

Weishu Zhang

626-566-4310 | weszhang@berkeley.edu | [LinkedIn](#) | [GitHub](#) | [Website](#)

U.S. work authorized; available December 2026. Backend-leaning full-stack engineer building production agent systems and enterprise workflow runtimes.

EDUCATION

University of California, Berkeley

Berkeley, CA

Bachelor of Science in Electrical Engineering and Computer Sciences

May 2027

– **GPA: 4.0/4.0**

– **Relevant Coursework:** Efficient Algorithms, Network Science, Artificial Intelligence

TECHNICAL SKILLS

Languages: Python, TypeScript/JavaScript, SQL, Java, Go, C/C++

Full Stack: React, Next.js, Node.js, FastAPI, PostgreSQL, Redis, SQLite, REST APIs

AI & Agents: LangGraph, MCP, LLM APIs, agent runtimes, structured generation, PyTorch, vLLM, QLoRA

Infrastructure: Docker, Kubernetes, gVisor, OCI, OpenTelemetry, pytest, distributed jobs, CI/CD

EXPERIENCE

Amazon

Incoming Software Development Engineer Intern

Sep 2026 – Nov 2026

ASML

San Jose, CA

Software Engineering in Test Intern

2026 – Present

- Designed and implemented **Regression Report Integration (RRI)**, an ontology-driven Python workflow CLI and agent-ready execution layer for SQA regression review, unifying governed entities and actions across Confluence review pages, Jira metadata, qTest synchronization, email, and compute jobs.
- Developed a **Test Plan Runtime Agent** and workspace harness for stateful workflows that synthesized EPDS and Jira documentation to create or enhance TPS artifacts, proposed test-task changes for human approval, then tracked asynchronous execution and reconciled results back into the TPS.
- Implemented deterministic workflow primitives with Confluence-backed approval state, SQLite-backed runtime state, reusable enterprise adapters, idempotent writes, job claiming, page-version conflict protection, and local sandbox replay for safe recovery.

Open Ludus

Remote

Co-Founder & Software Engineer, Cloud Agent Infrastructure

2026 – Present

- Designed a **Kubernetes-native agent sandbox control plane** using CRDs and controllers to reconcile isolated, stateful workspaces from reusable runtime templates, with warm-pool provisioning, suspend/resume, and TTL-based garbage collection.
- Hardened untrusted-code execution with **gVisor/Kata RuntimeClasses**, namespace-scoped service accounts and RBAC, default-deny NetworkPolicies, restricted pod security contexts, and CPU, memory, and ephemeral-storage quotas.
- Standardized reproducible agent environments using immutable OCI images, validated configuration and short-lived secret projection, repository bootstrap, and PVC-backed workspaces; instrumented provisioning and execution with OpenTelemetry traces and lifecycle metrics.

Berkeley Artificial Intelligence Research (BAIR)

Berkeley, CA

Undergraduate Research Assistant

Jan 2026 – Present

- Led expert task acquisition for the **Agent Human Last Exam** benchmark, recruiting **30+ domain experts** and contributing to the collection of **80+** complex, cross-domain agent tasks.
- Built an LLM-powered proposal agent that converts rough expert ideas into structured benchmark tasks, supporting iterative refinement, constraint checking, and higher-quality submissions.

University of California, San Francisco

San Francisco, CA

Software Research Assistant

Aug 2025 – Present

- Built a production-oriented long-context clinical information extraction pipeline, increasing inference throughput by **3x** and reducing VRAM usage by approximately **80%** using vLLM, paged attention, quantization, and batched execution.
- Created a **4,000+**-example domain dataset, fine-tuned Qwen models with QLoRA, and implemented validation and self-consistency checks that maintained **100% schema-valid** structured outputs.

OPEN SOURCE

PyTorch | [torch.compile](#), [torch.export](#), [Dynamo PRs](#)

- Contributed production fixes to **pytorch/pytorch** across compiler and export internals, resolving eager-versus-compiled correctness gaps, tracing failures, and export regressions with targeted tests and maintainer-reviewed patches.

SELECTED PROJECTS

GradeView | [React](#), [TypeScript](#), [Python](#), [PostgreSQL](#), [Redis](#) [Live](#) | [Demo](#) | [Poster](#)

- Refactored a production learning-analytics platform used in Berkeley courses into containerized React, Node.js, and Python services; standardized PostgreSQL as the source of truth and added Redis cache prewarming to improve reliability and read performance.