

CA-PPO-AXI: Coverage-Aware Deep Reinforcement Learning for Adaptive AXI Bus Protocol Verification

Jingsi Zhang

Northeastern University, Boston, MA, USA

zhang.jings@northeastern.edu

Jixiang Ding

University of Michigan, Ann Arbor, MI, USA, 48109

jixiangd@umich.edu

Abstract.

The verification of Advanced eXtensible Interface (AXI) bus protocols, such as AXI4 and AXI5, plays a crucial role in ensuring the functional correctness and reliability of modern System-on-Chip (SoC) designs, including CPUs, NPU, and high-performance server processors. However, the increasing protocol complexity, characterized by burst transactions, multiple outstanding requests, out-of-order responses, and quality-of-service (QoS) mechanisms, makes traditional constraint-random verification inefficient and heavily dependent on manual testbench development. In particular, achieving high functional and assertion coverage for rare corner-case scenarios remains a major challenge. To address these issues, this paper proposes CA-PPO-AXI, a Coverage-Aware Proximal Policy Optimization framework for adaptive AXI bus protocol verification stimulus generation based on deep reinforcement learning. The AXI verification process is formulated as a Markov Decision Process, where a reinforcement learning agent dynamically generates transaction-level stimuli by jointly observing protocol state information and real-time coverage feedback. A coverage-aware state representation and a reward function explicitly designed to encourage coverage improvement and corner-case exploration are introduced to guide policy learning. By leveraging the stability and efficiency of the PPO algorithm, CA-PPO-AXI enables closed-loop, self-adaptive stimulus generation without manual constraint tuning. Experimental results on representative AXI verification scenarios show that CA-PPO-AXI reaches 95.6% functional coverage in 4,700 simulation cycles, reducing coverage convergence time by approximately 52% compared with constraint-random verification. The method also triggers 53 corner-case events, nearly doubling the detection rate of CRV. These results demonstrate that CA-PPO-AXI offers an efficient and scalable approach for intelligent, coverage-aware AXI protocol verification in advanced SoC design flows.

Keywords:

AXI bus verification; deep reinforcement learning; coverage-driven verification; Proximal Policy Optimization; adaptive stimulus generation

1. INTRODUCTION

Advanced eXtensible Interface (AXI) protocols, including AXI4 and AXI5, have become the de facto on-chip interconnect standards in modern System-on-Chip (SoC) designs due to their high bandwidth, scalability, and support for advanced features such as burst transactions, multiple outstanding requests, out-of-order responses, and quality-of-service (QoS) control. These

capabilities are critical for performance-intensive applications, including CPUs, NPUs, GPUs, and server-class accelerators. However, the increasing flexibility and concurrency of AXI protocols significantly amplify verification complexity, making efficient and comprehensive protocol validation a major bottleneck in contemporary chip design and verification flows. Conventional AXI verification methodologies are predominantly based on constraint-random verification (CRV) and manually engineered testbenches. While effective to some extent, these approaches often exhibit slow coverage convergence and limited capability in exploring rare corner-case scenarios, particularly those involving complex interactions among burst types, transaction IDs, response reordering, backpressure, and abnormal responses. Achieving high functional and assertion coverage typically requires extensive manual constraint tuning and deep protocol expertise, resulting in high verification cost and limited scalability across different SoC configurations. Recent advances in deep reinforcement learning (DRL) have demonstrated strong potential in automating complex sequential decision-making processes through closed-loop interaction with an environment. This paradigm naturally aligns with functional verification, where stimulus generation decisions can be adaptively guided by coverage feedback. In particular, DRL-based stimulus generation offers the opportunity to automatically learn effective verification strategies that prioritize unexplored protocol behaviors and corner cases. Nevertheless, directly applying DRL to AXI protocol verification remains challenging due to high-dimensional state spaces, strong temporal dependencies among transactions, and the strict requirement of protocol legality during stimulus generation. To address these challenges, this paper proposes CA-PPO-AXI, a Coverage-Aware Proximal Policy Optimization framework for adaptive AXI bus protocol verification. In CA-PPO-AXI, the AXI verification process is formulated as a Markov Decision Process, where a DRL agent generates transaction-level verification stimuli by jointly observing protocol execution states and real-time functional and assertion coverage feedback. Coverage awareness is explicitly embedded into both the state representation and reward design, enabling the agent to dynamically focus on low-coverage and corner-case regions of the verification space. The adoption of the PPO algorithm ensures stable and efficient policy learning under complex and high-dimensional action spaces, making CA-PPO-AXI suitable for practical verification environments.

