Minxuan Zhou — Research Statement

My vision is to build next-generation computer systems that leverage cutting-edge hardware technologies to tackle the extreme challenges of processing emerging applications on conventional systems. My research involves extensive co-design of hardware and software that innovates hardware systems driven by application needs and develops software to unlock the hardware's potential.

I am currently a Lead Postdoctoral Researcher at the Center for Processing with Intelligent Storage and Memory (PRISM), one of seven large, multi-disciplinary academic research centers in Joint University Microelectronics Program 2.0 (JUMP 2.0) funded by Semiconductor Research Corporation (SRC) and DARPA. I am leading a team of students at UCSD in multiple research projects that involve interdisciplinary efforts in algorithms, systems, architecture, and circuit design. I am actively collaborating with researchers from different universities and SRC member companies, including UCSD, UVA, PSU, UIUC, UW-Madison, Stanford, UCI, UCLA, Intel, IBM, Micron, SK Hynix, Samsung, etc. In my doctoral and postdoctoral research, I participated in writing several proposals that led to research funding from NSF, DARPA, SRC, and industrial companies.

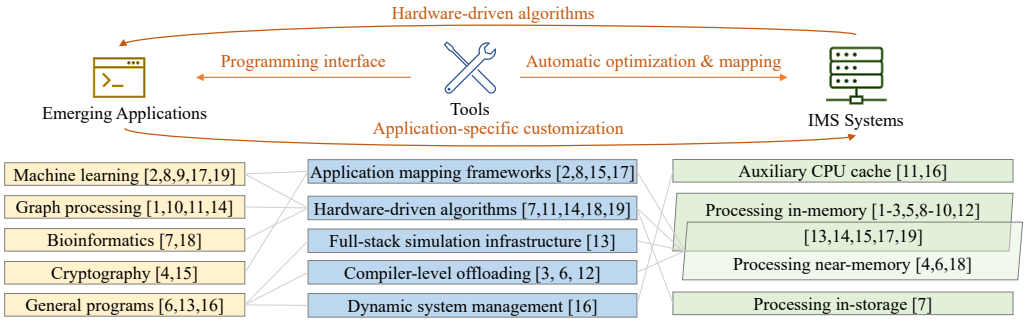

Figure 1: *The overview of my research that builds efficient and practical IMS systems.*

My research focuses on **intelligent memory and storage (IMS)**, a technology that promises to overcome the significant challenges of processing emerging data-intensive applications on conventional systems. However, as a novel technology supports non-conventional processing paradigms, building efficient IMS systems requires not only **holistic co-optimization of hardware architectures, system software, and application mappings** but also **tools that assist the system development and deployment**. Motivated by these challenges, my research (Figure 1) investigates the full-stack design of IMS-enabled systems to accelerate various emerging applications with an exploration of diverse IMS technologies. A significant part of my research is building efficient IMS systems for machine learning (ML) workloads. I developed the first compiler that optimizes the holistic mapping of ML models onto IMS systems [8] and the first IMS-based software-hardware co-design for Transformers [19]. In addition to ML workloads, my research explored various critical but challenging application fields including bioinformatics, graph processing, and post-quantum cryptography. A noteworthy outcome is that my collaboration with Intel on the DARPA DRIVE program contributed to a new chip product for accelerating post-quantum cryptography [4, 15]. Furthermore, I developed novel system tools to improve the efficiency, usability, and reliability of IMS systems. **Moving forward**, I aim to pioneer future systems, such as large-scale heterogeneous systems and embedded systems, that exploit cutting-edge hardware technologies for computing, data storage, and interconnection to accelerate emerging applications in different scenarios. I plan to develop software, such as AI-powered design automation tools and general compiler infrastructures, to help develop and deploy future systems. Furthermore, I am enthusiastic about innovating algorithms based on non-conventional hardware paradigms for emerging applications.

IMS Acceleration of Machine Learning with Mapping and Hardware Optimizations

Machine learning (ML) models have become increasingly important and require tremendous computing resources. Due to its data-intensive nature and regularized computation, ML workload is a promising target for **processing in-memory (PIM)**, an IMS technology that supports highly parallel arithmetic

operations with extensively large bandwidth. With collaborators and mentees, I designed novel software tools and hardware architectures for ML accelerators based on PIM technologies.

