Proceedings of The National Conference
On Undergraduate Research (NCUR) 2018
University of Central Oklahoma
Edmond, Oklahoma
April 5-7, 2018

Learning from AlphaGo Zero: Towards a Less Biased Approach for Autonomous Driving

Joshua Pachter
Computer Science
University of Rochester
Rochester, New York

Faculty Advisor: Professor Hayley Clatterbuck

Abstract

When imagining the ethical autonomous car, we place a burden on the creators to do the "right" thing through a thoughtful training of machine learning algorithms. In some cases, deliberately excluding human input from a training set can actually improve outcomes; for example, when supervised learning was omitted during training of DeepMind's AlphaGo Zero, the results surpassed that of any previous version of AlphaGo. AlphaGo Zero's approach relied on reinforcement learning and used Monte Carlo tree search (MCTS) as a policy improvement operator, allowing training to be done completely through self-play. Although this most recent iteration was ultimately more successful, prior versions of AlphaGo that used supervised learning were better at predicting expert human behavior. Through an ongoing review of existing literature and discussions with experts in human-robot interaction and ethical AI, I will investigate if using a similar strategy to train autonomous vehicles without any human "good driving" data can improve outcomes as it did for AlphaGo Zero. I will then discuss the ethical implications of such an approach in the context of driverless cars.

Keywords: Ethical Autonomous Driving, Value-Aligned Learning, AlphaGo Zero

1. Background

The ethical challenges of autonomous driving are innumerable, and many of the approaches being discussed today reflect our present technical limitations as one would expect. However, to achieve a future where self-driving vehicles are a reality, we must look past these limitations and consider what we will have to do eventually in order to meet society's stringent performance and safety requirements. We have to imagine a gold standard.

Achieving better performance for autonomous vehicles is possible by relying on one of the core enabling technologies behind modern artificial intelligence: machine learning. Machine learning techniques use statistics and logic to help solve abstract problems that were previously the sole purview of humans. They can play board games, diagnose some diseases, and even drive cars.[12]

This breakthrough technology does not come without any risks, of course. Any time we train a machine learning algorithm, we supply some degree of information innately.[4] With that information comes inevitable bias. Bias here refers to poor intuitions that the software could follow on its way to achieving its goal because of an extrapolation it made from the information provided by its creators. For example, bias could be a problem if poor driving data is accidentally included in a training set. Or even more elusively, if we neglect to consider that zip code census data may be tied to race, we may inadvertently create a racist AI even though we never explicitly specified race as a parameter[13].

There are further domain-specific challenges for autonomous driving. One of the main obstacles is that we do not know the moral rules that we want autonomous cars to abide by, and even if we did they would be so complex that creating a system that adheres to them in every imaginable instance would be an incredible implementation challenge. We cannot program moral rules top-down because a) we do not know them; philosophers still have not figured it out

after millennia, b) if we did know them they would be super complicated, and c) they would also be too rigid. Hence, we have to rely on machine learning. However, this may actually be to our advantage in more ways than one. Since we admit that moral rules are not so easily known, this could allow for a policy to be learned in a way that is more flexible and context-sensitive. Ultimately, we want self-driving cars that reflect our values without overfitting to human biases, and we want to machines to perform the task of driving safely even better than we do.

While many common machine learning approaches are plagued by various forms of bias, recent developments in artificial intelligence have shown promise when it comes to avoiding some types of harmful bias. Namely, DeepMind has constructed a version of their Go-playing AI, AlphaGo, that learns how to play the ancient game almost entirely on its own over just few days' worth of training.[1] Of course, we have to provide something for it to start with, and in this case it is the most basic rules of the game. There are some clear differences between a board game like Go and a far more complex problem like autonomous driving. For our purposes, two of the most relevant are 1) real-world driving is subject to randomness and uncertainty in a way that we do not need to account for in a board game, and 2) the consequences of messing up in a self-driving car are far higher than the consequences of losing a game.[2]

However, it is possible that the fact that the rules are unknown is to our advantage, and through iteration allows us to create a system that gets better and better without the need for an explicit standard to benchmark against. In other words, notice that if we knew the rules so absolutely, there would be no room for compromise or pluralism, while these aforementioned limitations lend themselves quite well to a moral framework that allows for some degree of ambiguity. The notion of accepting moral uncertainty may be frightening, but it is a necessary step in the progression of this technology and frankly not a risk in that us human agents also execute behavior with a requisite amount of moral uncertainty. The same will have to be true for machines. Furthermore, the key challenge is not to uncover a set of unquestionable ethical laws; such a task is not feasible anyways. Instead, we should consider some the most crucial technical and ethical limitations and try to come up with a clever solution that dodges as many bullets as possible.