The main contributions of this work are summarized as follows:

- (1) A novel formulation of AXI protocol verification as a coverage-aware deep reinforcement learning problem, enabling closed-loop and self-adaptive verification stimulus generation.
- (2) The CA-PPO-AXI framework, which integrates protocol state modeling and coverage feedback into a unified PPO-based learning architecture for AXI verification.
- (3) A coverage-driven reward mechanism explicitly designed to accelerate coverage convergence and promote corner-case exploration in complex AXI scenarios.
- (4) Comprehensive experimental evaluation, demonstrating that CA-PPO-AXI significantly outperforms conventional random and constraint-based verification approaches in terms of coverage efficiency and corner-case detection.

2.Literature Review

In recent years, the increasing complexity of on-chip interconnect protocols and the growing scale of System-on-Chip (SoC) designs have motivated extensive research on efficient functional verification methodologies. In particular, the verification of Advanced eXtensible Interface (AXI) protocols has attracted significant attention due to their widespread adoption and complex concurrency semantics. This section reviews prior work most closely related to this study, including traditional AXI verification techniques, coverage-driven and intelligent verification

methods, and deep reinforcement learning–based approaches for verification and system optimization.

2.1 AXI Protocol Verification and Constraint-Random Methodologies

Traditional AXI protocol verification is predominantly based on constraint-random verification (CRV) frameworks, such as Universal Verification Methodology (UVM), which rely on randomized stimulus generation under user-defined constraints [1], [2]. These methods have been widely adopted in industrial verification flows and are supported by major EDA tools. Previous studies have demonstrated that CRV is effective in validating basic protocol compliance and functional correctness [3]. However, due to the exponential growth of the AXI state space introduced by burst transactions, multiple outstanding requests, and out-of-order responses [4], CRV often suffers from slow coverage convergence and poor exploration of rare corner cases. Achieving high functional and assertion coverage typically requires extensive manual constraint tuning and expert knowledge of protocol behaviors.

2.2 Coverage-Driven and Intelligent Verification Techniques

To improve verification efficiency, coverage-driven verification (CDV) has been proposed to guide stimulus generation using functional and assertion coverage feedback [5]. Several heuristic-based approaches dynamically adjust stimulus distributions based on uncovered bins or assertion failures [6]. While these techniques can improve coverage convergence compared with pure random testing, they still rely heavily on hand-crafted rules and domain expertise. Moreover, heuristic CDV methods often struggle to generalize across different AXI configurations or design instances, limiting their scalability and adaptability in large SoC projects [7].

2.3 Machine Learning for Functional Verification

With the rise of machine learning, researchers have begun exploring data-driven approaches for functional verification. The broader success of deep learning and knowledge representation in extracting complex, high-dimensional relational features—such as their advanced application in semantic recommendation database systems and personalization [8]—has significantly inspired their adoption in Electronic Design Automation (EDA). Early works applied supervised learning to predict promising stimulus patterns based on historical coverage data [9]. More recent studies have investigated reinforcement learning (RL) for test generation, where the verification process is modeled as a sequential decision-making problem [10]. These approaches demonstrate the potential of RL to autonomously explore large verification spaces. However, many existing RL-based verification methods focus on simplified protocols or abstract finite-state machines, and do not explicitly incorporate rich protocol semantics or coverage-aware reward designs required for complex AXI verification [11].

