

# YUE FEI SHE/HER

Machine-Learning & Hardware-Aware Optimization • Parallel Programming/HPC • Data Pipelines & CI/CD

EDUCATION	<p><b>University of Toronto</b>  <i>Master of Engineering, Electrical Engineering</i> Sep. 2022 – Jun. 2024</p> <ul style="list-style-type: none"> <li>• Emphasis: Communications</li> <li>• Advisor: Prof. Raviraj Adve</li> <li>• MEng Thesis: “Pilot Training - Angle of Arrival and Channel Estimation in 5G Network”</li> </ul> <p><b>University of Toronto</b>  <i>Bachelor of Applied Science, Electrical Engineering</i> Sep. 2017 – Jun. 2022</p> <ul style="list-style-type: none"> <li>• Capstone Project: Convolutional neural network NPU Overlay (MobileNetV1) for FPGA (Intel Stratix 10 NX 2100)</li> <li>• Advisors: Prof. Vaughn Betz and Andrew Boutros</li> </ul>
PUBLICATIONS (PEER-REVIEWED CONFERENCE)	<ol style="list-style-type: none"> <li>1. Arash Ahmadian, Louis S.P. Liu, <b>Yue Fei</b>, Konstantinos N. Plataniotis; Mahdi S. Hosseini. Pseudo-Inverted Bottleneck Convolution for Darts Search Space. <i>IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)</i>, 2023.</li> <li>2. Abnash Bassi, <b>Yue Fei</b>, Gilead Posluns, Mark C. Jeffrey. Optimized Priority Scheduling for Faster Scalable Belief Propagation. <i>The Association for the Advancement of Artificial Intelligence (AAAI) [In Submission]</i>, 2026.</li> </ol>
AWARDS AND HONORS	<p><b>Dean’s Honour List</b> 2017 Fall, 2018 Winter, 2018 Fall, 2021 Fall, &amp; 2022 Winter  <b>Edward S. Rogers Sr. Department Betz Entrance Scholarship (\$5,000)</b> 2017</p>
CERTIFICATE	<p><b>Certificate in Engineering Business</b> Jun. 2022</p>
INVITED TALKS	<p><b>Panel: Demystifying Machine Learning</b> Mar. 2025</p> <p><i>Talk: From Channels to States — Machine Learning in the Language of Communication and Control</i>  <i>Q Women San Diego — Qualcomm Internal Panel Discussion</i></p>
TECHNICAL SKILLS	<p><b>Programming:</b> Python (Pandas for SQL-style data joins), C/C++, MATLAB, Julia, Arm Assembly, Perl</p> <p><b>Data Processing &amp; Automation:</b> CI/CD (Jenkins) pipelines for large IC/IP regressions — enabled same-day dashboards vs. 1–2-day manual; Makefile-based build/test flows</p> <p><b>ML &amp; Optimization:</b> PyTorch, GRU-RNN, MLP, Attention mechanism, Q-Learning (Reinforcement Learning), Convex Optimization (fractional &amp; quadratic-transform), Sampling-based Source Coding</p> <p><b>Parallel Programming:</b> Multithreading (OpenMP-style loops, SIMD), custom thread mgmt, multi-queue scheduling for scalable workloads</p> <p><b>Tools &amp; Environments:</b> Git, Linux/Unix shells, Vim/GVim, MurΦ Model Checker, SimpleScalar simulators</p>

