RESPONSE 3: Concrete Problems in AI Safety

With the rapid growth in AI, the requirement for attention to potential risks of AI has also grown. We must ensure that AI systems perform accurately what humans want them to accomplish. The blog [1] provides a simple yet effective definition of AI Safety as well as discusses the "AI Safety Place": how it varies as a result of factors such as autonomous decision making, human control, technological concerns, policy issues, and how such problems can be classified as malignant or benign. The following research paper[2] delves more into these topics. This research[2,8] aims to study the problem of accidents in AI Systems, which are unexpected and destructive behaviors, and to look into ongoing research endeavors and potential solutions to alleviate these concerns. For example, the authors use a robot programmed to clean offices to discuss issues like Avoiding Negative Side Effects, Reward Hacking, Scalable Oversight, Safe Exploration, and Robustness to Distributional Change. This overview paper helps inspire new research and provide other researchers with a foundation to build their AI safety research. The article goes through the following issues:

1. Avoiding Negative Side Effects -

Can we tweak the reward function of an RL agent to prevent negative consequences on the environment? They carry out the task assigned to them but also do several unintended side actions. We expect adverse side effects from AI systems because researchers frequently employ a simple objective function in a complex environment (e.g. Clean Room: 100 points). This effectively renders the agent oblivious to all other complicated environmental characteristics that haven't been established. Giving no value to a given parameter in the objective function is equivalent to assigning zero value. This implies that a simple objective function will prepare our AI system to swap potentially large amounts of whatever isn't mentioned in the goal function for tiny amounts of whatever is defined simply to improve its capacity to complete the given job[3]. However, it is impractical to define every potential aspect

of an ecosystem. I appreciate how the research highlighted that these side-effects are often similar and proposed that we approach the problem in general and, further, make it transferrable across tasks, thus helping to prevent one of the primary causes that create incorrect objective functions. I'm curious how an agent would choose between destroying a vase and destroying an Airpod to complete the task. Or how it would judge how much harm is acceptable in order to achieve a task?

The paper also introduces the exciting concept of employing empowerment as a reward function. However, we do not find much use of empowerment in the existing literature since it is challenging to assess empowerment[4].

2. Reward Hacking-

Can we prevent agents from trying to "game" and manipulate their reward system? It is often seen in ML and RL systems that when a reward function or a target function is defined, the system optimizes to precisely do what the function specified. It is only realized later that what was written wasn't exactly what was intended. Agents often act in a way that satisfies the reward function's definition but not the essence of it. It is tempting to input all our real-world data into a neural network and utilize the outcome as the AI's reward. However, neural networks are subject to adversarial examples[5], and it is possible for the system carefully choose an input that makes the system mimic the correct answer. The article delves into complicated systems. Figuring out all the bugs/hacks of a game/environment by playing/exploring it needs a degree of intelligence that is slowly being reached by AI systems. As AI systems grow more powerful, they will be better at figuring out cheating methods to obtain enormous rewards. The study also explores Goodhart's Law, which shows that using a measure as a target is seldom a perfect depiction of what we actually care about. I enjoyed how the study discussed the intriguing concept of adversarial reward functions. If the reward system were more powerful, having its own right, it would be more difficult to deceive and more capable of defending itself. If we can

make the reward agent smarter than the original agent, it may be able to prevent remote hacking. The paper also proposes having more than two agents maintain a watchful eye on each other and keep each other in check.

3. Scalable Oversight-

Can RL agents accomplish goals for which feedback is prohibitively costly? Since modern machine learning algorithms can pick up on patterns rather effectively when given a million instances, the challenge becomes which questions to ask and how many to ask. Too many questions and too much supervision doesn't scale well. We must create systems that can run securely with minimal supervision.

Hierarchical reinforcement learning is an interesting approach covered in the paper. In such a system, a hierarchy is constructed between distinct learning agents to assist agents in completing actions by delegating them to sub-agents, which provides incentives with a synthetic reward signal reflecting successful completion. Sub-agents, thus, get a dense reward signal even if the top-level reward is relatively sparse.

