CHAPTER 5

Complex Systems

5.1 OVERVIEW

Artificial intelligence systems and the societies they operate within belong to the class of *complex systems*. These types of systems have significant implications for thinking about and ensuring AI safety. Complex systems exhibit surprising behaviors and defy conventional analysis methods that examine individual components in isolation. To develop effective strategies for AI safety, it is crucial to adopt holistic approaches that account for the unique properties of complex systems and enable us to anticipate and address AI risks.

This chapter begins by elucidating the qualitative differences between complex and simple systems. After describing standard analysis techniques based on mechanistic or statistical approaches, the chapter demonstrates their limitations in capturing the essential characteristics of complex systems, and provides a concise definition of complexity. The "Hallmarks of Complex Systems" section then explores seven indications of complexity and establishes how deep learning models exemplify each of them.

Next, the "Social Systems as Complex Systems" section shows how various human organizations also satisfy our definition of complex systems. In particular, the section explores how the hallmarks of complexity materialize in two examples of social systems that are pertinent to AI safety: the corporations and research institutes pursuing AI development, and the decision-making structures responsible for implementing policies and regulations. In the latter case, there is consideration of how advocacy efforts are affected by the complex nature of political systems and the broader social context.

Having established that deep learning systems and the social systems surrounding them are best described as complex systems, the chapter moves on to what this means for AI safety. The "General Lessons" section derives five learnings from the chapter's examination of complex systems and sets out their implications for how risks might arise from AI. The "Puzzles, Problems, and Wicked Problems" section then reframes the contrasts between simple and complex systems in terms of the different kinds of

problems that the two categories present, and the distinct styles of problem-solving they require.

By examining the unintended side effects that often arise from interfering with complex systems, the "Challenges with Interventionism" section illustrates the necessity of developing comprehensive approaches to mitigating AI risks. Finally, the "Systemic Issues" section outlines a method for thinking holistically and identifying more effective, system-level solutions that address broad systemic issues, rather than merely applying short-term "quick fixes" that superficially address symptoms of problems.

5.2 INTRODUCTION TO COMPLEX SYSTEMS

5.2.1 The Reductionist Paradigm

Before we describe complex systems, we will first look at non-complex systems and the methods of analysis that can be used to understand them. This discussion sits under the *reductionist paradigm*. According to this paradigm, systems are just the sum of their parts, and can be fully understood and described with relatively simple mathematical equations or logical relations.

The mechanistic approach analyzes a system by studying each component separately. A common technique for understanding a system is to identify its components, study each one separately, and then mentally "reassemble" it. Once we know what each part does, we can try to place them all in a simple mechanism, where one acts on another in a traceable sequence of steps, like cogs and wheels. This style of analysis is called the *mechanistic approach*, which often assumes that a system is like a line of dominos or a Rube Goldberg machine; if we set one component in motion, we can accurately predict the linear sequence of events it will trigger and, thus, the end result.

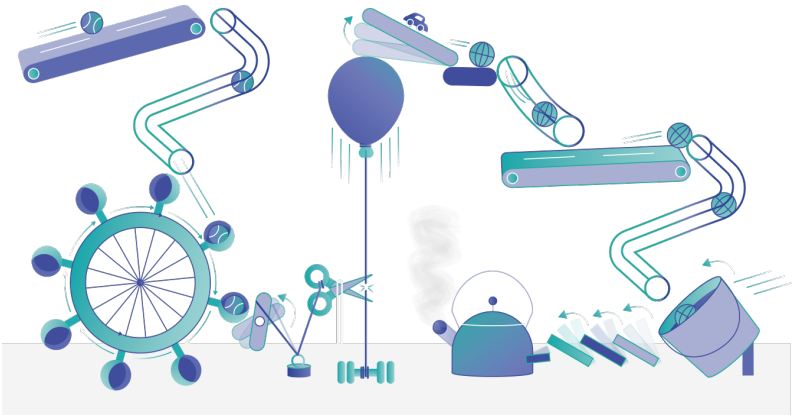

FIGURE 5.1. A Rube Goldberg machine with many parts that each feed directly into the next can be well explained by way of mechanisms.

Many human artifacts can be understood mechanistically. Devices like bicycles, clocks, and sewing machines are designed with specific mechanisms in mind, where one component directly acts on another in a cause-and-effect way to perform an intended function. For example, we can look at a bicycle and understand that turning the pedals will pull on a chain, which will turn the wheels, which will move the bicycle forward.

We can often derive mathematical equations that govern mechanistic systems. If we can successfully model a system's behavior mechanistically, then we can usually find mathematical equations that describe its behavior. We can use these equations to calculate how the system will respond to different inputs. With this knowledge, we can control what the system does by controlling the inputs. For example, if we know how quickly the pedals on a bicycle are rotating then we can calculate the speed at which it is traveling. Conversely, we can control the bicycle's speed by controlling how quickly the pedals rotate.

Many conventional computer programs can also be understood mechanistically. Simple algorithmic computer programs involving for-loops and "if... else..." constructions can be understood in this way too. Given any input, we can trace through the program's operations to predict the output. Similarly, for any given output, we can trace the steps backward and deduce information about the input.

Functions in computer programs can also be understood mechanistically. We can create functions within programs and give them names that are readable and intuitive to humans. For instance, we can name a function "add(x, y)" and define it to return the sum of x and y. We can then write a computer program using various functions like this, and we can analyze it by understanding how each function works on its own and then looking at the sequence of functions the program follows. This enables us to predict reliably what output the program will give for any input.

If there are a large number of components, we can sometimes use statistics. Suppose we are trying to predict the behavior of a gas in a box, which contains on the order of 10^{23} particles (that is, 1 followed by 23 zeros). We clearly cannot follow each one and keep track of its effects on the others, as if it were a giant mechanism.

However, in the case of a system like a gas in a box, the broader system properties of pressure and temperature can be related to averages over the particle motions. This allows us to use statistical descriptions to derive simple equations governing the gas's coarse-grained behavior at the macroscopic level. For example, we can derive an equation to calculate how much the gas pressure will increase for a given rise in temperature.

The mechanistic and statistical approaches fall within the reductionist paradigm. Both mechanistic and statistical styles of analysis seek to understand and describe systems as combinations or collections of well-understood components. Under the mechanistic approach, we account for interactions by placing the components in a mechanism, assuming they only affect one another in a neat series of direct

one-to-one interactions. Under the statistical approach, we assume that we do not need to know the precise details of how each interaction plays out because we can simply take an average of them to calculate the overall outcome.

Summary. Reductionist styles of analysis assume that a system is no more than the sum of its parts. For a reductionist analysis to work, one of the following assumptions should often apply: There either needs to be a simple, traceable mechanism governing the system's behavior, or we need to be able to relate the broader system properties to statistical averages over the components.

Limitations of the Reductionist Paradigm

Having discussed simple systems and how they can be understood through reductionism, we will now look at the limitations of this paradigm and the types of systems that it cannot be usefully applied to. We will look at the problems this presents for understanding systems and predicting their behaviors.

Many real-world systems defy reductionist explanation. Imagine that, instead of looking at a bicycle or a gas in a box, we are trying to understand and predict the behavior of an ecosystem, weather patterns, or a human society. In these cases, there are clearly far too many components for us to keep track of what each one is doing individually, meaning that we cannot apply the mechanistic approach. Additionally, there are also many complex interdependencies between the components, such that any given component might behave differently in the context of the system than it does in isolation. We cannot, therefore, use statistics to treat the system's behavior as a simple aggregate of the components' individual behaviors.

In complex systems, the whole is more than the sum of its parts. The problem is that reductionist-style analysis is poorly suited to capturing the diversity of interdependencies within complex systems. Reductionism only works well if the interactions follow a rigid and predictable mechanism or if they are random and independent enough to be modeled by statistics. In complex systems, neither of these assumptions hold.

In complex systems, interactions do not follow a rigid, structured pattern, but components are still sufficiently interconnected that they cannot be treated as independent. These interactions are the source of many novel behaviors that make complex systems interesting. To get a better grasp of these systems, we need to go beyond reductionism and adopt an alternative, more holistic framework for thinking about them.

We can sometimes predict general short-term trends in complex systems. Note that we may be able to predict high-level patterns of behavior in some complex systems, particularly if we are familiar with them and have many observations of their past behavior. For example, we can predict with a high degree of confidence that, in the northern hemisphere, a day in January next year will be colder than a day in June. However, it is much more difficult to predict specific details, such as the exact temperature or whether it will rain on a given day. It is also much more challenging

to predict the longer-term trajectory of the system, such as what the climate will look like in several centuries or millennia. This is because complex systems often develop in a more open-ended way than simple systems and have the potential to evolve into a wider range of states, with numerous factors influencing the path they take.

New or unfamiliar complex systems are even more difficult to predict. The challenges in predicting how complex systems will behave are compounded when we face newly emerging ones, such as those involving AI. While we have plenty of historical information and experience to help us predict weather patterns, we have little past data to inform us on how AI systems and their use in society will develop. Nevertheless, studying other complex systems and paying attention to their shared properties can give us insights into how AI might evolve. This might offer clues as to how we can avoid potential negative consequences of using AI.

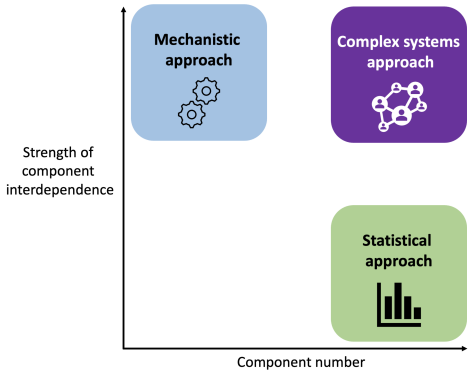

FIGURE 5.2. Often, we use mechanistic or statistical approaches to analyzing systems. When there are many components with strong interdependence, these are insufficient, and we need a complex systems approach.

Summary. Reductionist styles of analysis cannot give us a full understanding of complex systems, whose components neither function like a deterministic mechanism nor behave randomly and independently enough to use statistics—shown in Figure 5.2. This lack of a full understanding presents challenges for predicting the system's behavior on two levels: which component will perform which action at what time, and how the whole system might change over the long term.

5.2.2 The Complex Systems Paradigm

Now that we have seen that many systems of interest are inscrutable to the reductionist paradigm, we need an alternative lens through which to understand them. To this end, we will discuss the *complex systems paradigm*, which takes a more holistic view, placing emphasis on the most salient features shared across various real-world complex systems that the reductionist paradigm fails to capture. The benefit of this paradigm is that it provides "a way of seeing and talking about reality that helps us better understand and work with systems to influence the quality of our lives."

Complex systems exhibit emergent properties that are not found in their components. As discussed above, some systems cannot be usefully understood in a reductionist way. Studying a complex system's components in isolation and doing mental reassembly does not amount to what we observe in reality. One primary reason for this is the phenomenon of *emergence*: the appearance of striking, system-wide features that cannot be found in any of the system's components.

The presence of emergent features provides one sense in which complex systems are "more than the sum of their parts." For example, we do not find atmospheric currents in any of the molecules of nitrogen and oxygen that make up the atmosphere, and the flexible intelligence of a human being does not exist in any single neuron. Many biological concepts such as adaptation, ecological niche, sexuality, and fitness are not simply reduced to statements about molecules. Moreover, "wetness" is not found in individual water molecules. Emergence is so essential that we will use it to construct a working definition of complex systems.

