Minh Dao

656-214-9550 | tuanminhdao@usf.edu | Tampa, FL | https://github.com/Daominh09 | linkedin.com/in/minhdao09

Education

University of South Florida — Expected Graduation December 2026

Bachelor of Science in Computer and Information Science — *Cumulative GPA: 3.91/4.00*

- **Awards:** USF Green & Gold Presidential Scholarship (2023), Gold Prize in International Creative Papers Conference & Olympic (2022)
- **Coursework:** Machine Learning Specialization, Large Language Model Operations (LLMOps), Data Structures & Algorithms, Operating Systems, Parallel Computing, Database Design, Probability & Statistics, Linear Algebra

Technical Skills

Languages: Python, SQL, C/C++, JavaScript, TypeScript, HTML/CSS
Frameworks: PyTorch, TensorFlow, Langchain, FastAPI, React, Node.js, Express.js, Next.js, Vue.js, Flask
Libraries: Scikit-learn, OpenAI, OpenCV, SciPy, Pandas, Numpy, Matplotlib, Huggingface, Shiny, SQLAlchemy
Tools: Git, AWS, Azure, Apache Airflow, MLflow, Databricks, DVC, Docker, ONNX, WandB, Postman, Netlify, Vercel

Experience

Moffitt Cancer Center — Feb. 2025 – Present

ML Research Intern — *Tampa, Florida*

- Designed a pipeline using **Python**, **Shiny**, and **Scikit-learn** to identify immune and pharmacokinetic predictors of post-transplant complications in **93 thalassemia patients**
- Boosted model accuracy by **30%** through engineering **124** clinical features via univariate analysis with **SciPy**, validated separability with **PCA/t-SNE**, and identified **10** key predictors using permutation importance
- Implemented **9** Scikit-learn models to predict post-transplant immune rejection risk, achieving **0.89 F1** and 0.97 AUC with **AdaBoost**, and **0.80 F1** with **Logistic Regression**; tracked experiments and performance using **MLflow**
- Utilized **Pandas** and **Kaplan-Meier curves** to analyze post-transplant survival, revealing a statistically significant difference of **0.06 p-value** between patients with and without immune rejection

USF Computational Biophysics Lab — Oct. 2024 – Present

Research Assistant — *Tampa, Florida*

- Developed an **AI model** for ion binding site prediction using **PyTorch**, **Biopython**, and **ESM2** Protein Language Model with few-shot learning; fine-tuned final **transformer** layer with single-layer **MLP**, achieving **0.8 F1** score
- Improved **15%** data quality by recovering missing residues using internal libraries, removing **90%-similar** sequences via CD-HIT, and addressing class imbalance with **Focal Loss** and **ReduceLROnPlateau** scheduler to prevent overfitting
- Accelerated preprocessing by **15%** through parallelizing labeling and cleaning of **4,000** PDB files to **JSON** using multiprocessing; saved **10% GPU memory** by enhancing data loader with dynamic batching based on sequence length

FPT Software AI Center — May. 2024 – Sep. 2024

AI Engineer Intern — *Hanoi, Vietnam*

- Optimized a **text-image prompt** model using **PyTorch**, **Flask**, and **Stable Diffusion**; developed and deployed a client supporting dynamic user input with custom style templates on a high-traffic site serving **50K+** users
- Enhanced **UNet** with a decoupled cross-attention layer for image features and integrated **object localization loss** to enable **multi-face ID** generation, boosting image quality, consistency, and identity preservation by up to **20%**
- Built a scalable image-text pipeline with **Apache Airflow** and **DVC** to preprocess **70K** images using BLIP-2 and Mask2Former; deployed the model as a low-latency microservice via **TorchServe** on **AWS ECS** with **API Gateway**

Projects

STEM Bot | *Python, NextJS, FastAPI, Langchain, SQLAlchemy, PostgreSQL, Microsoft Azure, OpenAI* — May. 2025

- Engineered a **multi-modal** STEM chatbot with real-time chat capabilities using **NextJS** and **FastAPI**
- Increased response relevance by **25%** by deploying a **RAG** system with **Langchain** on Microsoft Azure, orchestrating **OpenAI** and **Hugging Face** models for text-image routing in **Dockerized** environments
- Reduced **40%** message latency by parallelizing data processing across asynchronous pipelines with **Redis Streams**

GenFlow | *Python, PyTorch, Numpy, Hugging Face, NVIDIA NeMo, ONNX, WandB* — Oct. 2024

- Assisted Prof. Tassef Rahman in research on **generative flow models** for protein conformation design
- Improved **12%** structure quality by fine-tuning **Transformer blocks** as a flow model using pre-trained **ESM** and a custom trunk decoder, integrating embedded structural features to guide the generation process.
- Enhanced model inference speed by **10%** by integrating a **D-Flow** module that **optimizes noise distribution**