December 2021

Key Concepts in AI Safety: Specification in Machine Learning

CSET Issue Brief

AUTHORS
Tim G. J. Rudner
Helen Toner

This paper is the fourth installment in a series on "AI safety," an area of machine learning research that aims to identify causes of unintended behavior in machine learning systems and develop tools to ensure these systems work safely and reliably. The first paper in the series, "Key Concepts in AI Safety: An Overview," outlined three categories of AI safety issues—problems of robustness, assurance, and specification—and the subsequent two papers described problems of robustness and assurance, respectively. This paper introduces specification as a key element in designing modern machine learning systems that operate as intended.

Introduction

Specification is the task of conveying to a machine learning system what exactly its designers would like it to do.[1] For some tasks—such as choosing which tiles in a CAPTCHA test contain a traffic light—it is relatively straightforward for the designer of such a system to write a precise description of what they are looking for. For many other tasks, however, it is difficult to capture the nuances of the intentions in precise, mathematical language.

In some ways, this type of challenge is not unique to machines. One prominent computer scientist describes specification problems in terms of familiar fictional analogues: As in the cases of King Midas, the Sorcerer's Apprentice, or the genie in the lamp, you get exactly what you wished for, which may not necessarily be what you wanted.[2] Principal–agent problems in economics deal with related situations, where a task is delegated from one person (the principal) to another (the agent), but the agent's understanding or incentives may diverge from the principal's. Specification problems in machine learning systems arise due to similar dynamics: In all but the simplest settings, it is challenging to convey and incentivize desired behaviors, which may in turn lead to undesired behaviors.

Ensuring that a given specification of a machine learning system results in a specific desired behavior and is in accordance with its designer's intentions is a key challenge for machine learning

research. To guard against failures, the goal is for machine learning systems to be robust to potential errors or inaccuracies in the specification. Getting this right will become increasingly critical as machine learning systems are deployed in higher-stakes and more complex settings. This brief provides an overview of specification problems for a policy audience, introducing key concepts, offering real-world examples, and suggesting implications for policymakers.

Specification & Specification Gaming

Machine learning systems are designed to learn patterns and associations from data. Typically, a machine learning method consists of a statistical model of the relationship between inputs and outputs, as well as a learning algorithm.[3] To enable machine learning methods to learn patterns and associations, a human operator can specify an *objective function* to be optimized.[4] The objective function is a core part of the learning algorithm, which specifies how the model should change as it receives more information (in the form of data) about the input–output relationship it is meant to represent. Objective functions can be thought of as expressing how good a model is at reaching a human-specified objective. Put another way, an objective function can be thought of as a mathematical expression that takes on large numerical values if the model performs poorly (corresponding to a high error rate) or small numerical values if the model performs well (corresponding to a low error rate).

In some machine learning settings, the objective function is defined as the "reward" obtained by performing a sequence of predictions. In this case, the goal becomes to maximize the reward instead of minimizing an error rate. Conceptually, the process of "training" then corresponds to gradually tweaking the model parameters (that is, the set of numbers that make up the statistical model) in order to minimize the objective function. The lower the numerical value returned by the objective function, the closer the model is to achieving the human-specified objective. Returning to the CAPTCHA example, a learning algorithm trying to find a statistical model that can predict whether tiles in a new CAPTCHA contain a given object might try to find a model that has high accuracy on a

set of previously generated CAPTCHAs—that is, a model that minimizes the probability of misidentifying whether a tile contains the object in question.

The specification of an objective function for a given task is crucial for finding a machine learning model that works well. *Specification gaming* is a particular failure mode in specification that can occur after an objective function has been specified by a human designer. It refers to a phenomenon where machine learning algorithms "game" whatever specification they were given, finding ways to achieve the specified objective with techniques that are totally disconnected from what the operator wanted. This behavior can look like cheats or workarounds.

