Sketching: a Cognitively inspired Compositional Theorem Prover that Learns to Prove

Motivation: Advances in Deep Learning have accelerated Artificial Intelligence (AI) in an unprecedented way but these methods are unable to do higher level reasoning. In the past, reasoning was done with logic and search methods [5]. Although useful, these algorithms are unable to learn or adapt. Instead, I suggest building an AI system that learns and tackles this problem by leveraging insights from cognitive science & meta-learning through a technique called sketching [2].

Background: Sketching implements the cognitive science principle of compositionality for building programs by composing high-level concepts and then completing the details with a synthesizer. It's powerful because it avoids low-level reasoning and allows the synthesizer to avoid a combinatorial explosion during its search to complete the sketch.

Broader impact: Sketching has the potential to pioneer a new direction in AI where researchers explore systems that reason at a high level and create abstractions like humans do. In addition, Theorem Proving (TP) is important because it can be used for verification of hardware and software.

Research Question: Are cognitively inspired methods using sketching, combined with artificial neural networks, able to create useful abstractions that improve state of the art performance on the modern TP benchmarks like HOList [1]?

Research Plan & Methods: I propose extending the sketching framework suggested by Nye et at. [2] and extend it to TP. I will need: 1) a way to construct a formal proof sketch 2) an interactive TP to build a synthesizer to complete and verify the proof 3) a way to jump-start the process by processing the theorem to be proved. For the first part, I propose using Miz3 [3], a declarative proof language to construct formal proof sketches together with a sketcher Recurrent Neural Network (RNN) [2]. For the second part, I plan to use the prover used by HOList [1] to build the synthesizer and verify completed proofs. For the third part, I propose using Graph Neural Networks (GNNs) [4] to compute a vector from the target theorem to initialize my whole proving system. The system will work as follows: first, I will compute a vector from the target theorem using the GNN and use it to initialize the sketcher RNN that will produce a formal proof sketch (using Miz3). Given a formal proof sketch, I will then use a trainable Monte Carlo Tree Search (MTCS) method to complete the proof. After the proof is complete, the system will be trained to maximize the probability of proof completion under a distribution of time thresholds using gradient ascent as specified by Nye et al. [2] using the HOList proof data set [1]. Crucially, the verification of the proof will be done with HOList, thus allowing me to compare my progress with previous work [1,4].

The University of Illinois at Urbana-Champaign is the ideal place to combine machine learning and TP because 1) prof. Koyejo's (my advisor) expertise is in machine learning applied to neuroscience and 2) the strong presence of faculty in formal methods (over 20 faculty). I plan to take courses intense in this area like; CS476 Program Verification & CS576 Topics in Automated Deduction. These have a large project component where I can integrate my research and get additional guidance from experienced researchers in TP like prof. Gunter and prof. Parthasarathy.

Career Goals: My career objective is to become a professor in the impactful field of AI. This is the ideal career because I can have broad impact with AI and have my own research lab and be involved with students' projects where I give them individual guidance to be great scholars of the future.

References: [1] K. Bansal, S. M. Loos, M. N. Rabe; C. Szegedy and Wilcox, S. 2019. Holist: An environment for machine learning of higher-order theorem proving. ICML, 2019. **[2]** M. Nye, L. Hewitt, J. Tenenbaum, and A. Solar-Lezama. Learning to infer program sketches. 2019. **[3]** F. Wiedijk. A Synthesis of the Procedural and Declarative Styles of Interactive Theorem Proving. 2012. **[4]** A. Paliwal, S. M. Loos, M. N. Rabe, K. Bansal, and C. Szegedy. Graph representations for higher-order logic and theorem proving. CoRR, 2019. **[5]** Nikolaj Bjørner and Leonardo de Moura. z310: Applications, enablers, challenges and directions. In Sixth International Workshop on Constraints in Formal Verification Grenoble, France, 2009.