# GUYUE HUANG

Email: hguyue1@gmail.com • Phone: (+1) 571-523-6855 • Linkedin • Google Scholar • Github • Homepage

## EDUCATION

**University of California, Santa Barbara**, Santa Barbara, CA 09/2020 - Present
Ph.D. Degree, Department of Electrical and Computer Engineering.

**Tsinghua University,** Beijing, China 09/2016 - 07/2020
Bachelor Degree, Department of Electronic Engineering.
Graduate with honor (Top 10%).

## EXPERIENCES

**Deep Learning Architect Intern**, NVIDIA, Santa Clara, CA. 06/2023 - 09/2023
• Work on a software feature that uses PyTorch 2.0 compiler to accelerate huggingface Parameter-Efficient Fine-Tuning (PEFT) models. The key method is to debug the torch dynamo tracing process to reduce graph breaks.
• Design and implement a software feature that leverages activation CPU offloading to reduce the memory requirement of context extension fine-tuning.
• Participate in the development and maintenance for the PEFT feature in the NVIDIA NeMo software package. Work on bug fixing related to the distributed training package (Megatron-LM).

**Deep Learning Architect Intern**, NVIDIA, Santa Clara, CA. 06/2022 - 09/2022
• Work on a software feature that reduces the memory usage of the Deep Learning Recommendation Model in Tensorflow.

**Research Intern**, Alibaba DAMO Academy, Shanghai China 08/2020 - 09/2021
• Work on a research project that develops an algorithm-software co-optimization framework for accelerating sparse DNN inference on GPUs. Publish a first-author paper in DAC'22.
• Participate in the designing and modeling of the sparse acceleration hardware feature on internal DNN accelerator.

## RESEARCH INTERESTS

My PhD research focuses on **enhancing the system and architecture of GPU for sparsity in deep learning**. I investigate software and architecture techniques to efficiently support sparsity forms including weight/activation sparsity, embedding operators and graphs. I am also interested in the general field of Deep Learning systems and accelerators, especially Deep Learning Compilers.

## PUBLICATIONS

• **Huang G.**, Wang Z., Tsai P., Zhang C., Ding Y., Xie Y. RM-STC: Row-Merge Dataflow Inspired GPU Sparse Tensor Core for Energy-Efficient Sparse Acceleration. In Proceedings of 56th IEEE/ACM International Symposium on Microarchitecture (MICRO), 2023.
• Wang Y., Feng B., Wang Z., **Huang G.**, Ding Y. TC-GNN: Bridging Sparse GNN Computation and Dense Tensor Cores on GPUs. *To appear* in the USENIX Annual Technical Conference (ATC), 2023.
• **Huang G.**, Bai Y., Liu L., Wang Y., Yu B., Ding Y., Xie Y. ALCOP: Automatic Load-Compute Pipelining in Deep Learning Compiler for AI-GPUs. In Proceedings of the Sixth Conference on Machine Learning and Systems (MLSys), 2023.
• Wang X., Wei Y., Xiong Y., **Huang G.**, Qian X., Ding Y., Wang M., Li L. LightSeq2: Accelerated Training for Transformer-based Models on GPUs. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2022.
• **Huang G.**, Li H., Sun F., Qin M., Ding Y. and Xie Y. Shfl-BW: Accelerating Deep Neural Network Inference with Tensor-Core Aware Weight Pruning. In Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC), 2022.
• Dai G., **Huang G.**, Yang S., Yu Z., Zhang H., Ding Y., Xie Y., Yang H. and Wang Y. Heuristic Adaptability to Input Dynamics for SpMM on GPUs. Design Automation Conference (DAC), 2022. (**Best Paper Nominee**).
• Zhang H., Yu Z., Dai G., **Huang G.**, Ding Y., Xie Y. and Wang Y. Understanding GNN Computational Graph: A Coordinated Computation, IO, and Memory Perspective. In Proceedings of Fifth Conference on Machine Learning and Systems (MLSys), 2022.

