ChatLS: Multimodal Retrieval-Augmented Generation and Chain-of-Thought for Logic Synthesis Script Customization

Haisheng Zheng ♠ Haoyuan Wu ♡ Zhuolun He ♡♣

♠ Shanghai Artificial Intelligence Laboratory
♡ The Chinese University of Hong Kong
♣ ChatEDA Technology

Jun. 25, 2025

SPONSORED BY CEDA sigda

Outline

1. Introduction
2. Algorithm
3. Evaluation

Introduction

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda ACM

Background

Recent advancements in Large Language Models (LLMs) have been leveraged to enhance EDA tool usability:

- Generating EDA Tool Scripts [1].
- Query-Answering for EDA Tool Documentation [2].

[1] **Haoyuan Wu et al. (2024). "ChatEDA: A Large Language Model Powered Autonomous Agent for EDA".** In: *IEEE TCAD*.

[2] **Yuan Pu et al. (2024). "Customized Retrieval Augmented Generation and Benchmarking for EDA Tool Documentation QA".** In: *Proc. ICCAD*.

Take a Further Step

Commercial logic synthesis tools provide numerous optional commands to improve netlist quality; these should be selected based on [3][4]:

- Circuit Design Characteristics.
- User Preferences.

[3] **Matthew M Ziegler et al. (2016). "A Synthesis-Parameter Tuning System for Autonomous Design-Space Exploration". In: *Proc. DATE*.**

[4] **Himanshu Bhatnagar (2002). *Advanced ASIC Chip Synthesis: Using Synopsys® Design Compiler™ and PrimeTime®*.**

Challenges: Comprehensive Analysis of Design

- Recent research [5] demonstrates the capability of LLMs to understand RTL code snippets, potentially aiding in analyzing circuit design for script customization.
- LLMs have limitations with long-context inputs, affecting their effectiveness in analyzing large codebases [6] and executing lengthy reasoning sequences [7].

---

[5] **Xufeng Yao et al. (2024). "RTLRewriter: Methodologies for Large Models aided RTL Code Optimization". In: *Proc. ICCAD*.**

[6] **Carlos E Jimenez et al. (2024). "SWE-bench: Can Language Models Resolve Real-world Github Issues?" In: *arXiv preprint*.**

[7] **Nelson F Liu et al. (2024). "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics*.**

Challenges: Selecting Suitable Synthesis Commands

Consider a scenario where a design requires timing optimization:

- Retiming [8] reduces delays in designs with extended critical paths and unbalanced register placement.
- Buffer balancing addresses timing issues in high-fanout nets.

[8] **Charles E Leiserson and James B Saxe (1981). "Optimizing Synchronous Systems". In: *Proc. FOCS*.**

Problem Formulation

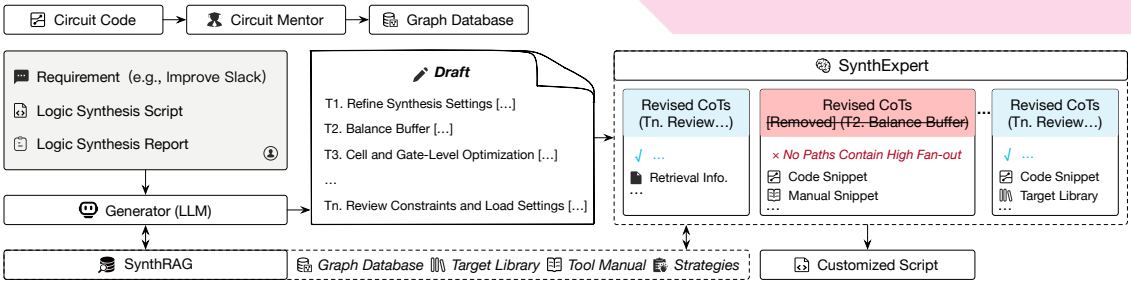
Logic Synthesis Script Customization

Design methodologies to enhance the capability of LLMs in customizing logic synthesis scripts to meet user requirements.

Algorithm

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda ACM

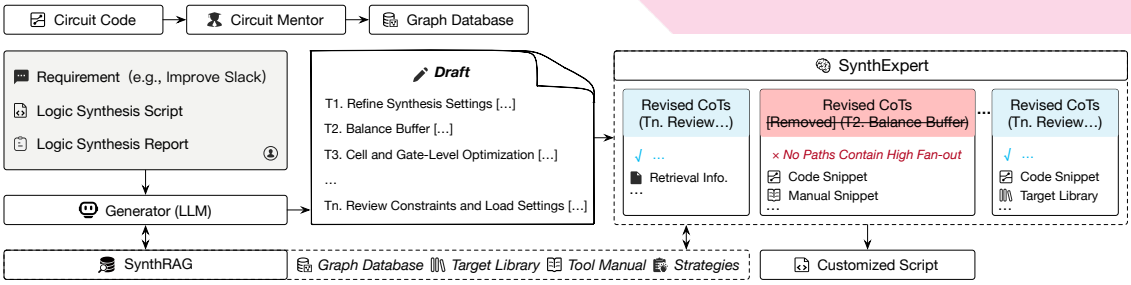
Overview



The Overall Flow of ChatLS.

- **CircuitMentor** is a circuit design analysis tool that transforms the circuit design into a graph representation and employs graph neural network (GNN) to extract high-level features from the circuit design.

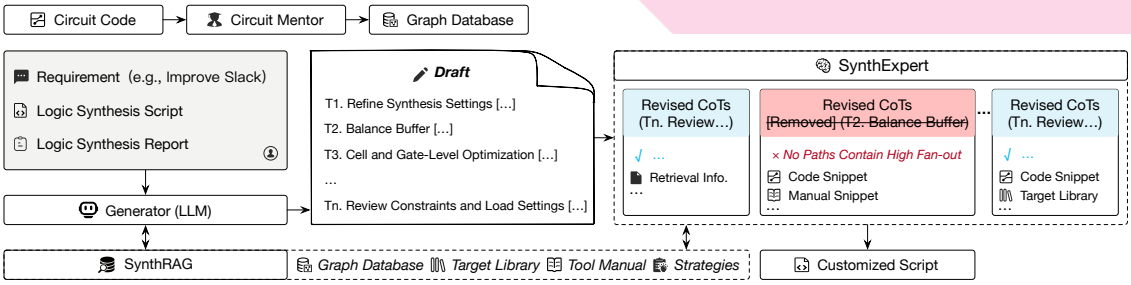
Overview



The Overall Flow of ChatLS.

- **SynthRAG** is a multi-modal retrieval-augmented generation (RAG) framework used to retrieve relevant information for LLMs when customizing logic synthesis scripts.

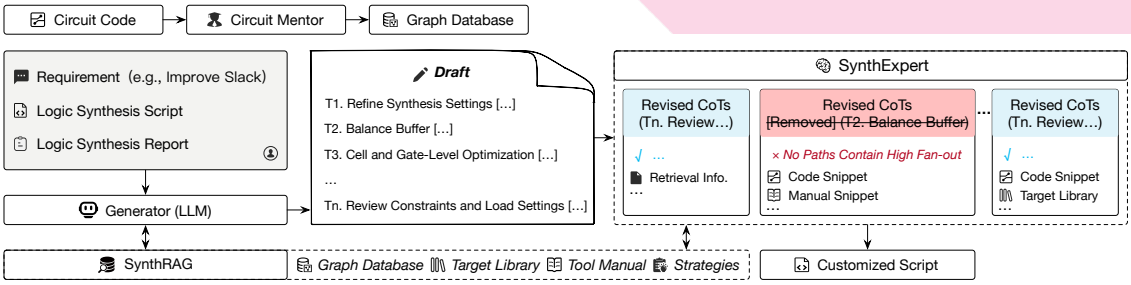
Overview



The Overall Flow of ChatLS.

- **Generator (LLM)** drafts a customized logic synthesis script based on user requirements, integrating synthesis tool reports, relevant data from **SynthRAG**, and high-level information matched using embeddings from **CircuitMentor**.

Overview



The Overall Flow of ChatLS.

- **SynthExpert** is a Chain-of-Thought (CoT) reasoning mechanism that uses **SynthRAG** to fetch step-relevant information and iteratively evaluates the suitability of the drafted logic synthesis script for the circuit design.

Circuit Mentor: Motivation

- LLMs encounter significant challenges in comprehending complex circuit code.
- Numerous studies [9][10][11] to leverage graph neural networks (GNNs) to extract meaningful characteristics from complex circuits.

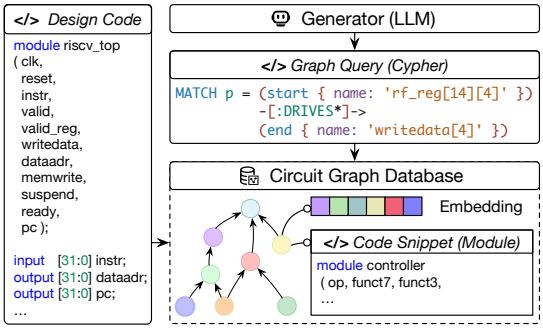
---

[9] **Haoyuan Wu et al. (2025). "Circuit Representation Learning with Masked Gate Modeling and Verilog-AIG Alignment".** In: *Proc. ICLR*.

[10] **Zhengyuan Shi et al. (2024). "DeepGate3: Towards Scalable Circuit Representation Learning".** In: *Proc. ICCAD*.

[11] **Ziyi Wang et al. (2024). "FGNN2: A Powerful Pre-Training Framework for Learning the Logic Functionality of Circuits".** In: *IEEE TCAD*.

Circuit Mentor: Graph-Based Assistant for Circuit Analysis



Workflow Visualization of CircuitMentor.

- **Graph Representation**: Transforming the circuit code into a graph and stored it in the Neo4j database.
- **Global Circuit Feature Extraction**: Hierarchical GNN based on GraphSAGE [12].
- **Local Circuit Feature Extraction**: LLMs could generate query requests using Cypher to extract code snippets for analysis.

[12] **Will Hamilton, Zhitao Ying, and Jure Leskovec (2017). "Inductive Representation Learning On Large Graphs".** In: *Proc. NIPS*.

Circuit Mentor: Metric Learning

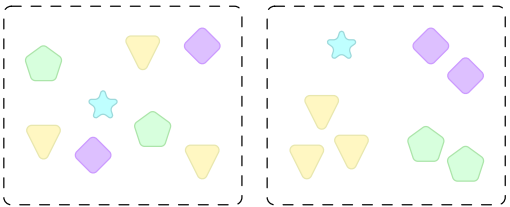


Illustration of the Metric Learning Process.

- **Motivation**:
  GNN is a topology-based approach, which may result in failing to retrieve the same type of components.
- **Metric Learning**:
  Similar designs are pulled closer together, while dissimilar ones are pushed farther apart.

SynthRAG: Multimodal RAG for Domain-Specific Retrieval

Category	Representation	Query Method	Retrieval Content
High Level Info. of Circuit Design	Graph Embedding	Join	Compile Strategy Optimization Strategy
Code of Circuit Design	Graph Structure	Direct	The code of the module where the path is located
Target Library	Graph Structure	Direct	Gate Cell Information
Logic Synthesis Tool User Manual	LLM Embedding	Direct	Command Usage Command Requirement

Table: Summary of Query Methods.

SynthRAG: Graph Embedding-Based Retrieval

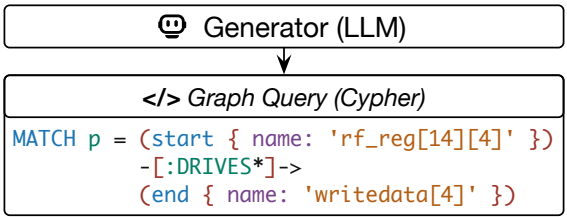
- Circuit Embeddings Similarity-Based Retrieval:

$$z_{Retrieved} = \arg\min_{z_{db}} \left(\text{sim}(z_{query}, z_{db})\right) \tag{1}$$

- Circuit Performance-Based Reranking:

$$\text{Score}(z_i) = \alpha \cdot \text{sim}(z_{query}, z_i) + \beta \cdot c_i \tag{2}$$

SynthRAG: Graph Structure-Based Retrieval



An Example of Cypher Query.

- Cypher Queries[13] are used to retrieve information from circuit graphs.

[13] **Nadime Francis et al. (2018). "Cypher: An Evolving Query Language for Property Graphs".** In: *Proc. SIGMOD*.

SynthRAG: LLM Embedding-Based Retrieval

- **Embedding Generation:**
  The `text-embedding-3-large` model [14] is used to convert the user manual into text embeddings.
- **Retrieval Reranking:**
  `GPT-4o` [15] is employed as a reranker to improve retrieval accuracy.

[14] **https://platform.openai.com/docs/guides/embeddings**
[15] **Josh Achiam et al. (2023). "GPT-4 Technical Report".** In: *arXiv preprint*.