Holistic optimization of mapping ML models onto PIM accelerators

Each operator (i.e., layer) in ML models contains extensive parallel computations that require different input data segments. Therefore, the scheduling of computations on ML accelerators directly impacts the hardware data flow, resulting in the ML mapping problem. However, mapping an ML model onto PIM accelerators is significantly different from and more complicated than traditional ML accelerators. First, unlike conventional ML accelerators that process a small number of layers at a time, PIM accelerators require a holistic mapping of all layers to utilize the memory fully. Furthermore, each PIM mapping requires a unique data layout in the memory, resulting in mutual impacts between the mappings of different layers in holistic optimization. To tackle these challenges, I developed `PIM-DL` [8], the first ML-PIM mapping framework that efficiently explores the design space via a comprehensive data layout modeling and an efficient optimization algorithm. I further developed several mapping tools [2, 17] that consider more fine-grained mapping (e.g., computation overlaps between layers) to improve the throughput. These are **the earliest efforts that systematically investigated the mapping of ML models on IMS accelerators**, significantly easing the efficient deployment of ML-PIM acceleration.

PIM acceleration for emerging ML models

My recent research focused on PIM acceleration for emerging ML models such as models with attention, one of the most significant algorithm innovations in recent years. The attention mechanism is computing- and memory-intensive with limited scalability due to extensive memory footprint and sparse data locality. When processing attention-based models on PIM, the layer-based data layout, considered by `PIM-DL` [8], introduced a significant overhead due to cross-layer data movements. This necessitates novel data flow optimizations for attention-based models. To tackle these challenges, I proposed `MAT` [9], a PIM accelerator with an optimized pipeline execution for long-sequence attention. In collaboration with biomedical experts at UW-Madison, `MAT` [9] outperforms GPUs by up to **40** $\times$ on high-resolution biomedical image classification. Furthermore, I proposed `TransPIM` [19], which targets Transformer, the most widely used attention-based model. `TransPIM` adopts a new token-sharding data layout to reduce the data movement overhead in Transformer. Furthermore, `TransPIM` features a novel hybrid PIM hardware, that integrates low-cost near-memory logic in memory to accelerate Transformer operations. Such software-hardware co-design enables `TransPIM`, **as the first PIM-Transformer accelerator**, to provide **22**$\times$ **to 115**$\times$ speedup over state-of-the-art GPUs. I also explored PIM to accelerate models with algorithmic optimizations. With collaborators at the Pennsylvania State University, I proposed an end-to-end PIM framework to support Transformers with dynamic token pruning [5]. In addition, I designed PIM accelerators for non-neural-network ML models, like hyperdimensional computing [1].

Software-Hardware Co-Design of IMS Systems to Accelerate Emerging Applications

In addition to ML-PIM acceleration, my research explored broad IMS technologies with hardware-software co-design for emerging applications that are critical for advancing life science (bioinformatics) [7, 18], processing big data for deep insights (graph processing) [10, 11, 14], and securing data privacy (fully holomorphic encryption) [4, 15].

Scalable near-data processing for accelerating bioinformatics

With collaborators from the University of Virginia and the Department of Bioinformatics at UCSD, I designed two near-data processing (NDP) systems [18, 7] for genome analysis which is an essential task in bioinformatics for detecting disease (e.g., cancer) and biological information in environmental samples (e.g., COVID virus). The critical concern of designing bioinformatics IMS systems is scalability that supports efficient data processing at scale. Furthermore, genome analysis requires intensive non-arithmetic operations. These facts make NDP, which supports scalable general-purpose processing, preferable to other IMS technologies. In [18], I designed a near-memory accelerator for *de Bruijn* graph, which assembles a whole genome sequence based on highly fragmented and duplicated short sequences

(i.e., reads) without a reference genome. I implemented an NDP-parallel *de Bruijn* graph algorithmic framework to utilize the memory bandwidth fully when scaling up. Furthermore, to accelerate genome analysis tasks on systems that require storage devices for large datasets, I designed `Abakus` [7], which accelerates k-mer (i.e., a DNA sequence with length k) counting, a performance bottleneck in most genome analysis applications. `Abakus` [7] is based on solid-state drive (SSD), which supports arbitrarily large bioinformatics data when memory scaling is too expensive. Overall, these accelerators are at least **10**$\times$ faster than conventional systems, and speedups significantly increase when scaling up the system.

