

# Factored Cognition

1. [A guide to Iterated Amplification & Debate](#)
2. [Hiding Complexity](#)
3. [Preface to the Sequence on Factored Cognition](#)
4. [Idealized Factored Cognition](#)
5. [Traversing a Cognition Space](#)
6. [Clarifying Factored Cognition](#)
7. [Intuition](#)
8. [FC final: Can Factored Cognition schemes scale?](#)

# A guide to Iterated Amplification & Debate

Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

This post is about two proposals for aligning AI systems in a scalable way:

- Iterated Distillation and Amplification (often just called 'Iterated Amplification'), or **IDA** for short,<sup>[1]</sup> is a proposal by [Paul Christiano](#).
- **Debate** is an IDA-inspired proposal by [Geoffrey Irving](#).

This post is written to be as easy to understand as possible, so if you found existing explanations of IDA confusing, or if you just never bothered because it seemed intimidating, this post is for you. The only prerequisite is knowing about the concept of outer alignment (and knowing about inner alignment is helpful as well). Roughly,

- Outer alignment is aligning the training signal or training data we give to our model with what we want.
- If the model we find implements its own optimization process, then inner alignment is aligning [the thing the model is optimizing for] with the training signal.

See also [this post](#) for an overview and [this paper](#) or my [ELI12 edition](#) for more details on inner alignment.

## 1. Motivation / Reframing AI Risk

Why do we need a fancy alignment scheme?

There has been [some debate](#) a few months back about whether the classical arguments of the kind made in [Superintelligence](#) for why AI is dangerous hold up to scrutiny. I think a charitable reading of the book can interpret it as primarily defending one claim, which is also an answer to the leading question. Namely,

- **It is hard to define a scalable training procedure that is not outer-misaligned.**

For example, a language model (GPT-3 style) is outer-misaligned because the objective we train for is to predict the *most likely* next word, which says nothing about being 'useful' or 'friendly'. Similarly, a question-answering system trained with Reinforcement Learning is outer-misaligned because the objective we train for is 'optimize how much the human likes the answer', not 'optimize for a true and useful answer'.

I'll refer to this claim as (\*). If (\*) true, it is a problem even under the most optimistic assumptions. For example, we can suppose that

1. progress is gradual all the way, and we can test everything before we deploy it;

2. we are likely to maintain control of AI systems (and can turn them off whenever we want to) for a while after they exceed our capabilities;
3. it takes at least another 50 years for AI to exceed human capabilities across a broad set of tasks.

Even then, (\*) remains a problem. The only way to build an outer-aligned AI system is to build an outer-aligned AI system, and we can't do it if we don't know how to do it.

In the past, people have given many examples of how outer alignment could fail (there are a lot of those in Superintelligence, and I've given two more above). But the primary reason to believe (\*) is that it has taken people a long time to come up with a formalized training scheme that is not clearly outer-misaligned. IDA and Debate are two such schemes.

If outer alignment works out, that alone is not sufficient. To solve the entire alignment problem (or even just Intent Alignment<sup>[2]</sup>), we would like to have confidence that an AI system is

1. outer-aligned; and
2. inner-aligned (or not using an inner optimizer); and
3. training competitive; and
4. performance-competitive.

Thus, IDA and Debate are a long way from having solved the entire problem, but the fact that they may be outer-aligned is reason to get excited, especially if you think the alignment problem is hard.

## 2. The Key Idea

Training AI systems requires a training signal. In some cases, this signal is easy to provide regardless of how capable the system is – for example, it is always easy to see whether a system has won a game of Go, even if the system plays at superhuman level. But most cases we care about are not of this form. For example, if an AI system makes long-term economic decisions, we only know how good the decisions are after they've been in place for years, and this is insufficient for a training signal.

In such cases, since we cannot wait to observe the full effects of a decision, any mechanism for a more rapid training signal has to involve *exercising judgment* to estimate how good the decisions are ahead of time. This is a problem once we assume that the system is more capable than we are.

To the rescue comes the following idea:

The AI system we train has to help us during training.

IDA and Debate provide two approaches to do this.

### 3. Iterated Distillation and Amplification

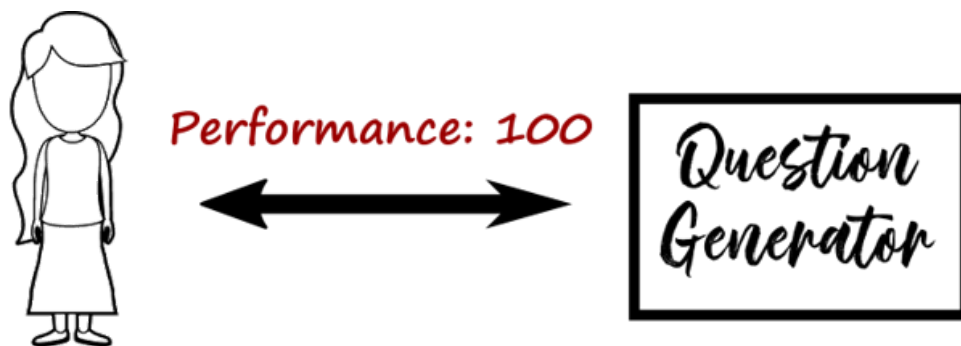
Before we begin, here are other possible resources to understand IDA:

