Statement of Purpose

Yichuan Wang, 2024 Fall Ph.D. Application

With the rapid advancement of machine learning in recent years, the importance of constructing more efficient and reliable computer systems tailored for it has grown exponentially. I am passionate about building next-generation computer systems, particularly those designed for machine learning. My approach will encompass perspectives from the levels of machine learning algorithms, systems, and hardware. By integrating insights from these levels, I aim to enhance the performance of current machine learning systems in terms of accuracy, efficiency, scalability, reliability, and cost-effectiveness through user-friendly methods.

I've always been fascinated by the rigor of computer systems and the power of artificial intelligence. During my undergraduate years, I also saw the success of some machine learning systems projects like Torch [5], TVM [1], and DGL [6].In my sophomore year, I decided to focus on researching machine learning systems. My subsequent engagements with renowned institutions like **EPCC lab at SJTU, MSRA, SystemLab@NYU, and AWS Shanghai AI Lab (ASAIL)**, have been instrumental in shaping my research ethos. I worked extensively on the development of efficient, large-scale, cutting-edge software systems, with a special emphasis on machine learning. At these places, I enhanced my skill set and was proud to publish my first paper (**OSDI '23**). Additionally, attending premier systems conferences like OSDI, ATC, and SIGCOMM further broadened my horizons and intensified my zeal to make a significant mark in this area. Beyond the scope of research, my role as an instructor has been equally enriching. Serving as an instructional assistant and leading the TA team has been a gratifying experience for me.

This blend of research and teaching has solidified my enthusiasm for becoming a systems researcher and an inspiring educator. These passions serve as the driving force for my aspiration to pursue a Ph.D. in Computer Science and Engineering at the University of Pennsylvania. My primary focus will be on the intersection of systems and machine learning (machine learning system) – a domain that nearly encapsulates all my research interests.

Past Research: Shanghai Jiao Tong University & Microsoft Research Asia

In June 2022, I joined the EPCC lab at SJTU to embark on a research journey in machine learning systems under the guidance of **Prof. Quan Chen**. We collaborated with MSRA(Microsoft Research Asia) to work on **optimization for dynamic neural network inference**. Currently, dynamic neural networks (such as Mixture of Experts, MoE) are being widely used in various large-scale models. However, there is still no effective framework to dynamically dispatch data to different paths during inference. Our insight was that deterministic execution, using the granularity of tensors, makes it difficult to trace these dynamic execution patterns, thereby reducing opportunities for optimization. Consequently, we built a compiler-like framework to improve the inference efficiency of dynamic neural networks with a concise API. I helped design the new abstraction, explored a larger optimization space with a profile-guided method, and was the main contributor to developing two important components. First, inspired by the IR optimization of traditional compilers, I designed and implemented "Speculative Routing." Utilizing torch.fx and profile data, I can manipulate the data flow graph, which introduces a suite of optimizations, including dead code elimination, constant propagation, operator reordering, etc. Building on this, I discovered that the same approach could also address the issue of the inability to overlap weight loading and computation in dynamic neural networks. To this end, I implemented a technique called "Speculative Weight Loading," which successfully reduced the GPU memory usage to just 43.5% of the original amount. Secondly, I designed and implemented an automatic dynamic mechanism for horizontal fusion. Based on Rammer(OSDI20) [3], I was able to first tune the vertical kernels using TVM, and then use profiled data along with heuristic algorithms to identify kernels suitable for horizontal fusion, thereby fully utilizing the GPU. These combined efforts resulted in up to an 11.7× speedup across our benchmark tests. I'm proud to share that our concerted efforts were acknowledged with our paper "Brainstorm" being accepted at **OSDI 2023** [2].

After the initial "Brainstorm", I noted the challenges in deploying large MoE models and managing runtime drift in distributed inference scenarios, which could be further explored as future work. This project is ongoing, and a key insight is the unique activation patterns of experts across layers. Understanding these patterns can help reduce communication costs by placing experts according to the profiled distribution. The impact of different datasets on the distribution of activated experts also presents a challenge. This is crucial for optimal expert placement and their real-time adjustment in practical LLM scenarios.

