Yiyang Wu

LinkedIn | GitHub | Website | wendywu2@andrew.cmu.edu | +1 (213) 275-6879

Skills

Languages & Tools: Python, C/C++, SQL, Bash, Docker, Git, Linux, Slurm, AWS, WandB

Frameworks: PyTorch, Transformers, Hugging Face, vLLM, veRL, LLaMA Factory, FAISS, NumPy

Focus: LLM Post-Training, Agentic RL, CoT Supervision, Coding Agents

Education

Carnegie Mellon University — **Pittsburgh, PA**
M.S. in Intelligent Information Systems — *Expected Dec 2026*
Coursework: PhD-level Machine Learning, Advanced NLP, Question Answering, Multimodal Machine Learning

University of Southern California — **Los Angeles, CA**
B.S. in Computer Science (GPA: 3.98/4.00) — *May 2025*

Industry Experience

TikTok Inc., Trust & Safety Team — **San Jose, CA**
Machine Learning Engineer Intern — *May 2026 – Aug 2026*

Owned the post-training loop that turned a shipped policy-token classifier into a model that reasons through each policy's decision tree and justifies its verdict

- Built a **million-scale decision-tree CoT corpus** via dual-LLM labeling with sampled human verification; a **Cohen's** κ audit showed the two label sources agreed at near-chance, so **human labels alone define the RL reward** while model labels supply scaled SFT supervision.
- Ran **dozens of SFT experiments** across several data generations, converging on a controlled **20-configuration matrix** (5 recipes $\times$ 4 verdict-token loss weights) and **40** clean models; selected checkpoints by balancing **precision, recall and leakage** against business priorities, with manual review of model outputs.
- Scaled to **multimodal GRPO** in ***veRL*** with **online self-distillation** (RLSD, OPSD) and a **decision-tree process reward**; best checkpoints reach **80.2% precision at 20.9% leakage**, and RL lifted recall from **72.6% to 83.5%** with format compliance at **99.6%**.
- Re-validated per-policy gains across many independently trained models per method, leaving **14%** of candidate policies with a robust improvement.

Research Experience

Measuring Credit Assignment in Agentic RL and On-Policy Distillation
Independent Research — *Apr 2026 – Present*

- Reproduced agentic-RL papers as matched baselines, such as **GiGPO**, **BEACON**, **ARPO**, **ReTool** and **RLAnything**, and scored the **progress detectors** they assign credit with against ALFWorld's exact planner over **3.4M** steps, obtaining a **gold-free** detector at **0.984** precision and **0.26** recall.
- Designed a new step-level credit signal, a **potential flux** on the GRPO advantage bounded to keep the outcome ranking and audited for cycle conservation, and evaluated it once on a **sealed, pre-registered** split, where it solved **49.3%** of tasks against **29.1%** for a matched outcome-only baseline.
- Isolated the role of credit *placement* in **on-policy distillation**: shuffling a teacher's per-token reward while keeping its sum and AUROC dropped MATH-500 **0.718**$\rightarrow$**0.212**.

USC LIME Lab — **Los Angeles, CA**
Research Assistant — *Aug 2024 – Dec 2025*

AdaRFT: Adaptive RL for Efficient Math Reasoning

- Designed **AdaRFT**, an adaptive curriculum method for **sample-efficient reinforcement learning** that updates task difficulty from reward signals during **PPO/GRPO** training using ***veRL*** on **Qwen2.5-1.5B** and **Qwen2.5-7B**.
- Estimated problem difficulty with **pass@128**, compared it against rubric-based **LLM-as-a-judge** evaluation, and found empirical support for **pass@128** as a strong difficulty metric.
- Achieved **2**$\times$ **faster convergence** and **3–4% accuracy gains** on competition-level math benchmarks; work published in **TMLR** (arXiv:2504.05520) with open-source code and dataset release.

Selected Projects

Coding Agent Trajectory Visualizer (*VS Code Marketplace*)
Personal Project — *Apr 2026 – Present*

- Built and published a **VS Code** extension that captures **Claude Code** API traffic and reconstructs whole sessions (prompt assembly, tool schemas, response events, timing, compaction) in an interactive webview.
- Added synchronized context, phase, tool-use, comparison and replay views, making production agent harnesses inspectable and connecting real coding-agent trajectories to process-reward research.

Workflow Memory for Coding Agents — **Pittsburgh, PA**
Research Developer — *Sep 2025 – Dec 2025*

- Ran **OpenHands + Kimi K2** on *SWE-bench* and analyzed failures to identify dominant failure modes.
- Built a **workflow-memory** layer over the pipeline, converting successful **CodeAct** repair trajectories into concise, reusable workflows via LLM-based cleaning, compression and abstraction.
- Designed **offline** and **online** settings: offline injects a static workflow library and improved the workflow-free baseline by **2%**; online incrementally compresses successful trajectories during inference to update memory.