Exploit PIM to parallelize computations with random data patterns in graph processing

Graph processing is known as challenging due to random access patterns on big data. In `GRAM` [10], I proposed a PIM-based vertex program algorithm that parallelizes the graph processing operations by exploiting the extremely long vector processing capability in the PIM accelerator. To tackle the challenges of random data dependency, the PIM-based algorithm adopts in-memory search operations to fetch a large batch of random data for computation efficiently. With such hardware-driven algorithm re-design, `GRAM` outperformed prior PIM-based accelerators on classic graph algorithms by 4$\times$. Furthermore, I proposed `HyGraph` [14], which adopted `GRAM`'s algorithm on a hybrid PIM-NDP accelerator with a graph-aware operation scheduling, achieving 2$\times$ speedup over PIM-only GRAM.

Memory acceleration and optimization for fully homomorphic encryption

In the last two years, at UCSD and Intel Labs, I have been collaborating with cryptography and chip design experts on hardware acceleration for fully homomorphic encryption (FHE), an important post-quantum cryptography algorithm. FHE introduces at least 4 orders of magnitude data and computation overhead, as compared to the plaintext computation. Such challenges necessitate specialized systems with high computation throughput and memory bandwidth to enable practical FHE applications. I proposed FHEmem [15], an FHE accelerator that adopted a novel PIM architecture with cost-efficient in-memory logic to support complicated FHE operations. With an FHE-PIM compilation framework that optimizes the data layout of pipeline execution, FHEmem is **4**$\times$ faster with **7**$\times$ better performance-cost efficiency than state-of-the-art FHE accelerators, leading to **5 orders of magnitude** improvement over conventional systems. Furthermore, I tackled issues in real systems equipped with FHE accelerators by architectural and compiler-level optimizations. For example, I proposed a new interconnect architecture and memory allocation optimizations to solve the memory under-utilization problem when scaling up the FHE accelerator system [4]. My research has contributed to **the chip design of an upcoming FHE accelerator product**, which will be used in future data centers for securing data privacy efficiently.

Software Tools for developing and deploying IMS architecture

The software tools, including simulator, compiler, and system runtime, are critical to the development and deployment of novel systems. In addition to the application-specific mapping frameworks and algorithms mentioned above, I developed several tools for emerging IMS systems to ease the design process [13], automate program scheduling [6], and improve system reliability [3, 12, 16]. `DP-SIM` [13] is a **full-stack simulation infrastructure** for exploring software-hardware co-design of PIM architectures. `DP-SIM` has a front-end library for PIM-enabled algorithm implementation, a middle-end memory allocation framework for compiler-level scheduling, and a back-end PIM simulator for architectural exploration. `PIMProf` [6] is an **LLVM-based PIM offloading tool**, that can automatically offload operations in off-the-shell programs to PIM-enabled main memory in conventional systems. `PIMProf` tackles the challenges of deciding appropriate program blocks that are offloaded to memory-side processors to improve the system performance. I proposed **compiler-level thermal optimization tools** that use static optimization and dynamic management of data layout in emerging memories, including PIM accelerators [3, 12] and 3D DRAM cache [16], to resolve reliability issues caused by temperature violations.

Future Directions

My previous research found that emerging IMS technologies hold great promise for use in a variety of application domains. I recognized the crucial role of software support in ensuring system efficiency and

reliability. In the future, I plan to deepen my research in designing novel systems for real-world emerging applications via hardware innovations, full-stack software tools, and technology-driven algorithms.

