

Kevin Xiang Li

<https://kevinx.li>

Email: kevinx.li@outlook.com

TEL: 734-510-0189

SKILLS

Machine Learning: SGLang, vLLM, Unsloth, JAX, Nsight, Gymnasium, HuggingFace, PyTorch, Python
Mobile Dev: Flutter, SQLite, Rust **AR/VR:** Unreal, Unity, C# **Web:** TypeScript, JavaScript, HTML, CSS

EDUCATION

- **Stanford University** Stanford, CA, U.S.A
M.S. in Computer Science; GPA: 4.0 2024. 9 – 2026. 3
- **University of Michigan** Ann Arbor, MI, U.S.A
B.S. in Computer Science, Minor in Linguistics; GPA: 3.87, Summa Cum Laude. 2020. 9 – 2024. 5

EXPERIENCES

- **SWE-smith-X: Scaling Multilingual Data for Software Engineering Agents** Stanford University
Researcher at Stanford 2025. 9 - Present
 - **Designing procedural modifications for 4 languages:** Wrote 40+ procedural modifications for efficient coding task synthesis across Java, JavaScript, C++, and Rust; PR merged into SWE-smith.
 - **Building end-to-end multilingual task synthesis:** Automating entire process from repo construction to task synthesis and trajectory generation across 4 languages, overcoming language-specific challenges in task validation.
- **Marin: Post-training LLMs in the Open** Stanford University
Researcher at Stanford 2025. 9 - Present
 - **Improving RL post-training:** Aligning in-house RL framework to match baseline on MATH-500, fixed training dataset, prompt format, reward function, advantage function, loss, training and sampling hyperparameters.
 - **Building eval pipeline:** Designing evaluation harness based on lm-eval-harness, tested on 32B base model.
- **LLM Inference Workload Performance** Santa Clara, CA
ML Engineer at Nvidia 2025. 6 - 2025. 9
 - **Benchmarked VLMs on large GPU clusters:** Measured throughput and latency of the Qwen 2.5 VL family (3B–72B) across H200 and B200 clusters.
 - **Pinpointed and reported inference bottlenecks:** Used Nsight, NVTX markers, and PyTorch Profiler to pinpoint kernel-level bottlenecks in SGLang and vLLM; dissected framework performance gaps under varying concurrency and provided detailed reports well received by both SGLang and vLLM multimodal teams.
 - **Submitted 5 PRs to SGLang that boosted Qwen 2.5 VL throughput by 1.6x end-to-end on MMMU:** (1) Doubled Qwen 2.5 VL vision prefill speed via automatic attention backend selection, (2) Accelerated rotary embedding with CUDA rotary kernels, boosting vision prefill throughput by 21%, (3) Identified and removed redundant device-to-host visual feature transfers to enable accelerated GPU hashing, yielding 7.5% end-to-end speedup on MMMU, (4) Fused SwiGLU in ViT to double peak TensorCore utilization, resulting in 4.5% vision prefill throughput gain, (5) Unified VLM benchmarking to support reliable cross-framework comparisons.
- **VideoMultiAgents: A Multi-Agent Framework for Video QA** Stanford University, U.S.A.
Researcher, in collaboration with Panasonic 2024. 10 - 2025. 3
 - **Designed multi-agent framework with modality-specific agents for video QA:** Enhanced video understanding by leveraging strengths of video, text, and graph modalities through multi-agent collaboration.
 - **Discovered that modality-specific multi-agent architectures benefit from structure and independence:** Showed that our Report architecture performs the best by aggregating opinions from independent modality-specific agents through an organizer agent and weighing strength of evidence from each modality.
 - **Achieved SOTA accuracy on popular video QA benchmarks:** Improved previous SOTA on Intent-QA by +6.2%, EgoSchema subset +3.4%, and NExT-QA by +0.4%.
- **Statically Contextualizing LLMs with Typed Holes** University of Michigan, U.S.A.
Researcher 2023. 9 - 2024. 8
 - **Enhanced code LLMs with static retrieval:** Leveraged semantic context and static error correction capabilities of language servers to enhance LLM code generation accuracy and stem hallucination.
 - **Boosted LLM coding performance significantly:** Static retrieval method resulted in 3.5x more unit tests passed on 5 realistic TypeScript benchmarks, compared to vector retrieval with GPT-4.
 - **Published at OOPSLA:** Research published at OOPSLA 2024 in Pasadena, California.