I'm currently finishing my undergraduate **CS degree with a minor in English**. Although this is an unusual combination, my English classes have greatly enhanced my ability to communicate difficult technical concepts in writing and in person. My career goals are (1) to further foundational knowledge in CS theory, specifically **complexity theory**, through research and collaborations and (2) to empower others through **teaching and research mentorship**. With these goals in mind, I aim to become a **CS theory professor**.

Research Interests I'm fascinated by complexity theory's investigation of inherent limitations, and I'm particularly interested in the **limits of what computational limitations we can prove**. There are several concrete barriers, such as relativization and algebrization, to proving important conjectured class separations and equalities. How to overcome these barriers is an open question. Relativization is a canonical example of limitations in proving lower bounds. We know that any complexity proof must not hold its structure relative to an oracle unless its conclusion holds relative to all oracles. For example, any proof that shows $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} \neq \mathsf{NP}$ must not relativize; otherwise, a contradiction would arise since there are oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ while $\mathsf{P}^B \neq \mathsf{NP}^B$ [1]. Further, natural proofs also seem unlikely to prove many important class separations because they imply the nonexistence of one-way functions [2]. Arithmetization overcomes both of these barriers and was successful in proving the major result $\mathsf{PSPACE} \subseteq \mathsf{IP}$. Arithmetization considers Boolean formulas as low-degree polynomials. This gives the verifier in an interactive proof increased power to catch the prover in a lie, enabling an interactive proof to verify the PSPACE-complete language TQBF [3]. Unfortunately, this promising technique relativizes with respect to algebraic oracles. $\mathsf{PSPACE}^A \subseteq \mathsf{IP}^{\tilde{A}}$ (the algebrized version of $\mathsf{PSPACE} \subseteq \mathsf{IP}$ in which $\tilde{A}$ is the arithmetized version of A) holds because we can write an interactive proof for TQBF^A in a similar manner as before with the addition of the algebraic oracle $\tilde{A}$; even though instances of A can be embedded in the QBF, the verifier can evaluate the arithmetized formulas by making calls to its $\tilde{A}$ oracle. The fact that this proof algebrizes is not a problem in and of itself. However, it indicates that this promising proof technique algebrizes, and for many open questions, their proofs must not algebrize. For example, any proof that $\mathsf{P} = \mathsf{NP}$ must not algebrize because there is an algebraic oracle $\tilde{A}$ such that $\mathsf{NP}^A \not\subseteq \mathsf{P}^{\tilde{A}}$, which would contradict $\mathsf{NP}^A \subseteq \mathsf{P}^{\tilde{A}}$, the algebrized version of $\mathsf{NP} \subseteq \mathsf{P}$. This algebrization barrier extends to many other central questions in complexity theory (including proofs of $\mathsf{P} \neq \mathsf{NP}$) [4]. As a recent example of proof barriers, Williams and Buss quantified the limit of alternation-trading proofs, a commonly used technique in proving time-space lower bounds for many NP-complete problems [5]. I'm intrigued by the "meta-discoveries" of the limitations of proof techniques, and I hope to develop creative approaches to overcoming them. One approach I'm particularly excited about has been coined "**ironic complexity theory**." This approach uses an upper bound for a problem in an algorithmic computational model (specifically, a circuit analysis problem) to prove a lower bound in a circuit model. This approach proved the remarkable result $\mathsf{NEXP} \not\subseteq \mathsf{ACC0}$, and it looks promising for proving several other circuit lower bounds [6].

At UW, I'm particularly interested in working with Dr. Beame because of his focus on computational complexity. Dr. Beame has contributed to known limits of lower bound proofs by showing the limitations of Nečiporuk's method for proving lower bounds on non-uniform models of computation [7]. Given an explicit function, Nečiporuk's method proves a lower bound by counting the number of subfunctions given restrictions on the input variables. The upper bounds on the lower bounds provable using Nečiporuk's method were already known for several types of explicit functions. Beame extended these results by proving the limit of the lower bounds on nondeterministic and parity branching program size obtainable by Nečiporuk's method. I'm interested in continuing this line of work by seeing if improvements to Nečiporuk's method could yield stronger lower bounds. In addition to Dr. Beame's work on the limitations of proving lower bounds, I'm intrigued by his work on lower bounds themselves. Dr. Beame has produced significant results regarding the complexity of SAT solvers, and I'm particularly interested in his time-space trade-offs for resolution proofs. This work is important because state of the art SAT solvers rely on conflict-direct clause learning (CDCL), which implicitly finds a resolution refutation – a proof that the input formula cannot resolve to true – for unsatisfiable instances. This means that lower bounds on the length of the resolution refutation imply time lower bounds on state of the art SAT solvers. Dr. Beame gives a time-space lower bound for superlinear space, proving that when memory is restricted to any polynomial in the input size, CDCL-based SAT solvers require superpolynomial time [8]. This result is important because it adds to the compelling evidence that SAT takes superpolynomial time (which of course would imply $P \neq NP$). Although I'm particularly interested in working with Dr. Beame, I'm also intrigued by Dr. Rao's research on computational complexity. Further, I'm excited to explore and open to researching other subfields of CS theory pursued across UW's theory group.

