8

FOUR CONCLUDING THOUGHTS

This book does not aim to be a comprehensive treatment of all issues relevant to AI interpretability. That would require much more than what we have provided here. We have tried to focus on a small subset of issues that is relatively self-contained: how metasemantic work in the externalist tradition can be used to create models for AI interpretability. In this final chapter, we will end with four scattered thoughts that we hope will illuminate and develop some of the ideas in the previous chapters:

1. The first issue we address is an important direction for further work: philosophers need to engage in more detail with the fact that AI systems have certain kinds of dynamic goals.
2. Second, we explore what happens when someone sympathetic to the views in this book also endorses a version of Clark and Chalmers' thesis of the extended mind.
3. We revisit an objection to our entire approach: that we have not sufficiently explored the idea that we should give up talk of AI systems being representational or treat such talk as a form of make-belief.

Making AI Intelligible: Philosophical Foundations. Herman Cappelen and Joshua Dever, Oxford University Press (2021).
 DOI: 10.1093/oso/9780192894724.003.0008

4. Finally, we return to the topic of explainable AI from the introduction, indicating what we think can be learnt about that movement from the metasemantic perspective taken in this book.

Dynamic Goals

The dynamic nature of neural networks gives rise to potential (and maybe actual) situations that makes the systems fundamentally uninterpretable. In the discussion above, we have conveniently ignored this feature of neural networks, but that's maybe cheating slightly. Below we give a brief outline of the sorts of issues that arise and require further investigation.

We start with a little story that illustrates what we have in mind. We should emphasize that this is not science fiction—it is, in effect, a partial feature of all neural networks. Our story is just meant to highlight something that we have not sufficiently focused on.

A Story of Neural Networks Taking Over in Ways We Cannot Understand

Suppose you've decided to build a machine learning system to help a bank run its business. You start with a very specific mandate. The bank makes many loans to individuals—some of these loans are repaid, but some are defaulted on and not repaid. The bank wants to minimize the number of defaulted loans. The difficulty, of course, lies in spotting, among loan applications, the likely repayers and the likely defaulters.

Naturally enough, you begin with a supervised learning project. The bank gives you access to thousands of actual prior loans and their eventual outcomes (repaid or defaulted). You train a machine learning system by giving it details about each loan and testing your system's classification against the real loan outcome. (Of course, what details to give it about loans will be one of the difficult points. Perhaps you begin by giving the information provided on the loan application—income, savings, and some basic demographic information. But then you discover you get better outcomes by providing more input data. Eventually, following the pattern of companies like SmartCredit, you use the full social media history of a loan applicant as part of the input data for classification.) With adequate training, your program gets very good at sorting loan applications into defaulters and non-defaulters.

After a while, though, it occurs to you that you might do better. Your current program is very good, but not perfect, at finding defaulters. It makes occasional mistakes in both directions: sometimes it flags a loan application as a likely default when in fact the applicant doesn't default (a false positive), and sometimes it doesn't flag a loan application as a likely default but the applicant does in fact default (a false negative). Both false positives and false negatives are costly. False negatives directly cost the bank through loans that aren't repaid; false positives cost the bank by denying it access to potential interest income. If your program could be perfect, eliminating all false positives and all false negatives, that would be ideal. But that's not realistic—the available data just doesn't definitively and perfectly reliably settle the outcome of every loan. Mistakes are going to happen.

Your goal so far has been to minimize mistakes. But you realize that that might not be ideal. Some mistakes are much costlier than

others. False negatives are costlier than false positives. And mistakes on large loans are costlier than mistakes on small loans. So you change the reinforcement learning pattern for your system. Now instead of just giving it a yes/no, default/no-default feedback on each loan application it evaluates, you give it a damage score feedback, telling it the amount of money that its evaluation has cost the bank, in light of the true outcome of the loan. The system is then trained to minimize damage scores.

It could well happen that the result is an increase in overall error rates in making default judgements. The machine learning system becomes sensitive to different patterns in the data, and those patterns aren't as well coordinated with the question of whether the loan will be defaulted, so it makes more mistakes of that sort. But the new patterns are well coordinated with something like *costly default*. One way to put this is that your program has changed from being a default detector to being a costly default detector. When we think of it in this way, there hasn't really been an increase in the error rate. It's true that more often now the machine learning system says 'yes' for a loan that goes on to be defaulted on, and 'no' to a loan for which there isn't any subsequent default. But those are only *errors* if we think that the program's 'yes' means 'yes, this loan is safe from default' and its 'no' means 'no, this loan isn't safe from default'. If the machine has changed, by virtue of the new reinforcement pattern, from being a default detector to being a costly default detector, then its 'yes' now means 'yes, this loan is safe from costly default', and it isn't, in fact, making more mistakes.

