

# Siqi Guo

✉: siqigu@andrew.cmu.edu

🌐: linkedin.com/in/siqigu047

☎: (+1) 412-612-0495

## EDUCATION

### Carnegie Mellon University

Master of Science in Electrical and Computer Engineering; GPA: 3.89/4.0

Pittsburgh, PA

Expected May 2025

Relevant Courses: Foundations of Computer Systems, Embedded System Software Engineering, Machine Learning  
Signal Processing, Distributed Systems, Autonomous Control System

### Tianjin University

Bachelor of Science in Computer Science and Technology; GPA: 3.84/4.0

Tianjin, China

June 2023

Relevant Courses: Data Structure, Computer Networks, Principles of Database, Parallel Computing, Natural Language Processing, Theory and method of Big Data Analysis, Pattern Recognition and Deep Learning

## PUBLICATIONS

Ding, Y., & **Guo, S.** “Conditional generative adversarial networks: Introduction and application” (November '22). In the 2nd *International Conference on Artificial Intelligence, Automation, and High-Performance Computing (AIAHPC 2022)* (Vol. 12348, pp. 258-266). SPIE.

Jin, D., Feng, B., **Guo, S.**, et al. “Local-Global Defense Against Unsupervised Adversarial Attacks on Graphs” (June '23). In the *Thirty-Seventh AAAI Conference on Artificial Intelligence* (Vol. 37, No. 7).

## RESEARCH EXPERIENCE

### Eigenbot - Bio-inspired Distributed Control for Modular Robot

Biorobotics Lab Research Assistant (Part-time)

Pittsburgh, PA

Dec 2023 - Current

Advisor: Professor Howie Choset, Research Scientist Lu Li

Carnegie Mellon University

- Modular Robot Bio-inspired Curve Walking Implementation
  - \* Implemented the curve walking algorithm on the Eigenbot (a hexapod robot) in CoppeliaSim, inspired by six-leg insects' behavior.
  - \* Fine-tuned the weights of my algorithm to achieve a smooth curve walking gait, and integrated curve walking / steering into Eigenbot's behavior tree.
- Eigenbot Firmware Development and Testing
  - \* Developed Eigenbot EEPROM parameters auto-setting tools, which accelerated Hexapod gaits tuning process.
  - \* Tested the firmware and debugged the module communication protocol to ensure the robot's stability and reliability. Conducted the experiments with Logan, analyzed the gaits and fine-tuned firmware parameters.
  - \* Integrated Foot Sensor Feedback into the robot's message protocol.

### MMPUG - Multi-Model Perception Uber Good

Biorobotics Lab Research Assistant (Part-time)

Pittsburgh, PA

May 2024 - Current

Advisor: Professor Matthew J. Travers

Carnegie Mellon University

- FAR-Planner Implementation in a Heterogeneous Agent System with RC cars and legged robots
  - \* Implemented A\* & Theta\* global planner with 2.5D orientation optimization for MMPUG architecture.
  - \* Integrated features to make the planner adaptive to dynamic obstacles and ensure a safety distance for the robots.
  - \* Deployed this Global Planner on the RC cars, and tested the planner both in simulation and in the real environment.

### Theory and Applications of Graph Convolutional Network

Research Assistant (Part-time)

Tianjin, China

Nov 2021 - May 2023

Advisor: Associate Professor Jin, Di

Tianjin University

- Graph Neural Network in Strategic Deployment
  - \* Uncovered the relationship between Graph Attention Transformer (GAT) and Graph Convolutional Network (GCN) applied in Deep Graph Infomax (DGI).
  - \* Learned about the theories proposed in Unsupervised Adversarially Robust Representation Learning on Graphs, combined DGI with GAT and introduced mutual information and unsupervised learning to DGI.
- Robustness Analysis of Graph Neural Networks
  - \* Used Metattack and Netattack to manipulate data to obtain multiple datasets of malicious attack matrices.
  - \* Reproduced and compared RoSA, Pro-GNN, PA-GNN, and other algorithms, and practically applied GraphCL and GraphSS to do defense analysis.
- Backdoor Attacks in Unsupervised Learning Graph Neural Networks
  - \* Designed a novel unsupervised backdoor attack on Graph Neural Networks based on GIN and Momentum Contrast. Built an unsupervised backdoor threat model, defined backdoor-oriented attack targets, and derived the loss function for our attack. Verified our specific backdoor triggers and implemented the trigger injection.