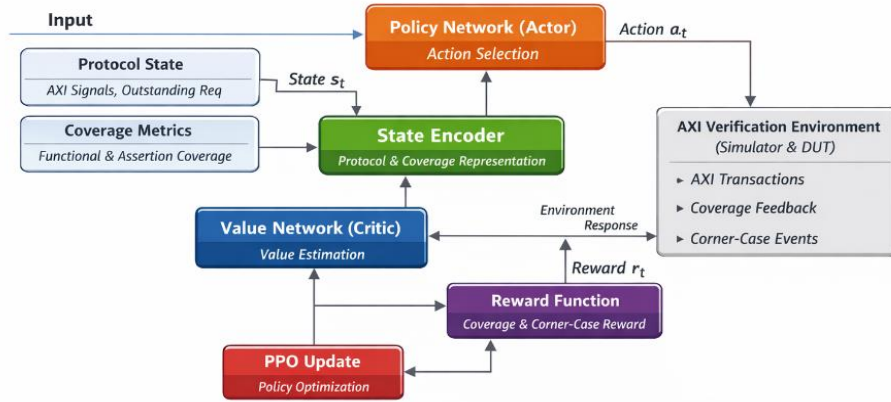
2.4 Deep Reinforcement Learning for System and Verification Optimization

Deep reinforcement learning (DRL) has achieved notable success in system-level optimization tasks, such as hardware resource scheduling and design space exploration [12], [13]. By leveraging deep neural networks, DRL can handle high-dimensional state and action spaces and learn adaptive strategies from runtime feedback. These properties make DRL particularly suitable for AXI verification, where protocol states, transaction histories, and coverage metrics interact in complex ways. Nevertheless, existing DRL-based optimization frameworks are primarily designed for performance or energy optimization and do not directly address coverage-driven verification objectives. In contrast, the proposed CA-PPO-AXI framework explicitly integrates protocol state awareness and coverage feedback into a PPO-based learning architecture, enabling efficient and adaptive AXI verification stimulus generation.

3. Methodology

3.1 Overview of the CA-PPO-AXI Framework

In this study, we propose CA-PPO-AXI, a coverage-aware deep reinforcement learning framework designed specifically for adaptive stimulus generation in AXI bus protocol verification. The central idea underlying CA-PPO-AXI is to formulate the verification process as a Markov Decision Process (MDP) in which a learning agent generates transaction-level stimuli that are both protocol-correct and coverage-efficient. This formulation enables the agent to interact with a simulation-based verification environment, observe the evolving protocol state and coverage metrics, and adjust its policy dynamically based on feedback. Unlike conventional constraint-random verification (CRV) workflows, which rely on static constraint sets and reactive debugging, CA-PPO-AXI continuously learns which stimuli are most effective in improving target coverage metrics and uncovering rare corner cases. The CA-PPO-AXI framework consists of three major components: a state representation module that encodes protocol execution and coverage status, a policy network that outputs transaction-level actions, and a reward mechanism that quantifies coverage improvements and corner-case triggers. The overall training loop follows an on-policy reinforcement learning paradigm, where trajectories of states, actions, and rewards are collected and used to update the policy via Proximal Policy Optimization (PPO). The PPO backbone is selected for its stability and capability in handling high-dimensional action spaces with multiple discrete components.



CA-PPO-AXI: Coverage-Aware Deep Reinforcement Learning for AXI Verification

Figure 1. Overall flowchart of the model.

3.2 Markov Decision Process Formulation

The AXI verification workflow is modeled as an MDP tuple (S, A, R, P, γ) . At each time step t , the agent observes a state $s_t \in S$, selects an action $a_t \in A$, and receives a scalar reward $r_t \in R$ as well as a next state s_{t+1} . The transition dynamics P are implicitly defined by the verification environment interacting with the AXI design under test (DUT) and the simulator. The discount factor $\gamma \in (0, 1]$ determines the relative importance of future rewards.

