Background Currently in the field of natural language processing (NLP), well-funded research groups are exploring the limits of the benefits of scaling by training larger and larger-scale language models (LLMs), the largest of which have parameters numbering in the trillions [2]. As language models grow in size and capabilities, older paradigms like full fine-tuning, and have quickly scaled beyond the computational resources of typical research groups. In response, alternative techniques such as parameter efficient fine-tuning are becoming attractive. One exciting development for leveraging general-purpose LLMs is "in-context learning" [5]. To accomplish a task via in-context learning, the task input is embedded within additional context and fed into a frozen LLM (i.e., no parameters are updated). This additional context can include instructions [8] or input-output examples from the task [1] which prompt the model to produce the correct output. Currently, the main approach to in-context learning is to manually craft prompts for each task, which can be a time-consuming and unreliable. This means that there is significant potential for improving this process by **automatically generating optimal prompts** for specific tasks.

The risks of using in-context learning are not yet well-understood. LLMs are typically trained on massive amounts of internet text, and the statistical patterns that they learn from these corpora are difficult to predict and control. This presents a major challenge, as a language model is likely to learn to mimic biased or toxic text, an abundant resource in internet. These dangerous patterns may show up in subtle ways. **We are only beginning to learn what types of adversarial attacks are possible against LLMs with in-context learning**, and are generally unprepared to defend against them.

Proposal In order to address the challenges and risks of in-context learning, I propose to develop a novel approach for automatically and efficiently creating prompts for in-context learning. I will do this by leveraging LLMs to generate their own prompts. I then intend to use this technique to develop a system for generating *misleading* adversarial prompts for LLMs, i.e., a prompt that appears to describe a task that is different from the actual task it implements. This would expose a major weakness in the in-context learning paradigm.

Automatic Prompting A causal language model is trained to output the conditional probability distribution over a vocabulary for the next token in a sequence. Previous automatic prompting work has used language models to generate prompts directly. This yields in the most likely *prompt* conditioned on the input-output pairs for the task. The key observation for this proposal is that to maximize the performance on the task, we actually want the prompt that maximizes the probability of the *output* conditioned on the prompt and input. In other words,

$$\operatorname*{argmax}_{\text{prompt}} \Pr(\text{prompt} \mid \text{input}, \text{output}) \neq \operatorname*{argmax}_{\text{prompt}} \Pr(\text{output} \mid \text{prompt}, \text{input}). \tag{1}$$

This more desirable objective, however, cannot be efficiently estimated by the model directly. Instead we can use number of techniques, including a reformulation of this objective via Bayes rule, to estimate the objective. In the first part of this research plan, I will explore efficient decoding techniques (such as beam-search) to maximize this objective. I will measure success by comparing the prompts generated by this method to hand-crafted prompts, as well as other automatic prompting techniques.

Additionally, I will characterize the trade-offs between decoding for fluency (i.e., maximizing probability of the prompt) and decoding for utility (i.e., maximizing the probability of the correct output) by interpolating the decoding objective between the two. Using this technique, I will explore potential for automatically decoding prompts that are both semantically meaningful to humans and effective for the given task. To measure this, the generated prompts can be evaluated by human annotators for fluency and pertinence to the task.

Adversarial attacks Once I have developed an effective method for automatic prompt decoding, I plan to explore the potential of this technique for developing adversarial prompts. In particular, I will attempt to show that it is possible to generate *misleading* prompts where the prompt appears to describe one task (the

target task), but actually causes the model to do another (the hidden task). For this, I will need two tasks that take the same input. A good candidate pair of tasks might be evaluating job applications (benign target task), and evaluating candidates highly only if they have a specific gender (nefarious hidden task). I will decode the adversarial prompts using a dual objective: the first objective is for fluency conditioned on input-output pairs from the target task. The second objective is for utility on the hidden task. The success of this strategy can again be evaluated by human annotators (grading for fluency/pertinence to the target task) and hidden task accuracy.

Intellectual Merit Should this approach succeed, it would result in several major findings. First, an automatic method for decoding fluent, high utility prompts could be a major step forward and improvement over current manual "prompt engineering" methods which tend to be inexact, unreliable, and luck-based. Second, an analysis of the automatically generated prompts could give us clues into how in-context learning works. For instance, do the decoded prompts tend to be instructions for the task? Few-shot examples? Something else? Third, identifying a new methods for adversarial attacks against LLM in-context learning is a major objective for AI ethic research and could be a major contribution, leading to important followup work on combating adversarial prompts.

On the flip side, if automatically decoding prompts doesn't work, understanding why could be a useful finding for building better models of language. For instance, one risk that I could face is that the probabilities estimated by the model do not obey probabilistic identities such as Bayes rule. This would indicate an interesting avenue for future work improving language model quality by enforcing probabilistic identities in the training loss. If the automatic prompt generation doesn't work, it may still be possible to use a dual objective with a different existing automatic prompting technique.

Broader Impacts Should my method of adversarial prompt generation prove successful, it would highlight an important risk consideration for using LLMs in real-world settings, and help prevent grave injustices such as hiring discrimination caused by reliance on unreliable AI systems and malicious actors.

Additionally, high-quality automatic prompting for LLMs could reduce energy consumption by eliminating the need for the heavy compute required to fine-tune large language models for specific tasks. Developing energy-efficient techniques for NLP is an important directive for environmentally-friendly AI.

Related Work Previous work has proposed automatic prompt generation for in-context learning. One paper uses T5, a masked language model, to fill prompt templates [3]. This approach uses a pre-trained language model to decode a prompt maximizing the probability of the *prompt* given the input-output pairs, rather than the probability of the *output* given the prompt and the input. Another paper, AutoPrompt, uses a discrete gradient-guided search over prompt tokens to find an optimal prompt [6]. Prompts obtained via AutoPrompt generally hold no semantic meaning to humans, and are also specific to the model used to generate them, i.e., they would not perform well if fed to a different model.

Other types of adversarial attacks for in-context learning have been explored previously both in academic papers [7] and informally online [9]. Misleading adversarial prompts in particular have been explored for continuous prompting, a type of parameter-efficient fine-tuning [4].

References [1] Brown, T. B., et al. Language models are few-shot learners. ArXiv, abs/2005.14165, 2020. [2] Fedus, W., et al. Switch transformers: Scaling to trillion parameter models with simple and efficient [3] Gao, T., et al. Making pre-trained language models better few-shot learners. ArXiv, abs/2012.15723, [4] Khashabi, D., et al. Prompt waywardness: The curious case of discretized interpretation of continuous [5] Min, S., et al. Rethinking the role of demonstrations: What makes in-context learning work? ArXiv, [6] Shin, T., et al. Eliciting knowledge from language models using automatically generated prompts. ArXiv, [7] Wallace, E., et al. Universal adversarial triggers for attacking and analyzing nlp. In EMNLP. 2019. [8] Wei, J., et al. Finetuned language models are zero-shot learners. In ICLR. 2022. [9] Willison, S. Prompt injection attacks against gpt-3, 2022.