- \* Experimented our graph-level attack on different agency models such as GIN and GAT. Gained higher Attack Success Rate and lower Clean Accuracy Drop than previous unsupervised graph attacks.

### Zero-shot Object Detection (Hitachi)

Research Assistant (Part-time)

Advisor: Associate Professor Tang, Lingjia

Remote

May 2022 - Sep 2022

University of Michigan

- Built a developed zero-shot object detection UI application from scratch using PyQt.
- Managed various tasks prior to training the model, such as embedding the results of labels, modifying matrix dimensions using the CLIP model, and writing Shell scripts to automate processes.
- Employed PDB to debug and reduced overheads in debugging time.
- Analyzed the COCO dataset and KITTI dataset proficiently using pycoco for statistical analysis.
- Utilized Netron and Tensorboard to visualize graphs and models in order to benchmark object detection models, such as YOLO and FasterRCNN.
- Implemented Tensorflow's Resnet50 object detection model as the baseline, and performed transfer learning on AlexNet, YOLOv5, and FasterRCNN.

### Research on Campus Uncivilized Behavior Recognition Based on Deep Learning

Research Assistant (Part-time)

Advisor: Associate Professor Yang, Liu

Tianjin, China

June 2020 - June 2021

Tianjin University

- Explored the mechanics of human skeletal information, and explored human behavior recognition algorithms.
- Attempted to improve the architecture of human skeleton recognition with federated learning.
- Recorded and generated datasets from various locations on campus, performed supervised classification learning on a series of images and videos.

## PROJECTS

### BusTub, a relational DBMS | Personal Project based on CMU 15-645

May 2024 - July 2024

- Implemented LRU-K policy, a disk scheduler and a buffer pool in the storage manager.
- Implemented disk-backed hash index in the DB system, using a variant of extendible hashing as the hashing scheme.
- Created the operator executors that execute SQL queries and implemented optimizer rules to transform query plans.
- Added transaction support by implementing optimistic multi-version concurrency control.

### Distributed File System Design | CMU 15-640 Course Project

Feb 2024 - Apr 2024

- Implemented the serialization protocol for the low-level RPC stub, supporting several RPCs in project's NFS.
- Designed a File-Caching Proxy based on RMI. Implement open-close session semantics for proxy operations, and the LRU replacement policy for cache management. Used version control, check-on-use and last-close-win to make sure the proxy's cache and server freshness.
- Implemented and Tuned a 3-Tier Architecture Scalable Web Service to satisfy dynamic or unexpected workloads by auto-scaling. Then, evaluated the performance of the system by benchmarking and figuring out the bottleneck.
- Designed a Two-phase Commit Protocol to ensure the consistency of the distributed file system. Customized a mechanism to recovering from nodes failures and handling lost messages.

### Blind Source Signal Decomposition & Analysis | CMU 18797 Research Project

Oct 2023 - Dec 2023

- Filtered the mixed HDEMG signals and detected peaks to slice the peak-wave windows. Used the PCA algorithm to reduce the dimensions of the data and K-Means to cluster the potentially same Motor Units' signals.
- Based on the clustered results, applied DTW, Covariance, Cosine in a multi-threshold way to compare the wave similarity. Verified the firing instances for each Motor Unit based on the comparison results.

### Cache Simulator with L1 and L2 level | Tianjin University Course Project

Oct 2022 - Dec 2022

- Implemented a simple Cache Simulator with L1 level Cache, applied various cache eviction strategies such as LRU, LFU, WBWA, and WTNA, and visualized the outcomes.
- Developed an advanced L2 Cache with an algorithm to effectively choose a victim cache entry.

## SKILLS

**Programming Languages:** C/C++, Python, C#, JavaScript, LaTeX, SQL, Bash

**Frameworks:** TensorFlow, PyTorch, ROS, NodeJS, Flask, .NET, React, Flutter

**Software and Tools:** GIT, Docker, MySQL, SQLServer, Qt Creator, Visual Studio Code, Xcode, Blender(Beginner)

**Platforms:** MacOS, Linux, Arduino, GCP, Kaggle