Arohan Agate

aagate@cs.washington.edu (425) 503-9856 linkedin.com/in/arohan-agate github.com/arohan-agate

Education

University of Washington, Seattle **Seattle, WA**

B.S. in Computer Science (Honors), 3.8 GPA *June 2027*

- **Selected Coursework:** Distributed Systems, Operating Systems, Machine Learning, Deep Learning, Computer Vision, Compilers, Computer Networks, Algorithms, Data Structures and Parallelism, Probability and Statistics, Linear Algebra

Experience

Meta **Menlo Park, CA**

Software Engineer Intern *June 2026 - Sept 2026*

- Instagram Search Recommendations

UW Reasoning, AI, Vision Lab **Seattle, WA**

Undergraduate Researcher *May 2025 - present*

- Built an end-to-end pipeline that converts raw internet videos into 3D object meshes through multiview filtering, object segmentation, pose estimation, and neural mesh reconstruction, processing large batches in parallel on GPU clusters.
- Designed and implemented a quantitative evaluation framework for 3D scene reconstruction, including automated camera trajectory generation, ground truth mesh extraction, and geometry/appearance metrics across multiple indoor environments.
- First-author paper to Computer Vision and Pattern Recognition Conference (CVPR) 2026 and International Conference on Machine Learning (ICML) 2026.

GEICO **Washington, D.C.**

Software Engineer Intern *June 2025 - Aug 2025*

- Developed a data-fix automation system with Azure DevOps API integration, Milvus **vector database** for RAG similarity search, and LLM orchestration with confidence filtering and response judging. Introduced **LangChain** agentic deployment – processing 7,500 monthly tickets with an efficiency gain of 70%, 90% success rate, and eliminating 50 weekly engineering hours.
- Designed and implemented backend field metadata services in Java **SpringBoot** with **gRPC** and **GraphQL**, adding schema versioning, URI validation, and persistence through Cassandra.
- Built a self-service interface for creating field URIs with **React** and **TypeScript**, including custom modal components, form validation, searchable dropdowns, and duplicate detection.

Allen Institute for AI **Seattle, WA**

Student Collaborator *March 2025 - June 2025*

- Worked with team to build a benchmarking pipeline to evaluate state-of-the-art language models on puzzles.
- Used Pandas, Selenium, and LLM APIs to automate scraping the Jane Street website for new puzzles and evaluating models.
- Established a benchmark for evaluating LLM reasoning on unseen problems, addressing leakage in existing tests (*project link*).

Paul G. Allen School of Computer Science **Seattle, WA**

Undergraduate Teaching Assistant *Sept 2024 - present*

- Teach content on data structures and algorithms, covering concepts like linked lists, binary trees, hashing, and sorting.
- Conduct recitations on object-oriented design and **Java** programming; help with debugging during office hours.

Altea Healthcare **Seattle, WA**

Data Engineer Intern *June 2024 - Sept 2024*

- Automated document processing using **Python**, **Beautiful Soup**, and **Selenium**, expediting workflow and accuracy by 97%.
- Developed and optimized a neural network using **TensorFlow** to predict hospital readmission rates. Achieved 92% accuracy by leveraging hyperparameter tuning, regularization, and batch normalization.
- Redesigned database schemas and transformation logic using DAX and Power Query, reducing storage requirements by 83%.

Personal Projects

LiveBid Auction System **December 2025**

- Developed **SpringBoot** auction platform with **PostgreSQL** persistence, WebSocket real-time updates, and 15ms response times.
- Implemented transactional bidding engine with automatic fund reservation and scheduled settlement for concurrent processing.
- Integrated **AWS S3** for image storage, **Redis** for caching, and Google OAuth with JWT-based authentication.
- Built **Next.js** frontend with live bid updates, user notifications, and auction search functionality.

xk Operating System **June 2025**

- Implemented an xv6-based OS in **C** with UNIX system calls open, read, write, dup, exec, and fork, as well as file descriptor tables and open file tracking.
- Enabled multiprocessing with fork, wait, and exit, supporting parent-child processes, cleanup, and copy-on-write optimization.
- Added heap and stack growth via sbrk and demand paging, allowing for dynamic memory allocation and user stack usage.
- Developed an interactive shell by enabling file I/O, pipes, and program execution, transforming to a multi-user shell environment.

Technical Skills

Languages: Python, Java, TypeScript, JavaScript, C, SQL
Frameworks & Libraries: Spring Boot, React, Flask, NextJS, PyTorch, TensorFlow, HuggingFace, NumPy, Pandas
Systems & Infrastructure: Linux, Git, Docker, PostgreSQL, Redis, WebSockets, SLURM (GPU clusters)
Cloud & Platforms: Amazon Web Services, Google Cloud