Google Scholar
github.com/YueFei0403
yuefei9943@gmail.com
(+1) 647-866-7311

Yue Fei she/her

Communication Systems & Digital Design & Computer Architecture

Education

University of Toronto
Master of Engineering, Electrical Engineering — *Sep. 2022 – Jun. 2024*
- Emphasis: Communications
- Advisor: Prof. Raviraj Adve
- MEng Thesis: "Pilot Training - Angle of Arrival and Channel Estimation in 5G Network"

University of Toronto
Bachelor of Applied Science, Electrical Engineering — *Sep. 2017 – Jun. 2022*
- Capstone Project: Convolutional Neural Network (CNN) NPU Overlay (MobileNetV1) for FPGA (Intel Stratix 10 NX 2100)
- Advisors: Prof. Vaughn Betz and Andrew Boutros

Publications (Peer-Reviewed Conference)

1. Arash Ahmadian, Louis S.P. Liu, **Yue Fei,** Konstantinos N. Plataniotis; Mahdi S. Hosseini. Pseudo-Inverted Bottleneck Convolution for Darts Search Space. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
2. Abnash Bassi, **Yue Fei**, Gilead Posluns, Mark C. Jeffrey. Optimized Priority Scheduling for Faster Scalable Belief Propagation. *The Association for the Advancement of Artificial Intelligence (AAAI) [In Submission]*, 2026.

Awards, Honors and Certificates

Edward S. Rogers Sr. Department Betz Entrance Scholarship ($5,000) *2017*
Dean's Honour List *2017 Fall, 2018 Winter, 2018 Fall, 2021 Fall, & 2022 Winter*
Certificate in Engineering Business *Jun. 2022*

Invited Talks

Panel: *Demystifying Machine Learning* *Mar. 2025*

Talk: From Channels to States — Machine Learning in the Language of Communication and Control
QWomen San Diego — Qualcomm Internal Panel Discussion

Technical Skills

Programming: Python, Pandas, C/C++, MATLAB, Julia, Arm Assembly, Verilog/SystemVerilog, Unix/Linux Shell, Perl
Data Processing & Automation: CI/CD (Jenkins) pipelines; Makefile
ML & Optimization: Scikit-learn, PyTorch, Stable-Baselines3 & Gymnasium for Reinforcement Learning (RL), GRU-RNN, MLP, Attention mechanism & visualization [Code], Q-Learning (Reinforcement Learning), Convex Optimization (fractional & quadratic-transform), Sampling-based Source Coding
Parallel Programming: Multithreading (OpenMP-style loops, SIMD), **multi-queue scheduling**
Tools & Environments: Git, Vim/GVim, SimpleScalar simulators

Industry Experience

Qualcomm – Design Verification Engineer *Jun. 2024 – Jul. 2025*
Markham, Canada

- Verified **UWB** receiver path and improved startup performance by reducing LNA charging delay **90%** (20ns → 2ns)."
- Validated **WLAN CP-PLL synthesizer** loop and built **UVM**-compatible test plans spanning 500+ channel indices, ensuring robust coverage across 2G, 5G, and emerging 5G alternative bands.
- Developed **multi-head GRU-based RNN** for receiver gain line-up optimization, where each head learns one analog block (LNA, GM, TIA, BQ, PGA). Transformed a complex combinatorial tuning problem into a scalable learning-based approach, easing designer effort.
- Built a physics-inspired **MLP** (*customized activation function* + *Pre-Normalization* + *Post-Normalization*) that predicts **VCO** capacitance from control inputs, removing the need for RF/analog designers to manually tune capacitors for 1000+ frequency targets.

Alphawave Semi - Digital Verification Engineer *May 2020 – Jun. 2021*
Toronto, Canada

- Developed **UVM** testbenches to verify **SerDes** (clocking, datapath, **SRAM**), expanding functional coverage across **50+ scenarios**.
- Enhanced CI/CD automation to support **15× growth** in regression testing (scaling from **4 to 60+ projects**), improving efficiency and reliability as the company expanded.

Selected Projects

Highlighted academic, research, and technical projects spanning ML, optimization, signal processing and hardware designs.

