Chenrui Gao

cg52638@my.utexas.edu — GitHub/cr-gao — LinkedIn/Chenrui-Gao — cr-gao.github.io

Education

University of Texas at Austin — Austin, TX
M.S. in Computer Science — *Aug 2026 – May 2028 (expected)*

University of Michigan, Ann Arbor — Ann Arbor, MI
B.S.E. in Computer Engineering ***(GPA: 3.93/4.00)*** — *Aug 2024 – May 2026*

Shanghai Jiao Tong University — Shanghai, China
B.E. in Mechanical Engineering ***(GPA: 3.70/4.00)*** — *Aug 2022 – Aug 2026 (expected)*

Publications

VAMP-MR: Vector-Accelerated Motion Planning and Execution for Multi-Robot-Arms
Philip Huang, **Chenrui Gao**, Jiaoyang Li — *IROS 2026*

Research Experience

Carnegie Mellon University, ARCS Lab — Pittsburgh, PA (Remote)
Research Intern; advised by Prof. ***Jiaoyang Li*** *— basis of the VAMP-MR publication* — *Nov 2024 – Oct 2025*

- Built the planner's throughput-critical core in C++, batching forward kinematics and collision checks across robot configurations into CPU SIMD lanes with a struct-of-arrays layout, which removed motion validation as the bottleneck and made multi-arm motion generation **10–100**× faster, down to milliseconds
- Implemented the CBS-MP multi-robot motion planning framework on top of the vectorized primitives

University of California, Irvine — Irvine, CA
Research Intern; advised by Prof. ***Sven Koenig*** *— Topological Pathfinding with Ray-Cut CBS* — *Jun 2025 – Aug 2025*

- Proposed a hierarchical approach for constrained multi-agent routing: computing homotopy classes over 2D space to guide Conflict-Based Search (CBS)

Open-Source Contributions

verl-omni — *diffusion/omni RL post-training* | **Committer, 15 PRs merged** — Jul 2026 – Present

- Implemented **On-Policy Distillation** for diffusion RL, the core feature of the official roadmap RFC: teacher-anchored KL and flow-matching losses plus a frozen-teacher runtime that runs as a third forward-only worker beside actor and ref, with fit-loop hooks and fail-closed config validation
- Verified it end to end on SD3.5-Medium with an OCR reward and saw student reward rise from **0.23 to 0.62–0.71**, matching the FlowGRPO RL teacher about **5**× faster than GRPO while distillation KL fell monotonically from 4.2e-4 to 1.8e-5; covered by 150+ mutation-tested unit tests, GPU smoke runs and a 40-step training-curve reproduction
- Extended OPD to **multi-teacher** distillation that routes each sample by a configurable batch column to teachers either co-located with the actor or in standalone per-teacher resource pools scored concurrently
- Wrote an RFC and shipped an **asynchronous one-step-off teacher scheduler** for the v1 separate-async trainer that overlaps teacher scoring of batch k with the actor update of batch $k-1$: teacher-phase wait fell from **3.6s to 0.02s**, two-teacher step time from **22.1s to 19.0s**, and single-teacher step time by 16.7–23.7% at full load, while training curves matched inline scheduling within 0.004 PickScore
- Fixed a Ray rendezvous port collision across worker groups by slicing ports per group, which unblocked standalone multi-teacher pools; two teacher-config corrections in the same PR, bf16 weights and the inference micro-batch, cut the teacher phase from **7.76s to 3.13s**
- Found and fixed a v1-trainer bug in which co-located reward outputs overwrote the whole batch
- Wrote a maintainer-endorsed RFC that measured rollout–training numerical consistency across three diffusion pipelines, SD3.5-Medium, Qwen-Image-20B and BAGEL-7B, over 8 experiment arms; its proposal shipped as default per-timestep monitoring

- Traced a replay crash and one-directional log-prob drift to a stale BAGEL step-count mapping and fixed it, cutting mean log-prob drift **157**× to bf16 noise in line with an analytical prediction
- Wired the torch profiler into both trainer fit paths and the reward-model rollout server, bringing profiled-step time from **616s to 70s** and trace size from 163MB to 32MB

vllm-omni — *SANA-Video acceleration stack & MAGI-2 torch.compile* | **5 PRs merged** Aug 2026 – Present

- Derived a **sequence-parallel scheme for ReLU linear attention**, where Ulysses and Ring do not apply: because the KV state sums over the sequence dimension, SP needs only one all-reduce of a single $d \times d$ state, so communication is $O(d^2)$ at any sequence length; measured **1.80**× at SP2 and **3.57**× at SP2+CFG2 on 4×A800
- Added tensor-parallel and CFG-parallel execution, measuring TP2 **1.52**×, CFG2 **2.04**× and TP2×CFG2 **3.02**×, with a 32-case correctness matrix and TP1 bit-exact against baseline
- Added Cache-DiT caching with CPU offload, **1.56**× faster with peak GPU memory down **7.3 GiB**
- Fixed a FLASH_ATTN cross-attention key-padding bug that silently mixed samples within a batch and affected every padded cross-attention diffusion model, raising real-weight output cosine from **0.9157 to 0.99996**
- Applied **regional torch.compile** to the MAGI-2 Preview native transformer, splitting each layer into compilable regions around the eager packed-attention and MoE-routing kernels and emulating eager bf16 rounding with precision casts; denoising step fell from **1.93s to 1.48s** at SP4/272p on A800 with per-GPU peak memory down **1.1–2.2 GiB**, and compiled-vs-eager differences sit at kernel reduction-order level; *in review*
- Fixed layerwise CPU offload that kept a nested 40-layer block fully resident and OOM'd single-GPU MAGI-2; *in review*
- Built a shadow-wrapper merge-on-load fast path for diffusion LoRA with fp32 delta fold-in and exact unload, bringing LoRA generation overhead from **+122% to +4.7%** while holding the PPO ratio at 1±1e-6; *in review*

sglang-omni — *Qwen3-Omni inference performance* | **1 PR merged** Aug 2026

- Rewrote the thinker-prefill multimodal embedding merge from a per-request Python loop into one batched GPU scatter: device-to-host syncs fell from **96 to 0** per batch for a **2.1–4.1**× speedup and cleared the path to prefill CUDA graphs
- Verified the rewrite three ways: 6000-seed differential fuzzing with element-wise identical output, byte-identical end-to-end serving logits, and an H800 performance matrix

Projects

LLM Training Systems from Scratch (Stanford CS336 Assignments) | *PyTorch, Triton, NCCL* 2026

- Wrote **FSDP from scratch** with backward-hook reduce-scatter, on-demand all-gather, fp32 master shards and a round-robin sharded optimizer, a **FlashAttention Triton kernel**, and a **KV cache**, all inside a Transformer LM built from the ground up with RoPE and a BPE tokenizer

Teaching

Teaching Assistant, Intro to Computers & Programming (Prof. Jigang Wu) *SJTU, Fall 2023*

Honors

Dean's Honor List, University of Michigan *Dec 2024, May 2025*

John Wu and Jane Sun Sunshine Scholarship *Nov 2023*

Skills

Languages: C++, Python, Triton, CUDA (basic), MATLAB

ML Systems: PyTorch (torch.distributed, torch.compile), vLLM, verl, FSDP/DDP + LoRA, FlashAttention, Ray, Hydra, torch profiler

Systems & Tools: Linux, Git/GitHub OSS workflow, CUDA 13/cu130, ROS, OpenCV, NumPy, Pandas

Coursework: Stanford CS336 (LLMs from Scratch), MIT 6.S081 (Operating Systems)