2. Machine Learning Options and Their Limitations

Modern machine learning approaches hold great promise but also come with some downsides. In the arena of self-driving cars, most employ some version of supervised or reinforcement learning, each of which has their potential uses as well as shortcomings. Supervised learning is demonstration-driven, and ultimately shows machines how to predict human behavior with no real sense of the underlying logic behind decision making. A common example of supervised learning is a system which is trained to classify images, such as distinguishing between images of cats and images of dogs. By adjusting parameter weights through training to fit some provided ground truth, the software eventually learns to classify dogs and cats. Supervised learning can be a quick way to make a car behave like human drivers by providing large amounts of good human driving data. Although this may be a quick and dirty way to get a car driving, it neglects all ethical considerations as it is only trained implicitly as a moral agent in that it is only ethical insofar as it follows our own behavior which is assumed to be also ethical. Imagine that some of the data provided contains a bad behavior or driving habit, the trained system will copy that behavior and have no choice but to believe it is perfectly acceptable given that it was part of its training data.[16] Additionally, using an explicit set of data with a supervised learning approach creates vulnerabilities to important omissions in the data set which will leave the system free to make mistakes upon encountering a scenario for which it has no training data. As you would expect, when AlphaGo Lee was trained using supervised learning, the results were that the machine eventually learned to play as well as kinds of human players on whose data it was trained. No matter how many hours of training are performed, this threshold will never be exceeded.[1]

It is clear that there needs to be more work on how to imbue a sense of reason into these systems instead of merely a blind capacity to follow. Another common approach, reinforcement learning (RL), succeeds in some ways where most supervised learning fails, but ultimately faces a different set of equally serious problems. Reinforcement learning uses cost functions that programmers provide instead of large amounts of human data. The advantage here is that there is no risk of creating accidental bias since there is no "human example" provided altogether. However, this approach does require that the desired behavior is formalized through cost functions. For an application like this, it makes sense to specify just the set of rules of the game and nothing more. Therefore, a great deal of work is left up to the algorithm to figure out how to apply those rules to actual gameplay. In the case of AlphaGo, the algorithm observes when it has either won or lost, and from there tries to figure out how it might generate a winning state and avoid a losing one. Alternatively, humans could take on a more involved role by providing not only the classic rules but also some clever strategies than one might find in a book on how to play Go. This would alleviate some of the training burden up front, but also potentially closes off avenues for the algorithm to learn new and innovative strategies on its own that humans

may not have been able to discover themselves. The latter case would be a perfect example of a sneaky kind of bias that ought to be avoided. Initially it may seem innocuous and expedient, but remember that the goal is to create something that drives better than a human, so we want to avoid this kind of training as much as possible as it will only pave a path to human-like driving. All in all, this makes some form of reinforcement learning a good candidate for applications like playing a board game such as Go where the rules are known and clear. By knowing the rules and not just a set of a priori data, we allow for innovation in a way that would not otherwise be possible. A perfect example of this can be found during the gameplay, where an earlier version of AlphaGo played non-standard moves that illustrated higher levels of insight than a human player.[1] We want this kind of innovation in our self-driving cars, too.

While certainly promising for board games like Go, what about applying reinforcement learning to the challenge of self-driving cars? One significant challenge with this transition is that there is no universally agreeable moral code that can be formalized using cost functions, as these moral rules are not so clearly known. Furthermore, even if everyone in the world could agree on a complete set of moral rules, there would still inevitably be mistakes in the transcription of these rules, and they would surely fail to capture important nuances.

A theme begins to emerge here, namely that providing explicit information is harmful because it will inevitably be tainted with human bias, since the aforementioned "rules" cannot be well known. How, then, can engineers create a reasonable system and train it to do something that they themselves admittedly do not know how to do? Humans' inability to be certain of moral law actually becomes a key advantage here, as it means they can create a system that does at least as good of a job at ethical decision making as any new driver.

Instead of trying to hardcode our moral standards, it is better to follow an approach not so different from raising children.[3] By acknowledging and even encouraging a certain degree of freedom on the part of the "child" (robot) and then trying to shape its behavior to align with their own, creators can allow for novel behaviors that are actually morally superior. What might this process look like? Although the domains have some important differences, we can learn how to build from the ground up by referring to an example like AlphaGo Zero, and use this approach to create a more ethical system for self-driving vehicles that in itself has a higher capacity to reason than do implementations that were trained on explicit sets of data.

