction, as illustrated in Fig. 23. A primary challenge is that MLLMs are inaccurate at assigning connection points (i.e. pins and ports) to their parent components, resulting in various connectivity errors.

Analysis capabilities: Some MLLMs demonstrate reasonable levels of circuit analysis capability. They accurately interpret circuit functionalities, which indicates a comprehensive and precise understanding of circuit knowledge, as well as a degree of generalization in visual recognition when dealing with stylized images. Nevertheless, for the reasoning tasks, the models occasionally produce correct answers despite evident errors in their analytical process, which undermines our confidence in their correct answers. For instance, when examining an operational amplifier circuit diagram comprising over 15 components, the model may misidentify the current mirror structure formed by two transistors M1-M2, and incorrectly state the structure is formed by two other transistors M3-M4. This misidentification aligns with the model's observed performance in tasks involving component counting and device classification. Possible reasons include: 1. The model exhibits hallucination phenomena when recognizing circuit diagrams with a high number of components [39]. 2. Models generally prioritize capturing global information, often at the expense of local information, which is crucial for nuanced understanding [40].

In the circuit design process, particularly during the sizing and layout auto-generation stages, a model with strong circuit analysis abilities can significantly reduce the parameter search space and enable partitioning of the circuit into macros for efficient layout generation. However, the current model lacks the ability to accurately quantify the trade-offs between circuit performance metrics, which limits its capability to recommend the appropriate circuit topology when given a target specification.

System-level circuit design capabilities: Evaluation findings demonstrate that current models exhibit satisfactory performance in simple circuit design tasks but reveal constrained proficiency in comprehending complex and system-level circuits, such as the Successive Approximation Register Analog-to-Digital Converter (SAR-ADC). For system-level circuits, the preponderance of model-generated designs failed to satisfy the netlist rule check, which validates the accuracy of component interconnections, and none fulfilled the stipulated circuit performance criteria. A comprehensive analysis of the SAR-ADC design produced by the GPT-4o model, which demonstrates comparatively robust performance across diverse tasks, is presented in figure 29 in the appendix. The circuit schematic indicates that GPT-4o accurately implements the differential input structure. Nevertheless, owing to an inadequate grasp of the operational principles underlying the SAR-ADC, the model was unable to extend the design beyond the differential input pair, resulting in an extremely incomplete circuit.

6 Conclusion and Future Work

This paper introduces AMSbench, a benchmark for evaluating the capabilities of Multimodal Large Language Models (MLLMs) in AMS circuit design. The benchmark assesses model performance across three key dimensions: schematic perception, circuit analysis, and circuit design, encompassing a variety of tasks.

The evaluation highlights significant limitations in current models, particularly in schematic perception and complex circuit design. While some models excel in basic component recognition and circuit analysis tasks, they struggle with more advanced tasks, such as system-level circuit design and accurate schematic interpretation. Notably, the models failed to generate correct testbenches for complex circuits, pointing to challenges in interpreting netlists and partitioning tasks.

Future research will prioritize the expansion of datasets to enhance the robustness and generalizability of multimodal models. It will investigate advanced methodologies, such as RAG and RLHF, to augment design capabilities. Additionally, efforts will be made to incorporate grounding modules or enriched datasets to improve performance on perception tasks, thereby advancing the model's analytical and design proficiencies. Furthermore, integrating topology generation, sizing, and floorplanning into the design process is planned to enable fully automated, end-to-end circuit design.

References

[1] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao, *et al.*, "A survey on multimodal large language models for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 958–979, 2024.

[2] Y. Hao, J. Gu, H. W. Wang, L. Li, Z. Yang, L. Wang, and Y. Cheng, "Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark," *arXiv preprint arXiv:2501.05444*, 2025.

[3] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, *et al.*, "Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

[4] R. Zhang, D. Jiang, Y. Zhang, H. Lin, Z. Guo, P. Qiu, A. Zhou, P. Lu, K.-W. Chang, Y. Qiao, *et al.*, "Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?," in *European Conference on Computer Vision*, pp. 169–186, Springer, 2024.

[5] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models," *CoRR*, 2023.

[6] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, *et al.*, "Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement," *arXiv preprint arXiv:2409.12122*, 2024.

[7] L. Zhong and Z. Wang, "Can llm replace stack overflow? a study on robustness and reliability of large language model code generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 21841–21849, 2024.

[8] J. Bhandari, V. Bhat, Y. He, H. Rahmani, S. Garg, and R. Karri, "Masala-chai: A large-scale spice netlist dataset for analog circuits by harnessing ai," 2025.

[9] J. Gao, W. Cao, J. Yang, and X. Zhang, "Analoggenie: A generative engine for automatic discovery of analog circuit topologies," *arXiv preprint arXiv:2503.00205*, 2025.

[10] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 379–387, 2025.

[11] 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)*, pp. 1–6, 2024.

[12] Z. Tao, Y. Shi, Y. Huo, R. Ye, Z. Li, L. Huang, C. Wu, N. Bai, Z. Yu, T.-J. Lin, *et al.*, "Amsnet: Netlist dataset for ams circuits," in *2024 IEEE LLM Aided Design Workshop (LAD)*, pp. 1–5, IEEE, 2024.

