

Skills

Languages: Python, Java, JavaScript, C/C++, TypeScript, Kotlin, C#, HTML, CSS, QML, SQL, Haxe, Lua, Bash, Emacs Lisp.
Frameworks: PyTorch, CUDA, Angular 2, Node.js Express, Ionic, React, Bazel, Vue.js, TensorFlow, OpenGL, Qt Quick.
Tools: Git, Perforce, Emacs, Vim, Linux, Unity3D, G-Suite, Microsoft Office.

Education

University of Toronto — Computer Science Specialist — Honors Bachelor of Science with High Distinction **Class of 2021**
CGPA: 3.98/4.0, Dean's List Scholar, President's Entrance Scholarship, Woodsworth College Scholarship, Mcnab Scholarship
Upper-level Courses: Neural Networks and Machine Learning Algorithm Design & Complexity Software Engineering
Natural Language Computing Introduction to Visual Computing Distributed Systems

Publication

- **Learning Deformable Tetrahedral Meshes for 3D Reconstruction** May 2020
 - Jun Gao, Wenzheng Chen, **Tommy Xiang**, Alec Jacobson, Morgan McGuire, Sanja Fidler
 - Accepted to Conference on Neural Information Processing Systems (**NeurIPS**) 2020

Professional Experience

- **Nuro, Software Engineer**, Perception August 2021 - Present
 - Design and develop novel onboard perception algorithms for autonomous delivery robotics as well as evaluation pipelines.
 - Technologies: **C++**, **CUDA**, **Python**, **TensorFlow**
- **NVIDIA, Deep Learning Research Intern**, Toronto Artificial Intelligence Lab September 2019 - April 2020
 - Supervised by **Sanja Fidler**, co-founder of the Vector Institute, Director of AI at Nvidia, and Canada CIFAR AI Chair.
 - Led research project for 3D asset generation from 2D images with minimal 3D supervision, using techniques including convolutional neural network, differentiable renderer, generative adversarial network, and Learned Perceptual Image Patch Similarity (LPIPS) loss
 - Technologies: **PyTorch**, **Python**, **CUDA**, **OpenCV**, **NVIDIA GPU Cloud**, **Docker**, **CNN**, **GAN**
- **Google LLC, Software Engineering Intern** May 2019 - August 2019, May 2020 - August 2020
 - Designed and implemented **full-stack** data quality alert system for internal intelligence visualization tool used by more than **30,000** Google employees
 - Deployed into production and immediately used to notify users of critical data anomalies, saving significant cost from **25-35** data errors annually
 - Technologies: **JavaScript**, **TypeScript**, **HTML/CSS**, **SQL**, **Protobuf**, **Angular 2**, **Java**, **Node.js**, **Python**
- **Google LLC, Software Engineering Intern** May 2018 - August 2018
 - Implemented **MVC structured**, accessibility-friendly user interface for layout formatting on Google Docs Android, reaching over **100 million users**

Technical Projects

- **MusicFlow - Music Generation with Flow-based Generative Models** Python, PyTorch
 - Researched ML-based music generation by applying **Continuous Normalizing Flow (CNF)** methods such as FFJORD on the Lakh MIDI dataset
 - Designed and implemented MIDI quantization encoding using 2-channel images, formulated model design and conducted experiments
 - Achieved generation of high-quality musical samples with the integration of Flow and **Variational Auto Encoder (VAE)**.
- **Distributed Key-Value Database** Java, Apache ZooKeeper, Consistent Hashing
 - Developed a replicated, scalable, distributed, transactional key-value storage server and client application in Java
 - Implemented consistent hashing, fault recovery with **ZooKeeper**, and eventual consistency with incremental continuous replication
 - Designed and implemented ACID transaction support with client interpreter for a custom transactional query language
 - Achieved less than **3ms** query latency with 20 concurrent servers and 20 concurrent clients
- **Cellular - 2D Procedurally-Generated Game** C/C++, OpenGL, Lua, SDL 2, Cpp-Taskflow, Fruit, Entt
 - Designed and developed procedurally-generated 2D action & adventure game, achieved **1st place** in UofT Game-Making Deathmatch 2017
 - Implemented a high-performance, ECS-based, multi-threaded game engine in **C/C++** and **OpenGL**. Designed concurrent renderer with fully automatic sprite batching. Designed lock-free task graph system allowing near-maximum hardware utilization using topological sorting and RLF vertex coloring heuristic. Utilized template metaprogramming to auto-detect race-condition. Achieved simulation of up to **200000+** animated entities at **60fps**
- **Emacs Client for TabNine** (github.com/TommyX12/company-tabnine) Emacs Lisp
 - Developed Emacs client for the code-completion system **TabNine**, which displays intelligent and relevant completions using deep learning
 - Implemented low-latency sub-process communication for candidate fetching, achieving **5ms** overhead per keystroke
 - Received more than **550 stars** on GitHub, and more than **15000** downloads on Emacs package archive

Awards and Contributions

- 1st Place - Bloomberg Codecon UofT 2017
- 2nd Place - Microsoft Code Competition UofT 2017
 - Solved one of the hardest problem
- 2nd Best Accuracy - (National) USC Competition 2017
 - Developed geo-tagging tool for drone mission
- Silver Medalist - Top 5 in Canada - (National) Canadian Computing Olympiad ([Link](#)) 2016
- Co-President of Game Design and Development Club 2017 - 2018
- 1st - UofT Game-Making Deathmatch 2017
 - Best Overall and Best Technical Achievement Award
- 3rd Place - Big Data Challenge 2016
 - Analyzed and visualized open data using Python
 - Journal Published on STEM Fellowship ([Link](#))
- Vision Subdivision Lead of University of Toronto Aerospace Team: Aerial Robotics division 2017 - 2018