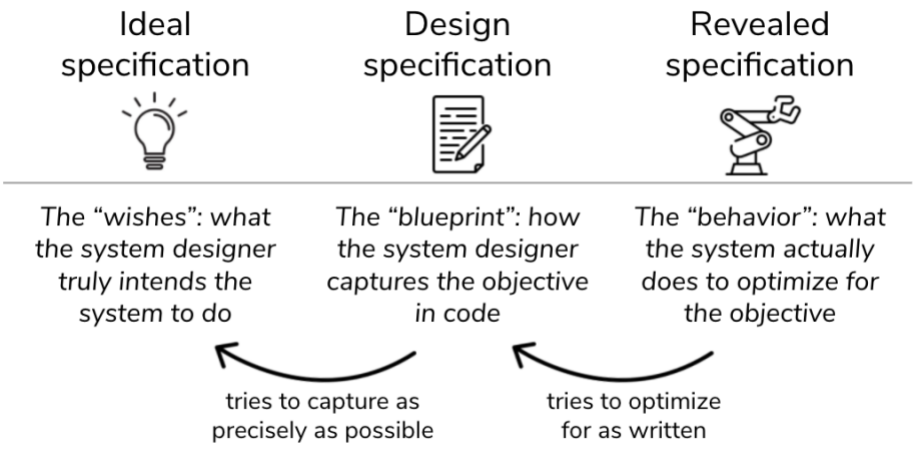
Specification gaming has been observed repeatedly in research settings. Examples include a boat in a racing video game learning that the best way to earn points is to loop endlessly in a harbor, repeatedly setting itself on fire rather than finishing the race;[5] a tic-tac-toe bot evolving to win games by making moves that crashed its opponent's software;[6] and simulated robots learning to exploit bugs in the simulation in order to move in physically impossible ways, such as by "wiggling" up walls or "surfing" on boxes.[7] This phenomenon is not specific to AI, and can arise in other settings as well. For example, a captive dolphin in Mississippi, upon learning it would be rewarded for bringing trash to its handler, was observed stowing trash in a corner of its habitat and tearing off small pieces to maximize the number of fish it could "earn."[8] Like humans and animals, machines respond to the incentives presented to them.

What makes specification in AI so challenging is that the typical design of machine learning algorithms forces system designers to write down an objective function that is only a simplified proxy of what they really want. The more complex the task, the harder it is to find a good proxy. Without a good proxy, there is a significant probability that the system will not operate as intended. Many of the tasks where machine learning has seen the greatest progress over the past decade are tasks where it is relatively easy to find good-enough proxies. In image classification, objective functions express the intention to "find a model that misclassifies the smallest number of images in a given set of training image–label

pairs" as a proxy for "find a model that classifies any image correctly."[9] In natural language processing, the last few years have shown that objective functions that express the intention to "find a model that predicts which word comes next in a text" can be a powerful, though far from perfect, proxy for "find a model that gives a sensible response to any text prompt."[10]

To illustrate what it means for a machine learning algorithm to be misspecified, consider three different levels of specification: (1) what task the designer of a machine learning system wants the system to perform; (2) the proxy objective the system designer specifies to enable the system to learn to perform the task; and (3) what the system actually does. The *ideal specification* of an objective function refers to the hypothetical description of an AI system's objective that is fully aligned with the human designer's desires—for example, wanting a robot to move a mug from one place to another, within a certain span of time, and without breaking the mug or breaking any other objects. The *design specification* of an objective function refers to the specification actually incorporated into the system—in other words, the proxy that the system designer chooses to implement. In the case of the mug example, this could be a mathematical expression that encourages the robot to keep a certain distance from other objects while incurring a penalty for performing the task too slowly or placing the mug in a wrong location. The design specification is typically an imperfect proxy of the ideal specification. Lastly, the *revealed specification* is the observed behavior when deploying a machine learning system in the real world. The revealed specification may make errors in the design specification apparent if the revealed behavior does not correspond to the desired behavior. For an illustration of the relationship between these different types of specification, see Figure 1 on page 5.

Figure 1. Three levels of specification



Source: CSET, with images from flaticon.com.

