DATE 26
DESIGN, AUTOMATION AND TEST IN EUROPE
THE EUROPEAN EVENT FOR ELECTRONIC SYSTEM DESIGN & TEST
20 – 22 APRIL 2026
VERONA, ITALY
PALAZZO DELLA GRAN GUARDIA

Towards Trustworthy LLM-Based Assertion Generation: A Data Augmentation Framework with Formal Check Approach

Qingchen Zhai†, Hao Yu‡*, Chen Bai‡, Charles Young§, Frank Qu¶, Dezhi Ran§, Yuan Xie‡, Tao Xie§*

†Institute of Automation, Chinese Academy of Sciences, Beijing, China

§School of Computer Science, Peking University, Beijing, China

‡Hong Kong University of Science and Technology, Hong Kong, China

¶Independent Research

中国科学院自动化研究所
INSTITUTE OF AUTOMATION
CHINESE ACADEMY OF SCIENCES

北京大學
PEKING UNIVERSITY

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Background

Functional Verification Bottleneck

Rapid growth of IC complexity makes verification the dominant cost in design cycles

Assertion-Based Verification (ABV) is widely adopted for specifying design properties and enabling formal checking

Limitations of Existing Automation

- Static analysis: misses complex temporal behaviors
- Dynamic mining: limited by simulation coverage
- Traditional ML/NLP methods: poor generalization due to data scarcity

Emerging Opportunity: LLMs

- Strong capability in understanding natural language specifications
- Early works focus on prompt engineering
- Key bottleneck: lack of large-scale, high-quality, verified datasets

Main Contribution

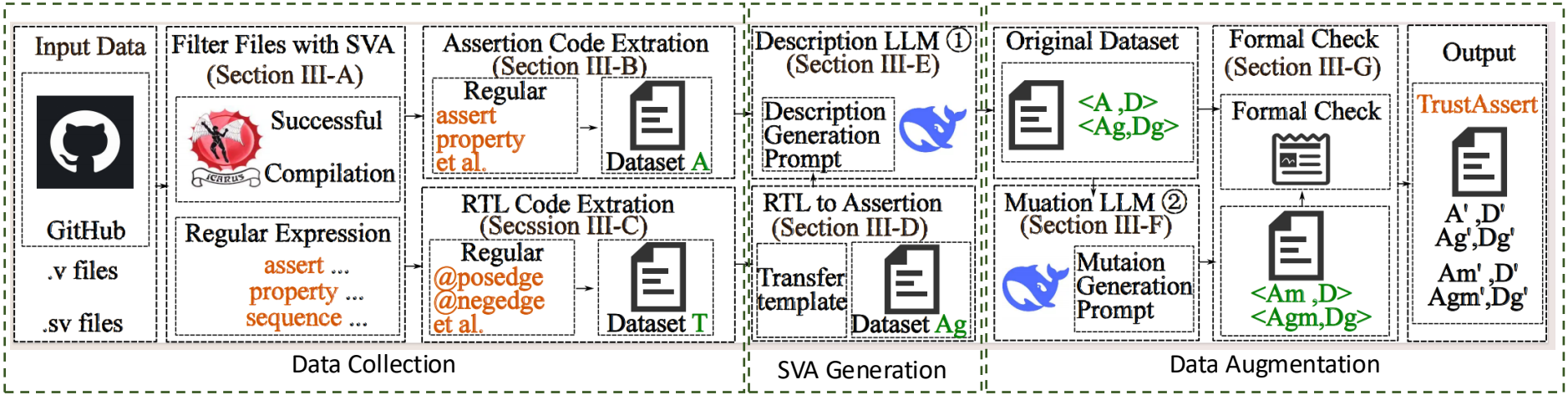
Our work introduces a rigorous methodology for constructing a scalable dataset of provably correct assertions, specifically designed to enable and accelerate AI research in this domain.

- **We present a comprehensive analysis of assertion patterns and distributions from a large corpus of open-source RTL projects (2020-2025), highlighting the current landscape and the need for curated data.**

- **We propose AutoAssert, a formal-verification-based framework that automatically generates high-quality assertion data. We release its output as the TrustAssertdataset, a corpus of 110K formally verified assertions.**

- **We fine-tune Llama2-7B on TrustAssert and achieve performance that surpasses GPT-4 in key scenarios. This result demonstrates the effectiveness of targeted fine-tuning with formally verified data.**