INDUSTRY EXPERIENCE	<b>Qualcomm – Design Verification Engineer</b> <i>Markham, Canada</i>	<i>Jun. 2024 – Jul. 2025</i>
	<ul style="list-style-type: none"> <li>• Verified <b>UWB</b> receiver path and improved startup performance by reducing LNA charging delay <b>90%</b> (20ns → 2ns)."</li> <li>• Validated <b>WLAN CP-PLL synthesizer</b> loop and built <b>UVM</b>-compatible test plans spanning 500+ channel indices, ensuring robust coverage across 2G, 5G, and emerging 5G alternative bands.</li> <li>• Developed <b>multi-head GRU-based RNN</b> 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.</li> <li>• Built a physics-inspired <b>MLP</b> that predicts <b>VCO</b> capacitance from control inputs, removing the need for RF/analog designers to manually tune capacitors for 1000+ frequency targets.</li> </ul>	
	<b>Alphawave Semi - Digital Verification Engineer</b> <i>Toronto, Canada</i>	<i>May 2020 – Jun. 2021</i>
	<ul style="list-style-type: none"> <li>• Developed <b>UVM</b> testbenches to verify <b>SerDes</b> (clocking, datapath, <b>SRAM</b>), expanding functional coverage across <b>50+ scenarios</b>.</li> <li>• Enhanced <b>CI/CD</b> automation to support <b>15× growth</b> in regression testing (scaling from <b>4 to 60+ projects</b>), improving efficiency and reliability as the company expanded.</li> </ul>	
SELECTED PROJECTS	Highlighted academic, research, and technical projects spanning ML, optimization, signal processing, HPC, and architecture.	
<b>ML &amp; Optimization</b> Modeling and optimization for wireless networks and semantic coding	<b>Convex &amp; Fractional Programming for Multi-Cell MIMO Beamforming</b>	<i>Sep. 2023 – Dec. 2023</i>
	<ul style="list-style-type: none"> <li>• Applied <b>fractional-programming</b> and <b>quadratic-transform</b> optimization to improve multi-cell <b>MIMO beamforming</b>, boosting convergence and power-constrained sum-rate performance.</li> </ul>	
	<b>Sampling-Based Semantic Source Coding (One-Shot Info Theory)</b>	<i>Jan. 2023 – May. 2023</i>
	<ul style="list-style-type: none"> <li>• Implemented <b>Poisson functional representation</b>, <b>rejection sampling</b>, <b>importance sampling</b>, and hybrid Poisson + dithered-quantization for 6G semantic source-coding in MATLAB.</li> </ul>	
	<b>Transformer &amp; Embedding Visualization (Research Assistant)</b>	<i>Jul. 2021 – Sep. 2021</i>
	<ul style="list-style-type: none"> <li>• Explored <b>RNN-based Transformer models</b> and <b>Attention mechanisms</b> for NLP tasks.</li> <li>• Applied <b>PCA-based embedding visualization</b>—as used in <b>GloVe</b>—to project high-dimensional embeddings into <b>2D/3D space</b> using <b>Python (NumPy, Matplotlib)</b>, enabling intuitive inspection of semantic clusters.</li> </ul>	
<b>Signal Processing</b> Estimation and detection for modern wireless systems	<b>Angle of Arrival (AoA) &amp; Channel Estimation</b>	<i>Jan. 2023 – May. 2023</i>
	<ul style="list-style-type: none"> <li>• Implemented <b>MUSIC</b>, <b>DFT</b>, and <b>Matrix-Pencil</b> eigenvalue methods in MATLAB for AoA estimation under large-scale and Rayleigh-fading channels; <b>Matrix-Pencil</b> delivered <b>≈2 dB gain in low-SNR regimes with 16-antenna arrays</b>, outperforming MUSIC and DFT for highly-correlated signals.</li> </ul>	
	<b>LTE Signal Processing</b>	<i>Jan. 2024 – Apr. 2024</i>
	<ul style="list-style-type: none"> <li>• Processed captured LTE signals for <b>time/frequency sync</b>, <b>OFDM demodulation</b>, and <b>pilot-power analysis</b> in MATLAB; resampled 40 MHz front-end data to 30.72 MHz LTE rate and validated <b>PSS/SSS</b> detection.</li> </ul>	

## Parallel & HPC

Multi-threaded  
acceleration and  
memory-coherence  
design

## Parallel Beamforming & Cache-Coherence Foundations for Scalable Compute

*Jan. 2024 – May. 2024*

- Accelerated medical-imaging by  $7\times$  ( $17\text{ s} \rightarrow 2.5\text{ s}$ ) via **16-thread data-parallel** ultrasound beamforming with SIMD intrinsics and memory optimizations (**restrict**, **single-write**); validated correctness (RMS error  $< 1\text{e-}16$ ) and scalability (1–16 threads) — a paradigm relevant to **data-parallel LLM training**.
- Designed and verified a **3-hop directory cache-coherence protocol** (MSI/MESI) in Mur $\alpha$ ; optimized with **Exclusive (E)** state to eliminate bus transactions (**0 MB** overhead vs 80 MB baseline), formalized **7 invariants** (e.g., single-writer ownership), and handled FSM edge-case scenarios for large-scale shared-memory systems.

## 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.

## Convolutional Codes & Viterbi Decoder

*Jan. 2023 – May. 2023*

- Built a rate- $\frac{1}{2}$  **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.

## Computer Architecture

Speculation mechanisms  
relevant to LLM  
decoding

## Computer Architecture Coursework

*Sep. 2021 – Dec. 2021*

- Developed a **5-stage pipelined CPU** with **hazard detection & forwarding**, implemented a **perceptron-based branch predictor**, **Tomasulo out-of-order execution**, **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.