The revised bit of financial software is a hit—bank profits go up as costly defaults are avoided. Encouraged, you look for further such modifications in your financial detector. You have a few ideas—maybe you could train it to minimize some product of

size-of-default and low-size-of-bank-financial-reserves, or maybe you could train it to minimize loss of potential interest earnings, so that false negatives are allowed to increase when interest rates go up. But you're a programmer, not a financial wizard—you worry that while you've got a few ideas about what loan features should be detected, you might be missing important features. (Or making horrible mistakes about what features to fixate on—maybe for subtle financial reasons you don't grasp, it would be a disaster to bias towards giving out more loans when interest rates are high.)

So you have another idea. Why not just use the overall financial state of the bank as the feedback mechanism for your program? Let it experiment with accepting and rejecting loan applications in various ways, and just let it know as it accepts and rejects how the bank is doing. That way, if increasing loans when interest rates are high is a good idea for overall bank health, the program can hopefully eventually get on to that pattern. But if that's not a good idea for overall bank health, the program will avoid looking for that pattern. No need for you to use your own defective financial understanding in picking a pattern for the program to detect.

After much training, the new system goes into effect, and it's a big success. Bank profit margins, when making loans following the advice of the new system, go up sharply. The bank CEO comes by to ask you about this great new piece of software, and asks you what the program is looking for when considering a loan application. This looks like a hard question to answer. You used to know what the program was looking for—originally it was looking for loan applications with a high probability of default, and then it was looking for loan applications that it would be costly to accept. But with your final revisions, there's some important sense in

which you don't know any more what the program is looking for. Maybe you were right and it's good to accept more loan applications when interest rates are high, and thus maybe the program is now looking for some feature involving interest rates. But you can't tell easily—you'd have to look over thousands of loan recommendations by the program to see if that pattern does indeed emerge. And that's only one thing the program might be looking for; one that happened to occur to you. Who knows what other subtle patterns might be hidden in the program's decisions?

Why This Story is Disturbing and Relevant

It now looks like you're in a somewhat disturbing situation, because in some important sense:

A. you no longer know what the program is looking for, and thus you don't know what the program means when it gives a positive or a negative verdict; and
B. control over what content the program is using, what category it is testing for, has been taken out of the hands of you, the programmer, and given over to the program.

We could try to avoid this conclusion by saying that the program is investigating some high-level abstract goal. You trained the final version of the program by giving it information about the overall financial health of the bank and then asking it to approve or reject loan applications depending on whether they improved that financial health. So maybe that's what gives the content of the program's verdicts. Maybe when the program says 'yes' to an application, it is characterizing that application as 'a loan that will

improve the overall financial health of the bank' (rather than as 'a loan unlikely to be defaulted on'). More generally, can't we always just ask what you trained the program to do, by asking what kind of scoring mechanism you used for its decisions, and then just read off from that what the content of its decisions are? If that's right, the contents of the states of a program can change, but not in any mysterious way—they change only when *we* change how we evaluate program outputs.

But here are two worries about this 'high-level content' response.

First, it seems like it's missing something important to attribute only the high-level content to the program. Set aside software for a moment. Suppose the bank hires a (human) financial advisor, and asks him to figure out rules for which loan applications should and shouldn't grant in order to maximize the financial health of the bank. The financial advisor hides away in his office for a while studying volumes of data about old loan applications, and eventually declares himself ready and starts evaluating loan applications. Things go very well—the loan decisions the advisor makes are working out to the advantage of the bank. So we ask him what the method is—what feature of applications does he look for in determining which ones to accept?

If he tells us that he looks for loan applications that have the property 'will improve the financial health of the bank' (mentioning, perhaps, that that is after all exactly what we asked him to find), we will feel that he is holding out on us, and not telling us the property that he's *really* looking for. What we want to say is that he's looking for some unknown property P, and he's looking for that property *because* having that property contributes to the overarching goal of improving the bank's financial health. The

overarching goal doesn't set the content of his rules—rather, it gives the reason for his rules having the content that they do (whatever that is).