AutoAssert Framework



- Data Collection

Data preprocess

- Assertion Generation from RTL

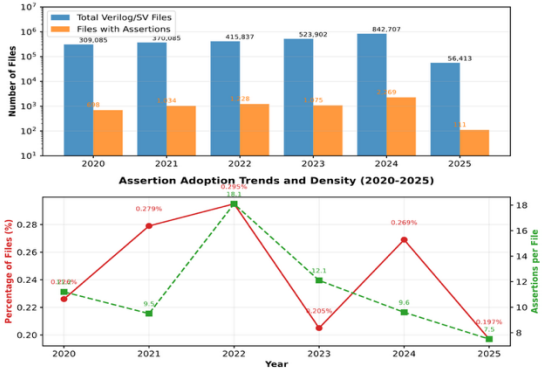
RTL Transfer to Assertion

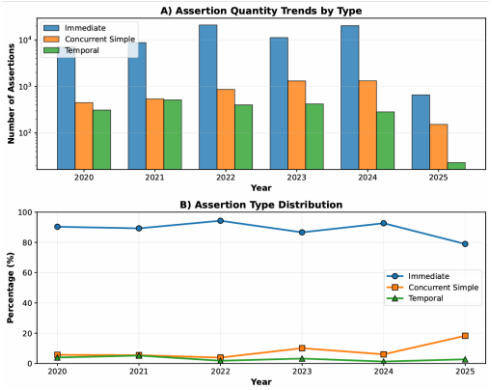
- Data Augmentation

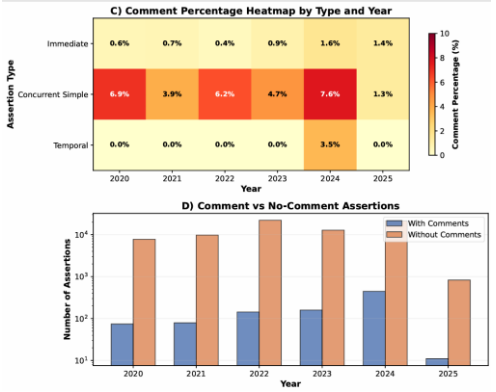
Data Mutation

Consistency Check

Data Collection: Addressing Data Scarcity



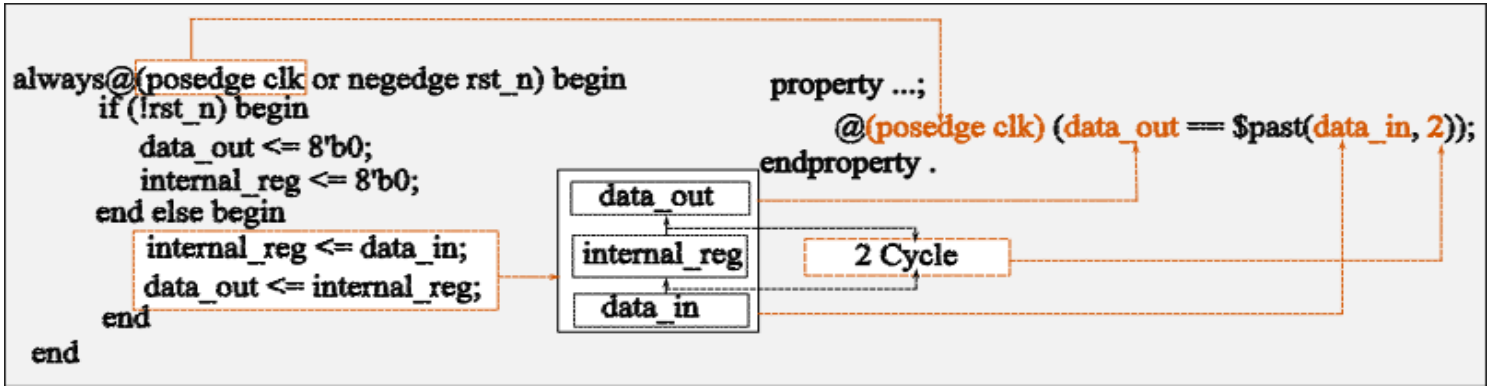




Mine 2.2TB open-source RTL repositories (2020–2025)

- **Extremely Low Adoption Rate:** Only 14.65% of 2.84 million Verilog/SystemVerilog source files contain assertions, highlighting a critical need for automated assertion generation in modern RTL design workflows.
- **Severe Imbalance in Assertion Types:** Immediate assertions dominate, accounting for over 85% of usage but offering limited verification value.
- **Identified Research Gap & Direction:** Increasing the quantity of temporal assertions (solving the “availability” problem). Ensuring semantic documentation (solving the “understanding and reuse” problem) to enable effective verification reuse and validation.

Generating Temporal Assertions from RTL



- Convert sequential RTL into multi-cycle temporal assertions
- Detect pipeline dependencies via:
  - always @(posedge clk) patterns
  - Signal propagation chains
- Generate complex temporal properties, not just trivial checks

Data Augmentation with Formal Guarantees

Problem: LLM-generated data is unreliable

- Descriptions may be incorrect
- Mutations may break semantics
- Noise can harm model training

Our Solution: Structured Data Augmentation Pipeline

1. Description Generation (LLM)
   - Assertion → Natural language
   - Capture functional intent
2. Mutation-based Augmentation
   - Generate semantically equivalent variants
   - Increase syntactic diversity

Outcome

TrustAssert Dataset 110K formally verified assertions

TrustAssert

Source	Immediate	Concurrent_simple	Temporal	Total
A'	20,000	4,216	1,827	26,043
A'_g	0	7,327	18,732	26,059
A'_m	22,530	4,216	1,827	28,573
A'_{gm}	0	10,483	20,863	31,346
Total	**42,530**	**26,242**	**43,249**	**112021**

Objective: Address type imbalance in open-source data (over-representation of immediate assertions) by constructing a balanced, high-quality dataset for assertion generation.