3. Yielding to Human Behavioral Boundaries as a Moral Framework

Concerns of bias are not limited just to human data-driven approaches. All systems, even Deep RL based ones like AlphaGo Zero, necessarily have innate knowledge.[4] Insofar as creators impart this knowledge onto a system and thereby provide innateness, there will be reasonable concerns about dangerous biases.

Let us accept that all current and future systems will be in some way predisposed to a certain kind of reasoning or thought, and consider what options there will be to mitigate this bias. Admitting that there is a degree of innateness necessarily, it is incoherent to say that human bias will be absent altogether. This is not the end of the road, however, it is merely an obstacle to consider carefully.

A logical starting place to avoid pitfalls here would be to think of "bad" biases, and proceed with this definition of bias from here on out. Not all forms of human intuition are biases and they are certainly not all bad. Many deliberate survival decisions are made for good reasons and follow human intuition, but are not an example of harmful bias. Examples of criterion to avoid imbuing include evaluating and making a decision based off the age, gender, or race of individuals who are outside the vehicle, and more simply bad driving habits such as hugging the outside of the road too closely, or not leaving enough time to brake for a red light. The question at hand becomes: "which biases are legitimate and reasonable to adhere to, and which are based on arbitrary distinctions that would only result in moral and social catastrophe (or minimally, suboptimal, merely human performance) if implemented?" This is where some common approaches go wrong. To answer this question completely means one has likely provided too much explicit information.

It may seem intuitive to intervene like this in dilemma scenarios. However, this is not necessarily true; all that is required is a decision, not human takeover. So long as the intuition behind this decision is formed from within the algorithm's own locus of control and not handed down from humans, the action will remain unbiased. Of course, creators necessarily have a role in adjudicating certain behavioral outcomes to inform the development of the algorithm through self-play into something behaves reasonably. This way, creators of the system are not implicated in explicitly assigning moral weight on the basis of something like ontology, a task that would implicate them in imbuing harmful bias, but instead have a reasonable pathway through which outcomes can be effected indirectly and ensure alignment with our values. More on how to intervene gently as to not allow for bias later.

How, then, can this gap be reconciled, and can a system arrive at the ideal solution without having to provide data verbatim that could be harmful if generalized? A system should only ever learn how to be the best possible proxy for a human driver. This does not mean that the solution merely requires training autonomous cars on sets of driving data from the best human drivers. Such an approach driven by deep supervised learning could be accomplished by observing steering wheel angle paired with images of road scenarios to learn how the human driver adjusts the former to account for the latter.[7] The problem is that what drivers should do and what they actually do are really different things, especially in quick decision scenarios. Therefore, this reliance on human reactions to various driving scenarios is misguided, and certainly not a substitute for any deeper decision making capacity for a machine. Recalling my earlier concerns about supervised learning, this system would merely learn to predict human behavior, which is suboptimal.

Humans can get away with making mistakes in their driving, but powerful computers will not be able to cite the same excuses, such as emotionality, or "not enough time to think." This is another reason why simply training systems on the behavior of humans (supervised learning), trying to apply a reward function (reinforcement learning), or any variation of the previous such as inverse reinforcement learning where a machine tries to learn the reward function implicit in a data set of human behavior, all fail, as all of these approaches rely too heavily on the idea that humans know what they are doing and are behaving in a way that is perfectly aligned with their most central moral beliefs.

By deferring the responsibility of coming up with specific moral weights to the machine, a huge pitfall of human bias is avoided. The risk of self-created bias, however, remains. Addressing this issue requires both a stringent array of testing, and also seems wanting for some ideological enforcement of why not-yet-tested instances will also be successful. It may seem counterintuitive to think that safest option is to ascribe more freedom and less restriction to the machine, but remember that the goal is to align with human values and perform better than they can. To that end, it would be impossible to specify a complete and explicit set of human data with the hopes that a machine would learn to outperform that standard. The obvious danger in having less explicit instruction is made up for by the manufactured boundaries and checkpoints that make it difficult to maintain any harmful habits.

Given that there has to be at least some degree of information provided by creators, a middle path becomes the best option. What degree of primitive information would be sufficient to allow for an autonomous vehicle to effectively teach itself the rules and fill in the blanks? This process would have to be carefully monitored, tested iteratively ad nauseam, and ultimately be evaluated with three questions:

> 1) Given a scenario, does a simulation of the vehicle's actions pass a behavioral test from the perspective of a diverse audience?
> 2) Is the primitive information we provided initially fundamental enough to constitute sufficient separation from our own harmful biases?
> 3) Can engineers prove that the system will continue to make good decisions for new scenarios that it has not yet encountered?