- The LessWrong/Alignment Forum [sequence](#) (written by Paul Christiano)
  - The [very long 80k hours podcast](#) with Paul Christiano
  - The attempted [complete explanation](#) of the scheme by [Chi Nguyen](#)
  - The [FAQ](#) written by Alex Zhu
  - A [video](#) by Robert Miles (who makes lots of AI-Alignment relevant youtube content)
- 

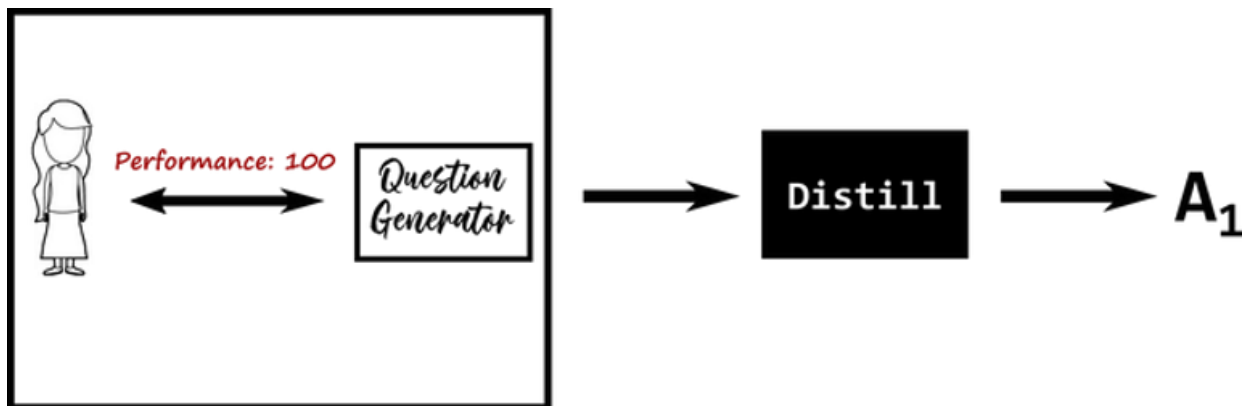
This is Hannah.



Hannah, or H for short, is a pretty smart human. In particular, she can answer questions up to some level of competence.



As a first step to realize IDA, we wish to **distill** Hannah's competence at this question-answering task into an AI system (or 'model')  $A_1$ . We assume  $A_1$  will be slightly *less* competent than Hannah, therefore Hannah can provide a safe training signal.



$A_1$  may be trained by reinforcement learning or by supervised learning of any form.<sup>[3]</sup>

The basic approach of IDA [leaves the distillation step as a black box](#), so any implementation is fine, as long as the following is true:

- Given an agent as input, we obtain a model that imitates the agent's behavior at some task but runs much faster.
- The output model is only *slightly* less competent than the input agent at this task.
- This process is *alignment-preserving*. In other words, if  $H$  is honest, then  $A_0$  should be honest as well.

If we applied  $A_1$  to the same question-answering task, it would perform worse:



However,  $A_1$  has vastly improved speed: it may answer questions in a few milliseconds that would have taken  $H$  several hours. This fact lets us boost performance through a step we call **amplification**:



In the general formulation of the IDA scheme, amplification is also a black box, but in this post, we consider the basic variant, which we call **stock IDA**. In stock IDA,

amplification is realized by giving H access to the model  $A_1$ . The idea is that this new 'agent' (consisting of H with access to  $A_1$ ) is more competent than Hannah is by herself.

If it is not obvious why, imagine you had access to a slightly dumber version of yourself that ran at 10000 times your speed. Anytime you have a (sub)-question that does not require your full intellect, you can relegate it to this slightly dumber version and obtain an answer at once. This allows you to effectively think for longer than you otherwise could.

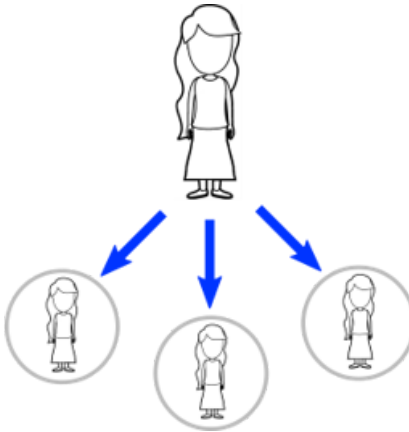
Thus, we conjecture that this combined 'agent' has improved performance (compared to H) at the same question-answering task.



Here is a different way of describing what happened. Our combined 'agent' looks like this:

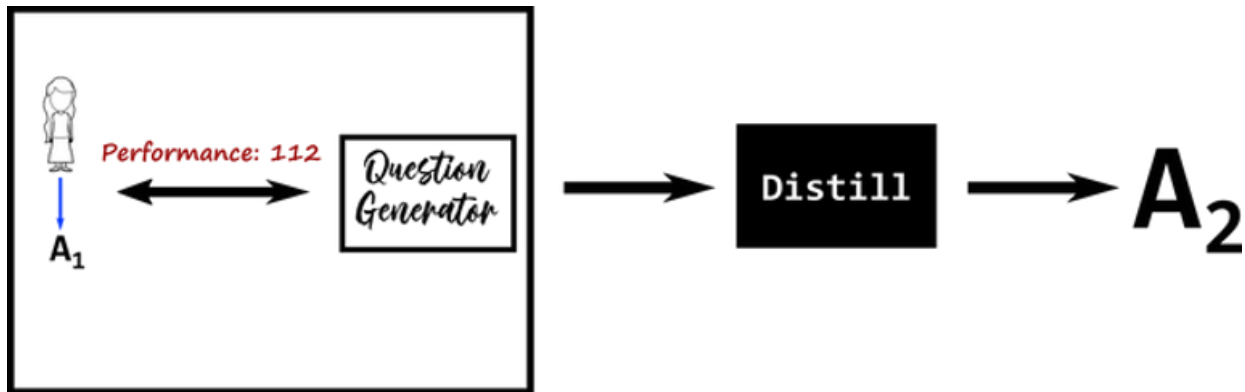


Since  $A_1$  tries to imitate H, we could think of Hannah as having access to an (imperfect) copy of herself. But since  $A_1$  thinks much faster than H, it is more accurate to view her as having access to many copies of herself, like so:



Where the gray circle means 'this is a model that tries to behave like the thing in the circle.'

At this point, we've covered one **distillation** and one **amplification** step. You might guess what happens next:



access

We train a new model  $A_2$  to imitate the agent  $[H \rightarrow A_1]$  on the question-answering

access

task. Since  $[H \rightarrow A_1]$  is more competent than  $H$ , this means that  $A_2$  will be more competent than  $A_1$  (which was trained to imitate just  $H$ ).



In this example,  $A_2$  is almost exactly as competent as  $H$ . This is a good time to mention of my performance numbers are made-up - the three properties they're meant to convey are that

- performance goes up in each amplification step; and



- performance goes down in each distillation step; but
- performance goes up in each (amplification step, distillation step) pair.

After each distillation step, we end up with some model  $A_k$ . While  $A_k$  was trained in a very particular way, it is nonetheless just a model, which can answer questions very quickly. Each  $A_k$  performs better than its predecessor  $A_{k-1}$  without a loss of speed.

The next amplification step looks like this:



Note that, in each amplification step, we always give *Hannah* access to our newest model. The  $A_k$ 's get better and better, but Hannah remains the same human.

This new 'agent' is again more competent at the question-answering task:



access

Now we could train a model  $A_3$  to imitate the behavior of  $[H \rightarrow A_2]$  on the question-answering task, which would then be less competent than the system above, but more competent than  $A_2$  (and in our case, more competent than  $H$ ). It would still be a model and thus be extremely fast. Then, we could give Hannah access to  $A_3$ , and so on.

One way to summarize this process is that we're trying to create a model that imitates the behavior of *a human with access to itself*. In particular, each model  $A_k$  imitates the

access

behavior of  $[H \rightarrow A_{k-1}]$ . Does this process top out at some point? It's conceivable (though by no means obvious) that it does not top out until  $A_k$  is superintelligent. If so,

and if distillation and amplification are both alignment-preserving, our scheme would be both aligned and performance-competitive.

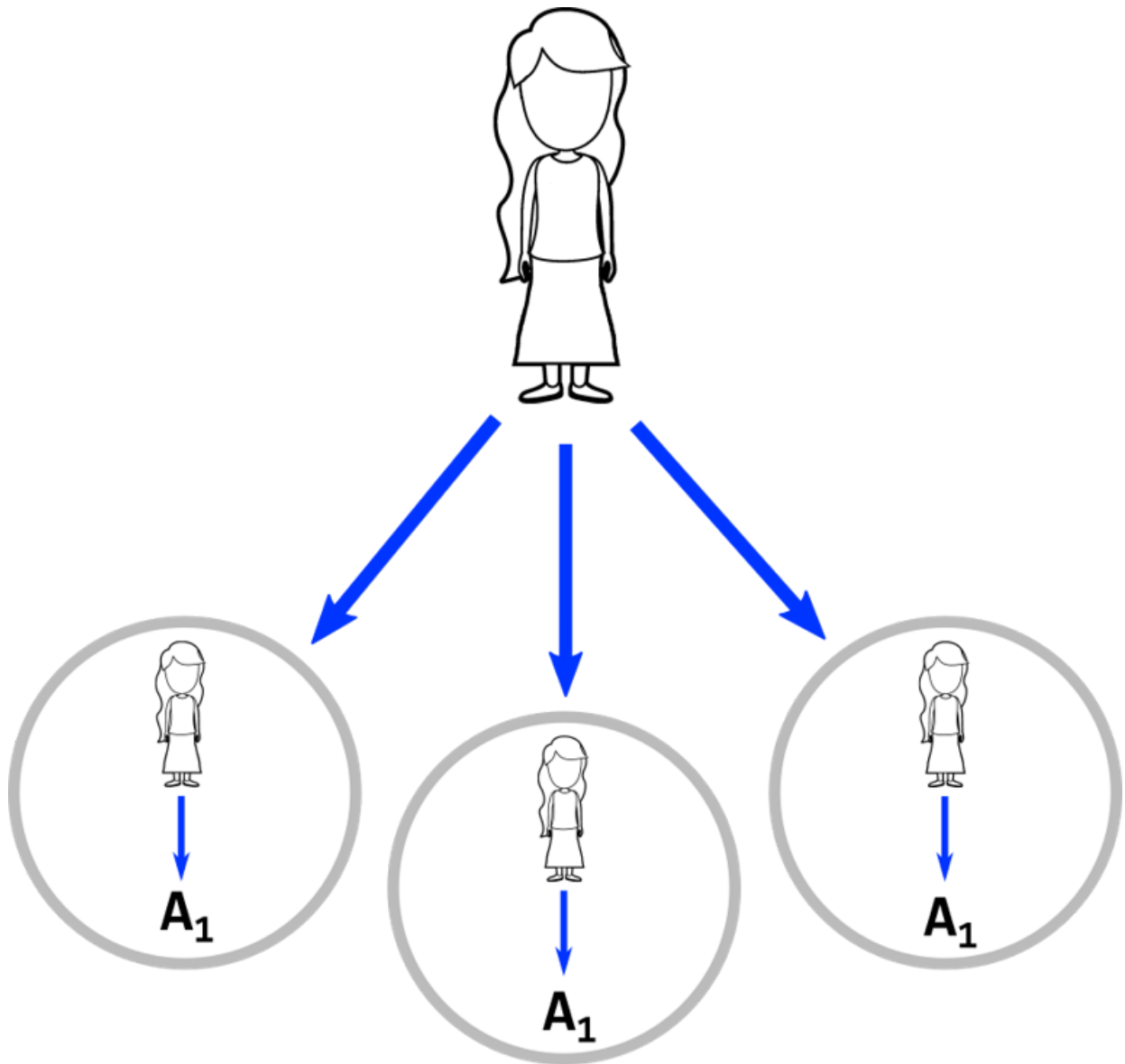
access

Recall that our 'agent'  $[H \rightarrow A_2]$  now looks like this:



access

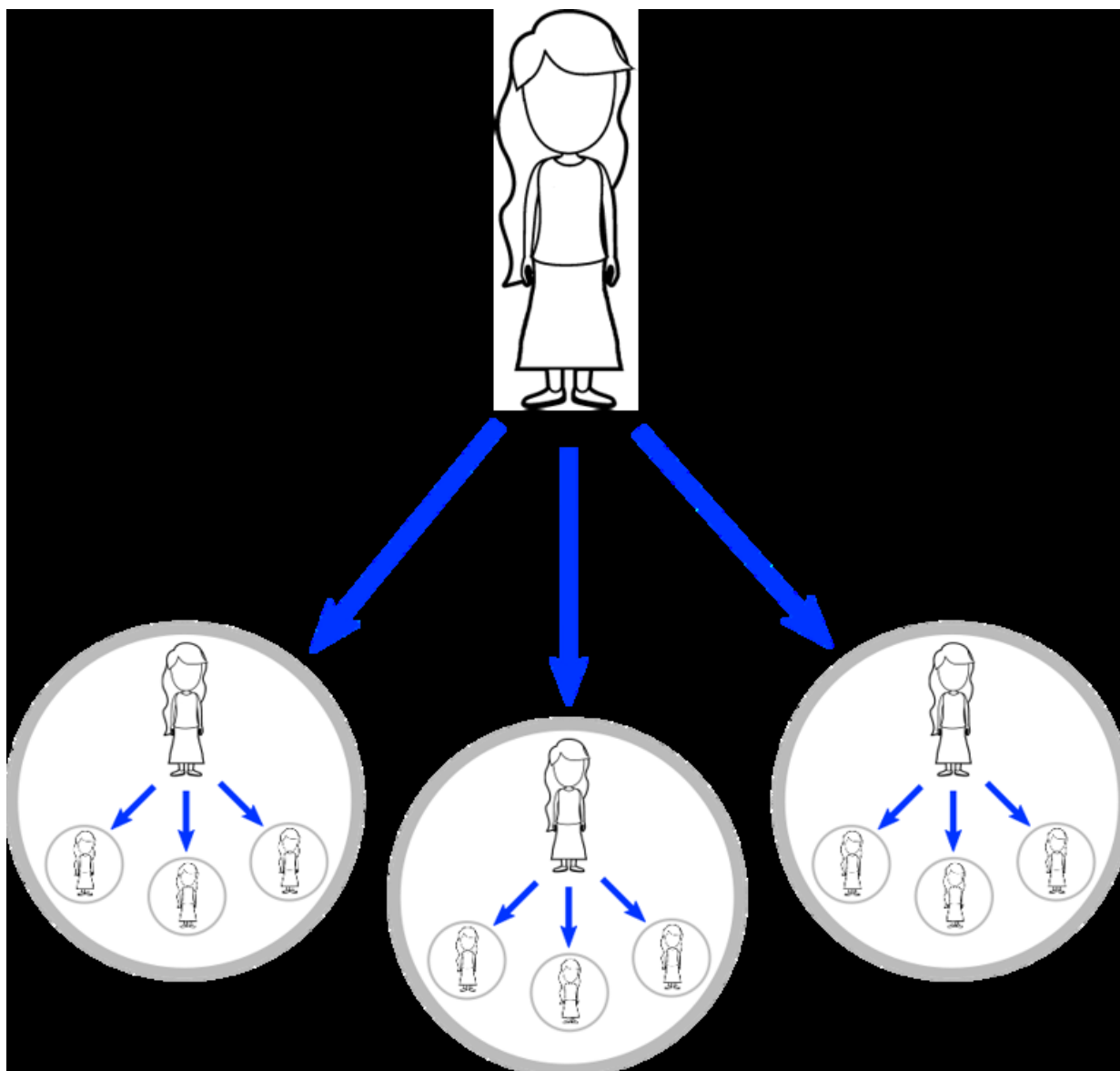
Since  $A_2$  tries to imitate  $[H \rightarrow A_1]$ , we can alternatively depict this as



access

Once again, we draw more than one of these since  $A_2$  is much faster than  $[H \rightarrow A_1]$ , so it is as if  $H$  had access to a lot of these, not just one. (Also not just three, but I only have that much space.)

Since each  $A_1$  tries to imitate  $H$ , we can depict this further like so:



Thus, insofar as the imitation step 'works' (i.e., insofar as we can ignore the circles), the resulting system will behave as if it were composed of Hannah consulting many copies of herself, each of which consulting many copies of herself. This is after precisely four steps, i.e., distillation → amplification → distillation → amplification. You can guess how it would look if we did more steps.

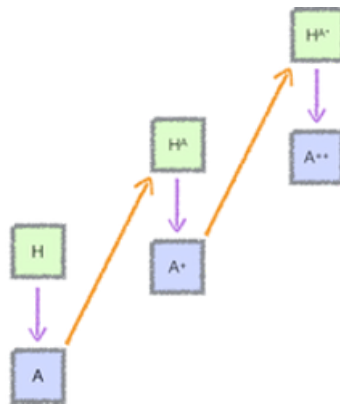
The name 'Hannah' is a bit on-the-nose as her name starts with 'H', which also stands for 'human'. Thus, the tree above consists of a human consulting humans consulting humans consulting humans consulting humans consulting humans...

We call the entire tree **HCH**,<sup>[4]</sup> which is a recursive acronym for **Humans consulting HCH**. Generally, HCH is considered to have infinite depth.

Note that there is an important caveat hidden in the clause 'insofar as the imitation step works'. In each distillation step, we are training a model to predict the answers of

a system *that thinks for much longer than itself*. Thus, each  $A_k$  is only more competent than  $A_{k-1}$  insofar as it is possible to solve problems *in less time through better algorithms*. There are strong reasons to believe that this is the case for a large class of tasks, but we know that it isn't possible for every task. For example, an HCH tree can play perfect chess (literally perfect, not just superhuman) by searching the entire chess tree.<sup>[5]</sup> A model trained by IDA cannot do the same.

In the aforementioned LessWrong sequence, the illustration for the Distillation → Amplification process looks like this:



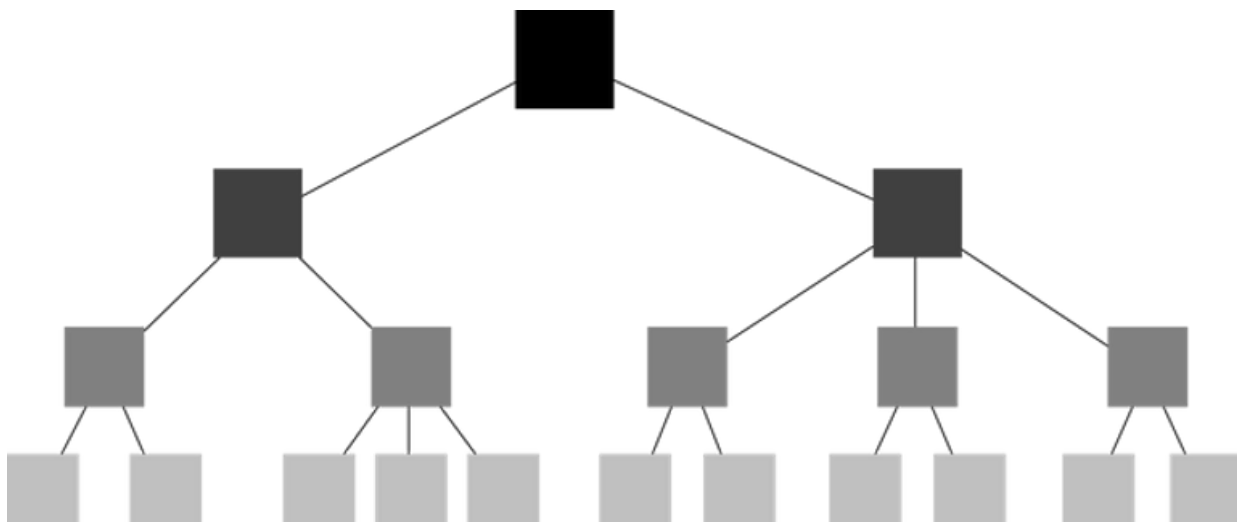
Alternatively, if we consider all of the  $A_k$ 's to be the same AI system that gets upgraded over time, we have the following (where  $r$  denotes a reward signal).



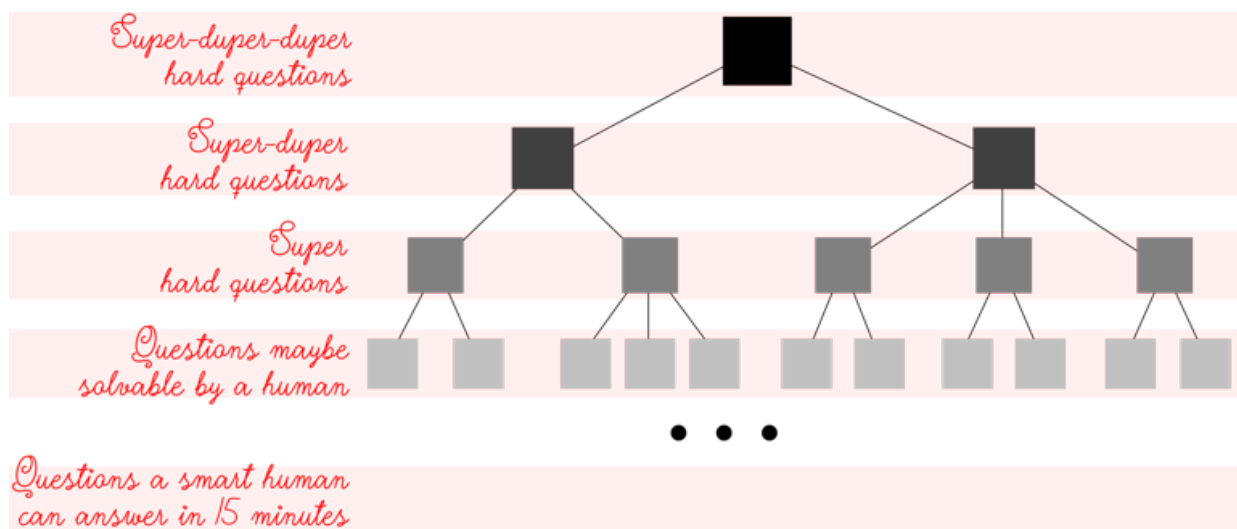
## 4. Factored Cognition

Informally, the **Factored Cognition Hypothesis** says that each question can be decomposed into easier subquestions such that the answer to the original question follows from the answer to the subquestions. Factored Cognition plays a crucial role for the applicability of both Debate and many instances of IDA.<sup>[6]</sup>

Here is an illustration, where the top block is a question, each layer below a block is a set of subquestions whose answers determine the top-level question, and darkness/size of the blocks corresponds to difficulty:



We might now hope that the absolute difficulties look something like this:



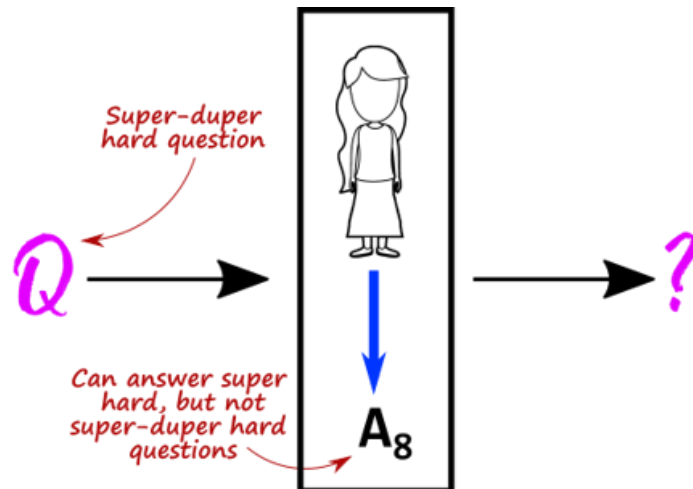
Where the lower part is meant to indicate that we can decompose all of the above questions such that they eventually bottom out in the lowest stripe of questions smart humans can answer in 15 minutes.

I see two ways to illustrate why Factored Cognition is important for stock IDA. One is the HCH picture - insofar as the imitations 'work', a model trained via stock IDA behaves just like a tree of humans consulting each other. Thus, if the model is supposed to be superintelligent, then we better hope that any question a superintelligent AI could answer can be recursively decomposed into subquestions, until we end up with something Hannah can answer by herself. (Otherwise, stock IDA may not be performance-competitive.) In other words, we better hope that the Factored Cognition Hypothesis holds.

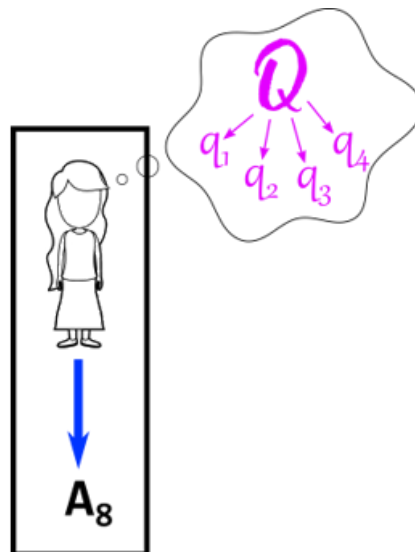
Another way is to look at just one amplification step in the procedure. Suppose that we have successfully trained model  $A_8$ , which is already smarter than  $H$ , and now want to

access

use this to create the smarter agent  $[H \rightarrow A_8]$ . Suppose that  $A_8$  is already smart enough to answer super hard questions. We want the new agent to be smarter than  $A_8$ , so we want it to be able to answer super-duper hard questions. In other words, we're in this position:



This means that, to answer this question, Hannah has to do the following:



She has to take the question  $Q$  and decompose it into subquestions  $q_1, q_2, q_3, q_4$ , such that the subquestions imply the answer to  $Q$ , and each  $q_i$  is at most super hard. Then, she can use  $A_8$  to answer the  $q_i$ , receive answers  $a_i$ , and, on their basis, output an answer  $a$  for  $Q$ .

This means that she requires the Factored Cognition Hypothesis to hold *for this particular step* (the one from super-duper hard to super hard). If the Factored Cognition Hypothesis fails for any one jump of difficulty, performance might grind to a halt at that level.

Both views point to the same phenomenon because they describe the same idea: HCH is idealized stock IDA, i.e., it is what stock IDA hopes to approximate in the limit. Both the concrete training procedure and the ideal utilize Factored Cognition.

It is also conceivable that a decomposition of the kind that Hannah needs to solve this problem does exist, but she is not smart enough to find it. This problem can be considered a motivation for Debate. <sup>[7]</sup>

## 5. Debate

*Before we begin, here are other possible resources to understand Debate:*

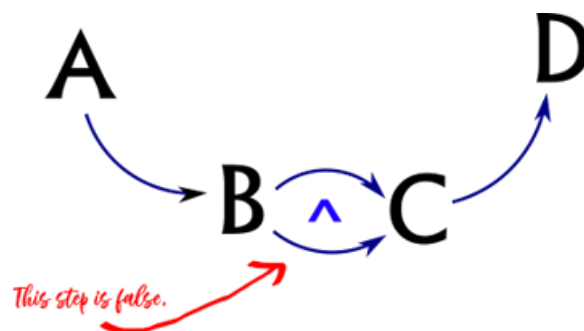
- The original [paper](#)
  - The [AI Alignment podcast episode](#) with Geoffrey Irving
- 

Suppose a smart agent X makes the following argument:



She wants to argue that D holds. Thus, she claims that A is true, that A implies B because {argument symbolized by leftmost arrow}, that B implies C because {conjunction of the arguments made by middle arrows} and that C implies D because {argument made by rightmost arrow}.