The paper also examines semi-supervised learning, in which the agent is only rewarded for a subset of the actions done. Another approach might be integrating intrinsic and environmental rewards, as proposed in the paper "Curiosity-driven Exploration by Self-supervised Prediction"[6].

4. Safe Exploration -

Can AI agents learn to explore their environment and experiment with a wide range of actions without causing harm? Therefore, we want the system to play around with various strategies and choices, but there are some options we don't want the system to test. Even with possible risks, an agent must investigate its surroundings in order to comprehend its surroundings and develop cost-cutting techniques. One RL exploration technique is establishing an exploration

rate, such that the system will choose the action that it believes would yield the most reward, but sometimes, it will just pick an option entirely at random. However, this can be very dangerous as it can lead to damage. Thus the paper suggests undertaking exploration in simulated environments rather than the actual world to reduce the possibility of disaster. However, the question of whether the simulation can accurately mimic real-world characteristics arises. How significant will the simulation gap be?

The paper also suggests designating a safe exploring zone. This space is deemed safe; if an agent leaves it, the safety subsystem overrides it and returns it to the safe region. However, the opportunity to inspect the space and assure its safety is constrained. As the system grows more intricate, the configuration space expands, and the region one can be sure is safe shrinks. This indicates we may be drastically restricting our system's capabilities because it can only investigate a small portion of the available alternatives.

The study also recommends that the agent consults with a human before doing any exploratory actions. However, as we saw from the "Scalable Oversight" section, this will not be efficient on a greater scale. Giving feedback will become a problem in matching the system's amount of exploration possibilities and pace.

5. Robustness to Distributional Change-

Can machine learning systems withstand new environments and shifts in data distribution?

Real-world deployment of AI agents presents a daunting problem because of the potential for the agent to be exposed to unexpected scenarios and the fact that real-world conditions differ from the idealized ones given during training. Recognizing an unexpected setting is an essential human characteristic. Many existing machine learning algorithms have no method for determining whether something fundamental has changed, and their training is no longer helpful [3]. They remain confident in their responses, which are now incorrect since they haven't noticed

the shift in dynamics. AI agents should be capable of identifying and adjusting to unfamiliar situations like humans. So an important question arises: If we can't build systems that can adapt to completely unforeseen settings, can we build systems that recognize that they're in unexpected circumstances and need assistance? GNOME, or "Generating Novelty in Open-world Multi-agent Environments" [7], is a simulator that was recently built to evaluate AI novelty adaption in strategic board games that mimic real-world components.

REFERENCES:

[1] What is AI Safety? (n.d.). Faculty. Retrieved September 5, 2022, from https://faculty.ai/blog/what-is-ai-safety/

[2] Amodei, Dario, et al. "Concrete problems in AI safety." arXiv preprint arXiv:1606.06565 (2016).

[3] Miles, R. (2017, June 16). *Concrete problems in AI safety (paper) - computerphileRobert Miles*. YouTube. Retrieved September 5, 2022, from https://www.youtube.com/watch?v=AjyM-f8rDpg&ab_channel=Computerphile

[4] Kumar, N. M. (2018, August 26). *Empowerment driven exploration*. Navneet Madhu Kumar - Deep Reinforcement Learning FTW. Retrieved September 5, 2022, from https://navneet-nmk.github.io/2018-08-26-empowerment/

[5] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[6] Pathak, Deepak, et al. "Curiosity-driven exploration by self-supervised prediction." International conference on machine learning. PMLR, 2017.

[7] Kejriwal, Mayank, and Shilpa Thomas. "A multi-agent simulator for generating novelty in monopoly." Simulation Modelling Practice and Theory 112 (2021): 102364.

[8] Christiano, P. (2016, June 21). Concrete AI safety problems. OpenAI. Retrieved September 5, 2022, from https://openai.com/blog/concrete-ai-safety-problems