潘浩林

(+86) 18749552933 · panhaolin21@mails.ucas.ac.cn · AI Infra / Compiler Optimization · 25 years old

Personal Profile

- Ph.D. candidate at the Institute of Software, Chinese Academy of Sciences, expected to graduate in 2027. My research focuses on **AI Infra, compiler optimization, compiler auto-tuning, and program representation learning**, with systematic work on LLVM / MLIR pass-sequence optimization, LLM/RL-driven tuning, quasi-dynamic program representation, and high-performance SIMD library design.
- First-author publications at leading international venues including **NeurIPS 2025, ICLR 2026, ASE 2025, and CGO 2025**. Several works form a complete research loop from **problem abstraction and algorithm design to system implementation and performance validation**.
- Strong background in compiler systems and performance engineering. Familiar with the **LLVM / MLIR / C++ / Python / Linux** stack, with hands-on experience in solving real-world compiler optimization problems and implementing deployable research systems.

Education

Institute of Software, Chinese Academy of Sciences, Computer Software and Theory, *Ph.D. Candidate* 2023.09 - 2027.06

- Published 6 papers at venues/journals including **NeurIPS, ICLR, ASE, TMC, and CGO**, including **4 CCF-A publications and 5 first-author papers**.
- Research interests: compiler optimization, AI for Compiler, program representation learning, auto-tuning, and SIMD optimization.

Institute of Software, Chinese Academy of Sciences, Electronic Information, *M.Eng.* 2021.09 - 2023.06

- Participated in the development of a multi-level compilation framework based on **MLIR**, covering DSL representation, IR design, and backend optimization.
- Worked on the design of a cryptography DSL and MLIR dialect, implemented lowering procedures, and conducted key optimization practices such as **SIMD vectorization**.

Zhengzhou University, Computer Science and Technology, *B.Eng.* 2016.09 - 2020.06

- Completed systematic training in core courses including computer systems, compiler principles, and programming languages, building a solid foundation in systems and software.

Selected Research Contributions

- **Compiler-R1（NeurIPS 2025, CCF-A）：** Proposed the first **Agentic RL** framework for compiler auto-tuning. Built nearly 20,000 high-quality reasoning and tool-interaction samples, and designed a two-stage SFT + reinforcement learning training pipeline, enabling LLMs to autonomously search optimization sequences and invoke external tools. On target programs, the method further reduces IR instruction count by **8.46%** over LLVM -Oz, with a task-interaction success rate of **96.71%**.
- **Behavioral-PQ（ICLR 2026, CCF-A）：** Proposed a **quasi-dynamic program behavior representation** framework that overcomes the limitations of traditional static code representations. The method captures scale-invariant behavioral spectra before and after optimization through probe sequences, and combines PQ with a multi-task Transformer to model the behavioral grammar of programs. It achieves **89.55% Top-5 accuracy** on best-pass prediction and **8.19% MAE** on long-sequence gain prediction, significantly outperforming baselines such as Inst2Vec and ProGraML.
- **Synergy-guided Auto-Tuning（CGO 2025, CCF-B）：** Addressed the high-dimensional and inefficient LLVM pass search space by proposing a **chained synergistic pass-pair** modeling method. By combining core subspace prediction with genetic algorithms, the method enables efficient second-level search. Across 10 public datasets, it reduces code size by an average of **13.9%** and IR instruction count by **7.5%** compared with LLVM -Oz.
- **HybridSIMD（ASE 2025, CCF-A）：** Designed a unified programming abstraction and auto-tuning framework for cross-SIMD-library co-optimization. On selected tasks, the method achieves up to **120**× performance improvement over LLVM -O3, and has resulted in two invention patents as the first inventor.

PUBLICATIONS

- **[NeurIPS 2025, CCF-A] Haolin Pan**, et al. *Compiler-R1: Towards Agentic Compiler Auto-Tuning with Reinforcement Learning*. [paper]
- **[ICLR 2026, CCF-A] Haolin Pan**, et al. *Behavioral Embeddings of Programs: A Quasi-Dynamic Approach for Optimization Prediction*. [paper]
- **[ASE 2025, CCF-A] Haolin Pan**, et al. *HybridSIMD: A Super C++ SIMD Library with Integrated Auto-Tuning Capabilities*. [paper]
- **[CGO 2025, CCF-B] Haolin Pan**, et al. *Towards Efficient Compiler Auto-Tuning: Leveraging Synergistic Search Spaces*. [paper]
- **[TMC, CCF-A, *IEEE Trans*]** Chao Zha, **Haolin Pan**, et al. *FlowXpert: Context-Aware Flow Embedding for Enhanced Traffic Detection in IoT Network*. [paper]
- **[SEKE 2025, CCF-C] Haolin Pan**, et al. *Navigating the SIMD Optimization Maze: A Reinforcement Learning Approach to Library and Compiler Co-Optimization*. [paper]

Submitted / In Preparation（all as first author）

- **[Submitted to ICML 2026]** *AwareCompiler: Agentic Context-Aware Compiler Optimization via a Synergistic Knowledge-Data Driven Framework*. [arXiv]
- **[Submitted to ICML 2026]** *ECCO: Evidence-Driven Causal Reasoning for Compiler Optimization*. [arXiv]
- **[Submitted to ASE 2026]** *GRACE: Globally-Seeded Representation-Aware Cluster-Specific Evolution for Compiler Auto-Tuning*. [arXiv]
- **[Submitted to ASE 2026]** *Synergy-guided Compiler Auto-Tuning of Nested LLVM Pass Pipelines*. [arXiv]
- **[Submitted to ASPLOS 2026]** *A Hybrid, Knowledge-Guided Evolutionary Framework for Personalized Compiler Auto-Tuning*. [arXiv]
- Research papers in progress: two papers on MLIR-based compiler optimization systems and AI-driven operator auto-scheduling.