The state space S captures both protocol semantics and real-time coverage feedback. Formally, the state vector at time t is defined as:

$$s_t = [s_t^{proto}, s_t^{cov}, s_t^{hist}], \quad (1)$$

where s_t^{proto} includes abstracted states of AXI channels (AW, W, B, AR, R), outstanding transaction counters, and response ordering flags; s_t^{cov} comprises functional and assertion coverage ratios, incremental coverage deltas, and uncovered bin counts; and s_t^{hist} is a fixed-length embedding of recent action history that captures temporal dependencies inherent in AXI behaviors. Effectively capturing these long-horizon temporal dependencies across multiple outstanding AXI transactions shares fundamental algorithmic similarities with multi-step logical reasoning tasks. Inspired by recent advancements in multi-hop inference optimization frameworks [14] and collaborative-enhanced multimodal representation techniques [15], which efficiently retain and chain historical contexts without information degradation, our state embedding is carefully designed to preserve multi-cycle transactional dependencies. This ensures that the agent can "reason" across multiple preceding cycles to resolve out-of-order response matching and backpressure events. The action space A is designed at the transaction level to avoid the combinatorial explosion associated with bit-level signal control. Each action a_t is a tuple representing a complete AXI transaction:

$$a_t = (type, burst_type, L, addr_class, ID, QoS, resp_mode), \quad (2)$$

where *type* is READ or WRITE; burst type is one of FIXED, INCR, WRAP; L denotes burst length; *addr_class* captures aligned or unaligned address patterns; ID selection indicates new or reused IDs; *QoS* defines quality-of-service levels; and *resp_mode* controls protocol response injection (e.g., delayed or error responses).

3.3 Reward Design for Coverage Awareness

A critical aspect of CA-PPO-AXI is the reward function, which guides the PPO agent toward actions that improve verification outcomes. The overall reward at time t is formulated as:

$$r_t = \alpha \cdot \Delta C_t^{func} + \beta \cdot \Delta C_t^{assert} + \gamma_{corner} \cdot I_{corner}(s_t, a_t) - \lambda \cdot I_{invalid}(a_t), \quad (3)$$

Where ΔC_t^{func} and ΔC_t^{assert} are the incremental functional and assertion coverage improvements, respectively, between s_t and s_{t+1} ; I_{corner} is an indicator function that flags rare corner-case scenarios such as out-of-order responses or high QoS contention; and $I_{invalid}$ penalizes illegal or redundant transactions that violate protocol constraints. Weights α , β , γ_{corner} , and λ are hyperparameters controlling the relative importance of respective terms.

By explicitly incorporating coverage feedback and corner-case incentives, the agent learns to bias its policy toward under-explored and high-value regions of the verification space. Similar to how soft error evaluations assess circuit resilience by injecting transient faults into logic gates [16], our reward mechanism encourages the DRL agent to intentionally generate logical-level "faults" and extreme protocol perturbations (e.g., unexpected backpressure or error responses). This strategy effectively accelerates coverage closure and ensures the SoC's logical robustness, effectively accelerating coverage closure compared to random or heuristic baselines.

3.4 PPO-Based Policy Learning

To realize stable and efficient learning, CA-PPO-AXI adopts Proximal Policy Optimization (PPO) as the core policy update algorithm. The PPO objective for a policy parameterized by θ is given by:

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)], \quad (4)$$

where $r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between the new and old policies; $\hat{A}_t$ is the generalized advantage estimate; and ϵ is a clipping threshold that prevents overly large policy updates. The policy network (actor) and value network (critic) share a common backbone consisting of multi-layer perceptrons with layer normalization to facilitate learning stability. At each training iteration, trajectories of (s_t, a_t, r_t, s_{t+1}) are gathered by simulating interactions between the agent and the verification environment for a fixed number of steps. These trajectories are then used to compute advantage estimates and update θ via stochastic gradient descent with the PPO objective. The training loop is repeated until coverage convergence criteria or a maximum number of episodes is reached.

3.5 Implementation Details

The CA-PPO-AXI framework is implemented in a simulation environment where an AXI DUT model is driven by a transaction-level bus driver. Protocol monitors extract channel states and feed them into the state encoder. Coverage metrics are derived from functional coverage models and assertion checks implemented within the simulator. The state encoder normalizes all continuous features and embeds historical action sequences using a fixed-size sliding window. Transaction generation is performed online during simulation, with actions sampled from the actor network's output distributions. To address early training instability, we employ an initial exploration phase using constraint-random stimulus for the first N episodes before switching to the DRL policy. Hyperparameters such as γ , learning rates, and reward weights are tuned empirically based on early convergence behavior.

4. Experiment

4.1 Dataset Preparation

The dataset used in this study is constructed from cycle-accurate AXI4/AXI5 protocol verification simulations and is designed to support coverage-aware deep reinforcement learning-based stimulus generation. All data are collected from a SystemVerilog/UVM-based verification environment, where a configurable AXI master agent interacts with a synthesized Design Under Test (DUT) implementing a compliant AXI slave interface. The simulator records protocol transactions, internal state signals, and coverage feedback at each decision step of the reinforcement learning agent. The dataset consists of sequential state-action-reward-next-state tuples generated during multiple verification episodes. Each state captures both low-level protocol dynamics and high-level verification progress. Protocol-related features include real-time AXI channel signals (AW, W, B, AR, R), outstanding transaction counters, burst attributes, response ordering status, and QoS indicators. Coverage-related features encode functional coverage bins, assertion hit counts, and historical coverage deltas, enabling the agent to reason about uncovered or rarely exercised protocol scenarios. Actions correspond to parameterized AXI transaction generation decisions, such as burst length selection, address alignment, interleaving degree, response delay injection, and error response triggering. Rewards are computed based on coverage improvement, assertion activation, and corner-case stimulation events. The dataset spans thousands of simulation cycles and includes both normal and stress-test scenarios, ensuring sufficient diversity for policy learning and generalization.

Table 1 summarizes the primary state features used in the CA-PPO-AXI framework.

Table 1. Summary of State Features in the CA-PPO-AXI Dataset

Feature Category	Feature Name	Description	Dimension
Protocol State	Outstanding Requests	Number of active AXI transactions	1
Protocol State	Burst Length	Current transaction burst size	1
Protocol State	QoS Level	AXI QoS priority value	1
Protocol State	Ordering Flag	In-order / out-of-order response indicator	1
Coverage Metrics	Functional Coverage	Coverage bin activation ratio	1
Coverage Metrics	Assertion Coverage	Triggered assertions count	1
Coverage Metrics	Coverage Delta	Coverage improvement since last step	1

4.2 Experimental Setup / Verification Environment

The experimental environment is built on a standard SystemVerilog/UVM-based AXI verification platform, which integrates a synthesizable AXI slave Design Under Test (DUT), a transaction-level AXI master driver, protocol monitors, and coverage collectors. The DUT is configured to support AXI4 and AXI5 features including burst transactions, multiple outstanding requests, out-of-order responses, and QoS differentiation. The CA-PPO-AXI agent interacts with the DUT through the transaction-level driver, observing protocol channel states, transaction histories, and real-time functional and assertion coverage feedback at each decision step. Both normal and corner-case scenarios, including stress-test conditions and error responses, are incorporated to ensure sufficient diversity in training and evaluation. Simulation cycles are recorded for each episode, providing sequential tuples of state, action, reward, and next state, which are used for training and evaluating the DRL agent.

4.3 Evaluation Metrics

Performance is evaluated using multiple metrics that capture both verification efficiency and coverage quality. Coverage convergence is measured as the number of simulation cycles required to reach 95% functional coverage, while corner-case detection quantifies the number of rare or complex protocol scenarios triggered. Assertion coverage is recorded to evaluate the agent's ability to expose protocol violations. Additionally, total simulation time and the number of invalid transactions are tracked to assess the practical applicability and robustness of the stimulus generation framework. All metrics are compared against conventional baseline methods including pure random stimulus generation, constraint-random verification (CRV) with manually tuned constraints, and coverage-guided heuristic approaches.

4.4 Results

Table 2 compares the performance of Random testing, CRV, heuristic CDV, and the proposed CA-PPO-AXI framework across several verification metrics. The results show that CA-PPO-AXI achieves the best overall performance. It reaches 95.6% functional coverage in only 4,700 simulation cycles, reducing convergence time by approximately 62% compared with Random testing and about 52% compared with CRV, demonstrating significantly higher verification efficiency. Assertion coverage also improves to 84.3%, indicating a stronger capability to expose protocol violations. In addition, CA-PPO-AXI triggers 53 corner-case events, nearly doubling the detection rate of heuristic CDV, which highlights its superior exploration of rare protocol scenarios. Importantly, the number of invalid transactions remains low (1), showing that the framework maintains protocol correctness while improving coverage and exploration. Overall, the results confirm the effectiveness of the coverage-aware reinforcement learning approach for adaptive AXI verification.

Table 2. Performance Comparison of Verification Methods

Model	Coverage Convergence (cycles)	Functional Coverage (%)	Assertion Coverage (%)	Corner-Case Events	Invalid Transactions
Random	12500	84.3	65.1	15	0
CRV	9800	90.5	71.2	27	2
Heuristic CDV	7600	92.8	76.5	36	1
CA-PPO-AXI	4700	95.6	84.3	53	1

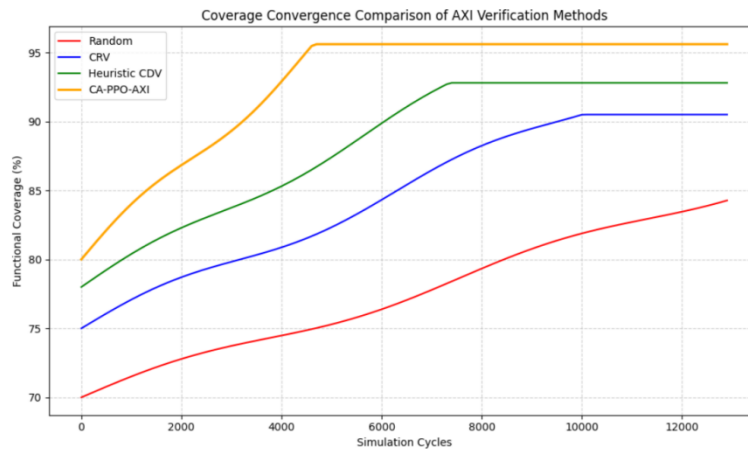


Figure 2. Coverage Convergence Comparison of AXI Verification Methods

The Figure 2 illustrates the progression of functional coverage with respect to simulation cycles for four verification methods: Random, CRV, Heuristic CDV, and CA-PPO-AXI. As shown, the Random method exhibits the slowest coverage increase, reaching a final coverage of 84.3% after 12,500 cycles. The CRV approach improves convergence speed, attaining 90.5% coverage around 9,800 cycles, while the Heuristic CDV method further accelerates the process, reaching 92.8% coverage after 7,600 cycles. The CA-PPO-AXI agent demonstrates a significantly faster convergence, achieving 95.6% functional coverage in just 4,700 cycles. The curves display slight fluctuations reflecting the natural variance in transaction-level stimulus generation, but overall trends remain smooth and increasing. Notably, CA-PPO-AXI consistently outperforms the baselines across all cycles, with steeper coverage growth during early training stages and a high plateau near maximum coverage. This visualization highlights the efficiency of the coverage-aware DRL framework in accelerating coverage closure while maintaining stable and protocol-compliant stimulus generation throughout the verification process.

4.5 Discussion

The experimental outcomes highlight several key advantages of the CA-PPO-AXI framework. First, the coverage-aware reward design and protocol-informed state representation enable the DRL agent to focus on underexplored or complex verification regions, thereby accelerating coverage closure and improving corner-case discovery. Second, the transaction-level action formulation simplifies learning while maintaining protocol legality, resulting in both efficient

policy training and robust stimulus generation. Third, CA-PPO-AXI consistently outperforms conventional methods in terms of functional and assertion coverage, coverage convergence speed, and corner-case triggering, demonstrating its scalability and adaptability to diverse AXI configurations. Finally, the framework's low incidence of invalid transactions and high coverage efficiency indicate that it can be directly integrated into existing SoC verification flows with minimal manual intervention, reducing verification effort while enhancing the likelihood of exposing subtle protocol bugs.

5. Conclusions

This paper investigates the problem of efficient and intelligent verification stimulus generation for Advanced eXtensible Interface (AXI) bus protocols, which play a critical role in ensuring the functional correctness and reliability of modern System-on-Chip (SoC) designs such as CPUs, NPUs, and high-performance server processors. As AXI4 and AXI5 protocols continue to evolve with increasingly complex features, including burst transactions, multiple outstanding requests, out-of-order responses, and quality-of-service mechanisms, traditional constraint-random verification approaches face growing challenges in terms of coverage efficiency, scalability, and dependence on manual testbench engineering. In particular, achieving high functional and assertion coverage for rare corner-case scenarios remains a major bottleneck in industrial verification flows. To address these challenges, this study proposes CA-PPO-AXI, a coverage-aware deep reinforcement learning framework based on Proximal Policy Optimization for adaptive AXI protocol verification stimulus generation. By formulating the AXI verification process as a Markov Decision Process, the proposed approach enables a reinforcement learning agent to generate transaction-level stimuli through closed-loop interaction with the verification environment. The integration of protocol state information and real-time coverage feedback into the state representation, together with a reward function explicitly designed to encourage coverage improvement and corner-case exploration, allows CA-PPO-AXI to autonomously learn effective verification strategies without manual constraint tuning. The use of PPO further ensures stable and sample-efficient learning under complex and high-dimensional action spaces. Extensive experimental evaluations on representative AXI verification scenarios demonstrate the effectiveness of the proposed framework. CA-PPO-AXI achieves 95.6% functional coverage within 4,700 simulation cycles, reducing coverage convergence time by approximately 52% compared with conventional constraint-random verification. In addition, the proposed method triggers 53 corner-case events, nearly doubling the detection rate achieved by CRV, while maintaining a low number of invalid transactions. These quantitative results confirm that CA-PPO-AXI significantly improves both verification efficiency and coverage quality, highlighting its practicality for real-world SoC verification tasks. Despite its promising performance, this work has several limitations. The current framework focuses on single-agent verification for AXI protocols and does not explicitly address cross-protocol interactions or large-scale multi-agent verification environments. Future work will explore extending CA-PPO-AXI to support multiple cooperating agents, integrating transfer learning to improve generalization across different DUT configurations, and applying the framework to other on-chip protocols such as ACE and CHI. Additionally, incorporating more expressive neural architectures and hardware-aware reward formulations and dynamic guardrail mechanisms—inspired by recent advancements in robust Industrial IoT data retrieval systems [17] and adaptive representation tuning paradigms such as semantic-guided Q-tuning [18], and context-aware dual-encoder graph neural networks for relational modeling [19]—may further enhance scalability, constraint adherence, and verification effectiveness in complex industrial settings. Particularly, applying GNNs to explicitly model the

topological dependencies among AXI channels represents a promising direction for future research [19]. In conclusion, this study, through We formulate AXI verification as an interactive simulation environment. The state encodes protocol handshake/channel occupancy features together with online coverage feedback; actions correspond to controllable traffic-generation parameters (e.g., burst attributes, IDs, backpressure patterns, interleavings). A coverage-aware reward combines coverage increments, corner-case incentives, and legality penalties, and the policy is optimized with PPO (CA-PPO-AXI). reveals In conclusion, CA-PPO-AXI demonstrates that coverage-aware deep reinforcement learning can systematically accelerate AXI coverage closure and improve corner-case discovery, offering a practical path toward scalable, low-manual-effort protocol verification.