Preparation Along with standard theory curriculum, I've taken courses focused on complexity theory and cryptography, and I'll be taking a graduate level course in approximation algorithms next semester. These **theory tech-**

nical electives have given me grounding in theoretical literature and experience applying mathematical principles and proof techniques. I've also developed a deep appreciation for the **collaborative nature of CS theory**; I've enjoyed many Sunday evenings puzzling through complexity problem sets with my classmates. Additionally, I completed a **term paper on Gödel's Proof of Incompleteness**, a proof that every consistent mathematic system has statements which are true but cannot be proven within the system. I condensed Gödel's 36-page proof into an 8-page overview understandable by advanced undergraduate students, demonstrating a firm handle of the mathematical concepts. It took me more hours to read and deeply understand Gödel's proof than there were pages in the paper, and I relished the challenge, finding it engaging and rewarding. My term paper can be seen here: `https://ryan-moreno.github.io/resources/godel_incompleteness_theorem.pdf`

My first research experiences were in the **OSU Walker Lab** and the **USC Interaction Lab**, in chemical and mechanical engineering, respectively. They taught me transferrable skills including experimental design, cross-discipline collaboration, and a willingness to ask questions. My current research is in artificial intelligence and natural language processing. I chose this area because USC doesn't have ongoing complexity research. Additionally, I wanted to work in AI and NLP for my **honors undergraduate thesis** because it combines my two areas of study (computer science and English). Last year I developed my thesis proposal for **automated poetry generation using deep learning** based on current techniques in creative image and sound generation. Next semester I will be turning this proposal into my honors undergraduate thesis under the advisement of Dr. Xiang Ren.

I'm also working with Dr. Ren in the **USC Intelligence and Knowledge Lab**, using machine learning to develop a sequence labeling module to recognize named entities in text. Our approach uses trigger phrases that signal named entities (such as the phrase "ate lunch at" signaling an entity of type "restaurant"). We use a small number of manually defined trigger phrases to label named entities in unlabeled corpora. The labels found by the trigger matching module are then used as weakly labeled data to train a separate sequence labeling module. This step is critical because existing approaches to sequence labeling are limited by the expense of human-annotated data, whereas our approach augments a small amount of human-annotated data with a large amount of weakly labeled data. I'm helping to implement the machine learning algorithms and write our manuscript (recently submitted for publication).

Along with research experience, I completed summer **internships at Facebook and Microsoft**. Although I hope to pursue academic research, understanding the engineering practices in the industry sector will help me communicate effectively with industry engineers during potential collaborations. Additionally, having a spectrum of experiences will be beneficial when mentoring my future students.

Broader Impacts As a professor, I want to engage students in CS theory. I've seen this field brushed off as academic and inapplicable, which I believe is due to a lack of effective communication. I'm passionate about communicating with students at their level of understanding because feeling like you're already missing background knowledge when learning a difficult subject is discouraging. This discouragement is exacerbated in undergraduate CS courses because some students enter college with years of programming experience while others, particularly underrepresented minorities in the field, enter with none. By promoting **effective communication** in our field, I hope to retain a broader spectrum of students. I've worked as an **undergraduate TA** since sophomore year, specifically helping teach theory courses since junior year. At the end of most discussion sections, I'll have students tell me they appreciated my enthusiasm and methodical approach to teaching problem solving skills. I strive to communicate at a student's level of understanding, so there's nothing more gratifying than hearing from my students that I've done exactly that.

I've also **taught 3rd-12th grade students** through tutoring and classroom assistance. From teaching a 5th grade mini-course on poetry to tutoring high schoolers in pre-calculus, these experiences have all boiled down to the same puzzle of how to effectively communicate in a manner appropriate to the students' experiences and understanding. I've also participated in **STEM outreach** programs such as Robogals, a club dedicated to engaging young women in STEM and empowering them to consider engineering careers. As a future educator, I will continue to participate in STEM outreach programs and will strive to **create inclusive learning and research environments**, which I've learned the importance of firsthand. Last year I participated in **Out 4 Undergrad**, an LGBTQ+ tech conference. The conference theme was "covering," when people feel the need to hide aspects of their identity in work or academic environments. Thinking about my own experiences and hearing others' reminded me that creating an environment in which people are comfortable being their full selves is necessary to maintain an intellectually vibrant community. I learned the importance of having a diverse group of colleagues when building a customer-facing feature at Microsoft. A coworker with experience in foreign languages pointed out that I hadn't accounted for the formatting of right-to-left languages. Someone else suggested ways to make keyboard navigation easier for people with visual impairments. Intellectual diversity is valuable because it allows collaborators to fill in each other's experiential gaps.

I'm excited to become a professor because of the opportunity to teach and share my enthusiasm with students while collaborating with other researchers to expand fundamental knowledge in complexity theory. I'm eager to make lasting impact to our field through intellectual contributions, student mentorship, and promoting effective communication.

References

[1] T. Baker, J. Gill, and R. Solovay, "Relativizations of P=?NP," *SIAM J. Comp.*, vol. 4, no. 4, pp. 431–442, 1975.

[2] A. A. Razborov and S. Rudich, "Natural proofs," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 24–35, 1997.

[3] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *J. ACM*, vol. 39, no. 4, pp. 859–868, 1992.

[4] S. Aaronson and A. Wigderson, "Algebrization: A new barrier in complexity theory," *ACM Trans. Comput. Theory*, vol. 1, no. 1, pp. 1–54, 2009, ISSN: 1942-3454.

[5] S. Buss and R. Williams, "Limits on alternation-trading proofs for time-space lower bounds," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 18, pp. 11–31, 2011.

[6] R. Williams, "Nonuniform ACC circuit lower bounds," *J. ACM*, vol. 61, no. 1, pp. 1–32, Jan. 2014.

[7] P. Beame, N. Grosshans, P. McKenzie, and L. Segoufin, "Nondeterminism and an abstract formulation of nečiporuk’s lower bound method," *ACM Trans. Comput. Theory*, vol. 9, no. 1, pp. 1–34, 2016.

[8] P. Beame, C. Beck, and R. Impagliazzo, "Time-space trade-offs in resolution: Superpolynomial lower bounds for superlinear space," *SIAM J. Comp.*, vol. 45, no. 4, pp. 1612–1645, 2016.