

Jonathan Stokes

Oakland, CA | 215-207-1143 | stokesj84@gmail.com | <https://sophist0.github.io/>
<https://github.com/sophist0> | <https://www.linkedin.com/in/jonathan-stokes-phd/>

Software Engineer

Experienced AI Software Development Engineer and Senior Software Engineer with a strong background in machine learning, contributing to innovative projects at Intel, General Dynamics, and Centry. Demonstrated expertise in developing and optimizing ML models, leveraging skills in deep learning, reinforcement learning, Gaussian processes, and various algorithms. Proven track record in project leadership, including automatic Homography estimation, generative AI Jupyter notebooks, and object tracking, complemented by a solid academic foundation in graph theory, applied probability, and stochastic processes. Published researcher with key papers in reputable journals and workshops.

Areas of Expertise

Applied Probability | CNNs | Combinatorics | Computer Vision | C++ | Deep Reinforcement Learning | Docker | Gaussian Processes | Graph Theory | Java | K-Nearest Neighbor | Linear Algebra | Linear Regression | Linux | Markov Chains Models | Naive Bayes | Optimization | PCA | Python | PyTorch | Random Forest | LSTM Models | Statistics | SQL | SVMs

Professional Experience

Intel: AI Software Development Engineer, Phoenix, AZ (Remote)

January 2022 - March 2024

Spearheaded innovative projects in automatic homography estimation, cloud-based generative AI Jupyter notebooks, object location tracking, and ML model benchmarking, showcasing expertise in software development, machine learning.

Project: Automatic Homography estimation

- Innovated the use of object detection algorithms for selecting calibration points in Homography estimation.
- Demonstrated the efficacy of the developed technique in overcoming large angular shifts, addressing a limitation of existing methods for Homography estimation.
- Filed an Innovation Disclosure Forum with Intel as a means of documenting the innovation and as an initial step in potentially patenting the technique, however after several rounds of reviews Intel declined to pursue a patent.

Project: Host generative AI Jupyter notebooks in the cloud

- Determined the system requirements to run several stable diffusion models and LLM's on a cloud server.
- Created a custom Jupyter notebook kernel for several LLM's including LLama2.
- Helped test and launch the cloud service, enabling Intel to show customers its hardware running gen. AI models.

Project: Track object location given a moving RGB camera

- Realized that the Homography transform is used in self-driving car applications to estimate object location without recomputing the transform for each camera pose.
- Used a synthetic dataset created with Unreal to validate an object location algorithm based on this realization.
- Developed a new online algorithm for stakeholders to generate an occupancy grid from a moving video stream.

Project: Benchmark ML models on baremetal and in a virtual machine

- Wrote scripts to automate OpenVINO model benchmarking using psutil and the OpenVINO benchmark app.
- Ran each ML model 10 times on baremetal and in the virtual machine to generate the benchmark results.
- Parsed and plotted the results, giving stakeholders the data required to understand the performance cost of running a model in a virtual machine vs on baremetal.

Project: Optimized a Kalman filter object tracker

- Wrote a framework to tune a Kalman filter based object tracker using the Bayesian Expected Improvement algorithm, allowing stakeholders to automatically tune tracker performance for each context.
- Modified the Expected Improvement algorithm to avoid resampling points in the parameter space.
- Demonstrated to stakeholders that the resulting optimization algorithm could outperform random sampling.

Project: Testing video object tracking

- Created an object tracking ground truth dataset using Unity to generate video and object location information.
- Wrote a framework in Python to calculate object tracking metrics relative to the Unity ground truth dataset.
- Made recommendations to stakeholders regarding improving object detection and tracking algorithms using the results from the testing framework.

General Dynamics: Senior Software Engineer, Groton, CT**January 2021 - December 2021**

Led projects focused on estimating construction times using Gaussian process regression models and applying reinforcement learning for construction process selection, while also excelling in document classification using gradient boosting and LSTMs.

Project: Estimating construction times and selecting construction methods

- Negotiated access, collected and cleaned construction data stored on relational databases using SQL and Java.
- Fit Gaussian process regression models to construction data using Pytorch and using Bayesian optimization.
- Experimented with applying Reinforcement Learning to select a construction method using Stable-Baselines implementations of the algorithms, finding the data lacked the counterfactuals required to train RL algorithms.

Project: Classifying documents

- Negotiated access, collected and cleaned construction data stored on relational databases using SQL and Java.
- Created an algorithm using a bag of words model combined with a gradient boosting model to solve a binary classification problem with 95% accuracy.
- Used a Pytorch implementation of an LSTM module to solve the same binary document classification problem with over 95% accuracy.

Centrly: Software Engineer (Contract), San Diego, CA (Remote)**January 2020 - April 2020**

Designed and tested web scrapers using Python, Selenium, Azure, and AWS to collect business intelligence data, demonstrating proficiency in web scraping and data acquisition.

Project: Designing and testing web scrapers

- Created several web scrapers using Python, Selenium, Azure, and AWS to collect business intelligence data, allowing the stakeholders to automate some of their data collection.
- Scoped web scraper design in regard to initial objectives to minimize the time and cost of implementing and testing the web scrapers.

Drexel University: PhD Student and Research Assistant, Philadelphia, PA**September 2013 - September 2018**

Conducted groundbreaking research in graph theory, applied probability, and stochastic processes, contributing to key papers and projects that estimate expected costs of using graph sampling and biased random walks to find high degree nodes in large graphs.

Project: Estimate the expected cost of star sampling to find a target node in a large graph

- Estimated the expected unit and linear cost to find a target node in Erdős Rényi (ER) graphs under three types of star sampling: Star sampling with replacement (SS-R), Star sampling with center removal (SS-C), Star sampling with star removal (SS-S).
- Proved asymptotically in the graph's size, the probability of finding a target node on a given sample under the three variants of star sampling is nearly identical.
- Coded simulations showing the estimates of the expected unit and linear cost of using star sampling to find a target node are accurate on ER graphs and can be accurate on real-world graphs.

Project: Estimate the expected steps for a biased random walk in a large graph to find a maximum degree node

- Developed a Self-Avoiding Walk Jump (SAWJ) algorithm to search large graphs for maximum degree nodes.
- Modeled a rough upper bound on the expected steps required by SAWJ to find a max degree node using a discrete time Markov chain model which is shown to be accurate if the joint degree dist. of the graph is known.
- Coded the simulations showing that SAWJ outperformed competing algorithms in the literature on degree assortative ER graphs and some degree assortative real-world graphs.

Education**Doctor of Philosophy (PhD), Electrical and Electronics Engineering - 2018****Masters of Science (MS), Electrical and Electronics Engineering - 2015**

Drexel University, Philadelphia, PA

Key Academic Papers

Graph search via sampling with and without replacement: Journal of Internet Mathematics 2020

Common greedy wiring and rewiring heuristics do not guarantee maximum assortative of given degree: IPL 2018

Star Sampling with and without Replacement: KDD Workshop MLG 2017

The Self-Avoiding Walk-Jump (SAWJ) Algorithm for Finding Maximum Degree Nodes in Large Graphs: IEEE BigData 2016