Lihan Hu

lihan-hu@uiowa.edu · (+1) 319-473-3260 · lihan hu

♡ Research Interest

Efficient Machine Learning Algorithm, GPU Programming, Parallel Computing, Compiler Optimization, Code Generation, Hardware-software Co-design, Graph Learning, Large Language Model Inference and Training

🎓 Education

The University of Iowa, Iowa, USA 2022 – Present

Ph.D. student in Computer Science **Advisor: Peng Jiang**

Hohai University, Nanjing, Jiangsu, China 2019 – 2022

Master in Pattern Recognition and Intellgent System **Advisor: Lixin Han**

Nanjing Audit University, Nanjing, Jiangsu, China 2014 – 2018

B.S. in Computer Science and Technology

ℹ Professional Experience

SK Hynix America San Jose, CA Nov. 2025 – Present

Research Intern on Memory Centric AI Machine Director: Jongryool Kim

The University of Iowa Iowa City, IA Sep. 2022 – Present

Research Assistant on Parallel Computing and Compiler Optimization Director: Peng Jiang

The City University of Hong Kong Kowloon, Hong Kong Sep. 2020 – Aug. 2021

Research Assistant on Imaging Processing Director: Hong Yan

Huazhong Univerisity of Science and Technology Wuhan, China Oct. 2016 – Feb. 2018

Research Intern on Disk Failure Detection Director: Ke Zhou

💬 Projects

Hardware-Software Co-Design for Sparse Attention Mechanism Oct. 2024 – Present

- Study current research on hardware-software co-design and PIM (Processing-In-Memory) architecture. We aim to build a heterogeneous system architecture and optimize the computation in transformer-based models, our system leverage the high memory bandwidth of PIM system for the attention layer and the powerful compute capability of the conventional system for the fully connected layer.

Compiler Optimization on Subgraph Matching Nov. 2023 – Jul. 2024

- Published in IPDPS'25. Matcha: A Language and Compiler for Backtracking-based Subgraph Matching.
- Extended our cuKE compiler for subgraph matching. Implemented apply operator for more flexible computation expression. Utilized kernel fusion, warp-level optimization, shared memory to improve the performance of generated code.

Performance Optimization on Knowledge Graph Embedding Training Sep. 2022 – Oct. 2024

- Published in IPDPS'24. cuKE: An Efficient Code Generator for Score Function Computation in Knowledge Graph Embedding.
- Analyzed the issue of computation, warp scheduling and synchronization overhead in current graph learning algorithm using GPU. Implemented PyTorch extension and custom CUDA operators to improve the performance of graph learning. Profiled the performance of knowledge graph embedding on Nvidia Nsight

Systems and Nvidia Nsight Compute. Replaced CUDA code with inline PTX code, used double buffering and vectorized data load on NVIDIA GPU to balance the overhead of data movement and computation.
- Built a compiler for score function computation in knowledge graph embedding. Proposed an aggressive operator fusion method to save memory cost and improve compute efficiency on GPU. Proposed a runtime inspection method to avoid redundant memory access.
- Published in IPDPS'25. Improving Accuracy and Efficiency of Graph Embedding Training with Fine-Grained Parameter Management.
- Proposed an analytical model to estimate the access frequency of embedding vectors to better arrange them on GPU and CPU memory. Proposed a fine-grained parameter duplication-and-precaching technique that improves the data access efficiency of knowledge graph embedding training with better accuracy. Our proposed method avoids write-conflict issues. Proposed a segmented parameter synchronization strategy to synchronize parameters across multiple GPUs to improve the embedding quality.

Structured Dynamic Sparse Training Mar. 2022 – Sep. 2022

- Published in NeurIPS'22. Exposing and Exploiting Fine-Grained Block Structures for Fast and Accurate Sparse Training.
- Studied the fine-grained dynamic sparse training algorithm with shuffled blocks to reach the accuracy with non-structured sparse training algorithm. Study the effects for accuracy and overhead of different block size setting and take a tradeoff between accuracy and performance.
- Implemented scalable and robust convolution computation CUDA kernel to accelerate training process in PyTorch. Our method achieved similar accuracy with lower time consumption after training with several models include both image classification model and language model.

C.elegans cell image processing Sep. 2020 – Mar. 2022

- Published in Quantitative Biology. CShaperApp: Segmenting and analyzing cellular morphologies of the developing Caenorhabditis elegans embryo.
- Designed and developed an automatic cell image segmentation and analysis tool for Caenorhabditis elegans cell images by using TensorFlow and PyQt5, improved the efficiency of annotating cells' MRI images for biologists. Implemented Delaunay triangulation to find the boundary of each cell and their nucleus, which brings more analyzable data.

Skills

- Programming Languages: Python, CUDA, C++, JAVA, SQL
- Platform: MacOS, Linux, Windows
- Tools: PyTorch, CUDA, Triton, TVM, Numpy, Pandas, cuBLAS, CUTLASS, PyQt

Publications

[IPDPS'25] Yihua Wei, **Lihan Hu**, Peng Jiang. *Matcha: A Language and Compiler for Subgraph Matching.*
[IPDPS'25] **Lihan Hu**, Peng Jiang. *Improving Accuracy and Efficiency of Graph Embedding Training with Fine-Grained Parameter Management.*
[IPDPS'24] **Lihan Hu**, Jing Li, and Peng Jiang. *cuKE: An Efficient Code Generator for Score Function Computation in Knowledge Graph Embedding.*
[Quantitative Biology] Jianfeng Cao, **Lihan Hu**, Guoye Guan, Zelin Li, Zhongying Zhao, Chao Tang, Hong Yan. *CShaperApp: Segmenting and analyzing cellular morphologies of the developing Caenorhabditis elegans embryo.*
[NeurIPS'22] Peng Jiang, **Lihan Hu**, Shihui Song. *Exposing and Exploiting Fine-Grained Block Structures for Fast and Accurate Sparse Training.*
[KES'20] **Lihan Hu**, Lixin Han, Zhenyuan Xu, Tianming Jiang, Huijun Qi. *A disk failure prediction method based on LSTM network due to its individual specificity.*