

# Yuang Xu

📍 Los Angeles, CA, USA    ✉ itwaix@outlook.com    🌐 github.com/yuangxu    🌐 yuangxu.com

## Education

---

<b>University of California, Merced</b> , Merced, CA <i>Incoming PhD in Computer Science</i>	<i>Starting Sep. 2026</i>
<b>University of Southern California (USC)</b> , Los Angeles, CA <i>Master of Science in Computer Science</i>	<i>Jan. 2024 – Dec. 2025</i>
<b>Shandong University</b> , China <i>Bachelor of Engineering in Computer Science and Technology</i>	<i>Sep. 2018 – Jun. 2022</i>

## Research Interests

---

Distributed ML training & inference systems on heterogeneous memory architectures (e.g., CXL-based tiered memory); AI compiler and operator optimization for LLM inference serving (e.g., vLLM).

## Publications & Patents

- 
- [1] Yuan Tian\*, **Yuang Xu\***, Jun Zhou. *Underwater Image Enhancement Method Based on Feature Fusion Neural Network*. **IEEE Access**, 09/2022. (\*Co-first author)
  - [2] Bin Zhang, **Yuang Xu**, Wenrui Luo. *A Signal Acquisition Optimization Method based on RAPID Tomography*. **CN109946384A**, 06/2019. (Patent)

## Services

---

**Artifact Evaluation Committee Member:** MLSys 2026, EuroSys 2026

## Research & Systems Projects

- 
- TinyLlama Acceleration using OpenAI Triton** *May. 2025 – Dec. 2025*
- Engineered a custom fused SwiGLU kernel in OpenAI Triton to integrate Linear, gating, and SiLU operations, significantly reducing global memory traffic and kernel launch overhead in MLP layers.
  - Profiled and optimized memory access patterns and block-level parallelism using NVIDIA Nsight Systems and Nsight Compute to eliminate architectural bottlenecks.
  - Achieved up to a 20% reduction in MLP latency on an RTX 3060 and a significantly lower GPU memory footprint compared to the PyTorch eager mode implementation.
- Vision Transformer (ViT) Operator Optimization via TVM TensorIR** *Aug. 2025 – Oct. 2025*
- Executed end-to-end model conversion from PyTorch to TVM via ONNX, identifying and targeting performance-critical subgraphs in Attention and Patch Embedding layers.
  - Optimized memory locality and execution efficiency by combining MetaSchedule auto-tuning with manual TensorIR scheduling, including tiling, loop reordering, vectorization, and cache read/write reuse.
  - Benchmarked optimized kernels on an NVIDIA RTX 3070 Ti, achieving superior inference performance and lower memory overhead compared to the PyTorch eager mode baseline.
- Mini-PyTorch: C++ Deep Learning Framework** *Jan. 2025 – May. 2025*
- Implemented lightweight deep learning framework in C++ to understand the internal mechanisms of systems like PyTorch.
  - Designed a Tensor class with proper memory management (using smart pointers) and stride handling to support n-dimensional array operations. And developed a basic Autograd engine that constructs a dynamic computational graph (DAG) and performs reverse-mode automatic differentiation (Backpropagation).
  - Implemented an operator dispatcher to separate the frontend interface from the execution backend.

## Work Experience

- 
- 4Paradigm (AI Unicorn)**, Shanghai, China – *Algorithm Engineer Intern* *Apr. 2025 – Sep. 2025*
- Built a multimodal content generation pipeline integrating ASR (Whisper), LLM (Qwen), and TTS models.
  - Optimized the inference backend service, implementing request batching and asynchronous processing, which reduced average response latency by 20% under high concurrency.
- TP-LINK**, Hangzhou, China – *Software Engineer* *Jul. 2022 – Jul. 2023*
- Designed a high-performance database connection pool for cloud services, utilizing lock-free data structures to handle high concurrency; contributed bug fixes to the open-source Kona JDK.

## Skills

---

**Languages:** C/C++, CUDA, Python, PTX Assembly  
**AI Systems:** OpenAI Triton, Apache TVM, PyTorch Internals, CUDA Optimization, TPU  
**Tools:** Docker, Git, Linux, GDB, Nsight Systems