Research Statement

Dong HUANG

Introduction

Generative models, particularly Large Language Models (LLMs), represent a paradigm shift in artificial intelligence, demonstrating remarkable capabilities in creating complex, human-like content. However, their opaque, black-box nature and tendency to produce plausible but incorrect, biased, or inefficient outputs pose significant risks. As these models are integrated into high-stakes domains—from scientific discovery and medical diagnostics to critical infrastructure—ensuring their trustworthiness is not merely an academic exercise but a societal imperative. My research program is dedicated to establishing the algorithmic foundations, theoretical principles, and evaluation frameworks necessary to build trustworthy generative AI.

To ground this mission in a rigorous and impactful context, my work uses automated software engineering as a primary testbed. Code is an ideal medium for studying trustworthiness: it is functional, its correctness can be formally verified, its efficiency is measurable, and its potential for bias is tangible. By tackling challenges in this domain, my research develops generalizable principles for building generative models that are **reliable, fair, and robust** by design. My work is structured around three interconnected thrusts: (1) advancing model reliability from plausible generation to provable correctness, (2) embedding fairness and computational responsibility into model design, and (3) developing rigorous, adversarial evaluation frameworks to ensure accountability.

Research Thrust 1: From Plausible Outputs to Provably Reliable Systems

The core challenge for generative models is reliability. An LLM might generate code that appears correct but contains subtle logical flaws that lead to critical failures. My research develops novel methodologies to bridge this gap between plausibility and correctness, moving towards systems with verifiable guarantees.

Integrating Reasoning with Execution Feedback. My early work showed that improving reliability requires grounding a model's abstract reasoning in concrete, real-world feedback. My **CodeCoT** framework guides an LLM to generate code through a structured process of reasoning and self-correction. Crucially, it executes the generated code against test cases, using the concrete compiler errors as feedback to fix syntactical flaws. To address more complex logical errors, **AgentCoder** [*Stanford AI Index Report 2024*] introduces a multi-agent system where a programmer agent, a test designer, and a test executor collaborate. By separating the roles of code generation and test creation, this system avoids confirmation bias and achieves state-of-the-art correctness (96.3% on HumanEval) with superior efficiency. These systems demonstrate a key principle: reliability emerges from a structured process of generation, execution, and iterative refinement.

Data-Centric Reliability. The reliability of a model is fundamentally tied to its training data. In **Seed-Coder** [*with ByteDance*], we pioneered a model-centric data curation pipeline. Instead of relying on brittle heuristics, we use models to assess and filter a massive 6-trillion-token dataset, training them to recognize high-quality code. By enabling the model to curate its own training data, we create a self-improving loop that enhances the foundational reliability of the resulting open-source models.

Future Direction: Neuro-Symbolic Methods for Provable Guarantees. The ultimate goal is not just code that passes tests, but systems that are provably correct. I plan to develop a new generation of neuro-symbolic techniques that integrate LLMs with formal methods. This involves designing models that co-generate code and formal specifications (e.g., in TLA+ or Dafny). A key innovation will be a tight feedback loop where failures from a symbolic solver or theorem prover are translated into natural language to guide the LLM in refining both the code and its proof. This research path aims to create generative models capable of producing entire software modules with verifiable safety and liveness properties, a critical step towards trustworthy AI in safety-critical domains.

Research Thrust 2: Ensuring Fairness and Responsibility in Generative AI

A trustworthy system must be both fair in its outputs and responsible in its use of resources. My research addresses these two facets of responsible AI, exposing how generative models can perpetuate societal biases and developing methods to create more equitable and efficient systems.

Detecting and Mitigating Algorithmic Bias. Generative models trained on vast internet corpora can inadvertently embed and amplify societal biases. My work [**TOSEM 2025**] was among the first to systematically demonstrate that LLMs inject social biases into generated code, with up to 84% of models exhibiting discriminatory logic. To combat this, I developed a novel testing framework that uses abstract syntax tree analysis to detect when code produces different outcomes based on protected attributes (e.g., gender, race). This work established standard metrics for measuring code bias and showed that while simple prompting offers limited improvement, systematic test-based feedback can dramatically reduce bias, demonstrating a viable path toward fairer generative models.

Promoting Computational Responsibility. Beyond fairness, responsibility includes efficiency. My work reveals that LLM-generated code is often highly inefficient, running 3-13x slower and using up to 43x more memory than human-expert solutions [**EffiBench, NeurIPS 2024**]. Such inefficiency translates into prohibitive financial and environmental costs. My research tackles this by creating a virtuous cycle of measurement and optimization. **EffiLearner** [**NeurIPS 2024**] pioneers a self-optimization framework where an LLM iteratively refines its code using profiling data. **EffiCoder** [**ICML 2025**] moves this upstream, creating models that are efficient by design through performance-aware fine-tuning. Finally, **Afterburner** uses reinforcement learning to discover optimizations beyond existing human knowledge. This line of work establishes efficiency not just as a performance metric, but as a core tenet of responsible and sustainable AI.

Future Direction: Holistic, Multi-Objective Optimization. I plan to develop models that can navigate the complex trade-offs inherent in real-world systems, optimizing for a spectrum of attributes including correctness, efficiency, fairness, security, and maintainability. This will involve a novel reinforcement learning framework where an agent is rewarded based on a holistic utility function derived from static analyzers, profilers, and fairness auditors. The goal is an AI collaborator that can reason about complex design trade-offs, a key capability for human-centric trustworthy AI.

Research Thrust 3: Rigorous and Adversarial Evaluation for Accountability

Trust cannot be claimed; it must be earned through transparent, rigorous, and continuous evaluation. A significant part of my research focuses on building the novel frameworks needed to hold generative models accountable, moving beyond simplistic benchmarks to comprehensive, adversarial testing.

Developing Robust Evaluation Frameworks. Traditional testing methods are ill-suited for the unique failure modes of LLMs. My research [**ICSE 2026**] shows that tests generated from problem specifications are significantly more effective at finding bugs than those derived from the generated code itself, arguing for evaluation based on intent, not implementation. To enable this at scale, I have developed a suite of specialized benchmarks. **EffiBench** and its multi-language successor **EffiBench-X** provide the first rigorous frameworks for evaluating computational efficiency. **DS-Bench** addresses the unique statistical and numerical stability challenges of code for data science. To combat benchmark contamination, I co-developed **CodeArena**, a dynamic platform that prevents memorization by continuously incorporating new problems and using adaptive scoring.

Future Direction: Autonomous Red-Teaming for Continuous Assurance. Static benchmarks are insufficient for ensuring long-term robustness. I will develop adaptive testing systems based on a co-evolutionary ”red teaming” dynamic. I will train an LLM-based agent rewarded specifically for finding inputs, edge cases, or adversarial prompts that cause a code-generating model to fail. This creates a continuous adversarial process where the generative model is constantly hardened against an increasingly sophisticated automated adversary. This approach directly addresses the need for adversarial robustness and aims to automate the discovery of unknown failure modes, ensuring that our trust in these complex systems is well-founded and persistent.