Working definition. Complex systems are systems of many interconnected components that collectively exhibit emergent features, which cannot, in practice, be derived from a reductive analysis of the system in terms of its isolated components.

Ant colonies are a classic example of a complex system. An ant colony can grow to a size of several million individuals. Each ant is a fairly simple creature with a short memory, moving around in response to chemical and tactile cues. The individuals interact by randomly bumping into each other and exchanging pheromones. Out of this mess of uncoordinated interactions emerge many fascinating collective behaviors. These include identifying and selecting high-quality food sources or nest sites, forming ant trails, and even constructing bridges over gaps in these trails (formed by the stringing together of hundreds of the ants' bodies). Ant colonies have also been observed to "remember" the locations of food sources or the paths of previous trails for months, years, or decades, even though the memory of any individual ant only lasts for a few days at most.

Ant colonies satisfy both aspects of the working definition of complex systems. First, the emergent features of the colony include the collective decision-making process that enables it to choose a food source or nest site, the physical ability to cross over gaps many times wider than any ant, and even capabilities of a cognitive nature such as extended memory. We could not predict all of these behaviors and abilities from observing any individual ant, even if each ant displays some smaller analogs of some of these abilities.

Second, these emergent features cannot be derived from a reductive analysis of the system focused on the properties of the components. Even given a highly detailed study of the behavior of an individual ant considered in isolation, we could not derive the emergence of all of these remarkable features. Nor are all of these features simple statistical aggregates of individual ant behaviors in any practical sense, although some features like the distribution of ants between tasks such as foraging, nest maintenance, and patrolling have been observed as decisions on the level of an individual ant as well.

This distinguishes a more complex system like an ant colony from a simpler one such as a gas in a box. Although the gas also has emergent properties (like its temperature and pressure), it does not qualify as complex. The gas's higher-level properties can be straightforwardly reduced to the statistics of the lower-level properties of the component particles. However, this was not always the case: it took many decades of work to uncover the statistical mechanics of gases from the properties of individual molecules. Complexity can be a feature of our understanding of the system rather than the system itself.

Complex systems are ubiquitous in nature and society. From cells, organisms, and ecosystems, to weather systems, cities, and the World Wide Web, complex systems are everywhere. We will now describe two further examples, referred to throughout this chapter.

Economies are complex systems. The components of an economic system are the individual persons, companies, and firms participating in the economy. These economic agents interact via various kinds of financial transactions, such as lending, borrowing, investing, and purchasing and selling goods. Out of these interactions emerge complex economic phenomena such as inflation, stock-market indexes, and interest rates. These economic phenomena are not manifested by any individual agent and cannot be derived by studying the behavior of these agents considered separately; rather, they arise from the complex network of interactions between them.

The human brain is a complex system. The human brain consists of around 86 billion neurons, each one having, on average, thousands of connections to the others. They interact via chemical and electrical signals. Out of this emerge all our impressive cognitive abilities, including our ability to use language, perceive the world around us, and control the movements of our body. Again, these cognitive abilities are not found in any individual neuron, arising primarily from the rich structure of neuronal connections; even if we understood individual neurons very well, this would not amount to an understanding of (or enable a derivation of) all these impressive feats accomplished by the brain.

Interactions matter for complex systems. As these examples illustrate, the interesting emergent features of complex systems are a product of the interactions (or interconnections) between their components. This is the core reason why these systems are not amenable to a reductive analysis, which tries to gain insight by breaking the system into its parts. As the philosopher Paul Cilliers writes: "In 'cutting up' a system, the analytic method destroys what it seeks to understand" [280].

Summary. Complex systems are characterized by emergent features that arise from the complex interactions between components, but do not exist in any of the individual components, and cannot be understood through or derived from a reductive analysis of them. Complex systems are ubiquitous, from ant colonies to economies to the human brain.

5.2.3 Deep Learning Systems as Complex Systems

An essential claim of this chapter is that deep learning models are complex systems. Here, we will briefly discuss what a reductionist approach to understanding deep learning systems would look like and why it is inadequate.

Consider a deep learning system that correctly classifies an image of a cat. How does it do this? The reductionist approach to this question would first try to break down the classification into a sequence of smaller steps and then find parts of the neural network responsible for executing each of them. For instance, we might decompose the problem into the identification of cat ears + whiskers + paws and then look for individual neurons (or small clusters of neurons) responsible for each of these elements.

The reductionist approach cannot fully describe neural networks. In some cases, it seems possible to find parts of a neural network responsible for different elements of such a task. Researchers have discovered that progressively later layers of deep neural networks are generally involved in recognizing progressively higher-level features of the images they have been trained to classify. For example, close to the input layer, the neural network might be doing simple edge detection; a little further into the hidden layers, it might be identifying different shapes; and close to the output, it might be combining these shapes into composites.

However, there is no clear association between an individual node in a given layer and a particular feature at the corresponding level of complexity. Instead, all the nodes in a given layer are partially involved in detecting any given feature at that level. That is to say, we cannot neatly attribute the detection of each feature to a specific node, and treat the output as the sum of all the nodes detecting their specific features. Although there have been instances of researchers identifying components of neural networks that are responsible for certain tasks, there have been few successes, and they have required huge efforts to achieve. In general, this approach has not so far worked well for explaining higher-level behaviors.

The complex systems paradigm is more helpful for deep learning systems. As these problems suggest, we cannot generally expect to find a simple, human-interpretable set of features that a neural network identifies in each example and "adds together" to reach its predictions. Deep learning systems are too complex to reduce to the behavior of a few well-understood parts; consequently, the reductionist paradigm is of limited use in helping us think about them. As we will discuss later in this chapter, the complex systems paradigm cannot entirely make up for this or enable a complete understanding of these systems. Nonetheless, it does give us a vocabulary for thinking about them that captures more of their complexity and can teach us some general lessons about interacting with them and avoiding hazards.

Summary. The difficulties involved in explaining neural networks' activity through simple mechanisms are one piece of evidence that they are best understood as complex systems. We will substantiate this claim throughout the next section, where we run

through some of the hallmark features of complex systems and discuss how they apply to deep learning models.

5.2.4 Complexity is Not a Dichotomy

In the previous section, we proposed a working definition of complex systems that suffices for an informal discussion, though it is not completely precise. In fact, there is no standard definition of complexity used by all complex-systems scientists. In part, this is because complexity is not a dichotomy.

Understanding system complexity. While we have described a distinction between a "simple" and "complex" system, labeling a system as inherently simple or complex can be misleading. Complexity is not always intrinsic to a system. Instead, it depends on our understanding. Certain phenomena in physics, for instance, have transitioned from being poorly understood "complex" concepts to well-explained "simple" mechanics through advanced analysis of the properties of the system. Superconductivity—the property of a material to conduct electricity without resistance when cooled below a certain critical temperature—is an example of this transition in understanding.

Superconductivity was originally perceived as a complex phenomenon due to the emergent behavior arising from electron interactions in metals. However, with the discovery of the Bardeen-Cooper-Schrieffer (BCS) theory, it became clear that superconductivity could be explained through the pairing of electrons. By considering these pairs as the components of interest rather than individual electrons, superconductivity was reclassified as a conceptually "simple" system that can be described by reductionist models.

Complexity, information, and reductionism. Current research in complex systems acknowledges the importance of interactions in determining emergent behavior but doesn't abandon the search for mechanistic explanations. Often, mechanistic explanations of systems can be found when considering a larger basic basic building block, such as pairs of electrons for superconductivity. This choice of scale is important for creating effective models of possibly complex phenomena.

Thus, rather than a binary classification, systems might be better understood as existing on a spectrum based on the scale and amount of information required to predict their behavior accurately. Complex systems are those that, at a certain scale, require a vast amount of information for prediction, indicating their relative incompressibility. However, they could still be explained mechanistically, if we understood them sufficiently well.

5.2.5 The Hallmarks of Complex Systems

Since complexity is not a dichotomy, it is difficult to pin down when exactly we can consider systems complex. In place of a precisely demarcated domain, complex-systems scientists study numerous salient features that are generally shared by the

systems of interest. While disciplines like physics seek fundamental mechanisms that can explain observations, the study of complex systems looks for salient higher-level patterns that appear across a wide variety of systems.

We consider seven key characteristics of complex systems. Chief among these is emergence, but several others also receive attention: self-organization, feedback and nonlinearity, criticality, adaptive behavior, distributed functionality, and scalable structure. We will now describe each of these hallmarks and explain their implications. Along the way, we will show that deep learning systems share many similarities with other complex systems, strengthening the case for treating them under this paradigm.

Emergence

We have already discussed emergence, the appearance of striking system-wide features that cannot be found in any of the components of the system. Ant colonies swarm over prey and build bridges over gaps in their trail; economies set prices and can crash; human brains think, feel, and sense. These remarkable behaviors are inconceivable for any individual component—ant, dollar, or neuron—existing in isolation.

Emergent features often spontaneously "turn on" as we scale up the system in size. A group of 100 army ants placed on the ground behaves not like an enfeebled colony but rather like no colony at all; the ants just walk around in circles until they starve or die of exhaustion. If the system is scaled up to tens of thousands of ants, however, a qualitative shift in behavior occurs as the colony starts behaving like an intelligent superorganism.

Emergent abilities have been observed in deep learning systems. Large language models (LLMs) are trained to predict the next token in a string of words. Smaller LLMs display a variable ability to output coherent sentences, as might be expected based on this training. Larger LLMs, however, spontaneously gain qualitatively new capabilities, such as translating text or performing three-digit arithmetic. These abilities can emerge without any task-specific training.

Summary. Emergent properties arise collectively from interactions between components, and are a defining feature of complex systems. These features often appear spontaneously as a system is scaled up. Emergent capabilities have already been observed in deep learning systems.

Feedback and Nonlinearity

Two closely related hallmarks of complexity are *feedback* and *nonlinearity*. Feedback refers to circular processes in which a system and its environment affect one another. There are multiple types of nonlinearity, but the term generally describes systems and processes where a change in the input does not necessarily translate to a proportional change in the output. We will now discuss some mechanisms behind nonlinearity, including feedback loops, some examples of this phenomenon, and why it makes complex systems' behavior less predictable.

In mathematics, a linear function is one whose outputs change in proportion to changes in the inputs. The functions $f(x) = 3x$ and $f(x) = 100(x - 10)$ linear. Meanwhile, the functions $f(x) = x^2$ and $f(x) = e^x$ are nonlinear.

Complex systems are nonlinear functions of their inputs. Complex systems process inputs in a nonlinear way. For example, when ant colonies are confronted with two food sources of differing quality, they will often determine which source is of higher quality and then send a disproportionately large fraction of its foragers over to exploit it rather than form two trails in proportion to the quality of the food source. Neural networks are also nonlinear functions of their inputs. This is why adversarial attacks can work well: adding a small amount of noise to an image of a cat need not merely reduce the classifier's confidence in its prediction, but might instead cause the network to confidently misclassify the image entirely.

Nonlinearity makes neural networks hard to decompose. A deep neural network with 10 layers cannot be replaced by five neural networks, each with only two layers. This is due to the nonlinear activation functions (such as GELUs) between their nodes. If the layers in a neural network simply performed a sequence of linear operations, the whole network could be reduced to a single linear operation. However, nonlinear operations cannot be reduced in the same way, so nonlinear activation functions mean that deep neural networks cannot be collapsed to networks with only a few layers. This property makes neural networks more capable, but also more difficult to analyze and understand.

