

XIAORUI HUANG

Always Fascinated 

 Preferred Name: Richard  hxr.richard@gmail.com
 +1 (289) 772-8682  Toronto, Canada
 xiaorui-richard-huang  Xiaorui-Huang

EXPERIENCE

Low Power AI Machine Learning Engineer

Qualcomm

 May 2023 – Aug 2023  Markham, ON, Canada

- Led efforts on **Neural Architecture Search (NAS)** and model compression within the **Edge AI R&D** team.
- Developed a NAS framework, leveraging Qualcomm's patented NAS techniques, to optimize **arbitrary models** for **any profiled hardware**, harnessing Pytorch's **torch.fx** extensively.
- Streamlined the NAS workflow for incoming client models, slashing **engineering time** by **80%**.
- Achieved a **50% reduction** in **model size** and a **60% drop in inference latency** without compromising accuracy across benchmark models.
- Engaged in lab paper-reading sessions focused on cutting-edge model compression research, particularly in **Quantization** and **efficient LLM**.

RPA Backend Developer

IBM

 May 2022 – Apr 2023  Markham, ON, Canada


- Worked on backend development for IBM's Robotics Process Automation (RPA) platform, written in C# **OOP**.
- Augmented IBM RPA's WAL programming language, introducing a reflection feature resembling Java and C#.
- Collaborated with cross-functional teams, achieving a **15% reduction** in customer issues and defects per release.
- Employed **agile methodologies**, showed both independent and collaborative competencies in a hybrid environment.
- Articulated and presented solution strategies to RPA's senior architects and product teams.

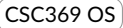
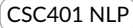
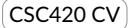
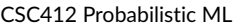
EDUCATION

University of Toronto

Honors BSc. in Computer Science

 Sep 2019 – Jun 2024 (*includes one year co-op*)

- CSC413 **Deep Learning** (96%) – GNN, Transformers, CNN, RNN, GAN, VAE, RL, Model Tuning techniques
- CSC317 **Computer Graphics** (97%) – Ray Tracing, Mass Spring Systems, BVH, Meshes, Kinematics, OpenGL Shaders in C++ using **Eigen** and **libgl**
- CSC367 **Parallel Computing** (83%) – CUDA Arch & Reduction Algo, Parallel Arch & Algo, threading & OpenMP, Distributed Computing w/ MPI, Cloud Computing
- ECE568 **Computer Security** (83%) – Buffer Overflow & Control Hijacking, Cache Side-Channel Attacks, Network Security, Cryptography, Web Security

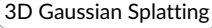



RESEARCH

Machine Learning Researcher

embARC Research Group



 Jan 2024 - Now  Xiaorui-Huang/DISORF

- Researched and Developed **DISORF**, a **real-time Gaussian Splatting & NeRF** framework for online 3D reconstruction and visualization of scenes captured by resource-constrained mobile robots and edge devices.
- Proposed a novel shifted exponential frame sampling method to address the degradation in rendering quality caused by naive image sampling during online training.
- Paper is under review for RA-L and available on arXiv

PROJECTS

CUDA Ray Tracing



 Nov 2023  Xiaorui-Huang/cuda-ray-tracing

- Implemented a **CUDA** ray tracer with **BVH** acceleration structure, with Blinn-Phong shading.
- Achieved **real-time** ray-tracing of **30 FPS** and **2000x Speedup** on RTX3060-Ti compared to CPU.
- Designed framework for scene construction, allowing for rendering of new scenes via config and existing assets.

Woodoku Learn

Reinforcement Learning Model



 Jul 2022  EdwardHaoranLee/WoodokuLearn

- Replicated the mobile game Woodoku for the terminal using Python, enabling both human and AI gameplay through dedicated environment APIs.
- Employed Q-Learning, a **Reinforcement Learning** approach with Pytorch, targeting top scores on the Woodoku leaderboard.

Doodle Jumps

Pure Assembly Game Dev










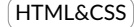

 Dev 2021  Xiaorui-Huang/doodle-jump

- Created a Minecraft-themed version of the **Doodle Jump** game using **MIPS Assembly**.
- Implemented game logic for player movement, collision detection, and scoring, key controls & graphic design.

SKILLS

Programming Languages

Skills, Frameworks & Development Environments