Brendan LaPuma

978-880-0790
brendanlapuma@gmail.com
Personal Website
LinkedIn Profile

Education

- **University of Maryland - College Park** *Expected May 2025*
  B.S. Computer Science GPA: 3.86

Experience

- **Teaching Assistant** *August 2023 - Present*
  University of Maryland - College Park College Park, MD
  – Collaborate with Algorithms, Data Science, and Intro to C Programming instructors for over 1,000 students
  – Develop scalable containerized project auto-grader, simplifying project submission process
  – Validate assessments pre-release for accuracy and fairness
  – Conduct weekly office hours for personalized student assistance
- **Infrastructure Engineer Intern** *May 2024 - August 2024*
  Liberty Mutual Insurance Boston, MA
  – Orchestrated Kubernetes chaos engineering event to maximize incident response efficiency
  – Spearheaded transition away from Docker Desktop, saving $300,000 annually
  – Reinforced upkeep of Amazon EKS clusters via Rancher used by over 2,000 developers
  – Collaborated in migration of company-wide CI/CD pipeline from Bamboo to GitHub Actions
- **Software Engineer Intern** *May 2023 - August 2023*
  Liberty Mutual Insurance Portsmouth, NH
  – Engineered full-stack application to automate role management tasks in Azure Active Directory
  – Minimized required time for service account provisioning by automating request process
  – Created appealing customer-facing front-end using React with AWS Lambda back-end
  – Worked as a team in an Agile development environment employing good development practices

Projects

- **GPT Selector** *July 2024*
  2024 Liberty Mutual Hackathon Winner
  – Collaborated on LLM query analyzer that automatically selects GPT-3 or GPT-4 to answer questions
  – Estimated thousands of dollars saved per year in compute costs, mitigating carbon emissions
  – Tools & technologies used: SQLite, Flask, OpenAI API, Python, React.js
- **Deep Learning Classification of NBA 3-Point Shots** *January 2024 - May 2024*
  Semester-long team research project
  – Trained recurrent neural network based on player statistics and court positions
  – Achieved 80% classification accuracy, surpassing results of similar papers
  – Tools & technologies used: PyTorch, Pandas, Numpy, Python
- **Containerized Gradescope Auto-Grader** *August 2023*
  Parallelized auto-grader run in Docker containers
  – Developed containerized auto-grader used by multiple classes at UMD
  – Streamlined and simplified project submission experience for students
  – Tools & technologies used: Docker, Python

Achievements

- **UMD Presidential Scholarship Awardee** Merit-based scholarship
- **6 Time Dean's List Awardee** Taking 12 or more credits with GPA 3.5 or higher
- **Liberty Mutual Hackathon Winner** *August 2024*
- **Liberty Mutual Hackathon Finalist** *August 2023*

Extracurricular Engagement

- **UMD Club Swim Team** *August 2021 - Present*
  University of Maryland - College Park College Park, MD
  – Participate in 4 practices per week, 3 meets per semester
  – Qualified for 2022 Nationals in Atlanta, GA

As an undergraduate I had the opportunity to work as the head teaching assistant for an introductory-level programming course with a professor who was teaching it for the first time. Since there was no prepared content or projects, I took an active role in designing assignments and guiding students through the material. Aware that the course's prior setup with a Linux-based submit server was not ideal for non-programmers, I developed a containerized autograder that integrates with Gradescope, allowing students to submit project files through a user-friendly browser interface and receive instant feedback. The tool significantly improved the submission process and has since become a standard part of the course. Balancing this with my own coursework was challenging, but I genuinely enjoyed the experience and the unique problems it presented. Since then, I have worked as a teaching assistant for courses in algorithms and data science, continuing to find the experience both challenging and rewarding.

During my final summer as an undergraduate, I interned at Liberty Mutual on a team of infrastructure engineers responsible for maintaining the company's enterprise Kubernetes clusters. I was able to experiment on my own development cluster the entire summer, suiting my trial-and-error approach to learning extremely well. Over the course of the internship, I proposed and implemented a development tooling change that is projected to save the company approximately $300,000 annually in licensing costs. Additionally, I won the company-wide intern hackathon with a large language model query analyzer that automatically routes user prompts to the appropriate GPT model, reducing both carbon emissions and associated costs.

Through these experiences, I discovered that I found fulfillment in supporting the tools and technologies that developers use to create solutions to complex problems. While a career in industry would allow me to support developers at a professional level, I am particularly drawn to research, where I believe I can make a more substantive impact. Research would enable me to

influence the industry on a broader scale, potentially benefiting the entire field rather than just a single company. A research environment is also more conducive to my preferred process of learning through experimentation, without many of the restrictions that exist in a corporate setting. While pursuing a PhD, I am particularly interested in investigating how to make cloud computing resources and other distributed systems more cost-effective and better for the environment. Through my background in container orchestration and managing distributed compute resources, I have seen firsthand both the financial and environmental costs associated with its use, especially on a global enterprise level. With the wide popularity of machine learning and large language models containing billions of parameters, having efficient and reliable computation is incredibly important. As such, I find that my background and interests most closely align with the software systems research area.

While working as a teaching assistant, I discovered that the professor I work for likes to hold small, unofficial lectures every Friday discussing general life challenges and how to overcome them. One of his most memorable lectures was on the subject of trying, discussing general strategies that can be used to complete any task. I learned that people tend to underestimate the extremes to which they are willing to go when they truly want to get something done, and tend to fail by not taking advantage of the tools at their disposal. Essentially, believing that a goal is unachievable is typically due to a lack of specific strategies used to complete that goal, rather than the goal itself being inherently impossible. I have since become a firm believer that anybody can achieve anything that they set their minds to when sufficient effort is applied. In the face of adversity, there is always another strategy, another tool, or another 1% of effort left to give when a goal truly matters.

During one of my office hours for an intro-level programming course, a student approached me and explained that it took her a few attempts to complete a fairly difficult project. She described a series of initial mistakes she had made along with her process of fixing them, and ended up asking “What is wrong with me?”. I was initially stunned, given that she had just described the typical learning process for most college students. Rather than valuing experimentation and learning through mistakes, she saw each error as proof of her own inadequacy. During the next lab session, I challenged the students to try and invent a sorting algorithm for a list of positive integers over the next week. If any of them managed to create an ideal algorithm, meaning one that ran in linear time, I would buy bagels for the entire class. Although one student managed to comically invent one of the slowest sorting algorithms I had ever seen in an academic setting (which she lovingly dubbed “flapjack sort”), none of them managed to complete my challenge. However, making an ideal sorting algorithm was not the point of the exercise. I was actually incredibly impressed that a group of individuals who had never written a line of code a month prior were able to invent a sorting algorithm at all. I was instead trying to demonstrate that there is nothing wrong with making mistakes during experimentation, especially in the process of learning something new.

Throughout my college experience, these two lessons are the most important I have learned, and have since guided nearly every major decision I make. I firmly believe that these lessons will guide me to a satisfying and fulfilling life, and it is for these reasons I decided to pursue a PhD. It is also through these principles that I strive to create healthy and inclusive learning environments for the classes I help teach, and I hope to continue this mission both as a doctoral student at Columbia and afterwards.