My research interest lies in using **Large Language Models (LLMs) to build trustworthy, robust, and reliable code-generation systems.** Interactive AI code-generation systems based on LLMs like GitHub Copilot or GPT are already being deployed and used by developers in professional and academic settings. Despite their prolific use, current code-generating models are held back by sparse reward spaces, limited context length, and a tendency to memorize syntactic rather than semantic meaning, leading to many cutting-edge code-generation models being significantly larger than similarly performant dialogue models. My research interest is leveraging code-generation LLMs, Natural Language Processing (NLP), and Reinforcement Learning (RL) to build robust, self-correcting, and interactive code-generation systems. I believe that seeking methods to de-sparsify code-generation reward spaces will be crucial in reducing memorization, making code-generation models smaller, and boosting the potential for real-world applications of LLMs beyond the realm of interactive dialogue tools.

Research in Natural Language Processing and Code Generation. My desire to deepen my knowledge of code-generation LLMs led me to take Prof. Baishakhi Ray's PhD seminar on applying generative models to code-generation tasks, where I studied the theoretical underpinnings of code-generation capable systems.

Under the supervision of Prof. Mark Santolucito at Barnard College Columbia University, I researched how formal verification methods improve the code-generation capabilities of LLMs. In particular, I explored how structured template-based formal languages enhance the understandability and context length of LLM-generated code. This project adopted Prof. Mark Santolucito's previous work on formal verification and specification language compilation to teach LLMs to generate compilable specifications from natural language input. We demonstrated that leveraging in-context learning and prompt engineering let us achieve high few-shot compilation rates on formal language specifications, that the inclusion of an intermediary specification generation step in the natural language to programming language pipeline improves the pass@k rate of LLM-generated code of open-source models, and that LLMs could leverage the condensed nature of specifications to describe long-context arbiter problems. Through this project, I deepened my understanding of formal verification methods, programming languages, verifiability, and generative code models. I then leveraged this knowledge to increase the understandability and capabilities of LLMs in code generation tasks.

Additionally, I am proud that my experience in code-generation and fine-tuning LLMs with RL methods let me join a new research project under the supervision of Prof. Baishakhi Ray, that seeks to improve the coding capabilities of LLMs through techniques presented in her previous work, Semcoder, by using new ways to de-sparsify the reward space.

Research in Reinforcement Learning and LLMs. As a research assistant in Prof. Xiadong Wang's lab in the Electrical Engineering department at Columbia University, I developed methods using massively parallelized Reinforcement Learning algorithms and LLMs for financial time-series prediction, risk modeling, and portfolio allocation. Markets have low signal-to-noise ratios which causes high variance in RL algorithms and harms the ability of LLMs to understand market information and make predictions. To address this, I built KL divergence-aware RL agents for multi-GPU massively parallel ensemble training to lower policy variance and overcome sampling bottlenecks. I also showed that LLM finetuning could be further aligned with markets by using market movements to compute a policy loss, increasing alignment and utilizing data better. By using market movements to compute a value loss over output logits, our model was able to improve its ability to predict short-term market movements, introducing a denser reward space for LLMs beyond syntactic and semantic matching. My research was used to host two competitions at the 5th ACM International Conference on AI in Finance (ICAIF), which saw 20 teams from industry and academia compete to develop novel RL agents to train market RL-enhanced LLMs.

Aided by my experience in LLMs, I helped author an accepted research proposal for Columbia's SIRS and STAR Program on Generally Capable Financial Agents from Open-ended Play. Further, I joined an ongoing research project with the Pytorch Foundation centered on using LoRA matrices to create interchangeable expert modules within LLMs that allow for domain knowledge updating of general-purpose LLMs.

Research in ML optimization. Under the supervision of Prof. Kaoutar El Maghraoui and Dr. Hadjer Benmeziane, I took the opportunity to develop a class project into a project with IBM Research. In this research project, I am extending Dr. Hadjer Benmeziane's work on analog neural architecture search (NAS). The goal of my project is to first understand the limitations of the XGBoost surrogate, which was trained on ResNet and CIFAR-10 performance pairs when generalizing to ResNet performance on other tasks. Subsequently, I am planning on designing new search spaces which will allow for more efficient NAS. This project demonstrates my theoretical and empirical skills in implementing low-level optimizations for efficient models. Working with experts from academia and industry presents me with exciting opportunities to learn more about systems-level high-performance AI and to apply my knowledge of low-level optimizations in both the digital and analog space.

I am excited by the prospect of continuing my research career at Columbia. I am interested in **Prof. Baishakhi Ray's** recent works SemCoder and TRACE, which show that a denser reward space can make smaller models match the performance of significantly larger ones trained on traditional objectives. **Prof. Suman Jana's** research on improving the security and robustness of LLMs aligns with my prior experience in LLM training, reward space densification, and code generation. **Prof. Zhou Yu's** work on interactive systems and language model alignment matches my interests and experience. **Prof. Mark Santolucito's** work also aligns with my interests, since I believe that formal verification methods may have the potential to improve existing program analysis methods.

My experience in conducting independent research in the space of LLMs, RL, and code-gen will pair well with my strong empirical skills and objectives and will allow me to capitalize on current trends and work on shaping research throughout my PhD. Obtaining a PhD in computer science at Columbia will allow me to pursue my interest in pushing the boundaries of performance, safety, and robustness of code generation in LLMs, both during and after my program.