Feedback loops in complex systems

A major source of nonlinearity is the presence of feedback. Feedback occurs when the interdependencies between different parts of a system form loops (e.g., A depends on B, which in turn depends on A). These feedback loops can reinforce certain processes in the system (positive feedback), and quash others (negative feedback), leading to a nonlinear relationship between the system's current state and how it changes. The following are examples of feedback loops in complex systems.

The rich get richer. Wealthy people have more money to invest, which brings them a greater return on investment. In a single investment cycle, the return on investment is greater in proportion to their greater wealth: a linear relationship. However, this greater return can then be reinvested. Doing so forms a positive feedback loop through which a slight initial advantage in wealth can be transformed into a much larger one, leading to a nonlinear relationship between a person's wealth and their ability to make more money.

Learning in the brain involves a positive feedback loop. Connections between neurons are strengthened according to Hebb's law ("neurons that fire together, wire together"). Stronger connections increase the probability of subsequent episodes of "firing together", further strengthening those connections. As a result of this feedback process, our memories do not strengthen or weaken linearly with time. The most efficient way to learn something is by revisiting it after increasing intervals of intervening time, a method called "spaced repetition."

Task distribution in beehives can be regulated by feedback loops. When a forager bee finds a source of water, it performs a "waggle dance" in front of the hive to signal to the other bees the direction and distance of the source. However, a returning forager needs to find a receiver bee onto which to unload the water. If too many foragers have brought back water, it will take longer to find a receiver, and the forager is less likely to signal to the others where they should fly to find the source. This negative feedback process stabilizes the number of bees going out for water, leading to a nonlinear relationship between the number of bees currently flying out for water and the number of additional bees recruited to the task.

AI systems involve feedback loops. In a system where agents can affect the environment, but the environment can also affect agents, the result is a continual, circular process of change—a feedback loop. Another example of feedback loops involving AIs is the reinforcement-learning technique of self-play, where agents play against themselves: the better an agent's performance, the more it has to improve to compete with itself, leading its performance to increase even more.

Feedback processes can make complex systems' behavior difficult to predict. Positive feedback loops can amplify small changes in a system's initial conditions into considerable changes in its resulting behavior. This means that nonlinear systems often have regimes in which they display extreme sensitivity to initial conditions, a phenomenon called chaos (colloquially referred to as the *butterfly effect*). A famous example of this is the logistic map, an equation that models how the population of a species changes over time:

$$x_{n+1} = rx_n(1 - x_n).$$

This equation is formulated to capture the feedback loops that affect how the population of a species changes: when the population is low, food sources proliferate, enabling the population to grow; when it is high, overcrowding and food scarcity drive the population down again. x_n is the current population of a species as a fraction of the maximum possible population that its environment can support. x_{n+1} represents the fractional population at some time later. The term r is the rate at which the population increases if it is not bounded by limited resources. When the parameter r takes a value above a certain threshold (~ 3.57), we enter the chaotic regime of this model, in which a tiny difference in the initial population makes for a large difference in the long-run trajectory. Since we can never know a system's initial conditions with perfect accuracy, chaotic systems are generally considered difficult to predict.

AIs as a self-reinforcing feedback loop. Since AIs can process information and reach decisions more quickly than humans, putting them in charge of certain decisions and operations could accelerate developments to a pace that humans cannot keep up with. Even more AIs may then be required to make related decisions and run adjacent operations. Additionally, if society encounters any problems with AI-run operations, it may be that AIs alone can work at the speed and level of complexity required to

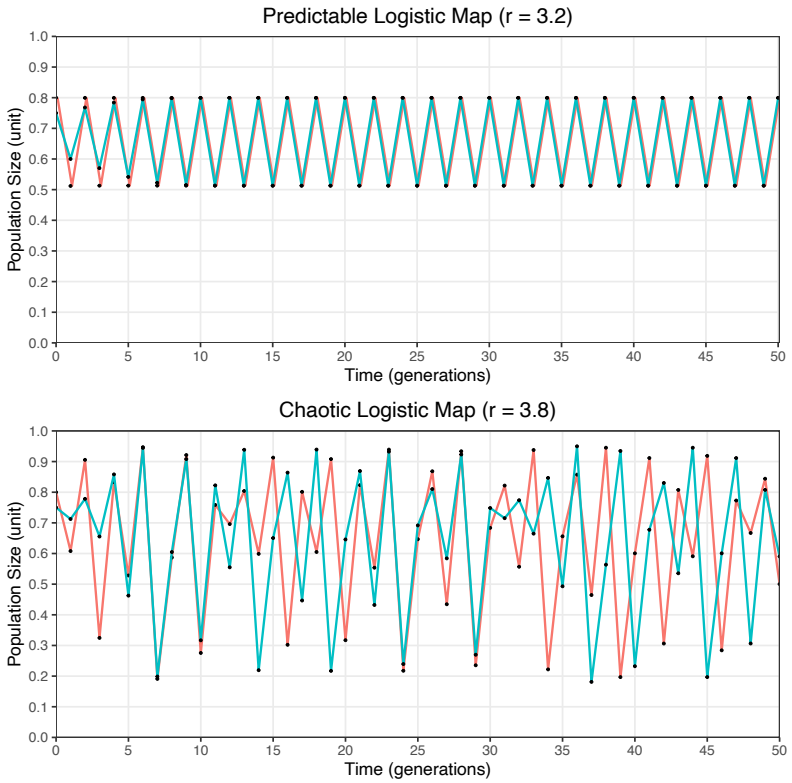

FIGURE 5.3. When systems are predictable, small changes in initial conditions can taper out. When they are chaotic, small changes in initial conditions lead to wildly different outcomes.

address these problems. In this way, automating processes could set up a positive feedback loop, requiring us to continually deploy ever-more AIs. In this scenario, the long-term use of AIs could be hard to control or reverse.

Summary. There are multiple ways in which complex systems exhibit nonlinearity. A small change in the system's input will not necessarily result in a proportional change in its behavior; it might completely change the system's behavior, or have no effect at all. Positive feedback loops can amplify changes, while negative feedback loops can quash them, leading a system to evolve nonlinearly depending on its current state, and making its long-run trajectory difficult to predict.

Self-Organization

The next salient feature of complex systems we will discuss is *self-organization*. This refers to how the components direct themselves in a way that produces collective emergent properties without any explicit instructions.

Complex systems sometimes organize themselves spontaneously. The forms and internal structure changes of complex systems are neither imposed by a top-down design nor centrally coordinated by "master components." The high-level

order and organization of a complex system is itself an emergent property that cannot be analyzed in terms of individual components. We will now look at some examples of self-organization.

Workers self-organize in ant colonies. In ant colonies, worker ants perform a variety of tasks, such as nest maintenance, brood care, foraging for food, and patrolling around the nest for signs of danger. Task allocation is partly determined by demand and opportunity in the environment. For example, the colony will shift to a more forager-heavy distribution if it discovers a large food source. The way in which individual ants are recruited to different tasks according to environmental demand and opportunity is self-organizing: a product of local stochastic interactions between the individuals, not set by a central controller (there's no ant commander).

The efficient market hypothesis states that economies self-organize to set prices. Increasing the price of a product leads to an increase in its supply (as profit margins for vendors are higher) and a decrease in its demand (as fewer consumers can afford it). Decreasing the price of a product has the reverse effect. In theory, the market price of a product will stabilize around the value at which the supply matches the demand. The system of vendors and consumers automatically "finds" the equilibrium market price without any centralized control or external help.

A neural network largely self-organizes during training. One could argue that there is an element of top-down control in the training of a neural network, in the way the backpropagation adjusts parameters to reduce the loss. However, there is not a predetermined plan specifying which parts of it are supposed to perform the different functions needed to carry out the task. Instead, the training process starts with a disordered system and its ultimate shape is determined by many interactions between components, resulting in a highly decentralized organization throughout the network. To a large extent, therefore, the training process resembles self-organization.

Summary. In a complex system, each component responds to conditions and directs its own actions such that the components collectively exhibit emergent behaviors without any external or central control. Neural networks arrange themselves in this way during training.

Self-Organized Criticality

Through self-organization, complex systems can reliably reach configurations that might seem improbable or fine-tuned. We will now look at the phenomenon of *self-organized criticality*, which is an important example of this.

Criticality is when a system is balanced at a tipping point between two different states. In nuclear engineering, the "critical mass" is the mass of a fissile material needed for a self-sustaining nuclear chain reaction. Below the critical mass, the chain reaction quickly dies out; above the critical mass, it continues at an ever-increasing rate and blows up. The critical mass is a boundary between these two regimes—the point at which the system "tips over" from being subcritical (stable

and orderly) to supercritical (unstable and disorderly). It is therefore referred to as the tipping point, or critical point, of the fissile system. Under normal operations, nuclear reactors are maintained at a critical state where the ongoing reaction ensures continual energy generation without growing into a dangerous, uncontrolled reaction.

Systems at their critical point are optimally sensitive to fluctuating conditions. In the nuclear case, an internal fluctuation would be the spontaneous fission of a nucleus. Below the critical point, the consequences of this event invariably remain confined to the neighborhood of the nucleus; above the critical point, the knock-on effects run out of control. Precisely at criticality, a local fission event can precipitate a chain reaction of any size, ranging from a short burst to a cascading reaction involving the entire system. This demonstrates how, at a critical point, a small event can have the broadest possible range of effects on the system.

The concept of criticality applies far beyond nuclear engineering: one classic example is the sandpile model. A sandpile has a critical slope, which is the tipping point between a tall, unstable pile and a shallow, stable pile. Shallower than this slope, the pile is relatively insensitive to perturbations: dropping additional grains onto the pile has little effect beyond making it taller. Once we reach the critical slope, however, the pile is poised to avalanche, and dropping extra grains can lead to avalanches of any size, including system-wide ones that effectively cause the whole pile to collapse. Again, we see that, at criticality, single events can have a wide range of effects on the system.

The freezing point of water is a critical temperature between its solid and liquid phases. In ice, the solid phase of water, there is long-range order, and fluctuations away from this (pockets of melting ice) are small and locally contained. In the liquid phase, there is long-range disorder, and fluctuations away from this (formation of ice crystals) are likewise small and locally contained. But at the freezing point of water—the critical point between the solid and liquid phases—the local formation of an ice crystal can rapidly spread across the whole system. As a result, a critically cooled bottle of beer can suddenly freeze all at once when it is perturbed, for example by being knocked against a table.

Neural networks display critical points. Several studies have found that certain capabilities of neural networks suddenly 'switch on' at a critical point as they are scaled up. For example, grokking is a network's ability to work accurately for general, random datasets, not just the datasets used in training. One study trained neural networks to recognize patterns in tables of letters and fill in the blanks, and found that grokking switched on quite suddenly [281]. The study reported that this ability remained near zero up to 10^5 optimization steps, but then steeply increased to near 100% accuracy by 10^6 steps. This could be viewed as a critical point.

Self-organized criticality means systems can evolve in a "punctuated equilibrium". According to the theory of *punctuated equilibrium*, evolutionary history consists of long periods of relative stasis in which species experience very little change, punctuated by occasional bursts of rapid change across entire ecosystems.

