HADA: Hardware Assertion through Data Augmentation

Leveraging Multi-Source Knowledge for LLM-Based Security Assertion Generation

Weimin Fu[1], weiminf@ksu.edu
Yiting Wang[2], ywang144@umd.edu
Zelin Lu[2], zelinlu@umd.edu
Xiaolong Guo[1], guoxiaolong@ksu.edu
Gang Qu[2], gangqu@umd.edu

Background

- **Security assertions** are critical for detecting hardware vulnerabilities during pre-silicon verification, ensuring early detection and reducing costly post-silicon fixes.
- **Manual assertion writing** requires deep domain expertise, is labor-intensive, and often misses subtle vulnerabilities due to the complexity of modern SoC designs.
- **Traditional assertion generation tools** often lack adaptability to evolving threat models and design changes, leading to gaps in security coverage and delayed detection.

Motivation

- **Automation** enables broader vulnerability coverage, improves verification efficiency, and reduces human error.
- **HADA** leverages multi-source knowledge (CWE, version control, FPV) and formal validation tools to generate reliable security assertions automatically.
- Domain-specific LLMs fine-tuned with verified assertions achieve superior performance over traditional methods.

Workflow

- **1–2:** Generate assertions and hardware design from CWE with GPT4o.
- **3–4:** Extract versioned design pairs and generate assertions from their diffs in the version control system.
- **5-6:** Use AutoSVA2 to generate tool-based assertions from open source SoC designs.
- **7–8:** Validate syntax using VCS and Verilator; only passing assertions are retained. Explain the Design and assertion with GPT4o.
- **9, X:** Construct fine-tuning triplets and train domain-specific LLMs. (LLaMA, FT GPT4mini)

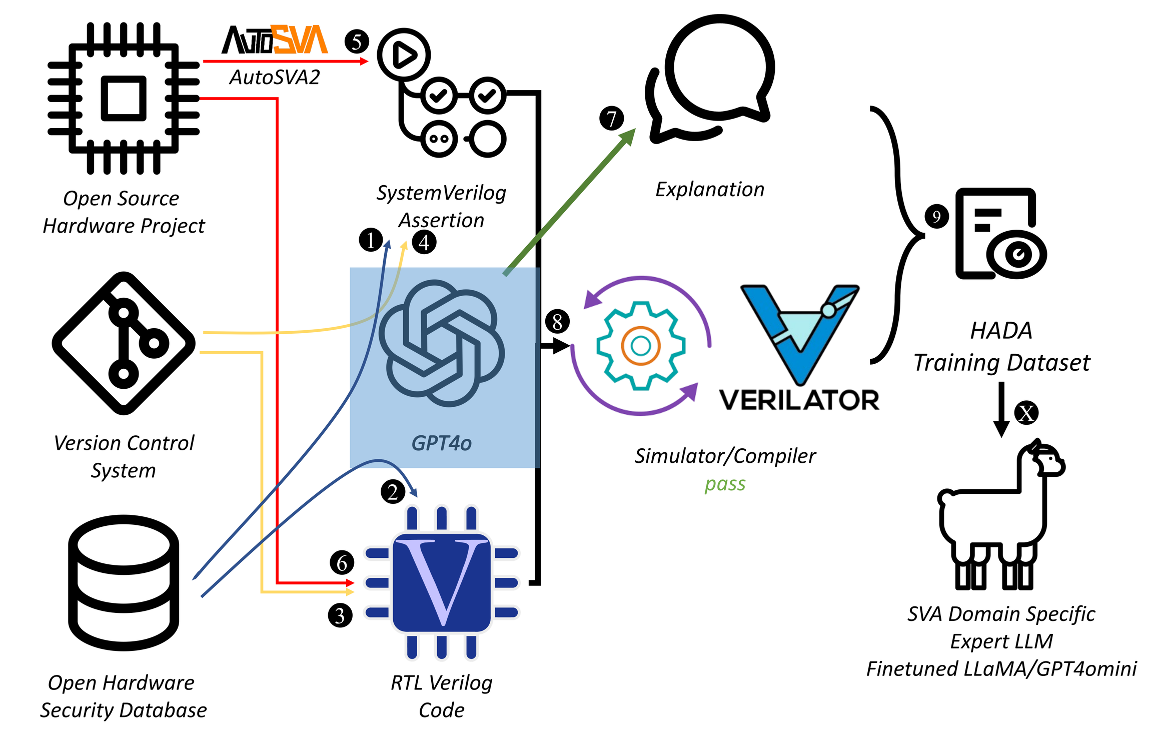


Figure 1: HADA workflow: integrating domain sources, generating & validating assertions, and producing fine-tuning data.

Evaluation Results

Table 1: Performance Comparison of LLMs Pre- and Post-Fine-Tuning on FVEval and HSAEval (Func $pass@1,5$)

Model		FVEval: Nvidia FV Real World Benchmark				HSAEval: Benchmark from open source SoC	
		FSM		Pipeline			
		Functionality					
		pass@1	pass@5	pass@1	pass@5	pass@1	pass@5
GPT4o-mini	base	10.11%	41.37%	8.81%	37.01%	11.96%	23.91%
	HADA	9.42%	39.08%	34.52%	88.03%	15.22%	32.60%
LlaMA3 70B	base	17.08%	60.89%	12.03%	47.39%	11.96%	17.39%
	HADA	30.58%	83.95%	23.19%	73.35%	17.39%	34.78%
LlaMA3.1 70B	base	24.26%	75.16%	18.93%	65.06%	7.17%	15.22%
	HADA	30.58%	83.95%	23.19%	73.35%	12.60%	30.43%
LlaMA3.2 3B	base	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	HADA	2.41%	11.49%	4.10%	18.91%	1.30%	2.17%
SOTA General Proprietary LLM							
GPT4o		10.40%	42.70%	37.30%	90.00%	26.09%	28.26%
Prompt Methods - base on GPT4o mini							
RTLFixer		0.00%	0.00%	0.00%	0.00%	0.22%	1.19%
DIVAS		0.00%	0.00%	0.00%	0.00%	0.22%	1.19%
LAAG		0.00%	0.00%	0.00%	0.00%	0.22%	1.19%
Fine Tuned LLM							
LLM4SecHW		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Evaluation Highlights

