# Dr. RTL: Autonomous Agentic RTL Optimization through Tool-Grounded Self-Improvement

Speaker: Zhiyao Xie, Assistant Professor, HKUST
Email: eezhiyao@ust.hk

***Abstract*—Recent advances in large language models (LLMs) have sparked growing interest in automatic RTL optimization for better performance, power, and area (PPA). However, existing methods are still far from realistic RTL optimization: their evaluation is often unrealistic, based on manually degraded, small-scale RTL designs and weak open-source toolchains; their methods are also limited, relying on coarse design-level feedback and simple pre-defined rewriting rules. To address these limitations, we present Dr. RTL, an agentic framework for RTL timing optimization in a realistic evaluation environment, with continual self-improvement through reusable optimization skills. We establish a realistic evaluation setting with more challenging RTL designs and an industrial EDA workflow. Within this setting, Dr. RTL performs closed-loop optimization through a multi-agent framework for critical-path analysis, parallel RTL rewriting, and tool-based evaluation. We further introduce group-relative skill learning, which compares parallel RTL rewrites and distills the optimization experience into an interpretable skill library. Currently, this library contains 47 pattern–strategy entries for cross-design reuse to improve PPA and accelerate convergence, and it can continue evolving over time. Evaluated on 20 real-world RTL designs, Dr. RTL achieves average WNS/TNS improvements of 21%/17% with a 6% area reduction over the industry-leading commercial synthesis tool. The original full paper is available at: https://arxiv.org/abs/2604.14989.**

## I. Introduction

Register-transfer level (RTL) design is a critical stage in modern digital IC development, as RTL structure directly shapes the downstream synthesis and implementation search space. Although commercial synthesis tools perform extensive Boolean rewriting, technology mapping, and implementation-level optimization, they are still constrained by the original RTL representation. Timing bottlenecks introduced at the RTL stage can therefore survive synthesis and require manual RTL rewriting guided by timing reports. This process is labor-intensive, time-consuming, and heavily dependent on implicit design expertise.

Recent large language models (LLMs) have shown promise in RTL generation and code transformation [2]–[4], [6], [8], [12]. However, existing LLM-based RTL optimization studies [1], [5], [7], [9]–[11] remain limited in both evaluation realism and optimization methodology. Many prior works evaluate on small or manually degraded RTL designs, rely on weak open-source toolchains, and use coarse design-level PPA feedback or predefined rewriting rules. As a result, it remains unclear whether current AI methods can optimize realistic human-written RTL beyond what commercial synthesis tools already achieve.

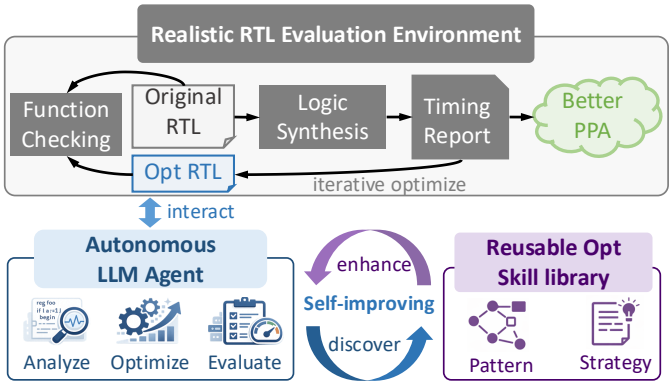



Fig. 1. Dr. RTL iteratively optimizes RTL PPA via closed-loop interaction with industrial EDA tools, while distilling trajectories into reusable skills for self-improvement.

In this work, we present **Dr. RTL**[1], an autonomous agentic framework for realistic RTL timing optimization through tool-grounded self-improvement. As shown in Fig. 1, Dr. RTL formulates RTL optimization as a closed-loop interaction between LLM agents and industrial EDA tools. The agent analyzes fine-grained timing feedback, proposes functionally equivalent RTL transformations, evaluates candidates through synthesis and sequential equivalence checking, and iteratively promotes the best verified candidate.