These sudden bursts can be understood through the lens of self-organized criticality. Ecosystems typically in equilibrium can slowly tend towards critical points, where they are optimally sensitive to perturbations from outside (such as geological events) or fluctuations from within (such as an organism developing a new behavior or strategy through a chance mutation). When the ecosystem is near a critical point, such a perturbation can potentially set off a system-wide cascade of changes, in which many species will need to adapt to survive. Similarly, AI development sometimes advances in bursts (e.g., GANs, self-supervised learning in vision, and so on) with long periods of slow development.

Summary. Complex systems often maintain themselves near critical points, or "tipping points". At these points, a system is optimally sensitive to internal fluctuations and external inputs. This means it can undergo dramatic changes in response to relatively minor events. A pattern of dramatic changes that sporadically interrupt periods of little change can be described as a punctuated equilibrium.

Distributed Functionality

As discussed earlier in this chapter, it is usually impractical to attempt to decompose a complex system into its parts, assign a different function to each one, and then assume that the system as a whole is the sum of these functions. Part of the reason for this is *distributed functionality*, another hallmark of complexity which we will now explore.

Complex systems can often be described as performing tasks or functions. Insect colonies build nests, forage for food, and protect their queens; economies calculate market prices and interest rates; and the human brain regulates all the bodily processes essential for our survival, such as heartbeat and breathing. In this context, we can understand adaptive behavior as the ability of a complex system to maintain its functionality when placed in a new environment or faced with new demands.

In complex systems, different functions are not neatly divided up between subsystems. Consider a machine designed to make coffee. In human artifacts like this, there is a clear delegation of functions to different parts of the system—one part grinds the beans, another froths the milk, and so forth. This is how non-complex systems usually work to perform their tasks. In complex systems, by contrast, no subsystem can perform any of the system's functions on its own, whereas all the subsystems working together can collectively perform many different tasks. This property is called "distributed functionality."

Note that distributed functionality does not imply that there is absolutely no functional specialization of the system's components. Indeed, the components of a complex system usually come in a diversity of different types, which contribute in different ways to the system's overall behavior and function. For example, the worker ants in an ant colony can belong to different groups: foragers, patrollers, brood care ants, and so on. Each of these groups, however, performs various functions for the colony,

and distributed functionality implies that, within each group of specialists, there is no rigid assignment of functions to components.

Partial encoding means that no single component can complete a task alone. The group of forager ants must perform a variety of subtasks in service of the foraging process: locating a food source, making a collective decision to exploit it, swarming over it to break it up, and carrying small pieces of it back to the nest. A single forager ant working alone cannot perform this whole process—or even any one subtask; many ants are needed for each part, with each individual contributing only partially to each task. We therefore say that foraging is partially encoded within any single forager ant.

Redundant encoding means there are more components than needed for any task. A flourishing ant colony will have many more ants than are necessary to carry out its primary functions. This is why the colony long outlives its members; if a few patroller ants get eaten, or a few foragers get lost, the colony as a whole barely notices. We therefore say that each of the functions is redundantly encoded across the component ants.

An example of distributed functionality is the phenomenon known as the "wisdom of crowds", which was notably demonstrated in a report from a village fair in 1906. At this fair, attendees were invited to take part in a contest by guessing the weight of an ox. 787 people submitted estimates, and it was reported that the mean came to 1,197 pounds. This was strikingly close to the actual weight, which was 1,198 pounds.

In situations like this, it is often the case that the average estimate of many people is closer to the true value than any individual's guess. We could say that the task of making a good estimate is only partially encoded in any given individual, who cannot alone get close to the actual value. It is also redundantly encoded because any individual's estimate can usually be ignored without noticeably affecting the average.

On a larger scale, the wisdom of crowds might be thought to underlie the effectiveness of democracy. Ideally, a well-functioning democracy should make better decisions than any of its individual members could on their own. This is not because a democratic society decomposes its problems into many distinct sub-problems, which can then be delegated to different citizens. Instead, wise democratic decisions take advantage of the wisdom of crowds phenomenon, wherein pooling or averaging many people's views leads to a better result than trusting any individual. The "sense-making" function of democracies is therefore distributed across society, partially and redundantly encoded in each citizen.

Neural networks show distributed functionality. In neural networks, distributed functionality manifests most clearly as distributed representation. In sufficiently large neural networks, the individual nodes do not correspond to particular concepts, and the weights do not correspond to relationships between concepts. In essence, the nodes and connections do not "stand for" anything specific. Part of the reason for this is partial encoding: in many cases, any given feature of the input data

will activate many neurons in the network, making it impossible to locate a single neuron that represents this feature. In addition, so-called polysemantic neurons are activated by many different features of the input data, making it hard to establish a correspondence between these neurons and any individual concepts.

Distributed functionality makes it hard to understand what complex systems are doing. Distributed functionality means that we cannot understand a complex system by attributing each task wholly and exclusively to a particular component, as the mechanistic approach would seek to. Distributed representation in neural networks is a particularly troubling instantiation of this insofar as it poses problems for using human concepts in analyzing a complex system's "cognition". The presence of distributed representation might be thought to substantiate the concern that neural networks are uninterpretable "black boxes".

Summary. Distributed functionality often means that no function in a complex system can be fully or exclusively attributed to a particular component. Since tasks are more loosely shared among components, this is one of the main reasons that it is so difficult to develop a definitive model of how a complex system works.

Scalable Structure and Power Laws

As discussed above, the properties of a complex system often scale nonlinearly with its size. Instead, they often follow power laws, where a property is proportional to the system size raised to some power that may be more or less than 1. We will now discuss these *power laws*, which are another hallmark of complex systems.

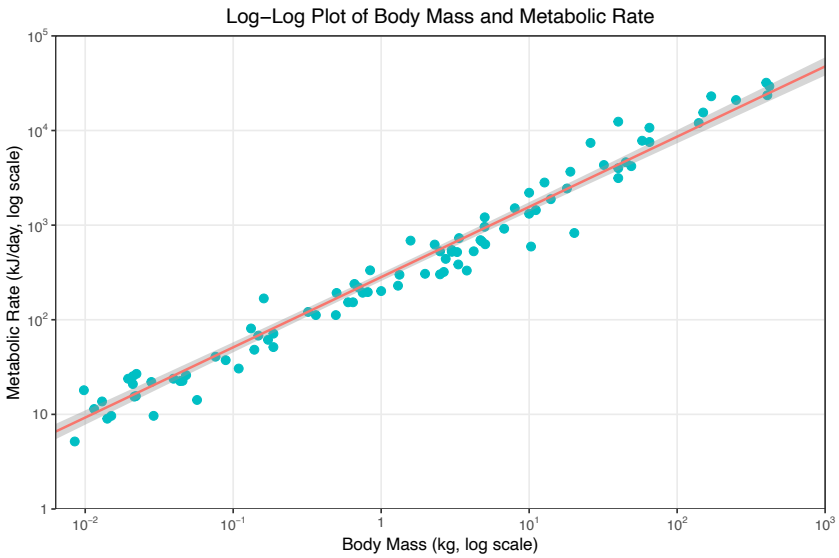

FIGURE 5.4. Data on mammals and birds demonstrate Kleiber's Law, with a power law relationship appearing as a straight line on a log-log graph.

Complex systems often obey power-law scalings of their properties with system size. Perhaps the most famous example of a power-law scaling is Kleiber's law in biology: across all mammals and birds, and possibly beyond, the metabolic rate of a typical member of a species scales with the three-quarters power of its body mass.

$$R \propto M^{\frac{3}{4}}$$

If we know that an elephant is five times heavier than a horse, we can guess that the elephant's metabolic rate will be approximately 3.3 times the horse's (since $5^{\frac{3}{4}} \approx 3.3$). There are several other documented cases of this power-law scaling behavior in complex systems. The average heart-rate for a typical member of a mammalian species scales with the minus one-quarter power of its body mass:

$$R \propto M^{-\frac{1}{4}}.$$

At the same time, the average lifespan scales with the one-quarter power of its body mass:

$$T \propto M^{\frac{1}{4}}.$$

This leads to the wonderful result that the average number of heartbeats per lifetime is constant across all species of mammals (around 1.5 billion).

Among cities within the same country, the material infrastructure (such as the lengths of pipes, powerlines, and roads, and the number of gas stations) scales with population as a power-law with an exponent of 0.85. Also among cities within the same country, socioeconomic quantities (such as incidents of crime and cases of flu) scale with the population size raised to the 1.15 power.

Experiments on LLMs show that their loss obeys power laws too. In the paper in which DeepMind introduced the Chinchilla model ([160]), the researchers fit the following parametric function to the data they collected from experiments on language models of different sizes, where N is the size of the model and D is the size of the training dataset:

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}.$$

The irreducible loss (E) is the lowest loss that could possibly be achieved. Subtracting this off, we see that the performance of the model as measured by the loss (L) exhibits a power-law dependency on each of model parameter count (N) and dataset size (D).

For more details on scaling laws in deep learning systems, see the Scaling Laws section in Artificial Intelligence & Machine Learning.

Summary. Certain important properties of complex systems often scale nonlinearly with the size of the system. This means that two separate systems will not behave in the same way as one single system of equivalent size.

Adaptive Behavior

The final hallmark of complexity we will discuss is *adaptive behavior*, which involves a system changing its behavior depending on the demands of the environment.

Complex systems often adapt flexibly to new tasks and environmental changes. Honeybees usually need to maintain their hives within an optimum temperature range of 32-36°C. When temperatures rise too high, bees engage in various adaptive behaviors to counteract this. They fan their wings at the hive entrance, increasing air circulation to cool down the hive. Additionally, more bees are sent out to gather water, which helps regulate the hive's temperature back to normal [282]. This ability to adjust their behavior to maintain homeostasis during environmental changes exemplifies one type of adaptive behavior.

The human brain, on the other hand, showcases a different form of adaptability. It possesses the remarkable capacity to navigate novel circumstances and solve unfamiliar problems. When faced with new challenges, the brain's ability to think about different things allows us to adapt and thrive in diverse environments. For example, London's taxi drivers ("cabbies") have been found to have larger-than-average memory centers. This adaptation enables them to navigate the complex maze of London's streets effectively. Furthermore, the brain can also adapt in response to injury. After a stroke or head injury, it can rewire itself, repurposing undamaged areas to compensate for the damaged ones. This adaptive behavior showcases the brain's remarkable plasticity and its ability to adapt and function even after experiencing trauma.

Some deep learning systems exhibit adaptive behavior. So-called "online models" learn from new data sequentially as they encounter it, rather than remaining fixed after an initial training phase. This enables these models to dynamically adapt to datasets that change over time, as well as continuing to perform well in the real world when the inputs they encounter differ from their training data, an ability known as "test-time adaptation" or simply "adaptation". While other deep learning systems such as Large Language Models remain fixed after their training phase, there are strong incentives to make these systems adaptive to overcome current limitations such as costs of re-training and lack of up-to-date information after the training date.

Another example of adaptive behavior in deep learning systems is *few-shot prompting*. This technique enables general deep learning models (such as large language models) to be used to perform certain tasks without any task-specific fine-tuning. It involves giving the model a few examples ("shots") of correct performance on the task, which stimulate the model to adapt its outputs to these examples and thereby carry out the desired task.

Summary. Complex systems can often undergo rapid changes in their structures and processes in response to internal and external fluctuations. This adaptive behavior enables the continuation of the system in a changing environment.

