Keyur Parag Joshi

Email: kpjoshi2@illinois.edu | LinkedIn: linkedin.com/in/keyurpjoshi | Website: kpjoshi.com

Education

University of Illinois Urbana-Champaign (UIUC), USA August 2017 – Spring 2024 (Flexible)
PhD in Computer Science
Research focus – analysis of uncertainty in programs and systems; effective application of approximations

Indian Institute of Technology, Hyderabad (IITH), India August 2013 – May 2017
Bachelor of Technology (Honours) in Computer Science and Engineering
Valedictorian

Skills

Programming: Extensive experience – Python, C/C++, Git, Make, Bash. Additional experience – LLVM, ANTLR, PyTorch, Lisp

Teamwork: Extensive experience collaborating on research and engineering projects in teams of 2-12 individuals

Documentation: Experience writing tool documentation for developers and end users

Coursework: Programming Languages, Compilers and Optimizations, Data Structures and Algorithms, Software Engineering, Approximate Algorithms, Software Verification, Machine Learning, etc.

Work Experience

Microsoft Research India – Research Intern May 2022 – August 2022

- Designed and implemented a library for training neural networks such as RNNs exclusively using fixed-point arithmetic
- Achieved accuracy similar to that of floating-point implementations of the same neural network
- Tool published at github.com/KPJoshi/Fixed-Point-RNN-Training
- *Skills used* – Python, PyTorch, Git, compilers, approximation, documentation, etc.

Open-Source Tools

Fixed-Point-RNN-Training: Tool for training RNNs using (almost) exclusively fixed-point arithmetic

- Tool, example, and tutorial available at github.com/KPJoshi/Fixed-Point-RNN-Training
- *Skills used* – Python, PyTorch, Git, compilers, approximation, documentation, etc.

AxProf: Tool for statistical analysis of the precision of approximate algorithms

- Used to successfully find bugs in multiple approximate algorithm implementations
- Tool, examples, and tutorial available at axprof.org
- *Skills used* – Python, Git, compilers, approximation, statistical analysis, etc.

Parallely: Tool for static analysis of quantitative error propagation in parallel programs

- Tool and instructions available at github.com/uiuc-arc/parallely
- *Skills used* – Python, Git, program analysis, compilers, etc.

Current Research

- **Efficient protection of programs against silent data corruptions:** Silent Data Corruptions (SDCs) incorrectly alter program data in an insidious manner. SDCs are increasingly common in large-scale systems due to transistor scaling. We propose a composable analysis of the effects of errors that cause SDCs in programs. Our analysis selects a set of vulnerable instructions to protect against SDCs that maximizes protection while minimizing runtime overhead. When the program is modified, our analysis saves time by only re-analyzing modified program sections.
  Under submission

- **Surrogate models for autonomous vehicle systems:** Modern autonomous vehicles use neural networks and other complex components to perceive the environment and/or to make control decisions. Simulating these systems to ensure they do not violate safety properties is costly. Our two-step approach enables the creation of cheap surrogate models which can be used to check safety properties. Using our surrogate models, we efficiently and precisely estimate the probability of a safety violation in multiple autonomous vehicle scenarios.
  Under submission; preprint available: https://arxiv.org/abs/2208.02232

Publications

- **[Under Submission] FastFlip: Compositional Error Injection Analysis of Programs**
  Keyur Joshi, Rahul Singh, Tommaso Bassetto, Sarita Adve, Sasa Misailovic, Darko Marinov

- **[Under Submission] GAS: Generating Fast and Accurate Surrogate Models for Autonomous Vehicle Systems**
  Keyur Joshi, Chiao Hsieh, Sayan Mitra, Sasa Misailovic
  Preprint available: https://arxiv.org/abs/2208.02232

- **Verifying Controllers with Vision-based Perception Using Safe Approximate Abstractions**
  Chiao Hsieh, Yangge Li, Dawei Sun, **Keyur Joshi**, Sasa Misailovic, Sayan Mitra
  Embedded Software (EMSOFT 2022)

- **Diamont: Dynamic Monitoring of Uncertainty for Distributed Asynchronous Programs**
  Vimuth Fernando, **Keyur Joshi**, Jacob Laurel, Sasa Misailovic
  International Conference on Runtime Verification (RV 2021)

- **ApproxTuner: A Compiler and Runtime System for Adaptive Approximations**
  Hashim Sharif, Maria Kotsifakou, Yifan Zhao, Akash Kothari, Ben Schreiber, Elizabeth Wang, Yasmin Sarita, Nathan Zhao, **Keyur Joshi**, Vikram Adve, Sasa Misailovic, Sarita Adve
  ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP 2021)

- **Aloe: Verifying Reliability of Approximate Programs in the Presence of Recovery Mechanisms**
  Keyur Joshi, Vimuth Fernando, Sasa Misailovic
  IEEE/ACM International Symposium on Code Generation and Optimization (CGO 2020)

- **Statistical Algorithmic Profiling for Randomized Approximate Programs**
  Keyur Joshi, Vimuth Fernando, Sasa Misailovic
  41st ACM/IEEE International Conference on Software Engineering (ICSE 2019)

- **Verifying Safety and Accuracy of Approximate Parallel Programs via Canonical Sequentialization**
  Vimuth Fernando, **Keyur Joshi**, Sasa Misailovic
  34th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2019)

- **ApproxHPVM: A Portable Compiler IR for Accuracy-Aware Optimizations**
  Hashim Sharif, Prakalp Srivastava, Muhammad Huzaifa, Maria Kotsifakou, **Keyur Joshi**, Yasmin Sarita, Nathan Zhao, Vikram S. Adve, Sasa Misailovic, Sarita Adve
  34th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2019)

- **Identifying Optimal Parameters for Randomized Approximate Algorithms**
  Vimuth Fernando, **Keyur Joshi**, Darko Marinov, Sasa Misailovic
  Workshop on Approximate Computing Across the Stack (WAX 2019) (Co-located with PLDI 2019)