## Personal Statement

As a new researcher, I was initially drawn to machine learning research and I was interested in investigating ways of automating simulating input modeling using neural networks. While working on this, I found myself continuously asking research questions that were not based in the theoretical aspects of my work, but rather in how we might improve the fundemental tools that we were using to build our models. I only thought about these questions more and more, and found myself compelled to learn about how we could possibly answer them. I was interested in how we might use language-level features to encode desired features into our programs, and this led me to my passion of programming languages and systems research.

While adding desired behaviors via language features can be successful, it can also be a challenging task if those language features are difficult for programmers to use. One application that suffers from this problem is serverless computing. Serverless computing is an emerging approach to cloud computing that makes server-side computing more accessible to a wider audience. While the name "serverless computing" implies an absence of servers, the reality is that programmers are just freed from the responsibility of server management. This means that web developers no longer have to be experts in server architectures, and those without access to server hardware can still host their applications. Most serverless applications are written in dynamic languages like JavaScript, which leads to slower performance and higher safety concerns. But JavaScript is widely used, so the difficultly is in how to remove the burden of using dynamic languages for serverless computing without placing that burden on developers. This is the research area I am passionate about exploring, how can we add these desired features while minimizing overhead on developers?

Equally as important as expanding the boundary of science is supporting students and younger emerging scientists. While my path to programming languages and systems research has not been the most straightforward, I have had amazing mentors and professors along the way that have supported my journey and helped me succeed. In turn, I've had the opportunity to serve as a teaching assistant to give this support back to others. Through this work, I discovered a passion for teaching and mentoring, and I want to pursue an academic career that combines research, teaching, and mentorship.

## Relevant Background

As an undergraduate, my first experience with research was through working with machine learning. I studied methods of style transfer for music using neural networks and completed an undergraduate thesis based on this work. Wanting to explore other uses of machine learning, I extended a class project to investigate mechanisms for simulating ant colony dynamics.

Excited to continue research, I began my graduate work at the University of Massachusetts Amherst the summer before my PhD studies officially began. Building on the ideas formed from my previous work studying ant colonies, I explored methods of using neural networks for simulation input modeling with Prof. Peter Haas and MS student Cen Wang. We researched this topic over the course of my first year, and our paper *NIM: Generative Neural Networks for Modeling and Generating of Simulation Input* was published at the 2019 Summer Simulation Conference, where I also gave a talk about our findings. While I enjoyed this work, the research questions I found myself asking were not congruent with the field I was exploring.

At the same time, I was enrolled in a programming languages overview course, which discussed a span of introductory topics. The material covered in class was very fun and compelling, and this piqued my interest to PL. After this course, I enrolled in a programming languages seminar that focused on working with the language Idris. I had grown profoundly aware of NumPy tensor dimension errors due to my machine learning research, and I used this seminar to remedy this. I prototyped an Idris tensor type that ensured dimension safety and compatibility of tensor operations during program compilation. Idris is able to compile to Python, so a complete system of this kind could be used to build dimension-safe machine learning models. My experience prototyping this tool demonstrated to me that the research questions I had been asking about my machine learning research were actually programming languages and systems research questions.

**Intellectual Merit.** Inspired by my background and motivated by the types of research questions I developed during my input modeling research, I enthusiastically entered into programming languages and systems research. In particular, I was interested in using programming languages and systems tools to improve programs in ways that are abstracted away from programmer themselves. This was considerably new territory for me, and in order to dive in head first, **I attended the 2019 Principles of Programming Languages (POPL) conference, funded by a travel grant from the co-located Programming Languages Mentoring Workshop (PLMW)**. Here I got a diverse introduction to programming languages topics and a survey of the current boundary of knowledge.