Success in this domain should be defined empirically by the performance of systems with varying degrees and kinds of innateness. Innateness is a necessary component of all machine learning systems and arguably also of all agents. Gary Marcus defends his perspective as a nativist on why this is true in his paper entitled *Innateness, AlphaZero, and Artificial Intelligence*. Marcus describes the "reductive" strategy, which starts with a small amount of innateness and then performs search iteratively to achieve some higher degree of knowledge as one way to achieve a sufficient level of intelligence. He also says that a "top-down" approach would be equally efficacious, wherein we perform a thorough cognitive study before setting innate primitives and use this as a starting point for AI, which involves a higher degree of innateness prior to any iteration.[4] Marcus is right in saying that both of these approaches and any number of other foreseeable approaches all require something to be innate, but the incentive for the reductive approach in the practical sense is that we are certain that we do not know which primitives will be the best to start with, even if we do conduct a thorough evaluation of our own cognitive systems. It makes more sense to place the blame on a highly iterated-upon and deeply simulated system instead of in the hands of some group of engineers. In reality it is unlikely that a system failure would completely and exclusively blame a poorly-chosen innate factor, but it still is logical to distance human creators from the role of providing explicit data to a model in lieu of self-learning.

4. Advantages of AlphaGo Zero

When DeepMind created AlphaGo Zero, they claimed that they had created a system that could start tabula rasa and achieve successful game playing ability. However, as Gary Marcus reminds us in his paper, no system can possibly

start completely from zero with no innate algorithms, otherwise there would be no way for it to grow and learn.[4] So, how exactly does AlphaGo Zero work, and what can we take from their approach when looking at the realm of autonomous driving?

AlphaGo Zero uses deep reinforcement learning and Monte Carlo Tree Search (MCTS) to predict the probabilities of success or failure if a specific move is played. Through many iterations of self-play, a neural network will be improved as it matches itself closer and closer to fit the move probabilities given by previous simulations, and the updated version is used in the next iteration and the improvement process continues. Crucially, there is a clear determination of the value of a given state. Using the unwavering rules of the game, the reward value of any node can be determined over the domain $r_T \in \{-1,+1\}$[1]. Through the process of self-play, simulation, and an updating of the neural network's parameters to match the new search probabilities as closely as possible, the system as a whole becomes better at playing the game over time, eventually becoming as good or better than any human or previous version of AlphaGo.[1]

Not only does this version become more powerful than other software and human players, but it is far cleverer than any other version as well. When playing Lee Sedol, the world champion Go player, a version of AlphaGo that shared the same underlying search and self-play functionality as AlphaGo Zero defeated the human champion with an unexpected move that shocked Go experts.[13] This innovative technology allows for creation of new knowledge, not just the blind following of human behavior.

In order to follow this approach and reap the clear benefits of starting even close to tabula rasa in the context of self-driving cars, we have to consider a few key domain differences. First, as said earlier, humans cannot know what (moral) rules to program in. Second, the number of "legal moves" from a given state is not enumerable or even known as is the case in Go. Third, since the reward value is for a given state cannot be a function of any discrete set of rules, there must be an alternative approach for imbuing a sense of value.

With these limitations noted, there is ample reason to be interested in AlphaGo Zeros approach as a good parallel for autonomous driving. Namely, starting with almost no human data proved to be a huge success for AlphaGo Zero both in terms of innovative capacity and ability to make reasonable decisions without needing to be provided with explicit information. This trait is invaluable in the arena of self-driving vehicles, as an AV can encounter any number of novel scenarios where either a predetermined index of moral value or a supervised learning-driven algorithm would could fail to provide a reasonable solution just by coincidental omission.

5. Accommodating our Approach

The aforementioned problems reaffirm that the machinery from AlphaGo Zero can be taken, modified, and applied to the self-driving car problem to create an optimal solution. The idea of a single combined policy and value reinforcement learning-driven neural network should be kept, as this is what allows for learning through self-play. Since allowing for self-play is key to the network's development, some form of MCTS will remain to serve as a policy improvement and evaluation. Here is where some changes need to be considered. The first task would be to create a representation of some form of value. Without value, there is no way for a process like MCTS to form any notion of higher or lower rewards, and therefore no way for the neural network to learn to match these rewards. These most fundamental values are what need to be innate, as they are the only logical ground truths for this system to operate on. As discussed earlier, we cannot claim to be the professors of our own moral code, so we have to do the next best thing: guess.