Digital Design & Computer Architecture
RTL design, parallel computing and memory-coherence protocols

Undergraduate Capstone Project (Team of 4) *Sep. 2021 – Apr. 2022*

- Collaborated to extend Microsoft Brainwave NPU on Intel Stratix 10 NX FPGA by replacing its matrix–vector unit with a **custom convolution unit (scheduler + tiles + accumulator)**, achieving **4.3×** speedup and **76.6 GOP/s** on MobileNetV1 inference.
- Designed a **pipelined accumulator** enabling 330 MHz operation; implemented three-stage **reduction-tree addition** and **opcode-based accumulation modes**, verified correct synchronization and 450 MHz Fmax in Quartus.

Computer Architecture Coursework (Team of 2) *Sep. 2021 – Dec. 2021*

- Collaborated to model **hazard detection** in a **MIPS**-based 5-stage pipelined CPU, implemented a **perceptron-based branch predictor**, **Tomasulo out-of-order execution**[Code], **Bouquet prefetcher**, and **MSI-directory cache-coherence protocol**; these experiences deepened my understanding of **pipeline parallelism for distributed LLM training** and how **speculative execution** in CPUs parallels **LLM speculative decoding** for faster inference.

Parallel Beamforming & Cache-Coherence Foundations for Scalable Compute *Jan. 2024 – May. 2024*

- Accelerated medical-imaging by **7×** (17 s → 2.5 s) via **16-thread data-parallel** ultrasound beamforming with SIMD intrinsics and memory optimizations (`restrict`, `single-write`); validated correctness (RMS error < 1e-16) and scalability (1–16 threads) — a paradigm relevant to **data-parallel LLM training**. [Code]
- Designed and verified a **3-hop directory cache-coherence protocol** (MSI/MESI) in Murϕ; optimized with **Exclusive** (**E**) state to eliminate bus transactions. [Code]

Coding Theory
Error-correcting codes for reliable communications

Graph-Based Error-Correcting Codes *Sep. 2023 – Dec. 2023*

- Implemented **LDPC**, fountain/LT, and **Polar encoders/decoders** over binary-erasure channel; developed custom simulators in Julia and MATLAB. [Code]

Convolutional Codes & Viterbi Decoder *Jan. 2023 – May. 2023*

- Built a rate-½ **convolutional encoder** and **Viterbi decoder** in MATLAB with custom trellis structures; validated against built-in tools and demonstrated 2-bit-error correction on a noisy BSC channel. [Code]

BCH Codes & Extended Euclidean Algorithm *Jan. 2023 – May. 2023*

- Developed a **binary BCH encoder/decoder** over $GF(2^m)$, applying the **extended Euclidean algorithm** to solve the key equation for syndrome decoding and implementing Chien search for error location. [Code]

Signal Processing
Estimation and detection for modern wireless systems

Master of Engineering Project *2024*

- Compared **Linear Minimum Mean Square Error** (**LMMSE**) approach and **Matrix-Pencil Method** (**MPM**) for uplink channel estimation in **multicell MIMO-OFDM systems** with four distributed units (DUs) arranged in a square of radius 100 m, using a 3GPP-compliant MATLAB model.
- Showed that with **pilot training** in time or frequency domain, MPM yields higher angle-of-arrival accuracy than DFT-based and MMSE methods even with only eight receiver antennas.

LTE Signal Processing *Jan. 2022 – Apr. 2022*

- Processed captured LTE signals for **time/frequency sync**, **OFDM demodulation,** and **channel equalization** in MATLAB; resampled 40 MHz front-end data to 30.72 MHz LTE rate and validated **PSS/SSS** detection.

ML & Optimization
Modeling and optimization for wireless networks and semantic coding

Convex & Fractional Programming for Multi-Cell MIMO Beamforming *Sep. 2023 – Dec. 2023*

- Applied **fractional-programming** and **quadratic-transform** optimization to improve multi-cell **MIMO beamforming,** boosting convergence and power-constrained sum-rate performance.

Sampling-Based Semantic Source Coding (One-Shot Info Theory) *Jan. 2023 – May. 2023*

- Implemented **Poisson functional representation, rejection sampling, importance sampling,** and hybrid Poisson + dithered-quantization for 6G semantic source-coding in MATLAB. [Code]

Transformer & Embedding Visualization (Research Assistant) *Jul. 2021 – Sep. 2021*

- Explored **RNN-based Transformer models** and **Attention mechanisms** for NLP tasks .
- Applied **PCA-based embedding visualization**—as used in **GloVe**—to project high-dimensional embeddings into **2D/3D space** using **Python (NumPy, Matplotlib)**, enabling intuitive inspection of semantic clusters.