New York University & AWS Shanghai AI lab

Having experienced the pleasure and success from my former research experience in Machine Learning systems, I continued my journey in that field with **Prof. Jinyang Li** at NYU. I am now actively collaborating with the Shanghai AWS DGL team (led by **Dr. Minjie Wang**) as well. I began to explore how to **scale up GNN training** from a very beginning stage. My motivation for using SSD in GNN training stems from observing that distributed training can't effectively scale up for GNNs with large input data patterns and often suffers from loss of accuracy. Additionally, the few existing works on scaling up GNN training on SSDs experience significant read amplification and are also ineffective at hiding data transfer time due to the low bandwidth of SSDs [4]. I have come up with an algorithm and system co-design method that allow for complete overlap of data transfer and computation without read amplification and can scale up well. From the algorithm perspective, considering the trade-off between randomness (corresponding to training quality) and system efficiency, I designed an offline mega batch sampling method that only slightly destroys the iid property of data loading. Combined with a recompute design in CPU memory, this method can share data in the CPU between every minibatch. Moreover, I prove that our scheme has no loss in training accuracy. From the system perspective, I have designed an effective IO engine and a data structure for SSDs that can achieve trade-offs between read amplification and storage space optimization. As for the results, it appears that we will be 13.0x faster than the baseline in data transfer, due to our efficient IO engine, and 4.1x faster in end-to-end performance, due to our overall system design.

During my time at NYU, I actively organized and participated in other side projects. Currently, one interesting project is on the topic of **LLM-based AI agents**. As the main developer, I designed a collaboration method for AI agents that can achieve a significant improvement in task accuracy over simple chain-of-thought approaches. This project is expected to be submitted to NAACL this year.

In addition to the two research experiences I mentioned earlier, to enhance my system knowledge and sharpen my engineering skills, I actively challenged myself with projects that I had no prior experience with and crafted things from scratch during my undergraduate study. In my sophomore year, I mastered Verilog HDL and digital circuit design within two weeks. Over the following two months, I dedicated myself to developing a simplified 5-stage-pipeline, user-mode RISC-V CPU on an FPGA. This project included dynamic speculation and a fully operational cache. In the same semester, I taught myself compilation techniques and advanced compiler optimizations and spent 5 months implementing a compiler, which translates a Java-like language to x86-64 NASM. I designed an IR myself, which mimics LLVM IR but is modified to achieve a better optimization effect on x86-64 CPUs.

Future Research and Plan In my future Ph.D. research, I aim to explore the creation of practical systems for efficient machine learning and data analysis, focusing on enhancing efficiency, scalability, and reliability.

- **Efficiency**: Build domain-specific end-to-end systems with improved performance by algorithmic innovation, system design optimization, and hardware-specific adaptations.
- **Scalability**: Propose advanced distributed ML frameworks that enhance GPU utilization, optimize communication overhead, and implement optimizations at the cluster or cloud level for ML.
- **Reliability**: Design reliable ML systems with better fault tolerance, and optimize machine learning tasks from system level like security, privacy, completeness, and other aspects.

I am thrilled to apply for the Ph.D. program at xxx. I am especially interested in working with Prof. xxx, who consistently conducts the best machine learning system research. I am familiar with and inspired by his achievement in distributed machine learning systems, including model serving and data center work. Additionally, Prof. xxx description on his homepage about the next generation of data systems is of great interest to me, not to mention his existing accomplishments in learned indices and databases. I am excited to learn from and collaborate with these distinguished faculty members and research communities, and I believe that the University of Pennsylvania will provide me with the necessary expertise to excel as an excellent researcher.

References

[1] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594, 2018.

[2] Weihao Cui, Zhenhua Han, Lingji Ouyang, Yichuan Wang, Ningxin Zheng, Lingxiao Ma, Yuqing Yang, Fan Yang, Jilong Xue, Lili Qiu, et al. Optimizing dynamic neural networks with brainstorm. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 797–815, 2023.

[3] Lingxiao Ma, Zhiqiang Xie, Zhi Yang, Jilong Xue, Youshan Miao, Wei Cui, Wenxiang Hu, Fan Yang, Lintao Zhang, and Lidong Zhou. Rammer: Enabling holistic deep learning compiler optimizations with {rTasks}. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 881–897, 2020.

[4] Yeonhong Park, Sunhong Min, and Jae W Lee. Ginex: Ssd-enabled billion-scale graph neural network training on a single machine via provably optimal in-memory caching. *Proceedings of the VLDB Endowment*, 15(11):2626–2639, 2022.

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[6] Minjie Yu Wang. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR workshop on representation learning on graphs and manifolds*, 2019.