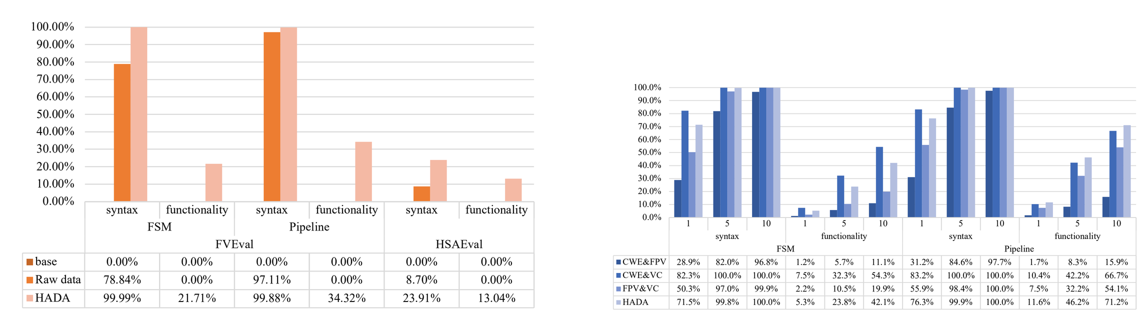
- Fine-tuning with HADA leads to consistent improvements across all evaluated LLMs, except a slight drop (5%) in FSM tasks for GPT-4o-mini.
- LLaMA-3.2B, initially unable to generate valid assertions, gained basic functionality after fine-tuning.
- Larger models (GPT-4o-mini, LLaMA-70B) show significant gains in functionality metrics after fine-tuning.
- HADA-trained models outperform existing prompt-based (RTLFixer, DIVAS, LAAG) and fine-tuned (LLM4SecHW) baselines by a substantial margin.

Data Source Ablation

Raw data improves syntax but lacks security depth. HADA's augmented data boosts both syntax and functionality.

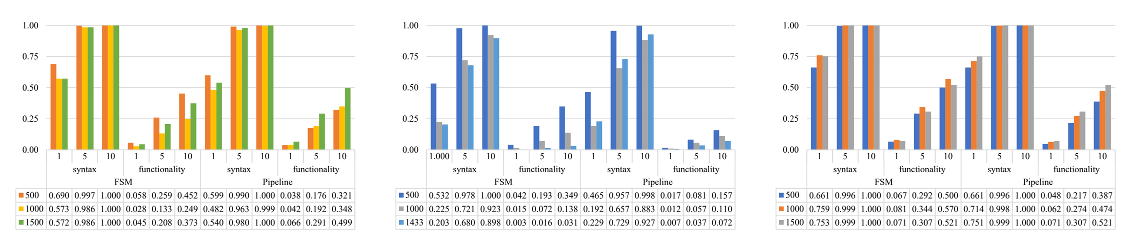
Version control yields the best results due to real bug-fix patterns. Formal verification is precise but repetitive; CWE adds structure but limited diversity.

Combining all sources performs best. CWE+FPV is the weakest due to redundancy.



(a) Raw vs. Augmented Data (b) Combined vs. Pairwise Composition

Figure 2: Performance comparison of data augmentation and source combinations. Bars represent pass@10 on syntax and functionality across FSM/Pipeline



(a) CWE only (b) FPV only (c) VC only

Figure 3: Comparison of fine-tuning results from individual data sources. VC leads to best functional assertions.

HSAEval Benchmark

- **Coverage:** 46 real-world security tasks derived from TrustHub, OpenPiton, and CWE vulnerabilities.
- **Formal Verification:** Each task includes a SystemVerilog testbench and is evaluated using JasperGold for assertion validity.
- **Metric:** Pass@k $= 1 - \binom{n-c}{k} / \binom{n}{k}$, measuring functional correctness under multiple generations.
- **Open Benchmark:** Publicly released for reproducible, community-driven evaluation.
- **Purpose:** HSAEval is specifically designed to benchmark security assertion generation, evaluating both syntactic validity and vulnerability-mitigation effectiveness in realistic SoC settings.

Takeaways & Future Work

- HADA demonstrates how domain-specific LLMs can reliably generate security assertions.
- Multi-source alignment and verification filtering are essential to training effectiveness.
- The VC-based data source provides high-value supervision signals, making it critical for practical assertion learning.
- Syntax validation alone is insufficient—functional correctness must be ensured via formal tools during data construction.
- Future: Expand benchmark with more SoC-level tasks and integrate simulation-based rewards.
- Explore instruction-tuning and RLHF with Verilog-aware reward functions on the HADA dataset.

Affiliations

[1] The Mike Wiegers Department of Electrical and Computer Engineering, Kansas State University

[2] Department of Electrical and Computer Engineering, University of Maryland, College Park

Funders

Portions of this work were supported by the National Science Foundation (2340949, 2419880).