Obviously, these are not just shots in the dark, there are two good reasons to feel confident with this approach: One, although an absolute moral code may be elusive, vague moral truths that will go largely undisputed are more commonplace. For example, human lives have value in themselves.[17] Two, since these trials are run through simulation before necessitating their performance in the real-world, there is no need to worry about anything bad happening should it mess up, which in this case means guessing a bad set of values higher-order values. Notice that this position is only attainable since there is no explicit cost function, but rather a set of starting points which will be iterated upon extensively and further refined in a manner which will be explained shortly. I refer to these most fundamental claims of value as "primitives."

After a sufficient amount of self-play, the preliminary results of the system are tested by providing a set of mundane and complex scenarios, and observing the resulting behavior. Ideally, this would happen after not more than a day or so of self-play, so that if a system is suggesting behaviors that are clearly undesirable, it can be abandoned without

having to invest any more time into its development. From there, overseers of this process can tweak the primitives slightly and try again.

In any other case where a system seems even remotely correct, overseers can engage in a sort of assisted training by providing small responses of binary approval or disapproval. Upon receiving this feedback, the algorithm would incorporate these changes into the structure by changing its choices starting with the lowest certainty value and re-outputting its response for the same scenario. This means that if the feedback provided by overseers was that the response was unsatisfactory, the logical change to make would be to pick the action with the next highest probability based off the self-play done thus far. This process is repeated until the response is satisfactory. Upon achieving approval, the rewards for the moves changed will be artificially inflated so that they will be chosen again should the same scenario be received. Then, the same process of updating the parameters of the neural network to match the newly generated rewards will take place, very much in the same way as the search probabilities are updated following a round of self-play. This process will be continued until a point is reached where a high number of scenarios are handled satisfactorily. Exactly how high this quantity need be is debatable and potentially contentious, but it should be high enough that the standard it enforces is probably a considerable amount higher than any previously accepted agent who is allowed to make decisions in these scenarios, i.e. human drivers.

Additionally, this process avoids an issue around conflicting moral norms. This is a challenge often had in these domains whereby a moral norm is set with the intention of it never being violated, but there can always be an instance where such an absolute imperative will result in a stalemate. For example, to explicitly define that a human life may never be sacrificed would be challenging in a scenario in which it seems the only options all involve the sacrifice of human life. In such a case, the system should still make the best decision possible. Although sometimes distracting, these "trolley cases" can help motivate an understanding of ethical edge cases, and they also will surely come up as some scenarios to use during testing in the previously described process. Another advantage of not supplying any explicit rules is that both these kinds of edge cases and mundane scenarios can be used in training without the risk of overfitting to either.

A relevant practical issue is that to train a system on a large portion of this huge state space would be inefficient and unrealistic. To mitigate that, there needs to be some sense of which scenarios are the most important so that the algorithm can choose to pursue searches that will yield the most useful results. This increases the chance that when the time comes to make a real-life calculation about some important dilemma or mundane situation, the algorithm will be confident in the suggestion it provides because it has traversed that territory before.

This can be accomplished in two ways. One, we could supply scenarios as part of the set of innate information that we think are the most relevant. However, this is suboptimal for a few reasons, namely because creators could accidentally omit a certain kind of scenario, which could bias our results. Also, the kinds of scenarios that are provided might all have something in common or conversely all be missing some important factor, also resulting in the teaching of harmful bias. Another solution involves a different subset of the system that is responsible for generating its own scenarios. Such an algorithm would have to have some understanding of what is ethically efficacious, so that it provides itself with a sufficiently diverse group of scenarios. Additionally, it would have to know what factors exist in the universe of possibilities so that it can provide scenarios that are nuanced with many different recognized characters and situational factors.

6. Generating Next States

There is another critical component of the challenge that will be far more difficult for an autonomous vehicle than it is for AlphaGo Zero. This is the second concern from earlier, which is that generating the children nodes from a given state is no longer as simple as enumerating all of the legal moves. Since the real-world is full of uncertainty and is non-discrete, figuring out what might come next is difficult and important to get right.

Creators will inevitably have to create an algorithm that understands not just what is likely to come next, but what kind of scenario is the most likely, and which are the most ethically consequential. These will be important in determining which states we even bother to generate in the first place, and which of those we evaluate first in an instance where computational expense is a barrier.