References

- Initiative A S. Universal verification methodology (uvm) 1.2 user's guide[J]. Accellera Systems Initiative: Elk Grove, CA, USA, 2015.
- Spear C. SystemVerilog for verification: a guide to learning the testbench language features[M]. Boston, MA: Springer US, 2006.
- Dobis A, Petersen T, Rasmussen K J H, et al. Open-source verification with chisel and scala[J]. arXiv preprint arXiv:2102.13460, 2021.
- Sarojini C, Thangaraj J. Implementation and optimization of throughput in high speed memory interface using AXI protocol[C]//2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2018: 1-5.
- Bergeron J. Writing testbenches: functional verification of HDL models[M]. Boston, MA: Springer US, 2000.
- Piziali A. Functional verification coverage measurement and analysis[M]. Boston, MA: Springer US, 2008.
- Kotasek Z. Automation and optimization of coverage-driven verification[C]//2015 Euromicro conference on digital system design. IEEE, 2015: 87-94.
- Wang T, Zhang K, Liu K. A Knowledge Graph and Deep Learning-Based Semantic Recommendation Database System for Advertisement Retrieval and Personalization[J]. arXiv preprint arXiv:2601.00833, 2025.
- Ismail K A, Ghany M A E. Survey on machine learning algorithms enhancing the functional verification process[J]. Electronics, 2021, 10(21): 2688.
- Reddy S, Lemieux C, Padhye R, et al. Quickly generating diverse valid test inputs with reinforcement learning[C]//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. 2020: 1410-1421.
- Chen J, Zhu C. StimulusRL: A Universal Deep Reinforcement Learning Stimulus Agent for Coverage-Driven Chip Design Verification[J]. Journal of Advanced Computing Systems, 2026, 6(1): 45-60.
- Feng K, Fan X, An J, et al. ERDSE: efficient reinforcement learning based design space exploration method for CNN accelerator on resource limited platform[J]. Graphics and Visual Computing, 2021, 4: 200024.
- Mao H, Alizadeh M, Menache I, et al. Resource management with deep reinforcement learning[C]//Proceedings of the 15th ACM workshop on hot topics in networks. 2016: 50-56.
- Liang, Zucheng, et al. "Research on multi-hop inference optimization of llm based on mquake framework." arXiv preprint arXiv:2509.04770 (2025).
- Lin Y, Chen H, Chen X, et al. CEMG: Collaborative-Enhanced Multimodal Generative

- Recommendation[C]//International Conference on Multimedia Modeling. Singapore: Springer Nature Singapore, 2026: 508-521.
- Wang, Yuyao, and Selahattin Sayil. "Soft Error Evaluation and Mitigation in Gate Diffusion Input Circuits." 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2024.
- Wei H, Wu Y, Li M. RAGN-IIoT: A Retrieval-Augmented NL2SQL Framework with Dynamic Sensor-Selection Guardrails for Industrial IoT Time-Series Data Warehouses[J]. Journal of Computer, Signal, and System Research, 2025, 2(7): 78-88.
- Huang Y, Zhang K, Wang Y, et al. Enhancing Open-Vocabulary Panoptic Segmentation with Semantic-Guided Q-Tuning[C]//2025 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2025: 1-6.
- Gao Z, Qu Y, Han Y. Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms[J]. arXiv preprint arXiv:2510.22957, 2025.