Constructing Secure Hardware Dataset to Support the Application of Large Language Models in Hardware Domain

Weimin Fu[1] and Xiaolong Guo[1]

Kansas State University, Manhattan, Kansas{weiminf,guoxiaolong}@ksu.edu

1 Abstract

The hardware domain is characterized by its high entry barriers and the need for experienced labor, making using Large Language Models (LLMs) to accelerate processes in this field a popular topic. Most LLM work in the hardware domain uses prompt engineering to solve problems. However, these approaches still heavily rely on experienced professionals, failing to lower the entry barrier effectively and typically translating natural language tutorials into LLM responses. Additionally, such prompt engineering cannot automatically address potential security vulnerabilities in hardware design, requiring users to define and link defects to the hardware design explicitly. LLM server providers can mitigate risk by constructing regular expression filters to avoid risky outputs. Still, in the hardware domain, the risks are often functional rather than textual, making input/output filtering ineffective for risk mitigation. Therefore, security in constructing datasets becomes crucial when training LLMs in the hardware domain.

Training LLMs to become experts within the domain is essential, yet suitable training datasets are often lacking in the hardware field. In general domains, datasets for LLM training are categorized into Supervised Fine-tuning (SFT) datasets, pretraining datasets, and datasets used to train reward models in Reinforcement Learning with Human Feedback (RLHF). The hardware domain, however, lacks open-source data, making it challenging to support pretraining-scale datasets. For SFT datasets, there are typically three categories: Dialog (continuous conversations), Pairs (input-output pairs), and Context (context text with related QA pairs). Such formatted data does not naturally exist in the hardware domain.

Inspired by methods for constructing multimodal datasets, many researchers use synthetic data and manual curation to build hardware datasets. Unlike image modalities, where large-scale open-source datasets are available, similar large-scale datasets do not exist for hardware. Additionally, general-purpose dataset construction methods struggle to eliminate bias and toxicity, and using similar techniques for hardware datasets cannot ensure the absence of hardware design security issues. Alternatively, datasets constructed through random generation tend to be low quality and highly repetitive, failing to provide hardware design safety.

While a manual approach to dataset construction might seem intuitive, the high barriers in the hardware field mean that only experienced hardware engineers can complete this task. This is neither cost-effective nor feasible for building large datasets. Therefore, the proposed method of synthesizing data to support LLMs in the hardware domain must ensure the safety of generated datasets by incorporating safety rules and feature linear cost increases that are unrestricted by the need for domain-specific expertise.

- **Inverse Generation**: This method analyzes and deconstructs existing hardware designs and implementations to extract valuable training data. This approach helps us understand hardware design's underlying logic and architecture and converts them into high-quality and safe datasets for training LLMs.
- **Version Control**: Version control is crucial for tracking and managing code changes. In the hardware domain, we apply version control methods to different versions of hardware designs and implementations, generating a series of progressively evolving datasets. This evolution represents changes in hardware design, such as bug fixes, performance improvements, or the introduction of new features. Using past hardware debugging information as a training set enables the LLM to correct input hardware designs and fix bugs.
- **End-to-End Formal Synthesis**: This method utilizes formal verification techniques to ensure the generated hardware designs meet specific specifications and requirements. We use formal synthesis to automatically generate compliant hardware designs and use these designs as training data. By symbolically representing hardware security information such as CWE and CVE, we ensure that the inputs in the end-to-end hardware generation datasets possess specific security features.

These three modeling methods allow us to automate the expansion of datasets at linear costs, thereby achieving the generalization and scalability of our approach. These methods enable us to train domain-specific LLMs to construct secure hardware datasets and implement effective debugging.

2 Speaker's Bio: Weimin Fu

Weimin Fu is a PhD candidate at Kansas State University in the Department of Electrical and Computer Engineering. He holds a Master's degree from George Washington University and a Bachelor's degree from Zhejiang University. Weimin's research interests focus on domain-specific large language models (LLMs), hardware security, and graph neural networks. Weimin has authored several publications in esteemed conferences and journals.

Weimin's work aims to lower the barriers to entry in the hardware domain by developing advanced LLM techniques and methodologies. His innovative approaches in synthesizing data for hardware LLMs hold promise for the future of hardware debugging and security.