Jeongjin Han

hjj22@kaist.ac.kr Jeong-jin-Han jeong-jin-han.github.io

Research Interests

- Architectural exploration of memory hierarchy design with interest in RTL-level hardware development
- Interest in CXL-based disaggregated memory systems and memory-pressure-driven smart routing
- Seamless resumption of distributed ML training through consistent multi-process state capture

Education

Korea Advanced Institute of Science and Technology (KAIST) — Daejeon, Korea
B.S. in Computer Science & Nuclear and Quantum Engineering — Feb. 2022 – Present

- **GPA:** 3.92 / 4.3 (96.20 / 100)
- **Minor:** Mathematical Sciences
- **Honors:** Dean's List, Spring 2025

Preprints

SHIFT: Sigmoid-Based Heuristic Invertible Fitness-Landscape Transformation — 2025
First author | CS454 AI-Based Software Engineering, KAIST `arXiv:2604.09171`

- Proposed and designed the core SHIFT algorithm: a sigmoid-based invertible fitness-landscape transformation for SBST that compresses flat basins to enable hill climbing to escape plateaus. Designed a three-rule basin detection framework and an N-dimensional compression manager storing metadata in original X-space to correctly handle saddle points under per-dimension, per-coordinate compression.
- Implemented AST instrumentation, constant folding, and SSA-like variable-shading for Python source transformation; time-budgeted parallel evaluation (per-branch 20 s timer) and constant-aware biased initialization (80% Gaussian sampling near AST-extracted constants); HC-SHIFT required only **2–8 trials** vs. hundreds/thousands for baseline HC/GA, demonstrating orders-of-magnitude efficiency gains in branch coverage.

VRAIL: Vectorized Reward-based Attribution for Interpretable Learning — 2025
Co-author (equal contribution) | CS377 Reinforcement Learning, KAIST `arXiv:2506.16014`

- Concretized and implemented a bi-level RL framework (Linear/Quadratic VRAIL) learning interpretable value estimators as potential-based reward shaping functions. Quadratic VRAIL converged ∼**10% faster** than DQN (538 vs. 600 epochs); Linear VRAIL achieved **100% convergence rate** (10/10 runs) vs. 80% for DQN.
- Led reward transfer experiments showing pretrained VRAIL reward functions generalize to vanilla DQN agents (100% vs. 80% convergence); conducted α-scheduling and environment variant ablation studies.

Research Experience

Nuclear I&C and Autonomous Operation Lab, KAIST — Winter 2025
Undergraduate Researcher | Advisor: Prof. Jonghyun Kim

- Identified **system-level bottlenecks** under memory and bandwidth constraints across heterogeneous CPU–GPU pipelines.
- Led pipeline redesign to resolve a **GPU underutilization bottleneck** caused by a CPU-bound simulator; decoupled policy optimization from environment interaction, significantly improving **hardware utilization** under system resource constraints.
- Reverse-engineered the **process memory layout** of a closed-source Windows VM simulator to extract and inject runtime state variables, enabling system integration without source access or binary modification.
- Designed and implemented a TCP/Docker-based distributed execution infrastructure bridging a Windows VM and a GPU server, managing cross-system state synchronization and execution control under **memory bandwidth and latency** constraints.

Reactor Physics and Transmutation Lab, KAIST — Winter 2024
Undergraduate Researcher | Advisor: Prof. Yonghee Kim

- Conducted Monte Carlo neutron transport simulations using **OpenMC**, analyzing neutron moderation behavior under Thermal Scattering Law (TSL) models from the ENDF/B-VIII.1 nuclear data library.
- Investigated computational and **memory characteristics** of large-scale Monte Carlo simulation workloads, and analyzed **scalability** behavior under **parallel execution**.

Projects

KECC: Optimizing C-to-RISC-V Compiler Spring 2025
CS420 Compiler Design

- Designed and implemented a full C-to-RISC-V compiler in **Rust**: SSA-based IR optimizer (Mem2reg, GVN, CFG simplification, DCE) reducing **memory traffic** and instruction count, and an Asmgen backend with struct/array/float/pointer support and phi-node parallel-move SSA destruction.
- Diagnosed and patched 10+ correctness bugs across all phases (non-deterministic GVN hash collisions, erroneous dead GEP elimination, float calling-convention violations, wrong-block phi-argument removal, etc.).
- Conducted independent cycle-level performance evaluation using **gem5 v25.1 MinorCPU** across 60 C benchmarks; achieved performance **comparable to Clang -O1** (geometric-mean cycle ratio 1.009×), with 37/60 outperforming Clang.

`Jeong-jin-Han/CS420-KECC`

MIPS 5-Stage Pipelined CPU Simulator Spring 2024 / Verilog: Spring 2026
CS311 Computer Organization

- Implemented a full MIPS 5-stage pipeline (IF/ID/EX/MEM/WB) in **C**: **forwarding unit** resolving **RAW hazards** via EX/MEM→EX and MEM/WB→EX paths without stalling; **hazard detection unit** detecting load-use hazards by freezing PC and IF/ID registers and inserting a NOP bubble into ID/EX.
- Designed a 1-bit branch predictor with BHT (1-bit taken/not-taken, PC-indexed) and BTB (target address + valid bit): prediction issued in **IF stage**, update applied in **EX stage** after branch resolves; re-implemented the full design in **Verilog** for **RTL-level** simulation with testbench verification; explored pipeline hazard handling and control flow mechanisms through cycle-accurate simulation.

Nuclear System Code: Overcooling Accident Simulation Fall 2025
NQE625 Nuclear System Computational Analysis

- Built a 1-D multi-channel nuclear system code from scratch in **MATLAB**: **upwind FVM** with **semi-implicit time integration** across coupled subsystems (fluid momentum/enthalpy, point kinetics with delayed neutrons, fuel heat conduction, steam generator primary–secondary coupling).
- Validated numerical solutions against the industry-standard **MARS** code across 10 physical quantities, demonstrating applicability of **coupled-system numerical methods** at scale.

Skills

Languages	C++, Rust, Python, MATLAB, Scala, Verilog
Tools & Frameworks	PyTorch, gem5, Docker, HuggingFace, LaTeX, Git, Linux

Relevant Coursework

Systems	Computer Organization (**CS.30101**), Operating Systems (**CS.30300**), System Programming (**CS.20300**), Computer Networks (**CS.30401**)
CS Theory	Introduction to Algorithms (**CS.30000**), Compiler Design (**CS.40200**), Programming Language (**CS.30200**)
AI / ML	Reinforcement Learning (**CS.30707**), Machine Learning (**CS.30706**), Introduction to Artificial Intelligence (**CS.40700**)
Numerical / HPC	Numerical Methods and Computer Simulation (**NQE.30011**), Computational Analysis in Nuclear System (**NQE.60025**)
Mathematics	Linear Algebra (**MAS.20012**), Analysis I / II (**MAS.20041/42**), Lebesgue Integral Theory (**MAS.40041**), Probability (**EE.20010**)