• **Huang G.**, Dai G., Wang Y., Ding Y. and Xie Y. Efficient Sparse Matrix Kernels based on Adaptive Workload-Balancing and Parallel-Reduction. Poster, ACM Student Research Competition (SRC), 2021.
• Yu Z., Dai G., **Huang G.**, Wang Y. and Yang H. Exploiting Online Locality and Reduction Parallelism for Sampled Dense Matrix Multiplication on GPUs. In Proceedings of the IEEE International Conference on Computer Design (ICCD), 2021.
• **Huang G.**, Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., Wu, J., Xu, Y., Zhang, H., Zhong, K., & others. Machine learning for electronic design automation: A survey. ACM Transactions on Design Automation of Electronic Systems (TODAES), 26(5), 1-46. 2021.
• **Huang, G.**, Dai, G., Yu, W., Huazhong, Y. GE-SpMM: General-purpose Sparse Matrix-Matrix Multiplication on GPUs for Graph Neural Networks. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2020.
• Li Q., Zhu H., **Huang G.**, Yu Z., Qiao F., Wei Q., Liu X., Yang H., Low-power in-pixel buffer circuit for smart image sensor. Sensor Review, 2020.

## AWARDS

- Best paper candidate, DAC 2022.
- ACM Student Research Competition (SRC), Graduate Student Third Place, 2021.
- MICRO'20 Student Research Competition (SRC), Undergraduate Student First Place.
- Travel grant of MLSys'23, MLSys'22.

## RESEARCH PROJECTS

### RM-STC: Row-Merge GPU Sparse Tensor Core for Energy-Efficient Sparse Acceleration

The goal of this project is to improve the state-of-the-art GPU sparse tensor core design towards better performance and higher energy efficiency. The proposed design, RM-STC, has two highlighted features: firstly, RM-STC supports sparse training (weight, activation, and gradient sparsity) without the overhead of sparse transposition; secondly, RM-STC achieves energy-efficiency and structure scalability via applying a row-wise dataflow. RM-STC achieves an average of 1.93× lower energy-delay product (EDP) than state-of-the-art STCs. This work is published at MICRO'23. (**Keywords: GPU Architecture, Tensor Core)**

### ALCOP: Automatic Load-Compute Pipelining in Deep Learning Compiler for AI-GPUs

The goal of this project is to bridge the gap between kernels generated by state-of-the-art deep learning compilers and hand-optimized by experts. We develop IR transformation for asynchronous copy and shared memory swizzling on NVIDIA Ampere GPU. We improve TVM auto-tuning efficiency by leveraging analytical hardware performance model. The method achieves an average of 1.23× speedup for dense kernels against out-of-the-box TVM. This work is published at MLSys'23. (**Keywords: DL Compiler, Software Pipelining)**

### Shfl-BW: Accelerating DNN Inference with Tensor-Core Aware Weight Pruning

The goal of this project is to develop a DNN pruning method that can achieve speedup over the tensor-core based dense neural network implementation, in contrast to prior work which mainly focused on theoretical computation reduction or non tensor-core baselines. We propose a novel parameter sparsity pattern, *Shuffled Block-wise sparsity*, to efficiently utilize tensor cores while minimizing the constraints on the weight structure. We optimize GPU kernels and achieve 4.18× speedup on NVIDIA T4 GPU for Transformer inference at 75% sparsity. This work is published at DAC'22. (**Keywords: GPU Sparse Kernel, DNN Pruning)**

### Optimizing GPU kernels and systems for Graph Neural Networks

I have conducted a series of projects for Graph Neural Network (GNN) systems, compilers and kernels (ATC'23, MLSys'22, DAC'22, ICCD'21, SC'20). Our fast sparse kernels and system optimizations have been integrated in popular GNN frameworks like PyTorch-Geometric and CogDL. In particular, my first-authored paper *GE-SpMM*, published at SC'20, presents an efficient Sparse-Dense Matrix Multiplication (SpMM) GPU kernel for GNNs. GE-SpMM uses thread coarsening to enhance data re-use, and pre-fetches a batch of sparse elements to the shared memory to enhance memory access coalescing. GE-SpMM was 1.41× faster than the cuSPRASE library at that time. My co-authored paper *DA-SpMM*, published at DAC'22, presents an input-adaptive SpMM GPU kernel design that can achieve 1.26-1.37× speedup over cuSPARSE library at that time. (**Keywords: GPU Kernel, SpMM)**