SynthExpert: Motivation

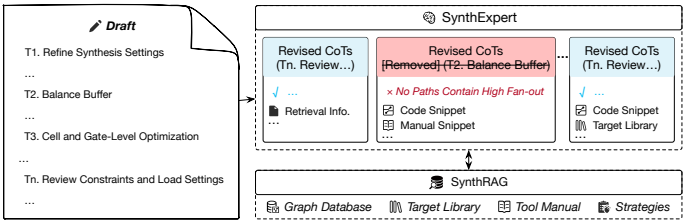
LLMs can be effectively assisted in complex decision-making tasks with:

- Retrieval-Augmented Generation (RAG) [16].
- Chain-of-Thought (CoT) [17] reasoning.

[16] **Patrick Lewis et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Proc. NIPS*.**

[17] **Jason Wei et al. (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Proc. NIPS*.**

SynthExpert: Iterative Customization with CoT and RAG



Workflow Visualization of SynthExpert.

- LLM initially generates a series of thought steps.
- SynthExpert employs SynthRAG to retrieve information for step revision, ensuring command suitability in customized scripts:

$$T^{\star}_{1:i} = p_{\theta}(\cdot|I, T^{\star}_{1:i-1}, T_i, R_i) \quad (3)$$

Evaluation

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda ACM

Evaluation Settings

- **Logic Synthesis Tool**: Design Compiler.
- **Circuit Designs**: Part of the OpenROAD [18] example.
- **Technology Library**: `Nangate 45nm` with the `5K_heavy_1k` wireload model.
- **Synthesis Script**: Configured according to the OpenROAD example.
- **Baselines**: `GPT-4o` [19] `(Version 2024-08-06),Claude 3.5-Sonet` [20] `(Version 2024-10-22).`
- **Resynthesis Iterations**: One.

[18] **T Ajayi et al. (2019). "OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain".** In: *Proc. GOMACTECH*.

[19] **Josh Achiam et al. (2023). "GPT-4 Technical Report".** In: *arXiv preprint*.

[20] **Anthropic PBC (2024).** ***Introducing Claude 3.5 Sonnet*.**

Evaluation Results

Design	Baseline				GPT-4o				Claude 3.5 Sonet				ChatLS (Ours)			
	Timing (ns)			Area	Timing (ns)			Area	Timing (ns)			Area	Timing (ns)			Area
	WNS	CPS	TNS	(μm^2)	WNS	CPS	TNS	(μm^2)	WNS	CPS	TNS	(μm^2)	WNS	CPS	TNS	(μm^2)
aes	-0.16	-0.16	-31.64	16577.12	-0.17	-0.17	-31.64	16408.21	-0.17	-0.17	-30.70	16126.78	0.00	0.00	0.00	15919.04
dynamic_node	-0.08	-0.08	-0.45	21155.51	0.00	0.01	0.00	16327.08	0.00	4.06	0.00	20048.95	0.00	4.85	0.00	18907.28
ethmac	-0.54	-0.54	-76.55	80533.36	-0.55	-0.55	-75.89	80502.77	-0.54	-0.54	-128.99	81993.70	-0.47	-0.47	-55.72	80349.56
jpeg	-1.17	-1.17	-439.66	107612.16	0.00	0.00	0.00	57227.24	0.00	0.00	0.00	66106.85	0.00	0.00	0.00	67290.02
risv32i	0.00	0.59	0.00	10241.00	0.00	0.59	0.00	10241.00	0.00	0.60	0.00	9711.66	0.00	0.74	0.00	9329.95
swerve	0.00	0.81	0.00	161551.64	0.00	0.79	0.00	143557.54	0.00	1.86	0.00	158364.70	0.00	2.05	0.00	143545.30
tinyRocket	-0.88	-0.88	-1057.89	44231.28	-0.21	-0.21	-42.22	35960.81	-0.33	-0.33	-484.71	41999.01	-0.09	-0.09	-14.74	36070.66

Table: Performance Comparison for Logic Synthesis Script Customization *(Pass@5)*.

Conclusion

Conclusion

In this paper, we proposed:

- A **Graph-Based Method** that enhances the ability of LLMs to analyze and understand circuit designs, with GNNs supporting the analysis process.
- A **Domain-Specific Multimodal RAG Framework** for efficient information retrieval to improve LLM-driven script customization.
- A **Task-Specific CoT Mechanism** integrated with the multimodal RAG framework, enabling iterative refinement of synthesis scripts through reasoning revision.

THANK YOU!