Review of the Hallmarks of Complexity

There are seven hallmarks of complexity that we can look out for when identifying complex systems. These hallmarks are:

1. Emergence: the appearance of novel properties that arise from interactions between the system's components, but which do not exist in any single component. These properties cannot be understood or predicted from reductive analysis of components.
2. Feedback and nonlinearity: the presence of feedback loops that can either amplify or quash changes in a complex system, and the multiple ways in which a change in the input to a complex system can produce a disproportionate change in the output.
3. Self-organization: the ability of a complex system to spontaneously self-organize through the self-directed behaviors of the components, without any external or centralized control.
4. Self-organized criticality: the tendency of complex systems to maintain themselves near critical points, at which they can undergo dramatic changes in response to even relatively minor perturbations.
5. Distributed functionality: the way in which tasks are shared loosely among a complex system's components. Tasks are both partially encoded—each individual contributes only partially to a task—and redundantly encoded—there are more individuals that can contribute to a task than are strictly necessary to complete it.
6. Scalable structure: the way in which properties of complex systems scale nonlinearly with size, so that a property of a single large system may be larger or smaller than the combined properties of two separate systems of half the size.
7. Adaptive behavior: a complex system's ability to change its structure and processes in response to perturbations, enabling it to continue functioning in a changing environment.

5.2.6 Social Systems as Complex Systems

So far, we have described how deep learning systems possess many of the classic features of complex systems. We have shown that they satisfy the two aspects of our working definition of complex systems and that they display all seven hallmarks discussed above.

We will now consider the organizations that develop AIs and the societies within which they are deployed, and describe how these systems also exhibit the characteristics of complex systems. We will argue that, on this basis, the problem of AI safety should be treated under the complex systems paradigm.

Worked Example: Corporations and Research Institutes as Complex Systems

The organizations developing AI technology are complex systems. Corporations and research institutes have multiple emergent properties that are not found in any of the individuals working within them. Brand identity, for example, does not exist in any employee of a company, but rather embodies and conveys the collective activities of all the employees and conveys the goals of the company as a whole. Similarly, the concepts of organizational culture and research culture refer to the general ways in which individuals tend to interact with one another within an organization or a research field.

Organizations developing AI are self-organizing. Although companies have CEOs, these CEOs are often selected by groups of people, such as board members, and do not generally dictate every single activity within the company and how it should be done. People self-select in applying to work at a company that interests them, and managers usually make decisions together on who to hire. Employees often come up with their own ideas for projects and strategies, and then decisions are made collectively on which ones to pursue. Likewise in academia, researchers investigate their own questions, keep up to date with the findings of their peers, and use those insights to inform their research directions, while experts form committees to decide which projects should be funded. There is very often no single central entity determining which researchers should work on which questions.

Both corporate and academic organizations can exhibit critical points. Often, a lot of an organization's effort is focused on a particularly consequential area or problem until a big breakthrough is made, representing a tipping point into a new paradigm. For this reason, research and development often progresses in the pattern of a *punctuated equilibrium*, with long periods of incremental advancements interrupted by windows of rapid advancements, following important breakthroughs.

Companies and research institutes show multiple forms of adaptive behavior. Examples of adaptation include organizations incorporating new information and technology to update their strategies and ways of working, and adjusting their research directions based on new findings. Additionally, they may adapt to the changing needs of customers and the changing priorities of research funders, as well as to new government regulations.

Companies and research institutes display distributed functionality. While there may be subsystems that focus on specialized tasks within a company or branches within a research field, in general, no employee or researcher single-handedly performs a whole function or advances an area alone. Even if there is just one person working in a particular niche, they still need to be informed by related tasks and research performed by others, and usually rely on the work of support staff. This illustrates *partial encoding*. There are also usually more people available to perform tasks than are absolutely needed, meaning that processes continue over time despite employees and researchers joining and leaving. This demonstrates *redundant encoding*.

There are multiple examples of feedback and nonlinearity in companies and institutes. A small disparity in investment into different projects or research directions may be compounded over time, with those that receive more initial funding also achieving bigger results, and therefore receiving even more funding. A small difference in support for a particular candidate in senior management can be decisive in whether or not they are selected, and thus have a large influence over future directions. More broadly, different organizations may imitate one another's successes, leading to a concentration of work in a particular area, while a small initial advantage of one organization may be amplified over time, allowing it to dominate the area.

Summary. The environments in which research and development occur display the hallmarks of complexity and are therefore best understood as complex systems. Research organizations and corporations possess emergent properties including safety culture, which is paramount for AI safety. Additionally, progress may have critical points and unfold in a nonlinear way that is difficult to predict. It is crucial that AI safety strategies are informed by these possibilities.

Worked Example: Complex Systems Applied to Advocacy

The social systems within which AI is deployed are complex systems. We find emergence and the hallmarks of complexity in all social systems, from political structures to economic networks to society as a whole. To illustrate this more specifically, we will now focus on the example of policymaking structures and advocacy. This example is particularly relevant to AI safety, because reducing risks from AI will need to involve the implementation of policies around its use. Advocacy will therefore be necessary to promote safety policies and convince policymakers to adopt them.

Social systems display emergence and self-organization. Patterns of governance and collective decision making, such as democracy, can be considered emergent properties of social and political systems. Although some individuals reach positions of power that might seem to centralize control, social systems are nonetheless partly *self-organizing*, in the sense that many individuals interact with one another and make decisions about whom to support, collectively determining which candidate is elected. Similarly, people who care about particular causes *self-organize* to form advocacy groups and set up grassroots campaigns. Policymakers interact with each other and various stakeholders, including advocates, to reach policy decisions.

Advocacy movements have critical points and often unfold as punctuated equilibria. Movements advancing different causes often display *critical points*, where a critical level of awareness and support must be reached before policymakers will pay attention. Social systems may self-organize toward this critical level of support and maintain it over time. However, the actual "tipping" from one state into another, wherein policies are implemented, may be dependent on other external factors, such as whether there are other urgent issues dominating decision-makers' attention. For this reason, advocacy efforts and their results tend to progress

as *punctuated equilibria*; there may be little apparent change for a long time, despite sustained work, and then a lot of sudden progress when momentum builds and the political climate is right for it.

Both advocacy groups and policymaking structures also exhibit adaptive behavior. Policymakers must continually adapt to the fluctuating political landscape and changing concerns of the public. Similarly, advocacy groups must constantly adjust their activities to capture the attention of the public and policymakers and convince them that a particular cause is relevant and important. They might, for instance, use new technology to innovate an original mode of campaigning, or link the cause to the prevailing zeitgeist—another emergent property of social systems.

Distributed functionality is evident on multiple levels in social systems. The various tasks involved in advocacy are *partially* and *redundantly* encoded across individuals within groups, allowing campaigns to continue even as people leave and join them. More broadly still, there are usually several groups campaigning for any given cause, meaning that the general function of advocacy is distributed across different organizations. Decision making is also partially and redundantly encoded among many policymakers, who interact with one another and various stakeholders to consider different perspectives and decide on policies.

There are many nonlinear aspects of processes like advocacy. There are numerous factors that affect whether or not an issue is included on a policy agenda. Public interest in a cause, the influence of opponents of a cause, and the number of other issues competing for attention are among the many factors that affect the likelihood that it is considered non-linearly; for instance, opponents with low influence may permit an issue being discussed, opponents with medium influence may try and block it from discussed, but opponents with high influence may permit it being discussed so that they can argue against it. Additionally, there is a degree of randomness involved in determining which issues are considered. This means that the policy progress resulting from a particular campaign does not necessarily reflect the level of effort put into it, nor how well organized it was.

Together with *distributed functionality* and *critical points*, this *nonlinearity* can make it difficult to evaluate how well a campaign was executed or attribute eventual success. It might be that earlier efforts were essential in laying the groundwork or simply maintaining some interest in a cause, even if they did not yield immediate results. A later campaign might then succeed in prompting policy-level action, regardless of whether it is particularly well organized, simply because the political climate becomes favorable.

Other examples of *nonlinearity* within advocacy and policymaking arise from various feedback loops. Since people are influenced by the opinions of those around them, a small change in the initial level of support for a policy might be compounded over time, creating momentum and ultimately tipping the balance as to whether or not it is adopted. On the other hand, original activities that are designed to be attention-grabbing may run up against negative feedback loops that diminish their

power over time. Other groups may imitate them, for instance, so that their novelty wears off through repetition. Opponents of a cause may also learn to counteract any new approaches that advocates for it try out. This dynamic was understood by the military strategist Moltke the Elder, who is reported to have said that "no plan survives first contact with the enemy".

Political systems and advocacy groups have scalable structure. Political systems usually have a hierarchical structure with multiple levels of organization, such as councils responsible for specific regions within a country and politicians forming a national government to address countrywide issues. Advocacy groups can also exhibit this kind of structure. There may, for example, be a campaign manager spearheading efforts, and then many regional leaders who organize activities at a more local level. This scalable structure is another indication of complexity.

Summary. The presence of these hallmarks of complexity in social and political systems suggests they are best described within the complex systems paradigm. Additionally, these observations can offer some insights into how we might approach advocacy for AI safety, suggesting it is not as simple as developing a good policy idea and making a convincing argument for it.

Instead, it is likely that successful advocacy over the long term will be characterized by adaptability to different political circumstances and changing public attitudes, as well as in response to opponents' activities. Advocates will need to invest in building and maintaining relationships with the relevant people and organizations, rather than just presenting the case for a policy. There may need to be a lot of work that is not immediately rewarded, but momentum should be maintained so that advocates are ready to capitalize on moments when the political climate becomes more favorable. It should also be understood that it might not be possible to attribute success in any obvious way.

It Is Difficult to Foresee How the Use of AI Will Unfold

Complex social systems mean the eventual impact of AI is hard to predict. As discussed earlier, the behavior of complex systems can be difficult to predict for many reasons, such as the appearance of unanticipated emergent properties and feedback loops amplifying small changes in initial conditions. This is compounded if a system is new to us; we may be able to predict certain high-level behaviors of complex systems we are familiar with and have a lot of historical data on, such as weather patterns and beehives, but AI systems are relatively new. Additionally, the deployment of AI within society represents a case of nested complexity, where complex systems are embedded within one another. This vastly increases the range of potential interactions and the number of ways in which the systems can co-evolve. As a result, it is difficult to predict all the ways in which AI might be used and what its eventual impact will be.

While this technology could have many positive effects, there is also potential for interactions to have negative consequences. This is especially true if AI is deployed

in ways that enable it to affect actions in the world; for example, if it is put in charge of automated decision-making processes.

5.3 COMPLEX SYSTEMS FOR AI SAFETY

5.3.1 General Lessons from Complex Systems

As we have discussed, AI systems and the social systems they are integrated within are best understood as complex systems. For this reason, making AI safe is not like solving a mathematical problem or fixing a watch. A watch might be *complicated*, but it is not *complex*. Its mechanism can be fully understood and described, and its behavior can be predicted with a high degree of confidence. The same is not true of complex systems.

Since a system's complexity has a significant bearing on its behavior, our approach to AI safety should be informed by the complex systems paradigm. We will now look at some lessons that have been derived from observations of many other complex adaptive systems. We will discuss each lesson and what it means for AI safety.

Lesson: Armchair Analysis Is Limited for Complex Systems

Learning how to make AIs safe will require some trial and error. We cannot usually attain a complete understanding of complex systems or anticipate all their emergent properties purely by studying their structure in theory. This means we cannot exhaustively predict every way they might go wrong just by thinking about them. Instead, some amount of trial and error is required to understand how they will function under different circumstances and learn about the risks they might pose. The implication for AI safety is that some simulation and experimentation will be required to learn how AI systems might function in unexpected or unintended ways and to discover crucial variables for safety.

Biomedical research and drug discovery exemplify the limitations of armchair theorizing. The body is a highly complex system with countless biochemical reactions happening all the time, and intricate interdependencies between them. Researchers may develop a drug that they believe, according to their best theories, should treat a condition. However, they cannot thoroughly analyze every single way it might interact with all the body's organs, processes, and other medications people may be taking. That is why clinical trials are required to test whether drugs are effective and detect any unexpected side effects before they are approved for use.

Similarly, since AI systems are complex, we cannot expect to predict all their potential behaviors, emergent properties, and associated hazards simply by thinking about them. Moreover, AI systems will be even less predictable when they are taken out of the controlled development environment and integrated within society. For example, when the chatbot Tay was released on Twitter, it soon started to make racist and sexist comments, presumably learned through its interactions with other Twitter users in this uncontrolled social setting.

Approaches to AI safety will need to involve experimentation. Some of the most important variables that affect a system's safety will likely be discovered by accident. While we may have ideas about the kinds of hazards a system entails, experimentation can help to confirm or refute these. Importantly, it can also help us discover hazards we had not even imagined. These are called unknown unknowns, or black swans, discussed extensively in the Safety Engineering chapter. Empirical feedback loops are necessary.

Lesson: Systems Often Develop Subgoals Which Can Supersede the Original Goal

AIs might pursue distorted subgoals at the expense of the original goal. The implication for AI safety is that AIs might pursue subgoals over the goals we give them to begin with. This presents a risk that we might lose control of AIs, and this could cause harm because their subgoals may not always be aligned with human values.

A system often decomposes its goal into multiple subgoals to act as stepping stones. Subgoals might include instrumentally convergent goals, which are discussed in the Single-Agent Safety chapter. The idea is that achieving all the subgoals will collectively amount to achieving the original aim. This might work for a simple, mechanistic system. However, since complex systems are more than the sum of their parts, breaking goals down in this way can distort them. The system might get sidetracked pursuing a subgoal, sometimes even at the expense of the original one. In other words, although the subgoal was initially a means to an end, the system may end up prioritizing it as an end in itself.

For example, companies usually have many different departments, each one specialized to pursue a distinct subgoal. However, some departments, such as bureaucratic ones, can capture power and have the company pursue goals unlike its initial one. Political leaders can delegate roles to subordinates, but sometimes their subordinates may overthrow them in a coup.

As another example, imagine a politician who wants to improve the quality of life of residents of a particular area. Increasing employment opportunities often lead to improvement in quality of life, so the politician might focus on this as a subgoal—a means to an end. However, this subgoal might end up supplanting the initial one. For instance, a company might want to build an industrial plant that will offer jobs, but is also likely to leak toxic waste. Suppose the politician has become mostly focused on increasing employment rates. In that case, they might approve the construction of this plant, despite the likelihood that it will pollute the environment and worsen residents' quality of life in some ways.

Future AI agents may break down difficult long-term goals into smaller subgoals. Creating subgoals can distort an AI's objective and result in misalignment. As discussed in the Emergent Capabilities section of the Single-Agent Safety Chapter, optimization algorithms might produce emergent optimizers that pursue subgoals, or AI agents may delegate goals to other agents and potentially have the

goal be distorted or subverted. In more extreme cases, the subgoals could be pursued at the expense of the original one. We can specify our high-level objectives correctly without any guarantee that systems will implement these in practice. As a result, systems may not pursue goals that we would consider beneficial.

Lesson: A Safe System, When Scaled Up, Is Not Necessarily Still Safe

AIs may continue to develop unanticipated behaviors as we scale them up. When we scale up the size of a system, qualitatively new properties and behaviors emerge. The implication for AI safety is that, when we increase the scale of a deep learning system, it will not necessarily just get better at doing what it was doing before. It might begin to behave in entirely novel and unexpected ways, potentially posing risks that we had not thought to prepare for.

It is not only when a system transitions from relative simplicity into complexity that novel properties can appear. New properties can continue to emerge spontaneously as a complex system increases in size. As discussed earlier in this chapter, LLMs have been shown to suddenly acquire new capabilities, such as doing three-digit arithmetic, when the amount of compute used in training them is increased, without any qualitative difference in training. Proxy gaming capabilities have also been found to "switch on" at a certain threshold as the model's number of parameters increases; in one study, at a certain number of parameters, the proxy reward steeply increased, while the model's performance as intended by humans simultaneously declined.

Some emergent capabilities may pose a risk. As deep learning models continue to grow, we should expect to observe new emergent capabilities appearing. These may include potentially concerning ones, such as deceptive behavior or the ability to game proxy goals. For instance, a system might not attempt to engage in deception until it is sophisticated enough to be successful. Deceptive behavior might then suddenly appear.

Lesson: Working Complex Systems Have Usually Evolved From Simpler Systems

We are unlikely to be able to build a large, safe AI system from scratch. Most attempts to create a complex system from scratch will fail. More successful approaches usually involve developing more complex systems gradually from simpler ones. The implication for AI safety is that we are unlikely to be able to build a large, safe, working AI system directly. As discussed above, scaling up a safe system does not guarantee that the resulting larger system will also be safe. However, starting with safe systems and cautiously scaling them up is more likely to result in larger systems that are safe than attempting to build the larger systems in one fell swoop.

Building complex systems directly is difficult. Since complex systems can behave in unexpected ways, we are unlikely to be able to design and build a large, working one from scratch. Instead, we need to start by ensuring that smaller systems work and then build on them. This is exemplified by how businesses develop; a business usually begins as one person or a few people with an idea, then becomes

a start-up, then a small business, and can potentially grow further from there. People do not usually attempt to create multinational corporations immediately without progressing naturally through these earlier stages of development.

One possible explanation for this relates to the limitations of armchair theorizing about complex systems. Since it is difficult to anticipate every possible behavior and failure mode of a complex system in advance, it is unlikely that we will be able to design a flawless system on the first attempt. If we try to create a large, complex system immediately, it might be too large and unwieldy for us to make the necessary changes to its structure when issues inevitably arise. If the system instead grows gradually, it has a chance to encounter relevant problems and adapt to deal with them during the earlier stages when it is smaller and more agile.

Similarly, if we want large AI systems that work well and are safe, we should start by making smaller systems safe and effective and then incrementally build on them. This way, operators will have more chances to notice any flaws and refine the systems as they go. An important caveat is that, as discussed above, a scaled-up system might have novel emergent properties that are not present in the smaller version. We cannot assume that a larger system will be safe just because it has been developed in this way. However, it is more likely to be safe than if it was built from scratch. In other words, this approach is not a guarantee of safety, but it is likely our best option. The scaling process should be done cautiously.

Lesson: Any System Which Depends on Human Reliability Is Unreliable

Gilb's Law of Unreliability. We cannot guarantee that an operator will never make an error, and especially not in a large complex system. As the chemical engineer Trevor Kletz put it: "Saying an accident is due to human failing is about as helpful as saying that a fall is due to gravity. It is true but it does not lead to constructive action" [283]. To make a complex system safer, we need to incorporate some allowances in the design so that a single error is not enough to cause a catastrophe.

The implication of this for AI safety is that having humans monitoring AI systems does not guarantee safety. Beyond human errors of judgment, processes in some complex systems may happen too quickly for humans to be included in them anyway. AI systems will probably be too fast-moving for human approval of their decisions to be a practical or even a feasible safety measure. We will therefore need other ways of embedding human values in AI systems and ensuring they are preserved, besides including humans in the processes. One potential approach might be to have some AI systems overseeing others, though this brings its own risks.

Summary.

The general lessons that we should bear in mind for AI safety are:

1. We cannot predict every possible outcome of AI deployment by theorizing, so some trial and error will be needed

2. Even if we specify an AI's goals perfectly, it may start not to pursue them in practice, as it may instead pursue unexpected, distorted subgoals
3. A small system that is safe will not necessarily remain safe if it is scaled up
4. The most promising approach to building a large AI that is safe is nonetheless to make smaller systems safe and scale them up cautiously
5. We cannot rely on keeping humans in the loop to make AI systems safe, because humans are not perfectly reliable and, moreover, AIs are likely to accelerate processes too much for humans to keep up.

5.3.2 Puzzles, Problems, and Wicked Problems

So far, we have explored the contrasts between simple and complex systems and why we need different approaches to analyzing and understanding them. We have also described how AIs and the social systems surrounding them are best understood as complex systems, and discussed some lessons from the field of complex systems that can inform our expectations around AI safety and how we address it.

In attempting to improve the safety of AI and its integration within society, we are engaging in a form of problem-solving. However, simple and complex systems present entirely different types of problems that require different styles of problem-solving. We can therefore reframe our earlier discussion of reductionism and complex systems in terms of the kinds of challenges we can address within each paradigm. We will now distinguish between three different kinds of challenges—puzzles, problems, and wicked problems. We will look at the systems that they tend to arise in, and the different styles of problem-solving we require to tackle each of them.

Puzzles and Problems

Puzzles. Examples of puzzles include simple mathematics questions, sudokus, assembling furniture, and fixing a common issue with a watch mechanism. In all these cases, there is only one correct result and we are given all the information we need to find it. We usually find puzzles in simple systems that have been designed by humans and can be fully understood. These can be solved within the reductionist paradigm; the systems are simply the sum of their parts, and we can solve the puzzle by breaking it down into a series of steps.

Problems. With problems, we do not always have all the relevant information upfront, so we might need to investigate to discover it. This usually gives us a better understanding of what's causing the issue, and ideas for solutions often follow naturally from there. It may turn out that there is more than one approach to fixing the problem. However, it is clear when the problem is solved and the system is functioning properly again.

We usually find problems in systems that are complicated, but not complex. For example, in car repair work, it might not be immediately apparent what is causing an issue. However, we can investigate to find out more, and this process often leads

us to sensible solutions. Like puzzles, problems are amenable to the reductionist paradigm, although they may involve more steps of analysis.

Wicked Problems

Wicked problems usually arise in complex systems and often involve a social element. Wicked problems are a completely different class of challenges from puzzles and problems. They appear in complex systems, with examples including inequality, misinformation, and climate change. There is also often a social factor involved in wicked problems, which makes them more difficult to solve. Owing to their multifaceted nature, wicked problems can be tricky to categorically define or explain. We will now explore some key features that are commonly used to characterize them.

There is no single explanation or single solution for a wicked problem. We can reasonably interpret a wicked problem as stemming from more than one possible cause. As such, there is no single correct solution or even a limited set of eternal possible solutions.

No proposed solution to a wicked problem is fully right or wrong, only better or worse. Since there are usually many factors involved in a wicked problem, it is difficult to find a perfect solution that addresses them all. Indeed, such a solution might not exist. Additionally, due to the many interdependencies in complex systems, some proposed solutions may have negative side effects and create other issues, even if they reduce the targeted wicked problem. As such, we cannot usually find a solution that is fully correct or without flaw; rather, it is often necessary to look for solutions that work relatively well with minimal negative side effects.

There is often a risk involved in attempting to solve a wicked problem. Since we cannot predict exactly how a complex system will react to an intervention in advance, we cannot be certain as to how well a suggested solution will work or whether there will be any unintended side effects. This means there may be a high cost to attempting to address wicked problems, as we risk unforeseen consequences. However, trying out a potential solution is often the only way of finding out whether it is better or worse.

Every wicked problem is unique because every complex system is unique. While we can learn some lessons from other systems with similar properties, no two systems will respond to our actions in exactly the same way. This means that we cannot simply transpose a solution that worked well in one scenario to a different one and expect it to be just as effective. For example, introducing predators to control pest numbers has worked well in some situations, but, as we will discuss in the next section, it has failed in others. This is because all ecosystems are unique, and the same is true of all complex systems, meaning that each wicked problem is likely to require a specifically tailored intervention.

It might not be obvious when a wicked problem has been solved. Since wicked problems are often difficult to perfectly define, it can be challenging to say

they have been fully eliminated, even if they have been greatly reduced. Indeed, since wicked problems tend to be persistent; it might not be feasible to fully eliminate many wicked problems at all. Instead, they often require ongoing efforts to improve the situation, though the ideal scenario may always be beyond reach.

AI safety is a wicked problem. Since AI and the social environments it is deployed within are complex systems, the issues that arise with its use are likely to be wicked problems. There may be no obvious solution, and there will probably need to be some trial and error involved in tackling them. More broadly, the problem of AI safety in general can be considered a wicked problem. There is no single correct approach, but many possibilities. We may never be able to say that we have fully "solved" AI safety; it will require ongoing efforts.

Summary. Puzzles and problems usually arise in relatively simple systems that we can obtain a complete or near-complete understanding of. We can therefore find all the information we need to explain the issue and find a solution to it, although problems may be more complicated, requiring more investigation and steps of analysis than puzzles.

Wicked problems, on the other hand, arise in complex systems, which are much more difficult to attain a thorough understanding of. There may be no single correct explanation for a wicked problem, proposed solutions may not be fully right or wrong, and it might not be possible to find out how good they are without trial and error. Every wicked problem is unique, so solutions that worked well in one system may not always work in another, even if the systems seem similar, and it might not be possible to ever definitively say that a wicked problem has been solved. Owing to the complex nature of the systems involved, AI safety is a wicked problem.

5.3.3 Challenges With Interventionism

As touched on above, there are usually many potential solutions to wicked problems, but they may not all work in practice, even if they sound sensible in theory. We might therefore find that some attempts to solve wicked problems will be ineffective, have negative side effects, or even backfire. Complex systems have so many interdependencies that when we try to adjust one aspect of them, we can inadvertently affect others. For this reason, we should approach AI safety with more humility and more awareness of the limits of our knowledge than if we were trying to fix a watch or a washing machine. We will now look at some examples of historical interventions in complex systems that have not gone to plan. In many cases, they have done more harm than good.

Cane toads in Australia. Sugarcane is grown in Australia as a valuable product in the economy, but a species of insect called the cane beetle is known to feed on sugarcane crops and destroy them. In the 1930s, cane toads were introduced in Australia to prey on these beetles, with the hope of minimizing crop losses. However, since cane toads are not native to Australia, they have no natural predators there.

In fact, the toads are toxic to many native species and have damaged ecosystems by poisoning animals that have eaten them. The cane toads have multiplied rapidly and are considered an invasive species. Attempts to control their numbers have so far been largely unsuccessful.

Warning signs on roads. Road accidents are a long-standing and pervasive issue. A widely used intervention is to display signs along roads with information about the number of crashes and fatalities that have happened in the surrounding area that year. The idea is that this information should encourage people to drive more carefully. However, one study has found that signs like this increase the number of accidents and fatalities, possibly because they distract drivers from the road.

Renewable Heat Incentive Scandal. In 2012, a government department in Northern Ireland wanted to boost the fraction of energy consumption from renewable sources. To this end, they set up an initiative offering businesses generous subsidies for using renewable heating sources, such as wood pellets. However, in trying to reach their percentage targets for renewable energy, the politicians offered a subsidy that was slightly more than the cost of the wood pellets. This incentivized businesses to use more energy than they needed and profit from the subsidies. There were reports of people burning pellets to heat empty buildings unnecessarily. The episode became known as the "Cash for Ash scandal".

Barbados-Grenada football match. In the 1994 Caribbean Cup, an international football tournament, organizers introduced a new rule to reduce the likelihood of ties, which they thought were less exciting. The rule was that if two teams were tied at the end of the allotted 90 minutes, the match would go to extra time, and any goal scored in extra time would be worth double. The idea was to incentivize the players to try harder to score. However, in a match between Barbados and Grenada, Barbados needed to win by two goals to advance to the tournament finals. The score as they approached 90 minutes was 2-1 to Barbados. This resulted in a strange situation where Barbados players tried to score an own goal to push the match into extra time and have an opportunity to win by two.

Summary. Interventions that work in theory might fail in a complex system. In all these examples, an intervention was attempted to solve a problem in a complex system. In theory, each intervention seemed like it should work, but each decision-maker's theory did not capture all the complexities of the system at hand. Therefore, when each intervention was applied, the system reacted in unexpected ways, leaving the original problem unsolved, and often creating additional problems that might be even worse.

Stable States and Restoring Forces

The examples above illustrate how complex systems can react unexpectedly to interventions. This can be partly attributed to the properties of self-organization and adaptive behavior; complex systems can organize themselves around new conditions in unobvious ways, without necessarily addressing the reason for the intervention.

Some interventions might partially solve the original problem but unleash unanticipated side effects that are not considered worth the benefits. Other interventions, however, might completely backfire, exacerbating the very problem they were intended to solve. We will now discuss the concept of "stable states" and how they might explain complex systems' tendency to resist attempts to change them.

If a complex system is in a stable state, it is likely to resist attempts to change it. If a ball is sitting in a valley between two hills and we kick it up one hill, gravity will pull it back to the valley. Similarly, if a complex system has found a stable state, there might be some "restoring forces" or homeostatic processes that will keep drawing it back toward that state, even if we try to pull it in a different direction. When complex systems are not near critical points, they exhibit robustness to external changes.

Another analogy is Le Chatelier's Principle, a well-known concept in chemistry. The principle concerns chemical equilibria, in which the concentrations of different chemicals stay the same over time. There may be chemical reactions happening, converting some chemicals into others, but the rate of any reaction will equal the rate of its opposite reaction. The total concentration of each chemical therefore remains unchanged, hence the term equilibrium.

Le Chatelier's Principle states that if we introduce a change to a system in chemical equilibrium, the system will shift to a new equilibrium in a way that partly opposes that change. For example, if we increase the concentration of one chemical, then the rate of the reaction using up that chemical will increase, using up more of it and reducing the extra amount present. Similarly, complex systems sometimes react against our interferences in them.

We will now look at some examples of interventions backfiring in complex systems. We will explore how we might think of these systems as having stable states and restoring forces that draw the system back toward its stable state if an intervention tries to pull it away. Note that the following discussions of what the stable states and restoring forces might be are largely speculative. Although these hypotheses have not been rigorously proven to explain these examples, they are intended to show how we can view systems and failed interventions through the lens of stable states and restoring forces.

Rules to restrict driving. In 1989, to tackle high traffic and air pollution levels in Mexico City, the government launched an initiative called "Hoy No Circula." The program introduced rules that allowed people to drive only on certain days of the week, depending on the last number on their license plate. This initially led to a drop in emissions, but they soon rose again, actually surpassing the pre-intervention levels. A study found that the rules had incentivized people to buy additional cars so they could drive on more days. Moreover, the extra cars people bought tended to be cheaper, older, more polluting ones, exacerbating the pollution problem [284].

We could perhaps interpret this situation as having a stable state in terms of how much driving people wanted or needed to do. When rules were introduced to try

to reduce it, people looked for ways to circumvent them. We could view this as a restoring force in the system.

Four Pests campaign. In 1958, the Chinese leader Mao Zedong launched a campaign encouraging people to kill the "four pests": flies, mosquitoes, rodents, and sparrows. The first three were targeted for spreading disease, but sparrows were considered a pest because they were believed to eat grain and reduce crop yields. During this campaign, sparrows were killed intensively and their populations plummeted. However, as well as grain, sparrows also eat locusts. In the absence of a natural predator, locust populations rose sharply, destroying more crops than the sparrows did [285]. Although many factors were at play, including poor weather and officials' decisions about food distribution [286], this ecosystem imbalance is often considered a contributing factor in the Great Chinese Famine [285], during which tens of millions of people starved between 1959 and 1961.

Ecosystems are highly complex, with intricate balances between the populations of many species. We could think of agricultural systems as having a "stable state" that naturally involves some crops being lost to wildlife. If we try to reduce these losses simply by eliminating one species, then another might take advantage of the available crops instead, acting as a kind of restoring force.

Antibiotic resistance. Bacterial infections have been a cause of illness and mortality in humans throughout history. In September 1928, bacteriologist Alexander Fleming discovered penicillin, the first antibiotic. Over the following years, the methods for producing it were refined, and, by the end of World War II, there was a large supply available for use in the US and Britain. This was a huge medical advancement, offering a cure for many common causes of death, such as pneumonia and tuberculosis. Death rates due to bacterial illnesses dropped dramatically [287]; it is estimated that, in 1952, in the US, around 150,000 fewer people died from bacterial illnesses than would have without antibiotics. In the early 2000s, it was estimated that antibiotics may have been saving around 200,000 lives annually in the US alone.

However, as antibiotics have become more abundantly used, bacteria have begun to evolve resistance to these vital medicines. Today, many bacterial illnesses, including pneumonia and tuberculosis, are once again becoming difficult to treat due to the declining effectiveness of antibiotics. In 2019, the Centers for Disease Control and Prevention reported that antimicrobial-resistant bacteria are responsible for over 35,000 deaths per year in the US [288].

In this case, we might think of the coexistence of humans and pathogens as having a stable state, involving some infections and deaths. While antibiotics have reduced deaths due to bacteria over the past decades, we could view natural selection as a "restoring force", driving the evolution of bacteria to become resistant and causing deaths to rise again. Overuse of these medicines intensifies selective pressures and accelerates the process.

In this case, it is worth noting that antibiotics have been a monumental advancement in healthcare, and we do not argue that they should not be used or that they are a

failed intervention. Rather, this example highlights the tendency of complex systems to react against measures over time, even if they were initially highly successful.

Instead of pushing a system in a desired direction, we could try to shift the stable states. If an intervention attempts to artificially hold a system away from its stable state, it might be as unproductive as repeatedly kicking a ball up a hill to keep it away from a valley. Metaphorically speaking, if we want the ball to sit in a different place, a more effective approach would be to change the landscape so that the valley is where we want the ball to be. The ball will then settle there without our help. More generally, we want to change the stable points of the system itself, if possible, so that it naturally reaches a state that is more in line with our desired outcomes.

Good cycling infrastructure may shift the stable states of how much people drive. One example of shifting stable states is the construction of cycling infrastructure in the Netherlands in the 1970s. As cars became cheaper during the 20th century, the number of people who owned them began to rise in many countries, including the Netherlands. Alongside this, the number of car accidents also increased. In the 1970s, a protest movement gathered in response to rising numbers of children being killed by cars. The campaign succeeded in convincing Dutch politicians to build extensive cycling infrastructure to encourage people to travel by bike instead of by car. This has had positive, lasting results. A 2018 report stated that around 27% of all trips in the Netherlands are made by bike—a higher proportion than any other country studied [289].

Instead of making rules to try to limit how much people drive, creating appropriate infrastructure makes cycling safer and easier. Additionally, well-planned cycle networks can make many routes quicker by bike than by car, making this option more convenient. Under these conditions, people will naturally be more inclined to cycle, so society naturally drifts toward a stable point that entails less driving.

It is worth noting that the Netherlands' success might not be possible to replicate everywhere, as there may be other factors involved. For instance, the terrain in the Netherlands is relatively flat compared with other countries, and hilly terrain might dissuade people from cycling. This illustrates that some factors influencing the stable points are beyond our control. Nevertheless, this approach has likely been more effective in the Netherlands than simple rules limiting driving would have been. There might also be other effective strategies for changing the stable points of how much people drive, such as creating cheap, reliable public transport systems.

Summary. Complex systems can often self-organize into stable states that we may consider undesirable, and which create some kind of environmental or social problem. However, if we try to solve the problem too simplistically by trying to pull the system away from its stable state, we might expect some restoring forces to circumvent our intervention and bring the system back to its stable state, or an even worse one. A more effective approach might be to change certain underlying conditions within a system, where possible, to create new, more desirable stable states for the system to self-organize toward.

Successful Interventions

We have discussed several examples of failed interventions in complex systems. While it can be difficult to say definitively a wicked problem has been solved, there are some examples of interventions that have clearly been at least partially successful. We will now look at some of these examples.

Eradication of Smallpox. In 1967, the WHO launched an intensive campaign against smallpox, involving intensive global vaccination programs and close monitoring and containment of outbreaks. In 1980, the WHO declared that smallpox had been eradicated. This was an enormous feat that required concerted international efforts over more than a decade.

Reversal of the depletion of the ozone layer. Toward the end of the 20th century it was discovered that certain compounds frequently used in spray cans, refrigerators, and air conditioners, were reaching the ozone layer and depleting it, leading to more harmful radiation passing through. As a result, the Montreal Protocol, an international agreement to phase out the use of these compounds, was negotiated in 1987 and enacted soon after. It has been reported that the ozone layer has started to recover since then.

Public health campaigns against smoking. In the 20th century, scientists discovered a causal relationship between tobacco smoking and lung cancer. In the following decades, governments started implementing various measures to discourage people from smoking. Initiatives have included health warnings on cigarette packets, smoking bans in certain public areas, and programs supporting people through the process of quitting. Many of these measures have successfully raised public awareness of health risks and contributed to declining smoking rates in several countries.

While these examples show that it is possible to address wicked problems, they also demonstrate some of the difficulties involved. All these interventions have required enormous, sustained efforts over many years, and some have involved coordination on a global scale. It is worth noting that smallpox is the only infectious disease that has ever been eradicated. One challenge in replicating this success elsewhere is that some viruses, such as influenza viruses, evolve rapidly to evade vaccine-induced immunity. This highlights how unique each wicked problem is.

Campaigns to dissuade people from smoking have faced pushback from the tobacco industry, showing how conflicting incentives in complex systems can hamper attempts to solve wicked problems. Additionally, as is often the case with wicked problems, we may never be able to say that smoking is fully "solved"; it might not be feasible to reach a situation where no one smokes at all. Nonetheless, much positive progress has been made in tackling this issue.

Summary. Although it is by no means straightforward to tackle wicked problems, there are some examples of interventions that have successfully solved or made great strides toward solving certain wicked problems. For many wicked problems, it may never be possible to say that they have been fully solved, but it is nonetheless possible to make progress and improve the situation.

5.3.4 Systemic Issues

We have discussed the characteristics of wicked problems as stemming from the complex systems they arise from, and explored why they are so difficult to tackle. We have also looked at some examples of failed attempts to solve wicked problems, as well as examples of more successful ones, and explored the idea of shifting stable points, instead of just trying to pull a system away from its stable points. We will now discuss ways of thinking more holistically and identifying more effective, system-level solutions.

Obvious problems are sometimes just symptoms of broader systemic issues. It can be tempting to take action at the level of the obvious, tangible problem, but this is sometimes like applying a band-aid. If there is a broader underlying issue, then trying to fix the problem directly might only work temporarily, and more problems might continue to crop up.

We should think about the function we are trying to achieve and the system we are using. One method of finding more effective solutions is to "zoom out" and consider the situation holistically. In complex systems language, we might say that we need to find the correct scale at which to analyze the situation. This might involve thinking carefully about what we are trying to achieve and whether individuals or groups in the system exhibit the behaviors we are trying to control. We should consider whether, if we solve the immediate problem, another one might be likely to arise soon after.

It might be more fruitful to change AI research culture than to address individual issues. One approach to AI safety might be to address issues with individual products as they come up. This approach would be focused on the level of the problem. However, if issues keep arising, it could be a sign of broader underlying issues with how research is being done. It might therefore be better to influence the culture around AI research and development, instead of focusing on individual risks. If multiple organizations developing AI technology are in an arms race with one another, for example, they will be trying to reach goals and release products as quickly as possible. This will likely compel people to cut corners, perhaps by omitting safety measures. Reducing these competitive pressures might therefore significantly reduce overall risk, albeit less directly.

If competitive pressures remain high, we could imagine a potential future scenario in which a serious AI-related safety issue materializes and causes considerable harm. In explaining this accident, people might focus on the exact series of events that led to it—which product was involved, who developed it, and what precisely went wrong. However, ignoring the role of competitive pressures would be an oversight. We can illustrate this difference in mindset more clearly by looking at historical examples.

We can explain catastrophes by looking for a "root cause" or looking at systemic factors. There are usually two ways of interpreting a catastrophe. We can either look for a traceable series of events that triggered it, or we can think more

about the surrounding conditions that made it likely to happen one way or another. For instance, the first approach might say that the assassination of Franz Ferdinand caused World War One. While that event may have been the spark, international tensions were already high beforehand. If the assassination had not happened, something else might have done, also triggering a conflict. A better approach might instead invoke the imperialistic ambitions of many nations and the development of new militaristic technologies, which led nations to believe there was a strong first-strike advantage.

We can also find the contrast between these two mindsets in the different explanations put forward for the Bhopal gas tragedy, a huge leak of toxic gas that happened in December 1984 at a pesticide-producing plant in Bhopal, India. The disaster caused thousands of deaths and injured up to half a million people. A "root cause" explanation blames workers for allowing water to get into some of the pipes, where it set off an uncontrolled reaction with other chemicals that escalated to catastrophe. However, a more holistic view focuses on the slipping safety standards in the run-up to the event, during which management failed to adequately maintain safety systems and ensure that employees were properly trained. According to this view, an accident was bound to happen as a result of these factors, regardless of the specific way in which it started.

To improve safety in complex systems, we should focus on general systemic factors. Both examples above took place in complex systems; the network of changing relationships between nations constitutes a complex evolving system, as does the system of operations in a large industrial facility. As we have discussed, complex systems are difficult to predict and we cannot analyze and guard against every possible way in which something might go wrong. Trying to change the broad systemic factors to influence a system's general safety may be much more effective. In the development of technology, including AI, competitive pressures are one important systemic risk source. Others include regulations, public concern, safety costs, and safety culture. We will discuss these and other systemic factors in more depth in the Safety Engineering chapter.

Summary. Instead of just focusing on the most obvious, surface-level problem, we should also consider what function we are trying to achieve, the system we are using, and whether the problem might be a result of a mismatch between the system and our goal. Thinking in this way can help us identify systemic factors underlying the problems and ways of changing them so that the system is better suited to achieving our aims.

5.4 CONCLUSION

In this chapter, we have explored the properties of complex systems and their implications for AI safety strategies. We began by contrasting simple systems with complex systems. While the former can be understood as the sum of their parts, the latter

display emergent properties that arise from complex interactions. These properties do not exist in any of the components in isolation and cannot easily be derived from reductive analysis of the system.

Next, we explored seven salient hallmarks of complexity. We saw that feedback loops are ubiquitous in complex systems and often lead to nonlinearity, where a small change in the input to a system does not result in a proportionate change in the output. Rather, fluctuations can be amplified or quashed by feedback loops. Furthermore, these processes can make a system highly sensitive to its initial conditions, meaning that a small difference at the outset can lead to vastly different long-term trajectories. This is often referred to as the "butterfly effect", and makes it difficult to predict the behaviors of complex systems.

We also discussed how the components of complex systems tend to self-organize to some extent and how they often display critical points, at which a small fluctuation can tip the system into a drastically different state. We then looked at distributed functionality, which refers to how tasks are loosely shared among components in a complex system, and scalable structure, which gives rise to power laws within complex systems. The final hallmark of complexity we discussed was adaptive behavior, which allows systems to continue functioning in a changing environment.

Along the way, we highlighted how deep learning systems exhibit the hallmarks of complexity. Beyond AIs themselves, we also showed how the social systems they exist within are also best understood as complex systems, through the worked examples of corporations and research institutes, political systems, and advocacy organizations.

Having established the presence of complexity in AIs and the systems surrounding them, we looked at what this means for AI safety by looking at five general lessons. Since we cannot usually predict all emergent properties of complex systems simply through theoretical analysis, some trial and error is likely to be required in making AI systems safe. It is also important to be aware that systems often break down goals into subgoals, which can supersede the original goal, meaning that AIs may not always pursue the goals we give them.

Due to the potential for emergent properties, we cannot guarantee that a safe system will remain safe when it is scaled up. However, since we cannot usually understand complex systems perfectly in theory, it is extremely difficult to build a flawless complex system from scratch. This means that starting with small systems that are safe and scaling them up cautiously is likely the most promising approach to building large complex systems that are safe. The final general lesson is that we cannot guarantee AI safety by keeping humans in the loop, so we need to design systems with this in mind.

Next, we looked at how complex systems often give rise to wicked problems, which cannot be solved in the same way we would approach a simple mathematics question or a puzzle. We saw how difficult it is to address wicked problems, due to the unexpected side effects that can occur when we interfere with complex systems. However, we also explored examples of successful interventions, showing that it is possible to make significant progress, even if we cannot fully solve a problem. In thinking about

the most effective interventions, we highlighted the importance of thinking holistically and looking for system-level solutions.

AI safety is not a mathematical puzzle that can be solved once and for all. Rather, it is a wicked problem that is likely to require ongoing, coordinated efforts, and flexible strategies that can be adapted to changing circumstances.

5.5 LITERATURE

5.5.1 Recommended Reading

- M. Gell-Mann. *The Quark and the Jaguar: Adventures in the Simple and the Complex*. St. Martin's Publishing Group, 1995. ISBN: 9780805072532. URL: https://books.google.com/books?id=l6aCe4zqZ˙sC
- D.H. Meadows and D. Wright. *Thinking in Systems: A Primer*. Chelsea Green Publishing, 2008. ISBN: 9781603580557. URL: https://books.google.com.au/books?id=CpbLAgAAQBAJ
- John Gall. *The systems bible: the beginner's guide to systems large and small*. General Systemantics Press, 2002
- Richard Cook. *How complex systems fail*. Jan. 2002. URL: https://how.complexsystems.fail