Applicant: Dacheng Li | Homepage: dacheng-li.info | email: dacheng2@cs.cmu.edu

Statement of Purpose

Machine Learning (ML) has made great advances in recent years, but adopting it in real-world applications imposes challenges beyond statistical methods. These challenges call for better system and infrastructure support for real-world performance and usability requirements in every stage of the ML pipeline. For instance, *training* systems need to consider scalability for large models, as well as heterogeneity for in-house clusters, while *serving* systems must minimize latency and protect users' privacy. Many challenges need to be addressed in the next few years, and I want to pursue a Ph.D. degree to tackle them.

In particular, I study large-scale Machine Learning in the intersection of **ML** and **distributed systems**. My goal is to codesign ML models, algorithms, and systems to support **performant** models at scale, and develop **easily usable** frameworks for people, to facilitate ML deployment in the real world. In my pre-doctoral research, I worked on democratizing distributed ML training, privacy-preserving ML serving, and large-scale ML algorithms. My research has been published in NeurIPS'22, CVPR'21, submitted to ICLR'23 (likely to be accepted), MLSys'23, and recognized by Amazon Research Awards. In my Ph.D. study, I hope to gain a more holistic view and work on the most impactful problems. Below I will summarize my research development.

Large-scale Machine Learning algorithms. I started ML research in my undergraduate with Prof. Zhuowen Tu. We found that existing generative models only specialized in a subset of ML tasks, which hindered more general usage. During experiments, I found that incorporating an instance-level contrastive loss was promising in well-rounded performance. Effectively using it, however, requires a large number of negative samples, which is far beyond what one GPU can store. To address this, I resorted to scalability literature and leveraged the data-parallel distributed training. The final model is the *first* generative model that can perform *simultaneous* image synthesis at the Generative Adversarial Nets (GAN) level, and image reconstruction and representation learnability at Variational Auto-Encoder (VAE) level. This work was accepted at **CVPR 2021** (co-first author). Reflecting on this experience, I found that besides better model designs, system support can provide fundamentally new opportunities for improvement.

Democratizing distributed ML training. With the curiosity of exploring ML Systems, I joined Prof. Eric Xing's group during my master's study at CMU. Through study of contemporary ML training systems, e.g. DeepSpeed, I found that they required experts to design model-parallel strategies but only provided solutions to a limited industry scenario. In practice, people are system outsiders but are training beyond these setups. To develop a general solution, I proposed a framework that can **automatically** find good model-parallel strategies, especially for **heterogeneous** models and clusters.

An automatic approach, however, requires *slow* real trials to evaluate model-parallel strategies and there are *exponentially* many. To enable automation, I took a two-step approach. Firstly, I developed a cost model as a fast proxy for real trials. During development, I decomposed model parallelism into different pieces, and examined each against the wall clock execution time. For instance, I examined that communication in the tensor model parallelism is asymptotically constant to the number of workers. Secondly, I developed a dynamic programming-based optimization procedure to bring down the number of evaluations to polynomially many. The end framework, AMP, is the first **automatic** approach in the complex and expressive space of data parallelism, tensor model parallelism, and pipeline model parallelism. It returns strategies that

are **1.54x** and **1.77x** better on heterogeneous clusters and models setup, and that match expert-designed ones on industrial setup. This work is supervised by Prof. Eric Xing and Prof. Hao Zhang, and accepted at **NeurIPS 2022** (first author).

When validating the cost model in AMP, I found that communication is a major bottleneck in model parallelism. I further studied communication compression as a potential solution with Prof. Shivaram Venkataraman. In this project, my collaborators and I conducted the *first* empirical study on model-parallel compression for a wide range of real-world setups, including different families of compression methods, hyper-parameters and hardware. We discovered that only *learning-based* compressors exhibit acceptable tradeoff between ML performance and the system throughput. Based on the throughput data, I further developed a cost model to help users predict the speed-up and decide whether to apply learning-based compressors for their training needs. This work is submitted to **MLSys 2023** (co-first author).

Privacy-preserving ML model serving. Positive feedback in previous projects encouraged me that both performance and usability are important. This led me to investigate the usability in another type of ML systems - serving systems. My supervisors and I discovered that many people were reluctant to use GitHub Copilot service, a powerful Transformer-based code generator, because it requires users to send codes to the cloud. To meet this privacy demand, I proposed an inference framework based on Secure Multi-party computation (MPC).

However, using MPC led to other real-world concerns. Many functions in Transformers, e.g. Softmax, require heavy communication in MPC, which leads to an unacceptably *slow* inference. Moreover, different service providers choose different serving systems, which requires a general and effective solution for many MPC systems. To address this, I formulated an approximation abstraction, which replaces slow functions with their MPC-friendly alternatives. This abstraction allows a *12.5x* speedup for Softmax, but introduces an ML performance drop. I further introduced a Knowledge Distillation(KD)-based solution, which positions the original Transformer as the teacher. The end framework, MPCFormer, co-designs the ML model, the training algorithm, with the MPC system, and achieves well-rounded performance in **privacy**, **latency**, and ML **performance**. In particular, it demonstrates a **5.93x** speedup compared to contemporary BERT models serving in MPC with comparable ML performance. This work is supervised by Prof. Eric Xing and Prof. Hao Zhang, and submitted to **ICLR 2023** (first author). This research has been well received by reviewers (scores: 8/8/8/6, borderline: 5.5, decision: Jan 21) and recognized by the **Amazon Research Awards**.

Engineering Efforts. Besides Research, I enjoy spending **engineering** efforts on impactful ML systems. In particular, I (co-)developed the NCCL backend for the Ray system, and several resource scheduling policies for the AdaptDL system. I also built the recommendation system for UniLink, which became an AI startup that attracted more than 200 users. These experiences significantly influenced my research problem choice, where I found iterating with real users, solving the problems they care about, and seeing my codes really being used rewarding.

Future Plans and fit. Ph.D. is my intermediate goal. In the long term, I hope to develop better ML system support to deliver ML techniques to people in the real world, and to really improve their lives. Besides developing generic ML systems, I am also looking into more concrete application domains, such as ML for science (for example, in the field of biology) and ML for enhancing human abilities (such as through program synthesis). My career goal is both professorship and entrepreneurship, as I believe academia and industrial environments offer unique opportunities for gaining insight into real-world needs and developing effective system support.