Similarly with the software. To say that it's detecting the property 'improves the financial health of the bank' seems like it's confusing what property it's detecting with why it is detecting that property. If that's right, then we *don't* know what property the program is detecting, and can't directly control what property it's detecting.

A second worry is this: the high-level content approach depends on us at least knowing what the scoring mechanism is for the program. But maybe that doesn't always happen. Suppose the financial software is designed so that in addition to changing its sorting procedures, it can also change its scoring mechanism. So *we* don't tell it to start favouring detection categories that maximize the overall financial health of the bank—it changes its own scoring mechanism to start favouring those categories. Of course, if the program isn't going to behave randomly, its own changes in scoring mechanism need to be rule-governed in some way. So perhaps the programmers give the program a second-order scoring mechanism for evaluating how well its choices of scoring mechanisms are doing. In that case there's an even-higher-level content that we could ascribe to the system: the program is detecting objects as 'being things that maximize fit with respect to some criterion that maximizes achievement of goal G', where goal G is what we've encoded in the second-order scoring mechanism.

And, of course, we can ascend another level to a third-order scoring mechanism, which lets the program pick its own second-order mechanism for assessing its own choices of first-order scoring mechanisms for assessing its own choices of classifications,

and then test that choice against our third-order criterion. Again, a very abstract higher-order characterization of the content of the machine verdicts can be given in this way, but any worries we already had about whether this abstract higher-order content is missing something important are only going to be made worse.

There's no limit to how many levels we can ascend. One limit point of this procedure has us switching over from a supervised learning software design to an unsupervised learning data mining design. The unsupervised learner starts with a kind of higher-order scoring rule, which just characterizes certain kinds of very abstract mathematical structure in the data as being good. It then looks for such structure in the data, and characterizes things in terms of that structure. Then it looks at that characterization and again looks for the desired kinds of structure in it. And so on, until, hopefully, some interesting large-scale patterns start to emerge. We might set such a data miner to work on a large history of loan applications and other financial information about the bank, and then try out using its classifications in making decisions about approving and rejecting loans. If things work out well using its classifications like this, we could then conclude that the machine is getting on to some feature worth attending to, without having any idea what that feature is, and thus without having any idea of what the program is telling us.

Taking Stock and General Lessons

It would be great to know even in these cases of complex dynamic shifting of program contents what these programs are telling us. After all, we may be handing off control over large aspects of our lives to such systems. If we're going to be denying someone a loan

to buy a house based on the output of some program, it would be nice if we could tell that person *something* about why their loan was denied, what it was about them that made them not loan-worthy. If we're going to begin an aggressive course of medical intervention on someone based on the output of some program, it would be nice if we could tell that person *something* about why that medical intervention was called for, what it was about them that was unwell or would be made better. In the limiting case, if we hand off control over judgements to machine learning systems with dynamically shifting goals that we can't understand, there may be no reason to expect that the things that we're told to do are things that we *ought* to do in any sense.

Dynamically shifting program contents, in short, give us special reasons for wanting a good story about what makes programs about the things they are about and a good story about how to find out what programs are about, but also special reasons for thinking that it may be particularly difficult to get the good stories that we want.

The Extended Mind and AI Concept Possession

Background: The Extended Mind and Active Externalism

In this book we have drawn heavily on the externalist tradition in metasemantics. It's a tradition that traces back to the work of Millikan, Kripke, Marcus, Putnam, and Burge. There is, however, another tradition that uses the term 'externalism'. In a brief but massively influential paper, Andy Clark and David Chalmers defend what they call 'active externalism'. They argue for the view

that the environment, what is found beyond the skull/bone boundary, can drive cognitive processes. Their form of externalism is one in which 'the human organism is linked with an external entity in a two-way interaction, creating a *coupled* system that can be seen as a cognitive system in its own right' (1998: 8). The result of this is a view according to which various external devices (which can include AI systems) should, under certain circumstances, be seen not just as cognitive tools, but as integral parts of human cognitive processes. They apply this view not just to cognitive processing, but also to, for example, beliefs. If a device external to skull/bone 'contains' information and an agent is appropriately related to that external device, then that information can be one of the agent's belief. For example, on the assumption that Lucie's phone is appropriately related to her, and that the phone contains the information that Nora lives in Pokfulam, then Lucie believes that Nora lives in Pokfulam, even if that information is inaccessible to her without the help of her phone. The phone, on this view, is an extension of Nora's mind, on par with the synapses and whatever else is doing work inside Nora's skull and bone. Clark and Chalmers say that this kind of view enables us to 'see ourselves more truly as creatures of the world' (1998: 18). It is a corollary of the view that we should also see the self as extended beyond skull and bone. The external tools that are parts of Lucie's mind are parts of her—they are her in the same sense as her brain or ear is her.

Clark and Chalmers emphasise that the external device plays 'an active causal role', in the following sense: the system as a whole (what's inside skull/bone + the device + the relationship between the device and what's inside skull/bone) jointly influence actions. This is, in all relevant respects, similar to what cognition usually

does: 'Our thesis is that this sort of coupled process counts equally well as a cognitive process, whether or not it is wholly in the head.'

The Extended Mind and Conceptual Competency

The kinds of externalisms we have relied on earlier in this book do not directly engage with action in the way, e.g. external electronic devices can do on Clark and Chalmers' view. For example, the distal sources of Kripkean causal chains (the dubbings) are not causing a speaker or thinker to turn left rather than right as she is walking down the street. Information on the iPhone, on the other hand, could have that kind of active impact on an agent's action. Hence the term 'active externalism'. As in the earlier part of this book, we will conditionally endorse this form of active externalism.

There has been a great deal of discussion and development of the Extended Mind Thesis and we will simply bypass that discussion. We want to focus on one potential corollary of the view that, to our knowledge, has not been extensively explored. Can the extended mind also have extended conceptual capacities? More specifically: suppose, as we have argued, that AIs can have conceptual content and conceptual competency. That view, combined with the Extended Mind Thesis, has as a corollary that we get/inherit that conceptual competency from those AIs that are part of our minds.

From Experts Determining Meaning to Artificial Intelligences Determining Meaning

There is a way into this that doesn't require appeal to the Extended Mind Thesis. Suppose you are sympathetic to Putnam or Burge's

style varieties of passive externalism. Putnam's slogan was 'Meanings ain't in the head'. That raises the question: where are they? The answer is either *nowhere* or *somewhere outside the head*. If we insist on something location-like, what we typically get is an appeal to *experts*. Experts, we are told, have the authority to determine the extension of e.g. predicates for natural kinds. In Burge's arthritis example, the community of medical experts have made it the case that the term 'arthritis' denotes ailments of the joints.

If you are on board with this view, and you are on board with our view that artificial systems could have contents, then the meanings could be located in artificial experts as well as human experts. This is a very natural move and it is independent of the endorsement of the Extended Mind Thesis. One source of objections to that view is that artificial agents don't have meanings or representational capacities. We have argued against that view. If you're on board with our arguments, the step from Putnam to AIs determining meanings isn't that radical.

Some New Distinctions: Extended Mind Internalist versus Extended Mind Externalists

Here are two interpretations of the core part of Putnam's externalism that are typically not clearly distinguished:

P1: meanings are not located inside the speakers skull/bone.
P2: meanings are not 'in' the speaker's mind.

A simplistic assumption to the effect that the mind is 'inside' the skull/bone would equate P1 and P2. If you endorse the Extended Mind Thesis, you could deny P1 and endorse P2. More generally, a

broader range of possible positions open up when thinking about meaning externalism. Here are some of those options:

- **Extended mind internalist**: meanings are located in (supervene upon) the extended mind.
- **Extended mind externalist**: meanings do not supervene on what's in the extended mind.
- **Skull/bone internalist**: meanings supervene on what's inside skull/bone.
- **Skull/bone externalist**: meaning do not supervene on what's inside skull/bone.

Note that a skull/bone externalist can endorse either extended mind internalism or extended mind internalism.

Kripke, Putnam, and Burge as Extended Mind Internalists

Classical externalists like Putnam, Burge, and Kripke all seem like they would most naturally be classified as extended mind externalists. The reason for this is that the communicative chains that Kripke appeals to don't play the same kind of active role as the various kinds of external devices that Clark and Chalmers use as their paradigms (e.g. notebooks and phones that are used to guide behaviour on a regular basis). The experts who play a meaning-constitutive role for Putnam and Burge are similarly distal.

However, on further reflection, this is not at all obvious. A lot will depend on how the relationship to external devices is understood. On that point, Clark and Chalmers are extremely open-minded—more so than is typically recognized. When

summarizing the relationship, R, that they suggest needs to obtain between an external device and an agent for that device to be part of the extended mind, they first mention four factors:

> **Constancy**: '…the notebook is a constant in Otto's life—in cases where the information in the notebook would be relevant, he will rarely take action without consulting it.' (17)
> **Ease of access**: '…the information in the notebook is directly available without difficulty.' (17)
> **Automatic endorsement**: '…upon retrieving information from the notebook he automatically endorses it.' (17)
> **Conscious endorsement in the past**: '…the information in the notebook has been consciously endorsed at some point in the past, and indeed is there as a consequence of this endorsement.' (17)

These four factors, however, are simply presented as salient generalizations of some features of the examples discussed in the paper and not given a theoretical justification. A full theory would need a justification for each of these, discussions of other options, and precisification. Clark and Chalmers are aware of this. Towards the end of the paper, their view becomes very liberal and open-ended. They say that what is part of the extended mind can be indeterminate. They say that being part of the extended mind might come in degrees: something can be a bit, but not fully, part of someone's mind. Finally, whether something is part of the extended mind could depend on context and in particular it could depend on the question under discussion. In a certain conversational setting, E might be part of A's extended mind, but in other conversational settings E might be excluded:

> In intermediate cases, the question of whether a belief is present may be indeterminate, or the answer may depend on the varying standards that are at play in various contexts in which the question might be asked. (17)

In another passage they say that other people could, in certain context, for certain purposes, when certain questions are under discussion, be part of an agent's extended mind:

> [T]he waiter at my favorite restaurant might act as a repository of my beliefs about my favorite meals (this might even be construed as a case of extended desire). In other cases, one's beliefs might be embodied in one's secretary, one's accountant, or one's collaborator. (17–18)

Putting aside Clark and Chalmers' view, it should be clear that the exact nature of the relation an external phenomenon needs to stand in to a person in order to be part of that person's extended mind is unsettled. It is unsettled not just in the sense that we know too little about it, and so haven't found the answer. It is also unsettled in that our concept of 'mind', 'belief', 'desire', 'memory', and so on are in flux. Those concepts will evolve in part with the way we interact and engage with technology. A full exploration of this would go very far beyond anything we can cover, but we end this section with a couple of conjecture/proposals:

1. For certain purposes having to do with attribution of conceptual competency, other people, e.g. experts, can be part of an agent's extended mind. Suppose the expert opinion is very easily available in a reliable way (say through a device for accessing information on the internet) and suppose the agent defers in various ways to those

experts. This doesn't look too different from the waiter or accountant case.

2. Speakers who are part of Kripkean communicative chains are in constant causal contact with that chain and, according to Kripke, intend to refer to whatever was at the beginning of the chain. The connection to the communicative chains is constant, easy, automatic, and deferential. So, when questions of conceptual competence comes up, causal communicative chains can be part of our extended mind.

If these hypotheses are correct, then Kripke, Putnam, and Burge should be classified as internalists in our new sense, i.e. they are extended mind internalists. Of course, we have made them internalists, by radically extending our notion of the internal. Rather than see the content-determining factors as factors outside the mind-determining content, we have extended the mind to include the content-determining factors.

Concept Possession, Functionalism, and Ways of Life

Here is a natural thought: the kinds of concepts we have are related to the kinds of creatures we are and our way of life. Our conceptual repertoire is, in part, determined by us being certain kinds of animals, with certain kinds of inputs (in large part determined by our perceptual capacities), and certain kinds of outputs (our actions often involve movements of our physical bodies). Functionalists pick up on this basic idea and tie contents to the kinds of inputs and outputs that are possible for us. These functional roles, that

are meaning determining, are fixed by the kinds of creatures we are and our way of life.

However, if extended internalism is true, then this is less of a limitation: we start out as certain kinds of animal with certain input and output capacities. Then we extend ourselves using, for example, artificial intelligence. This extension means that the range of contents we can entertain is extended because our possible input and output functions have been extended. This is because what we are has changed and our way of life has changed, as a result of ourselves being extended.

Implications for the View Defended in This Book

The strategy in this book has been to start with anthropocentric views in metasemantics, do some de-anthropocentrizing, and then try to apply the result to artificial intelligences. The Extended Mind Thesis suggests a complimentary strategy: incorporate! We have been assuming that whatever the AIs are doing isn't what we humans are doing and so we need to find some common process at a higher level of abstraction (the abstractive sweet spot). The alternative strategy just explored thinks of those AIs as potential (at least to some degree and in some contexts) parts of our minds. If those AIs are part of our minds (or rather: their processes are cognitive processes on par with what's happening inside skull/bone), then they are part of us (in the extended sense of us) and so what they are doing is what we are doing.

The effort to understand alien content determination thus becomes an effort to understand our own extended mind's content determination. That is a very useful perspective from which to approach these issues, but the issues that need resolution will

be roughly the same as those discussed earlier in this book. Our aim will be to get a grip on how hard to understand extended parts of us (e.g. the artificially created neural network that, to some degree and in some contexts, are parts of our extended mind) determine content. Seen in that light, the project pursued in this book is an exercise in extended-self-examination.

An Objection Revisited

We return briefly to an important thought that was behind some of Alfred's objections in Chapter 2. He was resisting the idea that questions about content had significance for his work in AI. We managed to persuade Alfred to take an interest in some of the philosophical issues we have outlined above, but of course, the way we wrote and ended that dialogue was self-serving: we gave ourselves what we needed. In some sense, we didn't do Alfred (or Alfred's position) justice. In particular, we didn't help him articulate an alternative to the content-focused picture that we have been pushing throughout. The objection we will now briefly address is this: Alfred should have focused on the notion of evidence or reliability—that's the alternative to a content-driven approach.

Here's a way to articulate that alternative:

> *The No-Content-Just-Evidence view:* Once StopSignDetector has been thoroughly trained on an initial sample of photographs pre-labelled as stop signs or not, we should then take the output of StopSignDetector as *evidence* that something is a stop sign, without thinking of StopSignDetector as having outputs whose *content* involves stop signs. StopSignDetector is, if nothing else, a *reliable detector* of stop signs. It is reasonable for us to form beliefs on the

basis of the outputs of reliable detectors. Reliable systems can serve as a form of evidence.

This view should be explored. It's an interesting alternative to the strategy defended in this book. So far, our aim has not been on developing direct objections to the No-Content-Just-Evidence view, but rather to make an indirect case against it by developing various pro-content alternatives. Those advocating for the No-Content-Just-Evidence should do the same: develop positive models that integrate theory of evidence and reliability with the nature and use of AI. Note that this is again a fundamentally philosophical project: it places philosophy at the centre of an understanding of AI. Those trained in computer programming, for example, are not trained to think about the nature of evidence, reliability, and how to apply theories of these phenomena to the output and use of AI. A defence of No-Content-Just-Evidence view would involve a shift in focus from the metaphysics of content to the theory of evidence and reliability. For some initial literature on this, see Kelly (2014) and references therein.

Reply to the Objection

While we welcome an exploration of the No-Content-Just-Evidence view, we are sceptical. We think it faces some serious obstacles, and in the next couple of pages we briefly outline some of these.

What Makes it a Stop Sign Detector?

The Evidence view, as we articulated it above, assumed that the system in question was a reliable stop sign detector. It is not,

however, clear that we are entitled to that assumption. In some important sense, we have no idea what the mechanism is by which StopSignDetector reacts as it does. All we really know is that StopSignDetector has some unbelievably complicated neural network, with connections and connection weights developed over millions of rounds of training, that somehow or other filter through the incoming data from a photograph (initially presented in some data form or other—an array of pixel values, for example) to work out activation levels culminating in the light blinking or not.

The best we can say is that there is some structural property or other of photographs that StopSignDetector is *really detecting*. That structural property presumably is some enormously complicated property about hugely computationally demanding relations among many different aspects of the incoming numerically given data—almost certainly a property that no human mind could ever really grasp, and possibly a property that there isn't even a way to express in a human language. Call that structural property S. (Now there's a way to express it in our language!) If the existence of S is helping us decide that StopSignDetector is genuinely disposed to react to stop sign pictures, then we must have some reason to think that S and stop signs are reliably correlated—that, in general, when we get a new photograph of a stop sign, it's probably going to have property S.

But why would we think that? The training of StopSignDetector doesn't look like it gives us a good reason to accept it. Here's what we learn from the training. There are two big piles of test cases that StopSignDetector was trained on—call them the positive cases and the negative cases. Property S, whatever it is, must then be a property that most of the positive cases have and most of the

negative cases don't have.[1] What reason, then, do we have to think that the particular property S that StopSignDetector got hooked onto is reliably correlated with stop signs?

Adversarial Perturbations

We don't need to rely on abstract theoretical considerations like these. There is a growing body of work on adversarial perturbations (see, for example, Goodfellow et al 2014) and their impact on machine learning image recognition systems. Adversarial perturbations provide methods of making small alterations in images that result in a machine learning system going from a very high success rate in classification to a very low success rate. Adversarial perturbations can involve small alterations in the photograph that don't significantly impact human identification abilities—adding a few bits of coloured tape here and there to the object to be identified, for example, or slightly rotating the angle of photograph. Or they can involve adding a masking layer of pixels over the original photograph that the human eye can't even detect, but that causes massive misclassification by the machine learning system.

Note that to say that adversarial perturbations cause *misclassification* isn't really right, in the current context. It's a misclassification only if the system was really (for example) a stop sign detector, so that the adversarial perturbation is causing the system to get

[1] If there were any doubt about this, note that two different machine learning systems can be trained on the same data, using neural network systems with a stochastic element, and go on to make slightly different distinctions among new cases. That shows that they are really detecting different underlying structural properties. And not necessarily slightly different such properties—the detected structural properties could be radically different from one another, but have only slightly different distributions among the cases we've tested so far.

things *wrong* by saying that things that really are stop signs aren't stop signs. But that way of putting things is loaded up with content-based talk about what the machine really detects and says, and the current dispositional line is meant to be a replacement for that talk. From the current perspective, what's going on is that adversarial perturbations are revealing what structural property S the machine is really tracking.

The worry, then, is that if StopSignDetector can be easily made to blink for non-stop-signs, or not to blink for stop signs, through using some adversarial perturbation that a human classifier would never even notice, then it's not clear that StopSignDetector is really disposed to blink when presented with a stop sign. The adversarial perturbation brings out the possibility of property S and the property of being a stop sign coming apart.

The general lesson: lots of things are statistically correlated with lots of other things. Dispositions require more than that. Dispositions require that the correlations be *reliable*, so that new cases will continue the correlation. When there's some underlying causal structure that created the correlation, we have a reason to think that it's reliable, even if we don't fully understand how the causal connection works. But in the machine learning case, we don't have any reason to think that there is a causal connection to support the disposition. That's because we do have reason to think that there's a different causal connection (one we don't fully understand) between some obscure structural property S and the machine learning outputs. Because we know there's that causal connection, it trumps the possibility of the causal connection we really want (but aren't getting). So we have to fall back on the coincidental convergence of weird structural properties and our target properties on the training cases—but the

coincidental convergence doesn't give us reason to treat the system as reliable for new cases.

This is in no way a conclusive argument to the effect that the No-Evidence view could work. It is, however, conclusive evidence that doing so is far from trivial. It requires deep engagement with philosophy. The final view will rely on a theory of what dispositions are, what reliability is, and the connection between evidence and reliability. We'll end this brief reply with a suggestive conjecture: When you have a satisfactory theory of that kind—one that responds to all these concerns—you have in effect come very close to constructing a theory of content again. According to this conjecture, the Evidence and the Content strategies will merge.

Explainable AI and Metasemantics

In the first chapter, we connected the topics of this book to the issues that come up in connection with the aim of achieving so-called explainable AI. 'Explainable AI' indicates a desire to ensure that decisions and other kinds of input made by AIs are not just handed down to us as from an oracle. If an AI system tells us that Lucie should not get a mortgage, she is entitled to understand why she should not get a mortgage. To answer the why-question by simply insisting that the decision was made by a reliable but incomprehensible algorithm isn't good enough. Lucie should also be able to understand why without having to perform the inhuman task of working through all the calculations made by an extremely complex neural network. She is entitled to receive a justifying reason for the rejection.

Even if everything we have said so far in this book is along the right track, we haven't succeeded in giving Lucie a procedure for getting a justification from SmartCredit. If we have succeeded, we have shown how SmartCredit can say that *Lucie is high risk*. That leaves us far short of getting a justification for *why* she is high risk. We have revealed nothing of the internal reasoning that might have gone into producing that output.

So did we engage in false advertising when we raised the prospect of illuminating explainable AI? Not really. We didn't claim we were going to show how explainable AI is possible. We did claim that our work could contribute to an understanding of how explainable AI could be possible. Here is how we see the connection:

1. Without content, there are no reasons. Reasons are things with content.[2] The natural way to think about the

[2] Actually, this is a bit controversial. While this book is not the place to get to grip with the vast and ever increasing literature on reasons, we'd just like to make clear that the above is a vast simplification that elides many important distinctions which any proper treatment of the topic will need to make room for. For example, a much-discussed question in the theory of reasons is taxonomic: how many different types of reason are there? To take an example from Alvarez's (2016) overview on the topic, reducing child obesity might be a reason for the government to tax sugary drinks in one sense, but a perfectly fine answer as to for what reason the government *in fact* taxed drinks in the winter of 2019 is because after the election the legislative body became filled with people who owned shares in bottled water companies. Roughly, the former would be a normative reason (something counting in favour of something in the abstract) while the latter a motivating one (something that in fact brought about a particular course in action). For a bit more on the distinction, see the opening pages of Dancy (2000).

Equally important are questions about the ontology of reasons. We've assumed they are items with representational content. It's the subject of further work exactly how that will work in the AI setting, because at least some of the literature has it that some reasons (normative ones) are non-representational entities like facts (Raz 1975 and Scanlon 1998). Moreover, a popular view about motivating reasons is that they are mental states (e.g. Audi 2001 and Mele 2003), and though this doesn't immediately cause a problem for us, there might be

explainability desideratum is this: the AI says something—e.g. that Lucie is high risk. Then Lucie is entitled to a justification of that claim. For that entitlement to make sense, the system must have said something (namely that Lucie is high risk). If there's no saying, there's nothing to justify. Moreover, the reasons themselves are contentful. So we need content both to have something to justify and in order to have something that can do the justifying. Here is what we have done: we have provided a strategy for establishing that the system can perform sayings. In so doing, we have shown how to take the first step towards explainability.

2. Explainability requires not just the generic possibility of content attribution, but also a procedure for determining specific contents. We need to know exactly what the system said before we can ask it to give a reason for what it said. The most central claim in this book is that such content cannot be found by looking at the internal computational structure of the system. It can only be found by looking at external factors of the kinds that the externalist tradition in metasemantics appeals to. It is hard to overemphasize this point. The story about AI content is not substantially different from the human story: in neither case do we find content by looking at internal computational architecture.

questions to be asked about how AI, presumably lacking mental states, can be present in the space of reasons. As mentioned, here isn't the place (and we aren't the authors) to decide these issues. We are just flagging some important distinctions an acceptable philosophical treatment of explainability will need to grapple with.

We end with a brief explanation of what an account of reason giving (and justification) for a decision (or output) would require. First a reminder of some issues that would have to be resolved in order to present such a theory.

- The philosophical tradition distinguishes between at least three kinds of reasons for actions: normative reasons, motivating reasons, and explanatory reasons.[3] How to characterize each of these is a matter of ongoing dispute. So a first question to be settled is whether it is any of these is what we are looking for. Should we, for example, be modelling AI explainability on explanatory or motivating reasons in humans? If the answer to either question is yes, then a theory of explainable AI could incorporate an existing theory of motivating or explanatory reasons.
- Alternatively, a theory of AI explainability could develop a new such theory, maybe by engaging in some form of de-anthropocentrizing, on analogy with what we have done for metasemantics in this book.

[3] Often motivating and explanatory reasons are often classified together. We are sympathetic to those who prefer to keep them apart. Maria Alvarez in the SEP entry 'Reasons for Actions' nicely summarizes one argument for the distinction: 'The fact that John knows that Peter has betrayed him is a reason that explains John's action. This is an explanatory reason. But that fact about John's mental state of knowledge is not the reason for which John punches Peter. That reason is a fact about Peter, namely that he has betrayed John. That is the reason that motivates John to punch Peter—his motivating reason. So in this case we have two different (though related) reasons: that Peter has betrayed John and that John knows that Peter has betrayed him, which play different roles. One reason motivates John to punch Peter (the betrayal); and the other explains why he does it (the knowledge of the betrayal)' (Alvarez 2016: section 3).

Back to the human case: we humans have asked for and provided reasons (and justifications) for a very long time. It is an activity that is at the core of how humans relate to each other. That is in part why it is something we care about in connection with AIs. Here is a basic fact about the cluster of activities that we call 'giving reasons' or 'explaining' or 'justifying':

> **Important and indisputable fact about human explainability**: Humans have been able to explain and justify their own (and others) actions/decisions without relying on any knowledge of the internal structure of the neural network that constitutes (part of) their brains. Human explainability succeeds in the absence of any knowledge about the internal computational structure of the human brain.

What this tells us is that knowledge of internal computational structure is unnecessary for explainability. It is, however, exceedingly tempting to conclude also more broadly that such knowledge is irrelevant to explainability. If so, it's a mistake to approach the goal of explainable AI by careful investigation into the computational structure of the AI's neural network. That kind of internalism is bound to fail and it will never lead to the space of reasons.