Starting in January of this year, I began exploring programming language and systems tools for improving serverless computing with Prof. Arjun Guha. First, we observed that Rust is amenable to serverless computing,[1] and then my research began by implementing an initial serverless function accelerator that compiles a subset of JavaScript to Rust. Such an accelerator could be used to leverage the safety features of Rust without adding additional burden to JavaScript developers. But, JavaScript language features and Rust language features have fundemental differences, and the research challenge is to create a mechanism that can compile these JavaScript features to their Rust counterparts without asking programmers to develop fundementally different code. I first approached this problem by studying the state-of-the art in compiling JavaScript.[2,3] Building off of methods presented by these systems, we implemented a mechanism that captures JavaScript program execution and assembles it into an interprocedural representation, allowing us to compile JavaScript language features to Rust language features.

We worked on this over the spring semester of this year, and at the beginning of the summer **I attended the 2019 Programming Languages Implemention Summer School (PLISS)**. This opportunity was particularly inspiring because it allowed me to connect with other early-career researchers in my field, and it was fun and encouraging to discuss our research ideas together. While there, Niko Matsakis gave a talk[4] about how the design and motivation behind the Rust compiler, and given my recent work with Rust, this was really exciting to me. Attending the PLISS summer

[1] Sol Boucher et al. "Putting the "Micro" back in microservices". In: *USENIX Annual Technical Conference (ATC)*. 2018.

[2] Andreas Gal et al. "Trace-based Just-in-time Type Specialization for Dynamic Languages". In: 2009.

[3] Michael Bebenita et al. "SPUR: A Trace-based JIT Compiler for CIL". in: 2010.

[4] Niko Matsakis. *Responsive compilers*. PLISS. 2019. URL: `https://nikomatsakis.github.io/pliss-2019/responsive-compilers.html#1`.

school solidified even further my enjoyment of programming languages and systems research and Matsakis' talk reinforced the appeal of Rust for our accelerator.

We have continued to develop the accelerator over the summer and through this semester, and we are now able to use it to compile JavaScript serverless functions to generated Rust functions. Our initial results show that the generated Rust functions result in lower latency and lower CPU utilization than their JavaScript counterparts. This work has the potential to serve as a method for improving serverless computing and demonstrates a mechanism of resolving languge feature discrepancy issues. We are actively pursuing this with hopes to publish our research in the near future.

**Broader Impacts.** Throughout this journey of exploring academia and research, I have been fortunate to have had the opportunity to teach others. As an undergraduate, I worked as a teaching assistant for several introductory computer science classes. I worked closely with the students, meeting regularly with those who were struggling and hosting open office hours around exams and project deadlines. I assisted over a span of two years, giving me the opportunity to work with a handful of the same students as they moved from course to course. One moment that stands out to me in particular is when I witnessed one of the students helping a peer with a CS concept that they themselves used to struggle with. It was incredibly rewarding to see someone that I had helped, helping others, and this experience helped introduce me to the joy of teaching. I've since continued this pursuit as a teaching assistant in graduate school.

Through working with students as a teaching assistant, I've been able to act as a mentor and resource for them to succeed. I remain in contact with several former students and have helped them navigate internship and career decisions. This school year I'm also serving as a PhD mentor to a first-year PhD student in the computer science department. The first year of a graduate program can be difficult, and I hope to provide resources that she can use to succeed and help her integrate into the graduate community. Being able to mentor others has really shown me how rewarding it can be to help others pursue their goals.

## Future Goals

My academic and teaching pursuits have given me opportunities to learn and grow, and I want to use these experiences to help contribute back to the community and to help others. Through my research, I hope to improve serverless computing such that it is better able to serve a wider audience, and I hope to advance the scientific community by investigating mechanisms of resolving language feature discrepancies without burdening programmers. I am passionate about programming languages and systems research, but I only arrived here through a process of reflection on my own research questions. I want to use this journey to help younger researchers explore their own pursuits, while simultaneously teaching them the skills that they need to succeed. My experiences have taught me that I greatly enjoy a combination of research, teaching, and mentorship, and a career in academia will enable me to pursue this. **I would like to be tenure-track professor at a university, where I can work with students and emerging scientists.** I am fortunate to have had opportunities to engage with my passions and grow personally and academically, and graduate funding support from the NSF GRFP will enable to pursue this even further.