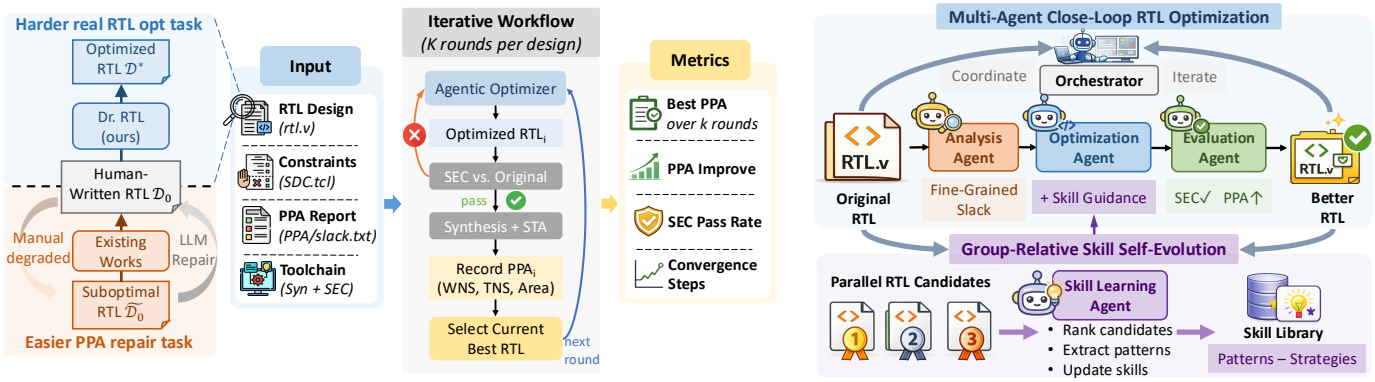
This work makes three main contributions. First, we establish a realistic RTL optimization benchmark and evaluation flow using human-written RTL designs, commercial synthesis, and sequential equivalence checking. Second, we design a tool-grounded multi-agent optimization framework that separates critical-path analysis, RTL transformation, and candidate evaluation. Third, we introduce group-relative skill learning, which distills reusable optimization knowledge from parallel candidate trajectories and enables continual self-improvement without LLM fine-tuning.

## II. Methodology

Dr. RTL consists of three tightly coupled components, as summarized in Fig. 2: a realistic industrial evaluation environment, a tool-grounded multi-agent optimization loop, and a group-relative skill learning mechanism.

**Realistic RTL optimization environment.** Dr. RTL evaluates RTL optimization using human-written Verilog designs, commercial synthesis, and sequential equivalence checking. Unlike manually degraded benchmarks, this setting directly

[1]The original full paper is available at: https://arxiv.org/abs/2604.14989.




(a) Evaluating agentic RTL timing optimization

(b) Overview of Dr. RTL agentic framework

Fig. 2. Proposed industrial-standard RTL optimization evaluation and overview of Dr. RTL for agentic RTL optimization.

TABLE I
COMPARISON WITH EXISTING SINGLE-SHOT LLMS AND LLM-BASED RTL OPTIMIZATION METHODS.

| Method | | WNS | TNS | Area |
|---|---|---|---|---|
| Single-Shot LLM | Claude Opus | -2.4% | -2.8% | -0.7% |
| | GPT 5.3 | -1.2% | 0.3% | 0.6% |
| SOTA Baselines* | RTLRewriter [11] | -4.9% | -6.3% | -3.1% |
| | SymRTLo [9] | -7.1% | -5.7% | -1.4% |
| **Ours** | **Dr. RTL** | **-21.3%** | **-16.9%** | **-5.8%** |

tests whether an agent can improve already strong RTL beyond standard synthesis optimization. The evaluation loop reports timing, area, and correctness information, including worst negative slack (WNS), total negative slack (TNS), area, and sequential equivalence checking (SEC) status.

**Tool-grounded multi-agent optimization.** Dr. RTL uses a multi-agent workflow to separate analysis, transformation, and evaluation. A timing analysis agent interprets EDA feedback, especially critical-path and register-level slack information. An RTL optimization agent proposes multiple candidate rewrites in parallel based on the current timing bottleneck and previous optimization experience. An evaluation agent invokes commercial synthesis and SEC tools to measure design quality and verify correctness. Only candidates that pass equivalence checking are considered valid, and the best candidate is promoted to the next iteration.

**Group-relative skill learning.** Dr. RTL introduces group-relative skill learning to extract reusable knowledge from optimization trajectories. Instead of judging each RTL transformation only by its absolute PPA result, the framework compares multiple candidates generated from the same parent RTL under the same timing context. This relative comparison provides cleaner feedback on which transformation strategies are effective. Successful and failed optimization attempts are summarized into reusable pattern–strategy knowledge, including timing bottleneck patterns, effective transformations, and strategies to avoid. This skill library is reused in later optimization rounds to guide future candidate generation.