Unfortunately, the revealed specification typically differs from both the design and ideal specifications, and it is often difficult to find a design specification that leads to the intended outcomes—that is, for the revealed specification to match the ideal specification—which can lead to unintended and harmful consequences. There are several factors that can make it challenging to find a good design specification. First, complex environments or objectives may be difficult to describe in terms of simple mathematical equations that reflect desired behaviors. Second, it is difficult for machine learning systems to learn to reach even simple objectives if those objectives take time or multiple sequential decisions to be reached. Third, machine learning systems may potentially encounter decisions or situations that were not foreseen by a human designer of the system, especially in situations where systems operate at a scale outside of human experience, or involve subtle but important downstream consequences that are difficult to foresee when designing the system. To design reliably safe machine learning systems, it is important to address these challenges.

Specification Problems in Practice

The simplest forms of specification problems in machine learning systems—such as a simulated robot exploiting a bug in the simulator to achieve physically impossible feats—usually become

obvious well before the system is deployed in the real world. Catching and remedying these is a challenge for engineers, but is less relevant for high-level decision-makers.

For policymakers and others deciding when, where, and how to use machine learning in real-world settings, the more concerning types of specification problems are those that do not make themselves known in test settings: more subtle, pernicious, slow-moving misspecifications, which may only become apparent over long timescales or when the system is deployed at large scale. Such misspecifications can be especially harmful when machine learning systems are deployed in high-stakes environments.

An example of slower, more pernicious effects is how misspecification has already been implicated in harms caused by social media platforms. The business model of companies such as Facebook and YouTube uses machine learning systems to recommend content and keep users engaged on their apps. User engagement—as measured by time spent on the site, probability of clicking a link, or similar metrics—may seem like an innocuous enough objective for a machine learning model to optimize. In practice, however, it appears that disinformation or extremist content can often be highly engaging for certain subsets of users, meaning that a platform's machine learning model learns to serve this content in order to keep customers active.[11] This is an example of a divergence between the ideal specification—which would presumably be to maximize user engagement without radicalizing subsets of users—and the design and revealed specifications.

A resume screening tool developed by Amazon could be seen as another real-world example of misspecification.[12] The machine learning-based tool, which gave applicants' resumes a rating from one to five stars, was trained on resumes of people Amazon had hired in the past—meaning that rather than giving high ratings to excellent candidates, it was designed to optimize for giving high ratings to candidates similar to people Amazon had already hired. The result, as engineers working on the project discovered after several months, was that the model learned to mimic the gender disparity in Amazon's hiring—giving lower ratings to resumes with female-coded language, such as "women's" in "women's chess

club captain." While it is unclear to what extent the tool was used in practice, this is an example where the proxy goal selected when designing the system ("give high ratings to resumes similar to those of candidates Amazon hired") was different from the actual goal ("give high ratings to strong candidates") in a subtle but harmful way.

This type of subtle misspecification with long-term consequences can also occur in settings where humans, not machines, are making the decisions. For instance, a common phenomenon in large organizations is so-called "check-the-box" training, when senior leadership decides that workers should undergo training on a topic and requires all teams to receive the training. If the only requirement is that a training is held, or that it meet some simple criteria, then the training is likely to become a box-checking exercise rather than a meaningful educational experience. In other words, if leadership's ideal specification of "increase employees' understanding of the topic" is translated into a design specification of "oblige employees to attend a training once a year," then the revealed specification is likely to be "employees sit through a session that they mostly ignore." In the best case, this wastes time; in the worst case, inadequate training can increase the risk of accidents or other harms.

It is worth noting that a system with a well-specified objective can still cause harm. For instance, the system's designer may have malicious intentions (or may simply be indifferent to potential harms) or the system could fail in some way that is not related to specification.[13] Achieving close alignment between the ideal specification and the revealed specification of a system is necessary but not sufficient for the development of responsible and trustworthy AI.

Avoiding Misspecification

Finding ways to convey nuanced and complex objectives to machine learning systems is an active area of research. Different approaches to avoiding misspecification tackle different types of specification challenges. Below, we describe three approaches that aim to create specifications that contain more nuance and

complexity than traditional approaches: learning from demonstrations, learning from human feedback, and inverse reward design. Each of these has unique strengths and limitations, and none provides a complete and easily usable solution to the challenges of specification.