Technology-driven design of future systems for emerging application scenarios

As emerging applications become increasingly challenging but different applications significantly vary in computation and data patterns, innovating computer systems based on various cutting-edge hardware technologies has become essential to continue speeding up computation. Given my experience in hardware-software co-design for emerging data-intensive applications, I am enthusiastic about designing the next-generation large-scale heterogeneous system, which consists of various accelerators, emerging memory and storage, and advanced interconnection technology (e.g., compute-express links). I believe a holistic innovation of the whole system is critical to boosting the performance of more and more challenging applications. I am interested in expanding my current work on machine learning, bioinformatics, graph processing, and cryptography to large-scale heterogeneous systems, that provide significant efficiency improvement on big data. In addition to application-specific acceleration, I plan to investigate more general-purpose heterogeneous systems that support various application domains with the minimum customization cost. This is inspired by GPU, where the high-throughput compute units can significantly accelerate both graphics and machine learning applications. In future heterogeneous systems, different applications can share hardware components, including accelerators for commonly used kernels and function-augmented general-purpose components such as cache, memory, storage, and interconnect. In addition to large-scale heterogenous systems, I am interested in conducting similar research for designing other systems such as low-power embedded systems for supporting emerging IoT applications by new hardware technologies.

Full-stack support for future heterogeneous systems with emerging hardware technologies

As systems become more heterogeneous with emerging hardware technologies for various applications, design optimization and efficient utilization of such systems are increasingly challenging. Given my experience in software-hardware co-design of accelerators and design/deployment automation tools, I plan to develop the end-to-end stack for next-generation heterogeneous systems. The full-stack support includes the **programming interface**, **compiler**, **system runtime**, and **design automation tools**. For example, one of my ongoing projects, collaborating with researchers at UIUC, is to develop the general compiler infrastructure and runtime management for heterogeneous and distributed systems with various IMS devices connected by novel interconnect. The infrastructure aims to provide the programming interface to exploit IMS operations in C++ programming using the LLVM facilities and adopt several compiler-level optimization passes to adaptively optimize the data layout and operation scheduling based on the system architecture. I plan to expand the compiler infrastructure to support heterogeneous systems with a variety of accelerators that share the same intermediate compiler layers. Furthermore, I plan to develop tools with **mathematical optimization** and **machine-learning** techniques to automate the design of hardware architecture and optimize system runtime management for emerging systems, for which I have started a collaboration with Intel Labs and plan to continue contributing. These tools can significantly speed up the development and deployment of next-generation systems.

Algorithm innovation with non-conventional computing on emerging hardware

Emerging hardware technologies support non-conventional computing paradigms such as processing using-memory, quantum computing, analog computing, and biological computing. Even though it is possible to exploit such emerging hardware to accelerate classical algorithms, the incompatibility between the hardware mechanism and conventional computing may offset the benefits of hardware innovation. As shown in my work on graph acceleration [10, 14], algorithm innovation might be critical to exploit the full potential of novel hardware with non-conventional computing. I plan to design hardware-friendly algorithms for emerging applications that use the native characteristics of cutting-edge technologies. Redesigning or approximating algorithms with such non-conventional operations can achieve significantly higher efficiency than directly computing the classic algorithms. Such hardware-driven non-conventional algorithms are possible game-changers in the future of computer science.

References

[1] KANG, J., ZHOU, M., BHANSALI, A., XU, W., THOMAS, A., AND ROSING, T. Relhd: A graph-based learning on fefet with hyperdimensional computing. In *2022 IEEE 40th International Conference on Computer Design (ICCD)* (2022), pp. 553–560.

[2] LIU, X., ZHOU, M., AUSAVARUNGNIRUN, R., EILERT, S., AKEL, A., ROSING, T., NARAYANAN, V., AND ZHAO, J. Fpra: A fine-grained parallel rram architecture. In *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (2021), pp. 1–6.

[3] LIU, X., ZHOU, M., ROSING, T. S., AND ZHAO, J. Hr3am: A heat resilient design for rram-based neuromorphic computing. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (2019), pp. 1–6.

[4] NAM, Y., ZHOU, M., GUPTA, S., DE MICHELI, G., CAMMAROTA, R., WILKERSON, C., MICCIANCIO, D., AND ROSING, T. Efficient machine learning on encrypted data using hyperdimensional computing. In *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (2023), pp. 1–6.