## III. EXPERIMENT

We evaluate Dr. RTL on 20 realistic human-written RTL designs covering buffers, processors, signal-processing modules, cryptographic circuits, and SoC-like systems. The designs are optimized under a commercial synthesis flow and verified using sequential equivalence checking. This setting is intended to reflect practical RTL optimization, where the goal is to improve timing while preserving functionality.

Across the 20 designs, Dr. RTL improves timing on all designs. It achieves average WNS and TNS improvements of 21.3% and 16.9%, respectively, while reducing area by 5.8%. Nine designs achieve Pareto improvements in both timing and area, eight designs incur only minor area overhead below 5%, and only three designs show noticeable area increase. The framework also maintains an average SEC pass rate of 86%, showing that the agent can explore non-trivial RTL transformations while preserving functional correctness.

We further compare Dr. RTL against representative LLM-based RTL optimization baselines, including single-shot rewriting and prior iterative optimization methods using the same backbone model. Dr. RTL achieves better WNS, TNS, and area results, indicating that the improvement comes primarily from the tool-grounded agentic workflow, fine-grained EDA feedback, and reusable skill learning rather than from the base LLM alone.

## IV. CONCLUSION

This work presents Dr. RTL, an autonomous agentic framework for realistic RTL timing optimization. By combining LLM-based reasoning with commercial EDA feedback, parallel candidate exploration, sequential equivalence checking, and group-relative skill learning, Dr. RTL moves beyond one-shot RTL generation toward tool-grounded agentic design automation. The results demonstrate that AI agents can improve human-written RTL designs beyond standard synthesis optimization while maintaining functional correctness. More broadly, Dr. RTL suggests a future direction in which AI agents interact with real EDA workflows, accumulate reusable design knowledge, and assist designers in exploring large RTL optimization spaces more effectively.

## REFERENCES


[1] C.-M. Chang, P. Vijayaraghavan, A. Jadhav, C. Mackin, H. Tsai, V. Mukherjee, and E. Degan, "Codmas: A dialectic multi-agent collaborative framework for structured rtl optimization," in *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, 2026, pp. 777–788.

[2] L. Chen *et al.*, "The dawn of ai-native eda: Promises and challenges of large circuit models," *Springer Science China Information Sciences (SCIS)*, 2024.

[3] 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.

[4] Z. He, Y. Pu, H. Wu, T. Qiu, and B. Yu, "Large language models for eda: Future or mirage?" *ACM Transactions on Design Automation of Electronic Systems*, vol. 30, no. 6, pp. 1–53, 2025.

[5] Y. Lu, S. Liu, H. Zhou, W. Fang, Q. Zhang, and Z. Xie, "A new benchmark for the appropriate evaluation of rtl code optimization," *arXiv preprint arXiv:2601.01765*, 2026.

[6] 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 (TODAES)*, 2025.

[7] H. Ping, P. Zhang, Z. Wang, S. Li, A. Cheng, W. Yang, P. Bogdan, and S. Nazarian, "Poet: Power-oriented evolutionary tuning for llm-based rtl ppa optimization," *arXiv preprint arXiv:2603.19333*, 2026.

[8] A. Ravindran, A. Patra, V. Babaey, and S. Purini, "Survey and benchmarking of large language models for rtl code generation: Techniques and open challenges," 2025.

[9] Y. Wang, W. Ye, P. Guo, Y. He, Z. Wang, B. Tian, S. He, G. Sun, Z. Shen, S. Chen *et al.*, "Symrtlo: Enhancing rtl code optimization with llms and neuron-inspired symbolic reasoning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[10] Z. Xu, B. Li, and L. Wang, "Rethinking llm-based rtl code optimization via timing logic metamorphosis," *arXiv preprint arXiv:2507.16808*, 2025.

[11] X. Yao, Y. Wang, X. Li, Y. Lian, R. Chen, L. Chen, M. Yuan, H. Xu, and B. Yu, "Rtlrewriter: Methodologies for large models aided rtl code optimization," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024.

[12] Z. Zang, Y. Song, A. Wang, B. W.-K. Ling, Q. Sun, Z. Lei, F. Yang, C. Zhuo, and J. Luo, "The dawn of agentic eda: A survey of autonomous digital chip design," *arXiv preprint arXiv:2512.23189*, 2025.