

# Jinsol Park

412-551-8580 | [jinsolp@cs.cmu.edu](mailto:jinsolp@cs.cmu.edu) | [LinkedIn](#)

## EDUCATION

### Carnegie Mellon University (CMU)

*Master of Science in Computer Science*

Relevant Courses (Ongoing): Distributed Systems, Parallel Computer Architecture and Programming

Pittsburgh, PA

*Dec 2024*

GPA: 4.0 / 4.0

### Seoul National University (SNU)

*Bachelor of Science in Computer Science and Engineering, Graduated Cum Laude*

Relevant Courses: Operating Systems, Internet Security, Principles of Software Development

Seoul, Korea

*Mar 2018 – Feb 2023*

GPA: 3.66 / 4.0

## SKILLS

**Languages:** Python, C/C++, Java, Go

**Frameworks / Libraries:** PyTorch, TensorFlow, Transformers, Onnx, DVC, Hydra, NumPy, Pandas, Matplotlib, Weights & Biases, Flink, Nsight Compute

**Developer Tools:** Git, Docker, Google Cloud Platform (GCP), Amazon Web Services (AWS), LaTeX

**Other Skillsets:** SIMD / ISPC parallelization, multi-thread optimization

## EXPERIENCE

### System Architecture Research Intern

*Architecture and Code Optimization Lab, SNU*

Jun 2022 – Dec 2022

*Seoul, Korea*

- Established distributed Linux environment using Docker on GCP, enabling easy experiment and deployment
- Optimized All-to-All latency for parallel MoE training with PyTorch on multiple GCP nodes by exploiting knowledge of network topology and computer architecture

### Software Systems Research Intern

*Software Platform Lab, SNU*

Jan 2021 – Jun 2022

*Seoul, Korea*

- Built specialized store backend for streaming workloads, improving throughput by 4.12x
- Devised a special prompt tuning method using PyTorch, increasing performance by 11.9%
- Developed custom TensorFlow Ops in C++ to import Triton kernels in Python environment
- Implemented disk compaction in C for system backend for stream data, reducing memory overhead by 12.4%
- Managed memory leaks within Java Native Interface by examining low-level data structure usages, enabling efficient memory consumption for scalability

### ML Engineer Intern

*Natural Language Understanding Team, NLP Lab, NCSoft*

Jul 2021 – Sep 2021

*Pangyo, Korea*

- Conducted error analysis on given dataset output for a safe software application
- Implemented model with modular code, exploiting object-oriented programming knowledge
- Directed a team of 4 to establish DL training scheme for zero anaphora resolution for company's chatbot

### Software Systems Research Intern

*Thunder Research Group, SNU*

Aug 2020 – Dec 2020

*Seoul, Korea*

- Analyzed runtime of different DL frameworks such as PyTorch and TensorFlow for high performance training

## PROJECTS

### FCR Detector | Python, PyTorch, Docker, Linux, Git

Sep 2022 – Dec 2022

- Led team of 4 to collaborate with Sherpa Space to develop a model predicting FCR disease using data mining
- Performed data augmentation on time series data in Python for training a DL model
- Exploited ARIMA algorithm and VAR for multivariate time series prediction

### Deepest Model Deployment | Weights & Biases, Hydra, DVC, Onnx, Docker, AWS

Mar 2022 – Jul 2022

- Developed a model predicting daily outlooks based on news articles on AWS

### Dataset Corruption Detection | Python, PyTorch, CrypTen, Linux

Mar 2022 – Jul 2022

- Applied Zero Knowledge Proof to detect dataset corruption in multi-party DL without harming privacy
- Experimented impact of corrupted dataset in multi-party DL, emphasizing need for detection methods

### LLVM Compiler Optimization | C++, LLVM Compiler, Git, Docker, Linux

Mar 2021 – Jul 2021

- Formulated optimization passes for customized LLVM compilers in C++, reducing 40% of cost on average
- Exploited knowledge of C++ data structures and low-level computation costs to implement optimizations
- Leveraged Github Actions and CMake to perform unit test for continuous integration
- Visualized cost reduction using matplotlib in Python