Construction:

•Curation (A): Selected 20,000 immediate assertions from 69,082 to balance types (total: 26,043).

•Generation (A_g): Synthesized 26,059 assertions from RTL code, significantly enriching complex types (7,327 concurrent simple, 18,732 temporal).

•Mutation (A_m): Applied formal-equivalence-preserving transformations to enhance diversity (A‘_m: 28,573, A’_gm: 31,346).

Key Achievements:

•+62% increase in total assertion volume

•+2,267% increase in temporal assertions (vs. original data)

•All assertions are formally verified for functional correctness

Experiments: Setting

Benchmark

- SOCKIT: an open-source socket controller implemented on an FPGA-based SoC;
- UART: a UART-to-bus interface from OpenCores
- I2C: an I2C master/slave logic controller from AssertLLM
- HATX:a custom high-throughput arbitration and crossbar module.

Metric

- Non-trivial/Total: counts assertions that capture non-trivial design properties
- Syntax Correct: verifies syntactic validity
- FPV Correct: measures assertions proven valid by formal property verification.

SFT Setting

- Our experimental setup processes 112,021 training examples over three complete epochs, ensuring thorough exposure to the diverse assertion patterns in the TrustAssert dataset. The training procedure employs a batch size of two per device with gradient accumulation over eight steps, resulting in an effective batch size of 64. We fine-tune our models on 16 Nvidia Tesla V100S GPUs.

Experiments: Results

Module	Metric	Goldmine [6]	SPEC2Assert [16]	AssertLLM [15]	Llama2-7B [18]	Llama2-7B-T	Qwen-3B [31]	Qwen-3B-T	GPT4 [19]
UART [32]	Non-trivial/Total	0/0	54/54	48/59	0/12	25/47	0/10	6/23	13/43
	Syntax Correct	0	47	42	3	16	0	7	8
	FPV Correct	0	17	9	0	6	0	1	5
HATX [33]	Non-trivial/Total	41/83	153/153	73/90	12/25	35/67	5/18	15/57	23/76
	Syntax Correct	41	152	17	5	30	3	10	21
	FPV Correct	2	15	2	0	12	0	2	13
I2C [15]	Non-trivial/Total	18/18	111/111	104/130	15/28	46/62	3/22	37/62	32/71
	Syntax Correct	18	99	88	12	42	3	23	23
	FPV Correct	11	33	26	3	18	0	12	6
SOCKIT [32]	Non-trivial/Total	37/906	161/161	95/109	2/25	47/62	0/32	32/52	43/83
	Syntax Correct	37	148	71	18	33	12	25	32
	FPV Correct	7	53	27	4	23	3	12	29

- **Substantial Gains Over Base Models:** The fine-tuned models show major improvements. For instance, Llama2-7b-T achieves a 191% increase in the non-trivial assertion ratio for HATX and a 400% improvement in functional correctness for I2C.
- **Reduced Performance Gap with GPT-4:** The gap to GPT-4 narrows significantly. In functional correctness for the I2C module, the performance difference between Llama2-7b-T and GPT-4 is reduced to 33%.
- **Outperforming GPT-4 in Specific Cases:** Qwen-3B-T achieves a 280% improvement in the non-trivial assertion ratio for SOCKIT and exceeds GPT-4 in functional correctness for the I2C module by 100%.

Experiments: Results

Ablation Study

Datasets: A 50K-sample Unverified Dataset vs. the formally verified TrustAssert (RVAssert) Dataset.

Method: Both datasets are used to fine-tune the same base model (Qwen-Coder-3B) and evaluated on the same hardware module (I2C).

Metrics: Performance is measured on Non-trivial Assertion Ratio, Syntax Correctness, and Functional Correctness (FPV).

Metric	Unverified Dataset	RVAssert (Ours)
Non-trivial/Total	13/72	**37/62**
Syntax Correct	10	**23**
FPV Correct	6	**12**

- The Non-trivial Assertion Ratio increased from 13/72 to 37/62, a 230% improvement.
- The number of Syntax-Correct assertions rose from 10 to 23, a 166% increase.
- The count of FPV-Correct assertions doubled from 6 to 12, a 100% improvement.

Thank You

中国科学院自动化研究所
INSTITUTE OF AUTOMATION
CHINESE ACADEMY OF SCIENCES

北京大学
PEKING UNIVERSITY

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY