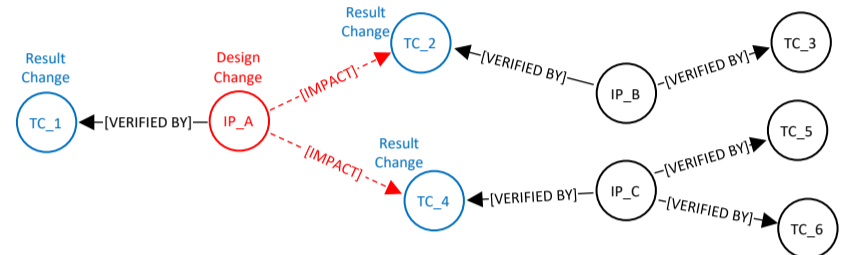


INTRODUCTION

- Modern SoCs integrate hundreds of blocks and IPs, significantly increasing verification complexity.
- Design changes in one IP often propagate test result variations to **indirectly related IPs** through functional and execution-level dependencies.
- Conventional verification focuses on **directly modified IPs and static hierarchies**, failing to capture such impact propagation.
- This results in **low targeting accuracy, redundant tests, and inefficient debug cycles**, motivating an adaptive verification framework.

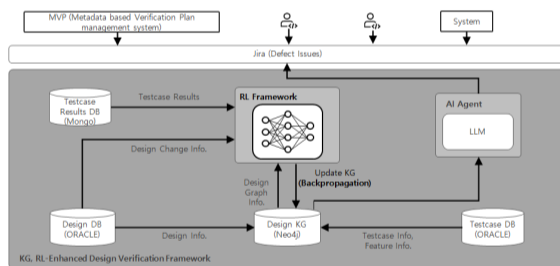
OBJECTIVES

- Build a **Knowledge Graph (KG)** that captures semantic and execution-level dependencies among Block, IP, and Testcase.
- Apply **Reinforcement Learning (PPO)** to dynamically weight impact paths and prioritize high-risk verification tasks.
- Enable **backpropagation-based debugging** to trace test result changes to their root-cause IPs.



PROPOSED FRAMEWORK

- We propose a **KG + RL framework** that learns execution-level impact propagation and prioritizes verification tasks.
- IPs are categorized by dependency scope: **Type 1** (intra-block dominant) and **Type 2** (cross-block dependent).
*e.g., abox IP of Audio block (Type 1), bus IP of USB block (Type 2)
- Reinforcement Learning (PPO) weights impact paths, enabling backpropagation-based root-cause tracing.

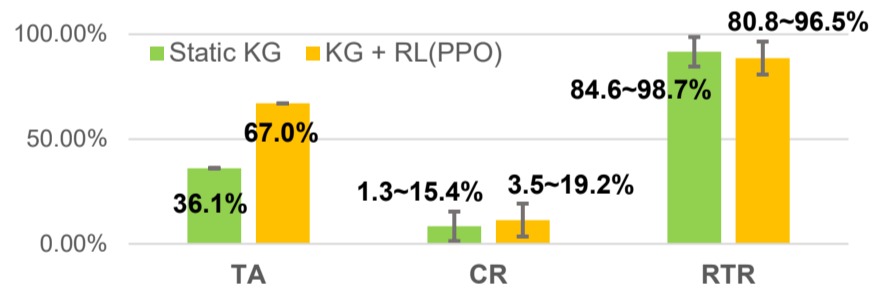


RESULTS

*Affected IPs: IPs derived from KG + RL that they were affected by the changed IPs

Quantitative Improvements (KG+RL vs Static KG)

- Targeting Accuracy (TA):** 36.1% → 67.0% (↑ ~31%)
*TA Effect = ((#Changed IPs + #Affected IPs) - #Changed IPs) / #Total IPs
- Coverage Recovery (CR):** ↑ 2.2–9.1%
*CR Effect = ((#Result-changed TCs @ changed IPs or affected IPs) - #Result-changed TCs @ changed IPs) / #Total TCs
- Redundant Test Ratio (RTR):** ↓ 2.3–9.2%
*RTR Effect = ((#Result-unchanged TCs @ changed IPs or affected IPs) - #Result-unchanged TCs @ changed IPs) / #Total TCs



CONCLUSIONS

- Static, hierarchy-based verification is insufficient to capture impact propagation caused by design changes. The proposed **RL-weighted KG** learns execution-level dependencies, improving targeting accuracy and reducing redundant tests. The framework enhances verification efficiency while **complementing full regression testing**, not replacing it.

Key Takeaways

- Design change impact extends beyond directly modified IPs
- Learned KG captures dynamic execution-level influence
- RL enables adaptive prioritization and efficient debugging

REFERENCES

- S. Kim and H. Park, "DRL-Verify: Deep Reinforcement Learning for Exhaustive SoC Verification," 2023 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1-9, Nov. 2023
- Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, Yu Su, "HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models" NeurIPS 2024
- Seonghee Yim, Hanna, Sunchang Choi and Seonil Brian Choi, "Accelerating SOC Verification using Process Automation and Integration" Design Verification Conference (DVCon), 2020
- J. Bergeron, E. Cerny, A. Hunter, and A. Nightingale, Verification Methodology Manual for SystemVerilog. Springer, 2006.