# Siyuan Chai

(314) 696-4596 ⋄ siyuanc3@illinois.edu ⋄ https://schai.me

## Education

**University of Illinois**, Urbana Champaign, IL                                                  Start Aug. 2021
Ph.D. Computer Science, Advisor: Prof. Tianyin Xu

**University of Illinois**, Evanston, IL                                                  Graduated June 2021
M.S. Computer Science, B.S. Electrical Engineering, GPA: 4.0/4.0

## Research Publication

1. Ramsey M Wehbe, Jiayue Sheng, Shinjan Dutta, **Siyuan Chai**, Amil Dravid, Semih Barutcu, Yunan Wu, Donald R. Cantrell, Nicholas Xiao, Hatice Savas, Rishi Agrawal, Nishant Parekh, Aggelos K. Katsaggelos. "Deepcovid-xr: An artificial intelligence algorithmto detect covid-19 on chest radiographs trained and tested on a large us clinical dataset." *Radiological Society of North America.* [Online]. Available: https://doi.org/10.1148/radiol.2020203511.

2. Brian Suchy, Souradip Ghosh, Aaron Nelson, Zhen Huang, Drew Kersnar, **Siyuan Chai**, Michael Cuevas, Gaurav Chaudhary, Alex Bernat, Nikos Hardavellas, Simone Campanoni, Peter Dinda. "CARAT CAKE: Replacing Paging via Compiler/Kernel Cooperation." *Submitted for ASPLOS 2022.*

## Experience

Research Assistant, **UIUC Xlab**, Advisor: Prof. Tianyin Xu                                     Aug. 2021 to Present
*Support Linux Kernel for Elastic Cuckoo Page Table*
  · Adapting Linux kernel, primarily the memory management portion, to support Elastic Cuckoo Page Table (ECPT), a hash page table that aims to replace paging by enabling memory-level parallelism
  · Extensively modified memory translation portion of QEMU to simulate ECPT's hardware behavior
  · Exploring the design space like page management, allocation and cache policy in linux running on ECPT

Software Engineering Intern, **Tencent Network Group**                                     June 2021 to Aug. 2021
*Service Driven Network Verification Tool*
  · Contributed to design a scalable network verification that supports quantitative query and covers all data plane with global formal modeling and local simulation
  · Designed easy-to-use geo-based intent language interface for network verification

Research Assistant, **NU Compilers Group**, Advisor: Prof. Simone Campanoni                       Jan. 2021 to July 2021
*Enhance Parallelism by Utilizing Commutative Loop iterations*
  · Coded a LLVM pass to tell the commutativity of <load, ALU operation, store> triplet across loop iterations
  · Extend the idea to develop tools for loop iteration commutativity for further utilization of parallelism

Research Assistant, **NU Parallelism Group**, Advisor: Prof. Peter Dinda                         June 2020 to May 2021
*CARAT CAKE: Replacing Paging via Compiler/Kernel Cooperation*
  · Designed and implemented CARAT CAKE, an allocation level address space which aims to replace virtual memory and paging with protection checks inserted at compile time and allocations tracked in runtime
  · Implemented a competitive paging address space with support for red black tree and splay tree data structures to track VA-PA mapping, Transparent Huge Pages, and PCID; performance measured with Performance Monitoring Counter
  · Designed runtime protection check with address mapping data structures

Research Assistant, **Image & Video Processing Lab**, Advisor: Prof. Aggelos Katsaggelos         June 2019 to July 2021
*DeepCOVID-XR*
  · Designed and implemented a CNN model to flag out positive COVID cases based on patients' chest X-ray images
  · Outperformed radiologists with an accuracy of 85% compared to 76 - 82% and AUC of 0.935 compared to 0.819 - 0.856

## Projects

**C-style Language Compiler**, CS 322 Compiler Construction
  · Created, from scratch, a compiler to translate C-style language to x86_64 assembly
  · Implemented backend optimizations including graph-coloring register allocation, liveness analysis, instruction selection with tiling, control flow graph, and memory access checking

**Middle End Analysis for a C-based API**, CS 323 Code Analysis & Transformation
  · Coded a LLVM pass to reduce calls to a custom C-based API by implementing analysis like reaching-definition, constant propagation and folding, alias analysis for the specific API, function inlining, and dead code elimination

## Skills

**Programming languages**: C/C++, Assembly, Python, Java, Go, JavaScript, MATLAB
**System-level Development**: Unix/Linux, QEMU, Docker, GDB, Make, Linker, LLVM, OpenMP
**Artificial Intelligence**: CUDA, PyTorch, Tensorflow, Keras, Image Processing, Computer Vision,
**Web Development**: HTML, CSS, Flask, Django, React