One well-established machine learning paradigm that circumvents specification problems is learning from demonstrations. Broadly speaking, this paradigm revolves around enabling machine learning systems to learn from the actions of humans. For example, a humanoid robot could watch a human trainer complete a household chore in order to learn how to perform the chore itself, or an autonomous vehicle could use data on how humans drive to imitate that behavior.

Imitative approaches can work fairly well in cases where it is straightforward to have a human demonstrate the desired behavior, such as in driving, carrying out tasks for a humanoid robot, and so on. This approach is inherently limited, however, by what humans are able to demonstrate. In many cases, the intention is for machine learning systems to operate in ways that humans cannot—whether due to speed, complexity, scale, safety, or other factors. In these situations, the designer of a machine learning system needs a different way to convey what they want the system to do.

Several research directions aim to achieve this goal. Two prominent approaches are learning from human feedback and inverse reward design. Each provides a framework for how humans can work with machine learning systems during training in order to provide more nuanced feedback than commonly used objective functions, while also not being limited by what humans can directly demonstrate.

Learning from human feedback uses direct feedback from human labelers to learn a reward signal, so the AI system tries to learn to predict what the labelers will prefer rather than over-fixating on a specified objective function. For example, in one experiment in learning from human feedback, a simulated robot learned to do a backflip based on receiving hundreds of pieces of feedback from

human labelers, who would compare two videos of the robot and choose which looked more like a backflip.[14] Other early work includes using human ratings to teach an AI system how to write accurate and useful summaries of text, a task for which it is difficult to specify a simple evaluation function.[15]

Another approach, called inverse reward design, is structured differently. The key insight in this approach is that it is possible to design machine learning systems to treat their objective function not as the absolute truth of what to aim for—as in the standard machine learning approach—but instead as just one piece of evidence about what is "good." This seemingly minor structural change has been shown to produce more risk-averse and correction-seeking behavior, as the system's built-in uncertainty about what it should really be doing prevents it from acting confidently in unfamiliar situations.[16]

Unfortunately, existing methods for addressing potential specification challenges fall short in two significant ways. Firstly, approaches developed to date are either difficult to implement in practice or too costly for a wide range of machine learning systems or application settings. Secondly, even approaches such as the two outlined above do not provide guarantees that no specification issues will occur, and so far no widely applicable evaluation protocol or theoretical tools exist to ensure that a system's revealed specification will not diverge from its ideal specification.

Outlook

At present, the vast majority of machine learning systems in use perform narrow, well-defined tasks, such as recommending products to customers or detecting credit card fraud. Machine learning models are also often deployed in tandem with explicitly defined rules to prevent undesired behavior. As long as machine learning is used in simple, circumscribed applications, specification problems are unlikely to cause major harm: Where they occur, they will often be detected quickly, and where they go undetected, the damage is likely to be limited.

As machine learning systems become more advanced, they will likely be deployed in increasingly complex environments to carry out increasingly complex tasks. This is where specification problems may begin to bite. Without significant progress in methods to convey intentions, machine learning systems will continue to carry out their instructions exactly as given—obeying the letter, not the spirit, of the rules their designer gives them.

To address the challenges posed by misspecification, more machine learning research needs to account for worst case scenarios and develop algorithms that more explicitly incorporate human supervision or provide theoretical guarantees for the worst case performance under a given specification.

In the meantime, policymakers grappling with the impacts of artificial intelligence would do well to keep specification challenges front of mind. For any potential use of machine learning, one can ask two questions: What objective has been specified for the system, and what kinds of perverse behavior could be incentivized by optimizing for that objective? Until significant progress has been made in research on how to convey nuanced, complex objectives and ensure systems will reliably work towards them, machine learning will only be suited to narrow, tightly prescribed settings.

Authors

Tim G. J. Rudner is a non-resident AI/ML fellow with CSET and a PhD candidate in computer science at the University of Oxford. Helen Toner is director of strategy at CSET.

Acknowledgments

For feedback and assistance, we would like to thank Corey Cooper, Rita Konaev, Igor Mikolic-Torreira, Larry Lewis, Vishal Maini, Paul Scharre, Adrienne Thompson, and Lynne Weil.

