

Alex Zhang

(509) 432-3135

✉ 7alexzhang7@gmail.com

🌐 <https://alexzhang77.github.io>

Education

2020 – 2023 **B.S. Computer Science**, *University of Washington*, **GPA: 3.95/4.0**
2024 – Present **M.S. Computer Science**, *University of Washington*, **GPA: 3.93/4.0**

Professional Experience

Jul – Dec 2021 **Software Engineer Intern**, *Schweitzer Engineering Laboratories*

- Jun – Sep 2022 ○ Enhanced a parsing and communications system for settings data in company's three most popular product lines using .NET Framework, automating over 95% of data processing rules:
- Designed and built a parsing algorithm and data storage structure capable of processing over 20,000 settings data using C++ and C#.
 - Deployed and automated product builds using Conan for dependency management and Jenkins for CI/CD, ensuring consistent and reliable deployment across various environments.
- Jun – Sep 2023 ○ Identified automation and optimization opportunities in end-to-end testing processes for company product protection systems. Designed and developed test automation systems which improved testing efficiency by upwards of 80%:
- Implemented an automated unit test suite generation system using Python and Jinja, expediting device settings test production time by 55%.
 - Built a Python and Selenium automated functional test system for company's more popular product line covering 80% of the end-to-end test processes.
- Streamlined and developed a prototype for the company's next generation automated electrical relay testing system, reducing 75% of manual processes:
- A pipeline parsed and transformed over 5,000 synthesized signals, which triggered a series of unit tests for the connected product under test and visualizes the test outputs.
- Technologies: C/C++, Python, C#, Java, JavaScript/TypeScript, MySQL, AWS, Conan, Docker, Jenkins, Kubernetes/EKS, Node.js, React, Selenium, Spark, xUnit

Research Experience

2024 – Present **Graduate Researcher**, *Robot Learning Laboratory (UW)*

- Developed a generalizable workbench environment for NIST assembly tasks in NVIDIA Isaac Sim and Lab. The environment supports novel reinforcement learning algorithms to complete long horizon and dexterous tasks for robotic manipulation.
- Designed baselines for sim-to-real pipelines, custom task definitions, and modular reinforcement learning frameworks to enhance reproducibility and scalability in robotic manipulation tasks.
 - Technologies: Python, PyTorch, TensorFlow, TFX, JAX, PyBullet, Gymnasium, Omniverse, PhysX

2024 – Present **Graduate Researcher**, *Control & Trustworthy Robotics Laboratory (UW)*

- Implemented a precise and accessible end-to-end computer vision pipeline for object detection and counting in aerial drone footage.
- Reduced detection and counting time for clients by more than 90% compared to previous method.
 - Integrated a processing method to outperform state-of-the-art aerial detection by over 10% in client's task.
 - Developed VLM integration to enhance previous detection and counting results by over 5%.
 - Technologies: Python, PyTorch, TensorFlow, TFX, OpenCV, NumPy, scikit-learn, Pillow, MMCV

Jun – Sep 2024 **Graduate Researcher**, *Personal Robotics Laboratory (UW)*

- Developed a continual imitation learning pipeline to leverage part-based reasoning through Vision Language Models (VLMs) to enhance robot manipulation tasks.
- Researched and benchmarked VLMs for part-aware image segmentation using Grounded-SAM-2, Llama 3.2, and LLaVA-NeXT.
 - Integrated dense part mask embeddings of VLMs into learning pipeline to solidify continual learning of object parts within robot manipulation tasks.
 - Enhanced simulation data image quality by 8-fold for improved downstream pipeline usability.
 - Technologies: Python, PyTorch, Keras, OpenCV, scikit-learn, Pillow, robosuite, MUJOCO, wandb