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: 3.9 *2024. 9 – 2026. 6*
- **University of Michigan** Ann Arbor, MI, U.S.A
  B.S. in Computer Science, Minor in Linguistics; GPA: 3.8, Summa Cum Laude. *2020. 9 – 2024. 5*

Experiences

- **SWE-smith Multilingual: Scaling Multilingual Data for Software Engineering Agents** Stanford
  Researcher at Stanford *2025. 9 - Present*
  - **Released largest dataset of software engineering tasks**: Synthesized 386K bug-fix pairs spanning 39 repos.
  - **Boosted issue resolve rate from 0% to 2.3%**: Fine-tuned Qwen2.5-Coder-32B using 142 GLM-4.6 teacher trajectories, improving SWE-bench Rust resolve rate from 0% to 2.3% and increasing completion rate by 2.4x.
  - **Slashed task synthesis time from days to hours**: Built a scalable pipeline using Modal to parallelize task synthesis across hundreds of concurrent sandboxes, reducing processing time from days to hours.
  - **Designed procedural modifications for 4 languages**: Wrote 40+ procedural modifications for efficient coding task synthesis across JavaScript, Java, C++, and Rust
  - **Submitted 10+ PRs to SWE ecosystem**: 6 PRs merged into SWE-smith; 2 pending at SWE-bench; 2 pending at SWE-ReX, 1 merged into mini-swe-agent.
- **Marin: Post-training LLMs in the Open** Stanford University
  Researcher at Stanford *2025. 9 - Present*
  - **Reproduced RL results on key reasoning benchmarks**: Aligning in-house RL framework to match baseline on MATH-500, fixed training dataset, prompt format, reward function, advantage function, and loss.
  - **Scaled training throughput**: Achieved a 3.7x speedup in async RL trainer throughput (from 4.1 to 15.4 requests/s) by implementing batch-dynamic max sequence lengths in the trainer.
  - **Sped up weight sync**: Doubled weight transfer speed in the RL pipeline by implementing fp32-to-bf16 conversion on the trainer side prior to transfer.
  - **Hardened system stability**: Authored and merged critical fixes for training stability, including resolving head node reservation issues during weight sync and ensuring consistent WandB logging across node pre-emptions.
  - **Open source contribution to vLLM**: Identified and resolved a critical bug in the vLLM-tpu sampling logic where default top K values incorrectly forced greedy sampling regardless of temperature settings.
- **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.