Document Identifier: doi: 10.51593/20210031

Endnotes

[1] "Specification" can also be used to refer to other parts of the process of designing and training a machine learning system, such as model specification, but these are not included for the purposes of this paper.

[2] Stuart Russell, "Of Myths and Moonshine," *Edge*, November 2014, https://www.edge.org/conversation/the-myth-of-ai#26015.

[3] While not all machine learning methods fall into this category, for simplicity of exposition we only consider machine learning methods that seek to learn input–output relationships.

[4] The terms "objective function," "cost function," "loss function," and "reward function" are all roughly equivalent, though each is used in slightly different settings. In reinforcement learning, for example, "reward function" is generally used. Reward functions are generally designed to be maximized, while each of the others is designed to be minimized.

[5] Dario Amodei and Jack Clark, "Faulty Reward Functions in the Wild," *OpenAI*, December 21, 2016, https://openai.com/blog/faulty-reward-functions/.

[6] Joel Lehman et al., "The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities," *Artificial Life* 26, no. 2 (Spring 2020): https://direct.mit.edu/artl/article/26/2/274/93255/The-Surprising-Creativity-of-Digital-Evolution-A.

[7] Lehman et al., "The Surprising Creativity of Digital Evolution," https://arxiv.org/pdf/1803.03453.pdf; Bowen Baker et al., "Emergent Tool Use from Multi-Agent Interaction," *OpenAI*, September 17, 2019, https://openai.com/blog/emergent-tool-use/#surprisingbehaviors; see Victoria Krakovna, "Specification gaming examples in AI," personal blog, April 2, 2018, https://vkrakovna.wordpress.com/2018/04/02/specification-gaming-examples-in-ai/ for more examples.

[8] Anuschka de Rohan, "Why Dolphins are Deep Thinkers," *The Guardian*, July 2, 2003, https://www.theguardian.com/science/2003/jul/03/research.science.

[9] Most methods actually use an objective function called "cross-entropy loss," which is similar to a model's accuracy, but has preferable mathematical properties when used with certain types of optimization routines.

[10] Tom Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems 33* (*NeurIPS 2020*),

https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html; note that even in this seemingly simple case, the difference between the proxy (predict the most likely next word based on a corpus of online text) and the desired behavior (generate useful or convincing text) can cause problems, for example if the system predicts that racist language is likely to occur after words relating to a marginalized racial group.

[11] See, for example, Max Fisher and Amanda Taub, "How Everyday Social Media Users Become Real-World Extremists," *The New York Times*, April 25, 2018, https://www.nytimes.com/2018/04/25/world/asia/facebook-extremism.html and Casey Newton, "How Extremism Came to Thrive on YouTube," *The Verge*, April 3, 2019, https://www.theverge.com/interface/2019/4/3/18293293/youtube-extremism-criticism-bloomberg. Social media companies have changed how their algorithms incorporate engagement metrics into their objective functions over time, in part to reduce these kinds of problems. However, it is not clear whether these changes have solved the problem. See, for instance, Eric Meyerson, "YouTube Now: Why We Focus on Watch Time," *YouTube Official Blog*, August 10, 2012, https://blog.youtube/news-and-events/youtube-now-why-we-focus-on-watch-time/ and Adam Mosseri, "Bringing People Closer Together," *Facebook*, January 11, 2018, https://about.fb.com/news/2018/01/news-feed-fyi-bringing-people-closer-together/.

[12] Jeffrey Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," Reuters, October 10, 2018, https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

[13] See other primers in this series for examples of such failures.

[14] Paul F. Christiano et al., "Deep Reinforcement Learning from Human Preferences," in *Advances in Neural Information Processing Systems 30* (*NIPS 2017*), https://papers.nips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

[15] Nisan Stiennon et al., "Learning to Summarize with Human Feedback," in *Advances in Neural Information Processing Systems 33* (*NeurIPS 2020*), https://papers.nips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

[16] Dylan Hadfield-Menell et al., "Inverse Reward Design," *31st Conference on Neural Information Processing Systems* (*NIPS 2017*), https://papers.nips.cc/paper/2017/file/32fdab6559cdfa4f167f8c31b9199643-Paper.pdf.