Hashem Elezabi

☎ (240) 708-3081 | ✉ hashem@stanford.edu | ⌂ hashemelezabi.github.io | hashemelezabi | in hashemelezabi

Education

Stanford University *Stanford, CA*

M.S. in Computer Science, AI Track | GPA: 3.99 *Jun 2024*

B.S. in Electrical Engineering, Minor in Mathematics | GPA: 3.75 *Jun 2022*

Coursework: [AI] *Machine Learning, Deep Learning for Computer Vision, NLP with Deep Learning, ML with Graphs, Deep Generative Models, Deep Reinforcement Learning.* [Systems] *Parallel Computing, Operating Systems, Computer Architecture, Compilers, Data-Intensive Systems, Digital System Design, Database Systems.* [Math] *Linear Algebra, Graph Theory, Applied Matrix Theory, Abstract Algebra, Formal Logic, Real Analysis.*

Honors & Awards

- **CS224N 2024 Outstanding Project Report** | Led project on improving LLM reasoning with a neurosymbolic approach, won among 509 students.
- **2022-23 Apple-Stanford Masters Scholarship** | 1 of 3 Stanford M.S. students in EE/CS chosen for this highly selective 1-year scholarship.
- **2022-23, 2021-22 Stanford School of Engineering Dean's Coterminal Fellowship** | This selective award covers a year of M.S. degree tuition.

Experience

Stanford Artificial Intelligence Lab *(ai.stanford.edu)*

Stanford, CA

Research Assistant, Stanford Vision and Learning Lab *Jan 2024 - Present*

- **Built a scalable data processing and rendering pipeline to prepare a training dataset for fine-tuning LLaVA**, a vision-language model.
- Researched 3D scene generation with LLMs. Built evaluator of 3D scenes using GPT-4V, validated with synthetic data from a text-to-image model.
- Curated data for synthesizing a question-answering benchmark for video understanding, built with prompt engineering and human validation.

Stanford Pervasive Parallelism Lab *(ppl.stanford.edu)*

Stanford, CA

Research Engineer *Jan 2023 - Dec 2023*

- Trained graph neural networks (GNNs) to predict the TPU runtime of AI models as part of a NeurIPS 2023 competition for improving ML compilers.
- **Fine-tuned a DistilGPT2 language model to generate more positive movie reviews** using RL with a reward based on BERT sentiment score.

Apple Inc.

Cupertino, CA

Software Engineering Intern, SoC Performance *Jun 2022 - Dec 2022*

- Developed new features in C++ performance models and ran simulations for improving the efficiency of Apple's iPhone and Mac chips.
- **Led new, cross-team effort developing algorithms for efficiently analyzing SoC memory bandwidth patterns** to improve performance.

NVIDIA Corporation

Santa Clara, CA

Software Engineering Intern, Deep Learning Library Performance *Sep 2021 - Dec 2021*

- Contributed to internal APIs for new architectural features used for delivering efficient deep learning primitives as part of the Fast Kernels team.
- **Integrated ~1000 new automated tests for NVIDIA's Hopper GPU architecture into Jenkins pipelines**, and caught several software bugs.

Gridspace *(gridspace.com)*

Los Angeles, CA

Machine Learning Engineering Intern *Jun 2020 - Sep 2020*

- **Implemented and trained generative speech AI models** in TensorFlow based on cutting-edge research for audio speech enhancement.
- Built a full AI pipeline, including complex data processing stages, and used it to enhance some of Gridspace's audio recordings.

Stanford Future Data Systems Lab

Stanford, CA

Undergraduate Researcher *Jun 2017 - May 2018*

- Wrote optimized parallel code in Python and C++ for efficiently processing large (>1TB) seismic time series data for earthquake detection.
- **Contributed to >100x speedup of algorithm**, enabling discovery of >6K new earthquakes. Results published at VLDB, top database conference.

Selected Projects

Language modeling from scratch [code] (ongoing)

- (PyTorch) **Trained my own byte-pair encoding (BPE) tokenizer**, wrote efficient priority-queue-based algorithm for quick BPE merges during training, and built memory-efficient tokenizer `encode` and `decode` functions. Now implementing the Transformer model training and inference.

Combining LLMs with a Z3 symbolic solver to improve their reasoning ability on AR-LSAT [paper, poster]

- **Proposed and implemented a new agentic LLM framework**, *Prototype-then-Refine*, that improves the ability of LLMs to generate correct logic programs using LLM-based *prototypers* and *refiners*. GPT-3.5 with our framework almost matches executable rate of GPT-4 (32.47% vs. 32.61%).

Transformer-based model for converting diagrams to source code [paper, poster]

- (PyTorch) Created a dataset of images of synthetic slides with diagrams and used it to **fine-tune a DEtection TRansformer (DETR) object detection model for common diagram shapes**. Achieved average precision of 89% on test data, significantly outperforming a baseline DETR model.

Neural networks and language models for machine translation and birthplace prediction (CS 224N)

- (PyTorch) (1) **Implemented and trained a Seq2Seq model (encoder-decoder RNN with attention) to translate Chinese to English.** Analyzed and discussed translation failures. (2) Pretrained char-level Transformer on Wikipedia data and fine-tuned it on a birthplace prediction task.

Open-domain question-answering system with retrieval-augmented generation (RAG) and the OpenAI API

- Built program with Stanford's DSP (DSPy V1) library that retrieves relevant context passages from a ColBERTv2 Wikipedia index and prompts GPT-3.5 to answer questions given the context passages. With prompt engineering, **improved F1 score from 0.34 to 0.51 on a challenging dataset.**

Parallel renderer in CUDA (CS 149)

- Wrote parallel renderer in C and CUDA that draws overlapping colored circles efficiently. **Wrote CUDA kernels that perform local computations in GPU shared memory**, avoiding costly data transfer and dramatically improving performance. Solution beat reference time by up to >100x.

Skills

Languages	Python, C/C++, CUDA, JavaScript, Java, SQL, Verilog, HTML, CSS, Matlab, LaTeX
Technologies	PyTorch, NumPy, Git, Unix/Linux, TensorFlow, Apache Spark, HuggingFace, Pandas, Docker, MapReduce, ReactJS, Jira, Tableau