Michael Nath

(646) 961-9184 | michael-nath.github.io | github.com/Michael-Nath | mnath@stanford.edu

Education

Stanford University, Stanford, CA

B.S. Mathematics and Computer Science, GPA: 4.0 09/2021 – 06/2025

[*] *- graduate level courses*

Highlighted Coursework: Matrix Theory, Operating Systems, Parallel Computing, Machine Learning Systems, Computer Networking, Bayesian Reinforcement Learning (RL)[*], Deep RL[*], Deep Computer Vision[*], Deep Natural Language Processing[*]

Stuyvesant High School, New York, NY 09/2017 – 06/2021

Professional Experience

Stanford Vision and Learning Lab (SVL), Stanford, CA 03/2023 – Present

Embodied AI Researcher under Dr. Jiajun Wu

- First-authoring work in intuitive physics and curiosity-driven planning for rapid adaptation in robotic fluid manipulation
- Engineered a processor fetching terabytes of robot data under a second, through memory-mapping and thread pools
- Dissected an optimized, open-source fluid simulator coded in C++ to re-design 3 tasks and invent 2 novel tasks from scratch
- Built many deep RL, vision, and graph models in PyTorch, involving VAEs, imitation learning, and particle networks
- Orchestrating lab Linux compute infrastructure to prepare 20+ experiments daily to be run concurrently on distributed GPUs
- Extending open-sourced graph codebases written by scientists at SVL, liaising with the authors daily to accelerate research
- Presented intermediary work in latent spaces and goal-conditioned RL as final projects in Stanford's vision and RL courses
- Employing: FluidLab, PyTorch3D, H5Py, VisPy, Open3D, OpenGL, WandB, PyG, PyTorch Lightning

Amazon Web Services (AWS AI), Seattle, WA 06/2022 – 09/2022

Software Development Engineer Intern

- Automated patching of outdated cloud resources in Anomaly Detection team, saving on-call engineers 36+ hours biweekly
- Helped DevOps engineers achieve continuous development by adding scheduled, automatic triggers to team pipelines
- Invented an on-demand clean-up algorithm enabling dynamic compute allocation, redesigning 2 cloud workflows used daily
- Hacked Python to digest metrics of pre- and post-cleanup performance, which proved new workflow benefits to team
- Employed: AWS Lambda, EC2, S3, CloudFormation, DynamoDB, Step Functions, EventBridge, Kotlin, Mockito

Brookhaven National Laboratory (BNL), Upton, NY 06/2021 08/2021

Computational Scientist Intern

- Tackled the optimal qubit control task by generating optimal pulse coefficients through deep reinforcement learning
- Co-authored robust qubit gym environment leveraging epsilon-greedy policy and temporal difference learning to train agent
- Improved quantum control fidelity by 10% by trying deep q network weights as pulse coefficients as moonshot idea
- Elucidated training results to department heads through interactive plots and demos, comparing tabular and deep performance
- Employed: PyTorch, OpenAI Gym, NumPy, Q Learning, Python

Projects

SarsaMCTS – *Generalized AI Player for Turn-Based Games* March 2023

- Invented an advanced AI system to compete in arbitrary turn-based games, fueled by Monte Carlo Tree Search
- Augmented core search by coupling a novel temporal difference learning algorithm with a reinforcement learning paradigm
- Heavily modularized agent design to compete under varying search heuristics, learning parameters, and playout policies
- Developed auxiliary framework for tree visualization and debugging, aiding in performance analysis & system improvement
- Employed: Eligibility Traces, UCB-1 Search Heuristic, NumPy, Python

CodeSage – *A Generative Approach to Improving Code Quality*

- Pioneered a code-to-code system with 2 Stanford engineers generating high-quality code from undesirable input snippets
- Leveraged Davinci GPT-3 to document raw code snippets, from which intent clusters are derived with DBSCAN clustering
- Finetuned Salesforce CodeT5 transformer to translate suboptimal code into exemplary from the same intent category
- Optimized system end-to-end in grid search permuting clustering algorithm, model sizes, and intent granularity
- Employed: HuggingFace, PyTorch, DBSCAN/K-Means Clustering, CodeT5/Davinci GPT-3