# Yash Patel

http://yashpatel5400.github.com/

| CONTACT INFORMATION | 816 E. Dana St.<br>Mountain View, CA 94041 | ypatel5400@gmail.com<br>(908) 670-8612 |
|---|---|---|

**EDUCATION**

**Princeton University**, Princeton, NJ

*Bachelor of Arts*      **September 2014 - May 2018**

▷ Major: Mathematics      Dept. GPA: 3.725/4.0, Cum. GPA: 3.642/4.0

▷ Certificates: Applications of Computing, Statistics and Machine Learning

▷ Selected Courses: Topology, Real Analysis, Complex Analysis, Theoretical ML (Graduate), Fairness in ML (Graduate), Machine Learning/Pattern Recognition (Graduate), Neural Networks: Theory & Applications, Theory of Algorithms, Analysis of Big Data, Computer Vision, Computer Graphics, Probability/Stochastic Systems

**RESEARCH EXPERIENCE**

**Princeton University**, Princeton, NJ

*Statistical Analysis Research (Senior Thesis)*      **September 2017 - May 2018**
https://yashpatel5400.github.io/files/deanonymization.pdf

Worked under the supervision of Matt Weinberg and Arvind Narayanan to investigate the prospects of partially deanonymizing Bitcoin transactions using graph clustering algorithms on a heuristics graph constructed atop the BTC transactions graph. Discovered hierarchical spectral clustering and METIS to have the best performance as measured by F-score, NMI, and purity, after which several BTC wallets were identified.

*Statistical Modelling Research (Junior Paper)*      **January 2017 - May 2017**
https://yashpatel5400.github.io/files/selfish.pdf

Worked under the supervision of Matt Weinberg to study the viability of selfish mining attacks in mining pools as an extension to "Majority is Not Enough: Bitcoin Mining is vulnerable" (Eyal) by taking price adjustments into account. Discovered selfish mining was viable for BTC/ETH through 2017.

**Columbia University (Mailman School of Public Health)**, New York, NY

*Statistical Modelling Research*      **May 2015 - September 2015**
https://github.com/yashpatel5400/SexualEqualityABM

Worked under the supervision of Professor Abdulrahmen El-Sayed to develop agent-based mathematical models for understanding the dynamics of self-efficacy for sexual minority populations from enrollment in exercise coach programs. Simulated dynamics in Python using Matplotlib, Numpy, and NetworkX.

**Princeton Plasma Physics Lab**, Princeton, NJ

*Numerical Simulations Research*      **June 2013 - January 2014**

Worked under the supervision of Ilya Dodin and Ammar Hakim to develop FTDT (RK4) numerical simulations in C++/Python to empirically study PDE governing plasma phase space evolution derived in https://arxiv.org/pdf/1006.3717.pdf (Eq. 88). Evolution behavior was verified on standard potential initializations (i.e. cos(x), $x^2$, $x^4$): https://yashpatel5400.github.io/files/cos.mp4.

**WORK EXPERIENCE**

**Facebook**, Menlo Park, CA      **July 2018 - Present**

*Undisclosed Projects (2019-)*
▷ Implemented real-time (72 FPS) HMD-rendered point cloud and TSDFs (KinectFusion) scene reconstruction in C++/OpenGL/OpenGL ES/OpenCL
▷ Implemented, optimized, and trained deep learning model for deployment on Portal platforms. Model was implemented in PyTorch and optimized via SNPE, quantization with QAT, and layer fusion to run at 30 FPS on Qualcomm SoC.
▷ Added translation support for quantized nodes in JIT-compiled PyTorch to Caffe2.

*Manifold (2018-19)*
https://research.fb.com/wp-content/uploads/2019/09/
An-Integrated-6DoF-Video-Camera-and-System-Design.pdf
▷ Added farm rendering through Docker, RabbitMQ, and Kubernetes. Improved depth estimation efficiency by 20% with "Gaussian funnel."
▷ Created test suite for open source release, extending coverage from 10% to 100% of binaries. Deployed with Travis CI.

**Amazon**, Seattle, WA                                      **June 2017 - August 2017**

Developed debugging service for Kiva Picking Optimization (Amazon team) developers. Deployed globally to all Amazon Robotics-enabled fulfilment centers. Implemented in Java with Spring MVC. Deployed on AWS (EC2, S3, SNS/SQS, DDB).

AWARDS | Siemens Westinghouse National Competition Semifinalist (2014)
1st Place & 2nd Place in Category, Delaware Valley Regional Science Fair (2013, 2014)
AIME Qualifier (2014)
2x Honorable Mention, SIAM Moody's Mega Math Challenge (2013, 2014)

NOTEABLE
PROJECTS

**FairTear (Fairness in Machine Learning: COS 597E)**
*Demo*: http://fairtear-demo.herokuapp.com/
*Report*: https://yashpatel5400.github.io/files/fairtear.pdf
Probabilistic fairness checker on hidden variables in machine learning models. Interfaces with decision trees, SVMs, and basic MLPs from scikit-learn. This work was an extension of Fairsquare (https://dl.acm.org/doi/pdf/10.1145/3133904).

**Optimal Charging Station Locations, (Optimal Learning: ORF 418)**
*Report*: https://yashpatel5400.github.io/files/tesla.pdf
Investigated policies to determine the optimal locations to place Tesla charging stations in a city setting. For this task, we developed a lookup-table model, with altered updating equations, and tested a few learning policies, in the forms of online and offline Knowledge Gradient Exploration (KG), Interval Estimation (IE), Boltzmann Exploration, and Pure Exploitation. Upon doing so, we found that the Knowledge Gradient Policy was the most effective in maximizing our total usage over our time horizon.

**NeuroPath (Great Moments in Computing: COS 583)**
*Code/Demo*: https://github.com/yashpatel5400/neuropath
Implemented two neural branch predictors (single neuron and path-incorporating network) in x86 Gem5 emulator environment. Compared with performance with standard BPs, such as Tournament and LTAGE, finding increased latency. This work was an extension of "Fast Path-Based Neural Branch Prediction" (Jimenez).

**Synalyze (Best Use of Machine Learning: HackPrinceton 2017)**
*Code*: https://github.com/yashpatel5400/synalyze
Business meeting-centric application for analyzing pain points and how to improve upon them. Produces analytics on voice audio recordings of business meetings using Watson NLP API. Implemented in Python/Ruby on server-side.

PROGRAMMING | Python, C++, C, Java, Bash, LaTeX, GLSL/OpenGL, C#