[5] PAN, Y., ZHOU, M., LEE, C., LI, Z., KUSHWAH, R., NARAYANAN, V., AND ROSING, T. Primate: Processing in memory acceleration for dynamic token-pruning transformers. In *29th Asia and South Pacific Design Automation Conference (ASP-DAC)* (accpeted).

[6] WEI, Y., ZHOU, M., LIU, S., ROSING, T., AND KHAN, S. Pimprof: An automated program profiler for processing-in-memory offloading decisions. In *2022 Design, Automation and Test in Europe Conference (DATE'22)* (2022).

[7] WU*, L., ZHOU*, M., XU, W., VENKAT, A., ROSING, T., AND SKADRON, K. Abakus: Accelerating k-mer counting with storage technology. *ACM Trans. Archit. Code Optim.* (nov 2023). Just Accepted.

[8] ZHOU, M., CHEN, G., IMANI, M., GUPTA, S., ZHANG, W., AND ROSING, T. Pim-dl: Boosting dnn inference on digital processing in-memory architectures via data layout optimizations. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)* (2021), pp. 1–1.

[9] ZHOU, M., GUO, Y., XU, W., LI, B., ELICEIRI, K. W., AND ROSING, T. Mat: Processing in-memory acceleration for long-sequence attention. In *2021 58th ACM/IEEE Design Automation Conference (DAC)* (2021), pp. 25–30.

[10] ZHOU, M., IMANI, M., GUPTA, S., KIM, Y., AND ROSING, T. Gram: Graph processing in a reram-based computational memory. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference* (New York, NY, USA, 2019), ASPDAC '19, Association for Computing Machinery, p. 591–596.

[11] ZHOU, M., IMANI, M., GUPTA, S., AND ROSING, T. Gas: A heterogeneous memory architecture for graph processing. In *Proceedings of the International Symposium on Low Power Electronics and Design* (New York, NY, USA, 2018), ISLPED '18, Association for Computing Machinery.

[12] ZHOU, M., IMANI, M., GUPTA, S., AND ROSING, T. Thermal-aware design and management for search-based in-memory acceleration. In *Proceedings of the 56th Annual Design Automation Conference 2019* (New York, NY, USA, 2019), DAC '19, Association for Computing Machinery.

[13] ZHOU, M., IMANI, M., KIM, Y., GUPTA, S., AND ROSING, T. Dp-sim: A full-stack simulation infrastructure for digital processing in-memory architectures. In *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)* (2021), pp. 639–644.

[14] ZHOU, M., LI, M., IMANI, M., AND ROSING, T. Hygraph: Accelerating graph processing with hybrid memory-centric computing. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)* (2021), pp. 330–335.

[15] ZHOU, M., NAM, Y., GANGWAR, P., XU, W., DUTTA, A., SUBRAMANYAM, K., WILKERSON, C., CAMMAROTA, R., GUPTA, S., AND ROSING, T. Fhemem: A processing in-memory accelerator for fully homomorphic encryption. `https://arxiv.org/abs/2311.16293`, 2023.

[16] ZHOU, M., PRODROMOU, A., WANG, R., YANG, H., QIAN, D., AND TULLSEN, D. Temperature-aware dram cache management—relaxing thermal constraints in 3-d systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39*, 10 (2020), 1973–1986.

[17] ZHOU*, M., WANG*, X., AND ROSING, T. Overlapim: Overlap optimization for processing in-memory neural network acceleration. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2023), pp. 1–6.

[18] ZHOU*, M., WU*, L., LI, M., MOSHIRI, N., SKADRON, K., AND ROSING, T. Ultra efficient acceleration for de novo genome assembly via near-memory computing. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)* (2021), pp. 199–212.

[19] ZHOU*, M., XU*, W., KANG, J., AND ROSING, T. Transpim: A memory-based acceleration via software-hardware co-design for transformer. In *The 28th IEEE International Symposium on High-Performance Computer Architecture (HPCA-28)* (2022).