To understand how this might happen, humans should appeal to their own predictive intuitions when it comes to anticipating the actions of those around them. Simply put, they can combine their knowledge of the laws that govern physical motion with their knowledge of the kinds of actions that a given object is capable of making to get a pretty good sense of this space. This should be done through a separate neural network which is trained ahead of time — meaning before self-play and trials are conducted.

Just like the previously discussed methodology, this neural network will be mostly self-taught to avoid bias for all the same reasons as before. In order to parse a given scenario, the first step will be to classify each of the relevant players. For example, two cars and one pedestrian could be the set of meaningful objects for which we need to anticipate moves. There will be an ontologically-derived weight that is static for each category of object. Those that are unclassifiable will be prioritized by some other set of properties such as size and/or motion. An object like a car will have a high predefined weight as it is generally capable of causing significant amounts of harm, whereas a pedestrian will have a low potential to cause significant harm and as such is assigned a lower weight. Note that although the pedestrian will likely not cause harm, it is crucial that it should not *be* harmed. So, by expanding the object with the highest weight first, (the car) this neural network generates its children nodes, which are the most likely actions it could take. Even though the harm that may be caused to a pedestrian or an object with lower weight is never explicitly evaluated, it happens implicitly by ensuring all the objects that may cause harm (such as the car) are expanded in an effort to predict cases where it might collide with a lower weight individual like a pedestrian.

Car manufacturers, insurance companies, and the government combined have large amounts of data on collisions and thus have a good sense of what tends to go wrong leading up to an accident.[9] Furthermore, work has been done to show the possibility of modeling complex traffic patterns using Dynamic Bayesian Networks.[8]

There are inevitable drawbacks of using such a system, even past the fact that it is non-trivial to generate the next most likely states with confidence. Namely, achieving sufficiently high look-ahead in a short amount of time, and to a lesser extent a near complete set of likely actions for all players in a scenario are computationally expensive. These challenges should not be overlooked, but there are a few small improvements that can be made right from the start.

First, assuming that object classification is performed robustly and consistently, the benefit of having a hard-coded weight for each type of object means that a constant time lookup can be performed given the object type revealing the weight that object carries, and the types of actions it is likely to make. Since this data is accumulated from a separate neural network that is trained prior to this system, any time-intensive computation will not get in the way of these lookups which are performed at runtime during self-play and therefore need to be done efficiently. As such, it holds true that the development of this ontology-driven neural network may take a significant amount of time, but insofar as it remains generalizable it need only be done once and can be used over and over again without the need for retraining.

Additionally, there is a less significant but still important challenge of calculating all likely permutations of the iterations of outcomes instead of just considering one move at a time. In other words, it is not permissible to create one child node where a vehicle B swerves left into another lane and another node where pedestrian A crosses the street, but not create a node that represents both of these actions occurring simultaneously. I see this as less significant of a challenge and merely a computational burden because to generate these states we need only generate every basic child node where only one action happens at a time, then cross the children with each of the outcomes represented by its siblings to create new nodes, as this represents a more complete sense of the space of all possible outcomes.

7. Towards an Evolutionary Approach

Using this evolutionary approach based off AlphaGo Zero achieves a few key advantages. Most importantly, much of the bias that other common machine learning approaches are plagued by is avoided. There is also an advantage in representing the moral landscape as pluralistic rather than absolute and unwavering, allowing for a nuanced and more accurate algorithm. For both mundane scenarios and "trolley cases," the system can handle both well without the risk of overfitting to the cases of high moral consequence.

However, there is a necessary piece of human feedback that is missing. Although primitives instill a sense of value such that simulation allows for this value to propagate and throughout the network, creators still need to make sure that the outcome is something that they actually want. In other words, they cannot simply trust that they picked a good set of primitive values, because those values could actually be catastrophic.

Luckily, outcomes can be simulated without having to perform trial-and-error testing in the real-world. When running tests of the system, attempts can be discarded if they are beyond recovery, meaning that a system has been trained for a while but has not converged closely to an acceptable set of values. More commonly, there might be apparent deviations from expected or desired behavior that are important but not catastrophic for the system as a whole. In these cases, overseers can provide a simple unit of yes or no feedback that will result in a no change/change operation respectively.

If a change should be made for the outcome chosen in a specific scenario, the system already has a good framework with which it can trace back and make an amendment. One of the most frequent heuristics for making a change upon

receiving negative feedback will simply be to choose the option the machine was second most certain about. Starting from the next highest likelihood to second most likely to third most likely and so on, an acceptable behavior will hopefully be found after only a few indications of dissatisfaction.

