# Boyao Wang

Email: bryanw2@cs.cmu.edu | Tel: (217)974-0197
Github | Google Scholar | LinkedIn

## EDUCATION

**Carnegie Mellon University** **Pittsburgh, PA**
*Master of Science* in **Machine Learning** | GPA: **4.00/4.00** *Dec. 2026*
• Relevant Coursework: Deep Learning Systems, Advanced Intro. to Machine Learning, Probability & Mathematical Statistics

**University of Illinois Urbana-Champaign** **Urbana, IL**
*Bachelor of Science* in **Computer Engineering** (Dual Degree) | **Highest Honors** | GPA: **3.94/4.00** *Jun. 2025*

**Zhejiang University** **Hangzhou, China**
*Bachelor of Engineering* in **Electronic and Computer Engineering** (Dual Degree) | GPA: **3.97/4.00** *Jun. 2025*

## SKILLS

**Core Skills:** ML/DL System Optimization, Applied LLMs (RAG, Fine-tuning), Feature Engineering, Full-stack Development
**Programming Languages:** Python (NumPy, Pandas, Scikit-learn, LightGBM, Optuna), C++, SQL, C, Java, R, x86 Assembly
**Tools & Platforms:** PyTorch, WandB, CUDA, DeepSpeed, SGLang, Git, Linux, Bash, CI/CD, PostgreSQL, Docker, AWS

## RESEARCH EXPERIENCE

**GlassTorch: Interpretable and Computation-Efficient Training via Dynamic Freezing** **Pittsburgh, PA**
**Project Lead** *Aug. 2025 - Dec. 2025*
- Spearheaded a 3-person team to build GlassTorch, a lightweight PyTorch-style training framework with custom autograd and pluggable NumPy / C++ / CUDA backends, enabling per-layer statistics tracking for training explainability
- Identified module-level training stability via per-module delta and norm tracking, observing that large portions of networks converge early and exhibit minimal parameter and statistic changes, including variance and L2 norms
- Researched adaptive, reversible auto-freezing policies that skip backprop on stable modules, reducing end-to-end training time by 26.7% (1331.7s $\rightarrow$ 976.0s) with $<$0.5% absolute test-accuracy drop on CIFAR-10 CNNs (30 epochs)

**Adapt-Pruner: Adaptive and Structured Pruning for Efficient LLMs** **Urbana, IL | Remote**
**Project Lead | Co-First Author** *Jun. 2024 - Feb. 2025*
- Engineered an automated model compression toolkit for LLMs that systematically prunes decoder layers by calculating tensor distances, assigning adaptive sparsity levels to optimize inference performance
- Evaluated performance on LLaMA-3.1-8B, Qwen2.5-7B, and Gemma-2-9B, demonstrated 5% improvement in common sense benchmark scores over state-of-the-art methods including SliceGPT at 50% sparsity
- Compressed LLaMA-3.2-3B to 1.3B and post-trained on 0.05B tokens, outperforming TinyLlama-1.1 pre-trained on 10B tokens (200x less), illustrating the effectiveness of pruning combined with post-training for generating small language models

**RL-Pruner: Structured Pruning Using Reinforcement Learning for CNNs** **Urbana, IL**
**Project Lead | First Author** *Jan. 2024 - Aug. 2024*
- Developed a model optimization framework (~2,000 lines of PyTorch code) that automates structured pruning for CNNs using Q-learning, with accuracy of pruned models as reward, refining layer-wise pruning distributions to minimize performance loss
- Applied response-based Knowledge Distillation to post-train compressed models, using original model as teacher to transfer extracted representations and efficiently recover accuracy lost during compression
- Experimented on ResNet, GoogLeNet and MobileNet, achieving 81% parameter reduction on VGG-19 (CIFAR-100) within a 1% performance drop

## PROFESSIONAL EXPERIENCE

**Tencent** **Hangzhou, China**
**Applied Research Intern** *Feb. 2025 - Jul. 2025*
- Architected and delivered an end-to-end generative AI pipeline for 3D scene generation, featuring a scalable Python backend service and integrating it with an Unreal Engine front-end, while refining features based on other teams' feedback
- Implemented an LLM agent leveraging prompt engineering and Retrieval-Augmented Generation to generate 3D layouts by selecting furniture from an existing database, and calling custom placement functions
- Optimized the backend service by fine-tuning the Qwen3-1.7B model via SFT and rule-based reward model RL training to replace the Claude-3.7 API in the Python backend, accelerating scene generation by ~9x (4.6s to 0.5s/request)

## PROJECTS

**ML4Investment** | Independent *Feb. 2025 - Current*
- Designed and built a comprehensive machine learning framework for U.S. stock price movement prediction, leveraging LightGBM for next-day price change forecasting, providing a cost-effective alternative to LLM-based solutions
- Implemented an advanced, end-to-end ML pipeline integrating a multi-stage optimization strategy to optimize sampling weights for monthly data, select optimal features, and tune hyperparameters via multi-objective Optuna and time-series cross-validation
- Engineered a rigorous backtesting system to simulate investment strategies and evaluate performance using composite returns against optimal benchmarks, producing a 138% actual gain over an 84-trading-day period