Scott Viteri

Address: 2023 Parker St, Berkeley, CA 94704
(480) 266-0242 | sviteri@stanford.edu

Profile

My goal is to reduce **existential risk** from artificial intelligence by **augmenting human intelligence**. Towards this goal I am thinking about how to train an ML model to bridge the gap between the worldviews of two different intelligences. I have been charting out the space of design choices of **ontology maps**, and I am excited about this research direction going forward. I am also interested in neural network interpretability and anything that would be useful to me in the scenario that I had read/write access to my own source code.

My background is in artificial intelligence and programming language theory. I was a computer science and electrical engineering major at MIT, where I contributed to research in AI and robotics. Afterwards I researched interactive theorem proving at CMU with Simon Dedeo, where I published on the use of abduction in mathematics to the Cognition journal. I am currently a 4th year PhD candidate at Stanford under Clark Barrett, where I have worked on various projects related to SMT solving and interactive theorem proving. I have changed my research direction to AI alignment as of this January due to my belief that it is the most morally significant project that I could be working on.

My primary character trait is curiosity, and I really love math.

Education

2014-2018	**Massachusetts Institute of Technology**	BS Computer Science and Electrical Engineering
2020-Current	**Stanford University**	Doctoral Candidate

Employment

Stanford, CS PhD Candidate in AI Safety, Stanford CA 1/2022 - Current

- Won 10K in Alignment Research Center contest for Eliciting Latent Knowledge solution proposals
- Gave a talk at the Topos Institute on computation, communication, and ontology maps
  - https://www.youtube.com/watch?v=WnvbPXykk8Y&t=1473s
- Gave a talk at the one-off Alignable Structures conference on trade-offs in ontology map design space
- Mentored for AGI safety fundamentals reading group, and TA'd for Stanford's Intro to AI Alignment course (STS 10SI)
- Currently applying to teach a graduate level course on AI alignment at Stanford in the Spring

Stanford, CS PhD Candidate in Formal Verification, Stanford CA 9/2019 – 1/2022

- Created method to flexibly shift between higher order and first order logic reasoning in verification of program correctness, with Oded Padon
  - Using verified performant key-value store Veribetrkv as a source of examples
- Created various projects while rotating between potential PhD advisors
  - Wrote algorithm to find common subroutines among combinator calculi programs for program synthesis
  - Implemented calculus for SMT solving in the Lean theorem proving language to allow external checking of CVC4 proofs
    - Published to IJCAR (https://link.springer.com/chapter/10.1007/978-3-031-10769-6_3)
- Software projects
  - Cowrote neural decoder auto-completion framework that uses left and right context

    - A SAT solver based on polynomial reduction in an XOR-AND calculus
    - Various Racket utilities for linear algebra, abstract algebra, and modular arithmetic

CMU, PL Researcher, Pittsburgh PA 10/2018 - 8/2019

Analyzed proof trees of major theorems in Coq in order to find patterns for automatic theorem proving

- Extended plugin to reify Coq AST https://github.com/scottviteri/CoqAST/blob/master/README.md
- Wrote repository for analyzing the resultant AST's for a large collection of famous mathematical proofs
    - https://github.com/scottviteri/ManipulateProofTrees
    - Compared generative tree models to simulate modularities and degree distributions of mathematical proofs
- Paper published in the Cognition journal (https://www.sciencedirect.com/science/article/pii/S0010027722001081)

New England Complex Systems Institute, Researcher, Cambridge MA 6/2018 – 10/2018

Created and experimented with a computational model of geographic wealth distributions

- This model is based on a continuous version of the Schelling Model for segregation
- This summer, I also attended the Oregon Programming Languages Summer School and the conference "From the Fundamental Lemma to Discrete Geometry, to Formal Verification"

MIT, Artificial Intelligence Researcher, Cambridge MA 1/2018 - 6/2018

Worked on a Coq framework for knowledge representation and planning

- Treated sub-goaling as automatic theorem proving in coherent logic
- Used coherent logical statements about the world as theorem-proving "hints"

MIT, Robot Locomotion Group Researcher, Cambridge, MA 1/2017

Worked with MIT's Robot Locomotion Group for a second time

- Used Vicon camera systems to analyse performance of state estimators for Atlas and Valkyrie robots

MIT, Hyperloop Team, Cambridge, MA 2/2016 – 9/2016

Competed to build the best high-speed transportation system

- Wrote brake controller and the pod's remote user interface
- Used ZCM (Zero Communication and Marshalling) networking protocol between various levels of computer systems

Amazon, Software Engineering Intern, Seattle, WA 6/2016 – 8/2016

Developed machine learning tools to predict Amazon Website user behavior

- Created prediction model software to reduce number of server requests
- Used Spark, Hadoop File Systems, and elastic map reduce techniques

Roambotics, Software Development Intern, Phoenix, AZ 6/2015 - 8/2015

Designed and implemented architecture for company's flagship product, an autonomous robot

- Created Ant build tools that allow general integration of external C and C++ libraries
- Designed a system general enough to add or subtract subsystems, states, and constants without recompiling

MIT, MASLAB Robotics Competition Entrant, Cambridge, MA 1/2015

Built from scratch a robot that autonomously navigated a field and completed tasks

- Implemented machine vision, path planning, and state estimation algorithms

MIT, DARPA Robotics Challenge (DRC) Team Member, Cambridge, MA 9/2014-6/2015

Worked with MIT's Robot Locomotion Group

- Only freshman and one of very few undergraduate members
- Created MatLab software bridge to natively visualize and utilize libraries of SDF files provided to DRC team

TECHNICAL SKILLS

Computer Languages: Racket, Lean, Python, Haskell
Other Tools: GNU/Linux (Ubuntu and Arch)
Hardware: Developed computer from transistor-level upward using the Beta architecture, implemented in software. Comfortable with laser cutters, lathes, mills, 3D printers, and other shop tools.

INTERESTS AND ACTIVITIES

Singing, guitar, brazilian jiu jitsu, zouk dancing, and math