For cases where descending down the chain of the "next most likely" option does not yield a good result, this might serve as an indication that the primitives chosen for that network were flawed and need to be reconsidered.

With this methodology, an important parallel arises between this approach and the process through which humans are raised. By supplying a most basic, domain-specific primitive as innate, a machine may create much of its own knowledge, with limited explicit intervention from humans. This is a parallel because humans are provided with some innate machinery as well, for example, the capacity to feel pain. Although they have this ability from birth, it still takes significant trial and error on their own accord to get them to understand this simple value on a more abstract level. In other words, just because humans have a capacity to feel pain does not mean that they also innately know all of the actions that could lead them to feel pain -- that requires them to explore their world and learn on their own. In the same way, an algorithm that is provided with some primitive amount of innate machinery also will need to go through lots of iteration, or self-play in this case, to extrapolate higher-order value from a mere primitive.

8. Humans as Effective Judges

An adjudicator of sorts is needed for this approach to work, as there has to be an origin for the feedback on whether or not a change should be made after an outcome is suggested. This input is different from approaches where human data is a foundational component, such as supervised learning driven methodologies. In the latter case, the *only* guiding force is data which is used to predict human behaviors, whereas in the evolutionary approach, there is a great deal of structure that does not demand input from human intuition. The interference of a human judge is necessary, and is only to ensure alignment mostly retroactively as it takes place well into the training process.

The most effective judge will probably be multiple people, on a panel comprised of traffic and ethics experts as well as normal citizens from a diverse set of backgrounds. Importantly, compromise will be necessary for certain difficult cases as this is inevitable, but the point is not to get one perfect idea of moral behavior but rather to gain a sense of the variation amongst valid moral behavior. Allowing for this amount of openness is necessary in that it is the equivalent of having two different judges hear a case in traffic court, and ultimately come to decisions that may be slightly different. So long as both judges are fully competent and have all their faculties, their decision are likely both somewhat reasonable, and the resulting variation is therefore morally legitimate.

Throughout the process, various scenarios will be suggested and tested on the at least somewhat trained version of the algorithm. Then, the outcome will be observed as a behavior. In the easy case, the behavior is deemed unanimously reasonable, and the process will repeat for another scenario. In the case that there is disagreement, people can register their binary approval or disapproval, and the system will make a change to try and change the outcome. As alluded to earlier, one of the most common avenues for creating this kind of change will involve changing a behaviorally efficacious decision to the node with the next highest-ranked likelihood. If this succeeds, meaning that, it helps gain further approval from a greater number of the human spectators, then the change can be made permanent. Permanent here does not refer to a hard-coded or special case that ensures this exact behavior given this exact scenario, but rather a lasting structural change involving the shifting of either a parameter weight or the policy for that node if the parameter weight is sufficiently high.

The point of this process is to make a system socially viable, so that potential consumers can see transparently and realize that they have reason to trust, and maybe even buy, such a system.

9. Optimizations

The state space of this problem is admittedly infinite, but as mentioned before, it becomes enumerable with a separate heuristic for making good guesses as to logical behavior for relevant players as described in section VI. Starting self-play completely with random weight as is done in AlphaGo Zero, it would be unlikely that weights for common scenarios or sequences of actions would be discovered naturally without any guidance. Similarly, iterating blindly over an entire space of actions without any consideration of which actions are more probable is a mistake and probably a waste of time.

Instead, creators should provide explicit scenarios from the start, or develop a complementary system that is capable of generating its own scenarios to use for self-play as alluded to earlier. In the former case, there is of course an additional risk of collateral bias in that certain scenarios are selected over others. This may result in unintentionally overfitting to a certain kind of scenario that is not actually representative or useful. While generating scenarios itself would avoid this bias problem, it is idealistic as it requires a higher level of understanding as to which scenarios are interesting or meaningful and which are not. This paper does not cover how such a system would work.

10. Conclusion

A lot of work needs to be done in order to achieve this gold standard of autonomous driving software. Currently, approaches are not structured like this because the work necessary to eliminate harmful bias to this extent would mean delaying testing and rollouts of this technology, which the industry is not willing to do. Instead, they rely on deep supervised learning, or at best try to combine temporal representations of our values with some notions of utility across an action space.[6]

In the meantime, it is useful to consider an approach that involves less compromise. Such an approach, maybe counterintuitively, involves providing less information explicitly and trusting a system to learn on its own. I contend that this methodology is actually safest. Unreasonable behavior can be prevented by trialing diverse groups of scenarios, which ensures compatibility with our commonly held values, without misrepresenting those values in the form of harmful bias. Not only would a top-down approach create more room for human error, but it would also disallow any avenue for recourse should something go wrong. After all, such a system would follow rules that humans specified explicitly, so if a mistake is made then the rules provided by humans, and thus the humans themselves, are likely at fault.