[13] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[14] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, *et al.*, "Qwen2. 5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[15] C. Liu, W. Chen, A. Peng, Y. Du, L. Du, and J. Yang, "Ampagent: An llm-based multi-agent system for multi-stage amplifier schematic design from literature for process and performance porting," *arXiv preprint arXiv:2409.14739*, 2024.

[16] J. Chaudhuri, D. Thapar, A. Chaudhuri, F. Firouzi, and K. Chakrabarty, "Spiced+: Syntactical bug pattern identification and correction of trojans in a/ms circuits using llm-enhanced detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.

[17] Z. Yan, Y. Qin, X. S. Hu, and Y. Shi, "On the viability of using llms for sw/hw co-design: An example in designing cim dnn accelerators," in *2023 IEEE 36th International System-on-Chip Conference (SOCC)*, pp. 1–6, IEEE, 2023.

[18] J. Blocklove, S. Garg, R. Karri, and H. Pearce, "Chip-chat: Challenges and opportunities in conversational hardware design," in *2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD)*, pp. 1–6, IEEE, 2023.

[19] Y. Fu, Y. Zhang, Z. Yu, S. Li, Z. Ye, C. Li, C. Wan, and Y. C. Lin, "Gpt4aigchip: Towards next-generation ai accelerator design automation via large language models," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, IEEE, 2023.

[20] J. Pan, G. Zhou, C.-C. Chang, I. Jacobson, J. Hu, and Y. Chen, "A survey of research in large language models for electronic design automation," *ACM Transactions on Design Automation of Electronic Systems*, 2025.

[21] W. Fang, J. Wang, Y. Lu, S. Liu, Y. Wu, Y. Ma, and Z. Xie, "A survey of circuit foundation model: Foundation ai models for vlsi circuit design and eda," *arXiv preprint arXiv:2504.03711*, 2025.

[22] 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. 27730–27744, 2022.

[23] Y. Yin, Y. Wang, B. Xu, and P. Li, "ADO-LLM: Analog Design Bayesian Optimization with In-Context Learning of Large Language Models," *arXiv preprint arXiv:2406.18770*, 2024.

[24] Y. Shi, Z. Tao, Y. Gao, T. Zhou, C. Chang, Y. Wang, B. Chen, G. Zhang, A. Liu, Z. Yu, *et al.*, "Amsnet-kg: A netlist dataset for llm-based ams circuit auto-design using knowledge graph rag," *arXiv preprint arXiv:2411.13560*, 2024.

[25] M. Liu, N. Pinckney, B. Khailany, and H. Ren, "VerilogEval: evaluating large language models for verilog code generation," in *2023 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023.

[26] Y. Lu, S. Liu, Q. Zhang, and Z. Xie, "Rtllm: An open-source benchmark for design rtl generation with large language model," in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 722–727, IEEE, 2024.

[27] L. Skelic, Y. Xu, M. Cox, W. Lu, T. Yu, and R. Han, "Circuit: A benchmark for circuit interpretation and reasoning capabilities of llms," *arXiv preprint arXiv:2502.07980*, 2025.

[28] B. Razavi, *Design of analog CMOS integrated circuits*. McGraw-Hill Higher Education, 2005.

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

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

[31] W. M. Sansen, *Analog design essentials*, vol. 859. Springer Science & Business Media, 2007.

[32] B. Wang, C. Xu, X. Zhao, L. Ouyang, F. Wu, Z. Zhao, R. Xu, K. Liu, Y. Qu, F. Shang, B. Zhang, L. Wei, Z. Sui, W. Li, B. Shi, Y. Qiao, D. Lin, and C. He, "Mineru: An open-source solution for precise document content extraction," 2024.

[33] ""grok 3 beta — the age of reasoning agents"." `https://x.ai/blog/grok-3`, 2025. Accessed: 2025-05-14.

[34] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[35] Anthropic, "Introducing claude 3.7 sonnet." `https://www.anthropic.com/news/`, 2024. Accessed: 2025-05-14.

[36] D. e. a. Guo, "Seed1.5-vl technical report," 2025.

[37] K. Team, A. Du, B. Yin, B. Xing, B. Qu, B. Wang, C. Chen, C. Zhang, C. Du, C. Wei, *et al.*, "Kimi-vl technical report," *arXiv preprint arXiv:2504.07491*, 2025.

[38] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[39] Y. Zhou, C. Cui, J. Yoon, L. Zhang, Z. Deng, C. Finn, M. Bansal, and H. Yao, "Analyzing and mitigating object hallucination in large vision-language models," *arXiv preprint arXiv:2310.00754*, 2023.

[40] Z. Li, Q. Xu, D. Zhang, H. Song, Y. Cai, Q. Qi, R. Zhou, J. Pan, Z. Li, V. Tu, *et al.*, "Groundinggpt: Language enhanced multi-modal grounding model," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6657–6678, 2024.