CORE PROJECT EXPERIENCE

AI for Compiler Auto-Optimization System, Core Lead 2024.01 - Present

- Built a systematic AI for Compiler research framework for high-dimensional discrete search, optimization-effect prediction, and automated decision-making in LLVM compiler optimization, focusing on **search-space modeling, program representation learning, and LLM/RL-based agentic tuning**.
- Led the construction of a complete research pipeline from **optimization-task definition, dataset construction, feature extraction, and search/learning algorithm design to evaluation and validation**, forming reusable experimental infrastructure and methodology for compiler auto-tuning.
- Developed several key infrastructures, including an **LLVM Pass Pipeline search platform, program-behavior probe generation pipeline, optimization-effect prediction datasets, LLM tool-interaction environment, and reinforcement-learning training pipeline**, supporting continuous research iteration across multiple directions.
- Related work systematically covers core problems such as **compiler search-space pruning, quasi-dynamic program representation, and agentic compiler optimization**, leading to first-author publications at **CGO 2025, NeurIPS 2025, and ICLR 2026**.

Super C++ SIMD High-Performance Library, Project Lead 2023.03 - 2023.08

- Led the design and implementation of the **Super C++ SIMD library** for high-performance computing and low-level performance engineering, building a unified abstraction interface and auto-tuning framework for cross-SIMD-library co-optimization.
- Addressed the fragmentation of existing C++ SIMD libraries by designing a unified programming interface that supports mixed use of multiple SIMD libraries at the single-function granularity, enabling flexible cross-implementation composition and optimization.
- Built an **auto-tuning framework** for searching performance-optimal SIMD interface combinations, systematically exploring different SIMD interface composition strategies and achieving up to **120**× performance improvement over LLVM -O3 on selected tasks.
- Conducted in-depth research on library design and compiler co-optimization, combining reinforcement learning and search strategies to explore joint library-level and compiler-level optimization mechanisms.
- Related work was published at **ASE 2025 and SEKE 2025**, and resulted in two invention patents, both as the first inventor: "Automatic Testing Method and Apparatus Based on SIMD Libraries"（Patent No.: ZL 2023

1 0375698.4）；“Hybrid SIMD Library and Programming Method Based on Hybrid SIMD Library”（Patent No.: ZL 2023 1 0958237.X）.
- Technical coverage: C++ template metaprogramming, SIMD programming, auto-tuning, performance modeling, and compiler co-optimization.

Wafer-Scale Multi-Level Compilation Framework Based on MLIR, Core R&D Member 2022.12 - 2023.12

- Participated in the overall design and implementation of a multi-level compilation framework for wafer-scale heterogeneous computing, building a complete technical pipeline from DSL representation and IR design to backend optimization based on the **MLIR multi-level compilation stack**.
- Responsible for the design and implementation of a **front-end cryptography DSL**, providing unified program abstractions for complex operators and multi-stage computation flows.
- Responsible for the design and development of the **MLIR dialect and operator** system, including operator conversion and lowering procedures, with support for cross-level optimization.
- Built an MLIR-based **multi-level compilation optimization pipeline**, and conducted targeted optimizations such as loop-structure rewriting and vectorization for cryptographic computing scenarios on the X86 platform.
- Technical coverage: MLIR pass development, dialect design, vector dialect optimization, and compiler optimization for cryptographic algorithms.

Internship Experience

PLCT Lab, Institute of Software, Chinese Academy of Sciences, Compiler Intern 2021.03 - 2021.09

- Participated in the implementation of **C++ standard library std::simd** in the LLVM/Clang ecosystem, working on cross-platform vectorization abstraction, API design, and low-level instruction mapping.
- Implemented multiple core interfaces of the **std::simd API**, and designed unified abstraction layers and functional adaptations across different hardware backends such as SSE and AVX, improving portability and maintainability across architectures.
- Built the overall performance evaluation system for the SIMD project, and conducted systematic benchmark comparisons against mainstream C++ SIMD libraries including **Highway, xsimd, nsimd, and VCL**.
- The implementation work has been accepted upstream by LLVM（LLVM PRs）.

Patents and Academic Service

- **First Inventor**, “Automatic Testing Method and Apparatus Based on SIMD Libraries”, Patent No.: ZL 2023 1 0375698.4
- **First Inventor**, “Hybrid SIMD Library and Programming Method Based on Hybrid SIMD Library”, Patent No.: ZL 2023 1 0958237.X
- **NeurIPS 2025 Reviewer**
- **NeurIPS 2026 Reviewer**
- **ICLR 2026 LLA Workshop Reviewer**
- **ICASSP 2026 Reviewer**

Professional Skills

- **Research Areas:** AI Infra, Compiler Optimization, Compiler Auto-Tuning, Program Representation Learning, SIMD Optimization
- **Technical Stack:** LLVM, MLIR, C/C++, Python, Shell, Linux
- **Core Competencies:** Problem abstraction, algorithm design, system implementation, experimental evaluation, paper writing, academic presentation

Honors and Awards

- Outstanding Student Model of UCAS
- Outstanding Student Leader of UCAS
- Outstanding Student of UCAS

Other Experience

- **Open Source Promotion Plan Mentor**: Supervised students in open-source project practice, focusing on the design and implementation of MLIR vectorization benchmarks.

- **MedLit: AI-Powered Medical Literature Learning Platform**: Independently developed an AI-powered medical literature learning platform, including backend system development, model API integration, deployment, and operations.