The technology we create today will have a lasting impact on the kind of world we live in in the future, because it will serve as the foundation upon which newer systems are built. As such, considering these matters is completely worthwhile and necessary. When we are faced with a choice between a top-down and bottom-up approach, we should be inclined to choose the latter so long as we do not feel confident enough to explicitly provide moral truths.

When it comes to defining a better standard for autonomous driving, our goals are clear. These systems will raise the ethical bar by making decisions that are more aligned with our most fundamental ethical values than we could have ever specified manually. They will also work faster than humans, and find more efficient and optimal solutions than any human driver could. It is a combination of these exciting feats that should make us confident and more comfortable with the possibility of a self-driving future.

11. Acknowledgments

The author would like to thank Professor Clatterbuck for her thoughtful advising of this research and continued encouragement, and the University of Rochester for supporting students in pursuing multidisciplinary research.

12. References

1. Silver, David, et al. “Mastering the Game of Go without Human Knowledge.” Nature, vol. 550, no. 7676, 2017, pp. 354–359., doi:10.1038/nature24270.

2. Thrun, Sebastian, et al. Probabilistic Robotics. MIT Press, 2005.

3. Rini, Regina. “Raising Good Robots.” Aeon, 18 Apr. 2017, aeon.co/essays/creating-robots-capable-of-moral-reasoning-is-like-parenting.

4. Marcus, Gary. “Innateness, AlphaZero, and Artificial Intelligence.” [1801.05667] Innateness, AlphaZero, and Artificial Intelligence, 17 Jan. 2018, arxiv.org/abs/1801.05667.

5. Kasenberg, Daniel.“Norm Conflict Resolution in Stochastic Domains.” Tufts HRI Lab, Tufts University, Feb. 2018, hrilab.tufts.edu/publications/kasenbergscheutz18aaai.pdf.

6. Thomas Arnold, Daniel Kasenberg, and Matthias Scheutz. Value Alignment or Misalignment–What Will Keep Systems Accountable?, 2017. Presented at the 3rd International Workshop on AI, Ethics and Society at AAAI 2017.

7. Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911, 2017.

8. Katrakazas, Christos. "Developing an Advanced Collision Risk Model for Autonomous Vehicles." Loughborough University Institutional Repository, Loughborough University, 2017, dspace.lboro.ac.uk/2134/27538.

9. Yanagisawa, M., Swanson, E., Azeredo, P., & Najm, W. G. (2017, April). Estimation of potential safety benefits for pedestrian crash avoidance/mitigation systems. (Report No. DOT HS 812 400). Washington, DC: National Highway Traffic Safety Administration.

10. Danks, David. "Learning." The Cambridge Handbook of Artificial Intelligence, pp. 151–167., doi:10.1017/cbo9781139046855.011.

11. Wolpert, David H. and Macready, William G. 1997. 'No Free Lunch theorems for optimization', IEEE Transactions on Evolutionary Computation 1: 67-82.

12. Sajda, Paul. "Machine Learning For Detection And Diagnosis Of Disease." Annual Review of Biomedical Engineering, vol. 8, no. 1, 2006, pp. 537–565., doi:10.1146/annurev.bioeng.8.061505.095802.

13. Veale, Michael, and Reuben Binns. "Fairer Machine Learning in the Real World: Mitigating Discrimination without Collecting Sensitive Data." Big Data & Society, vol. 4, no. 2, 2017, p. 205395171774353., doi:10.1177/2053951717743530.

14. Metz, Cade. "In Two Moves, AlphaGo and Lee Sedol Redefined the Future." Wired, Conde Nast, 3 June 2017, www.wired.com/2016/03/two-moves-alphago-lee-sedol-redefined-future/.

15. Arnold, T. & Scheutz, M. Ethics Inf Technol (2018) 20: 59. https://doi.org/10.1007/s10676-018-9447-7

16. Danks, D., and London, A. J. 2017. Algorithmic bias in autonomous systems. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 4691– 4697.

17. Kant, Immanuel, and James W. Ellington. Grounding for the Metaphysics of Morals. Hackett Pub. Co., 1983.