Then comes forth an equally smart agent Y to claim that

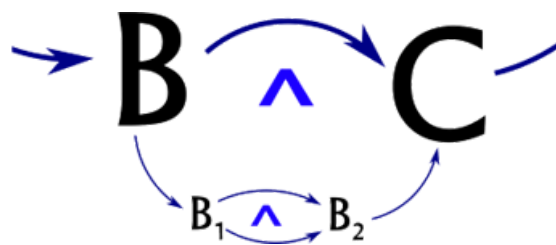




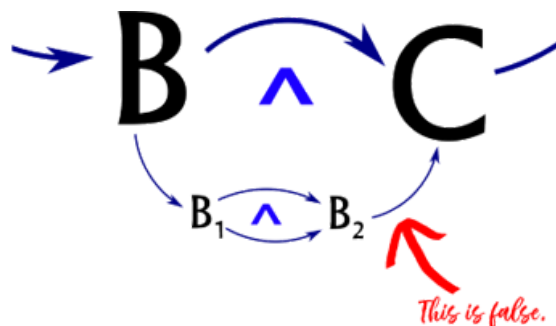
X cannot give up on the step since the entire argument depends on it, so she has to defend it. Unlike in normal debates, both X and Y now forget about the surrounding context: the steps from A to B and from C to D no longer matter (and neither does the first conjunct of the step from B to C). The remaining debate is entirely about the second conjunct of the step from B to C.

Thus, we zoom into this step. It turns out there is more going on; the step does itself has structure to it.

Then sayeth X:



Then sayeth Y:



Now, X has to defend this step, and so on. Eventually, the steps become so simple that Hannah can recognize the flaw for herself. The step from  $B_{23112}$  to  $B_{23113}$  was false; therefore the step from  $B_{2311}$  to  $B_{2312}$  was false; therefore the step from  $B_{231}$  to  $B_{232}$  was false; therefore the step from  $B_{23}$  to  $B_{24} = C$  was false; therefore the step from  $B_2$  to  $C$  was false; therefore the argument that A implies D was false. X was wrong; Y was right.

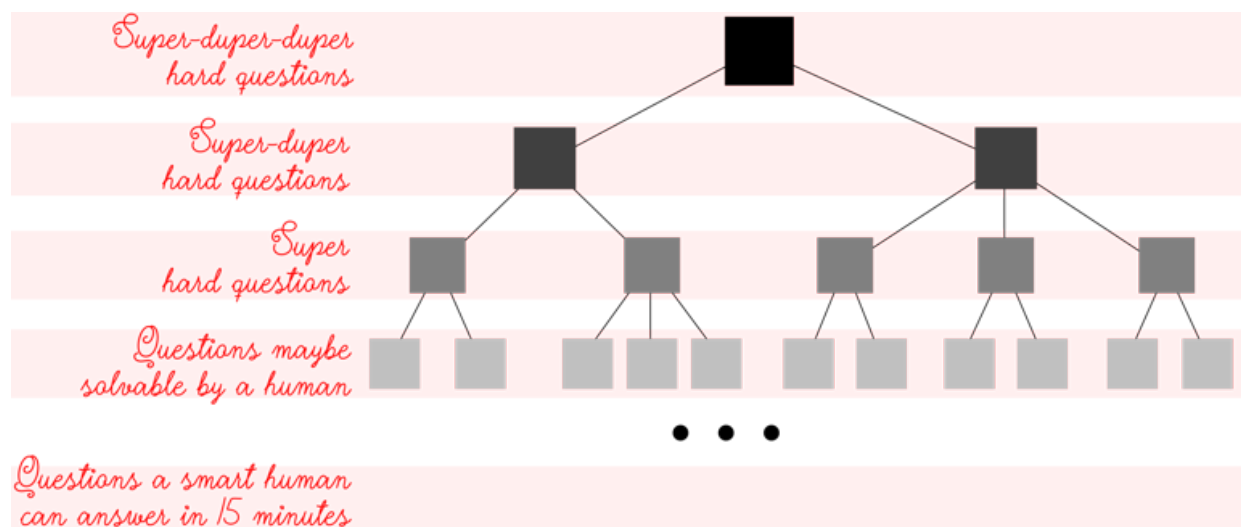
In the **Debate game**, an AI system X debates Y, an identical copy of itself. The input to the game is a question; the output is a transcript of the arguments made by X and Y. The game proceeds in rounds: first, X says something, then Y, then X, then Y, and so

on. Ideally, the first statement should always be the answer to the input question. In the end, a human  $H$  decides who won the game. In doing so,  $H$  should adhere to the following principles:

- Enforce recursive zooming into statements. If  $Y$  points to a particular part of  $X$ 's argument that she claims is wrong, and  $X$  responds by changing topics, then the game should end immediately, and  $Y$  should win.
- If a subquestion is resolved in favor of either party, always judge the entire debate based on who was right on that subquestion.

To illustrate why the second point makes sense, suppose that the debate above ends differently.  $X$  and  $Y$  still zoom into the problem until they hit step  $B_{23112} \rightarrow B_{23113}$ , but this time, Hannah thinks the step is valid. In this case, you might suspect that this is not conclusive since the argument could fail at another step. However, since  $Y$  was free to critique any part of the entire argument, chose this particular step, and failed to establish a flaw, we conclude that she could not have done better anywhere else.<sup>[8]</sup> Thus, based solely on verifying the validity of step  $B_{23112} \rightarrow B_{23113}$ , Hannah concludes that *every part of the argument was valid* and that  $X$  won the debate.

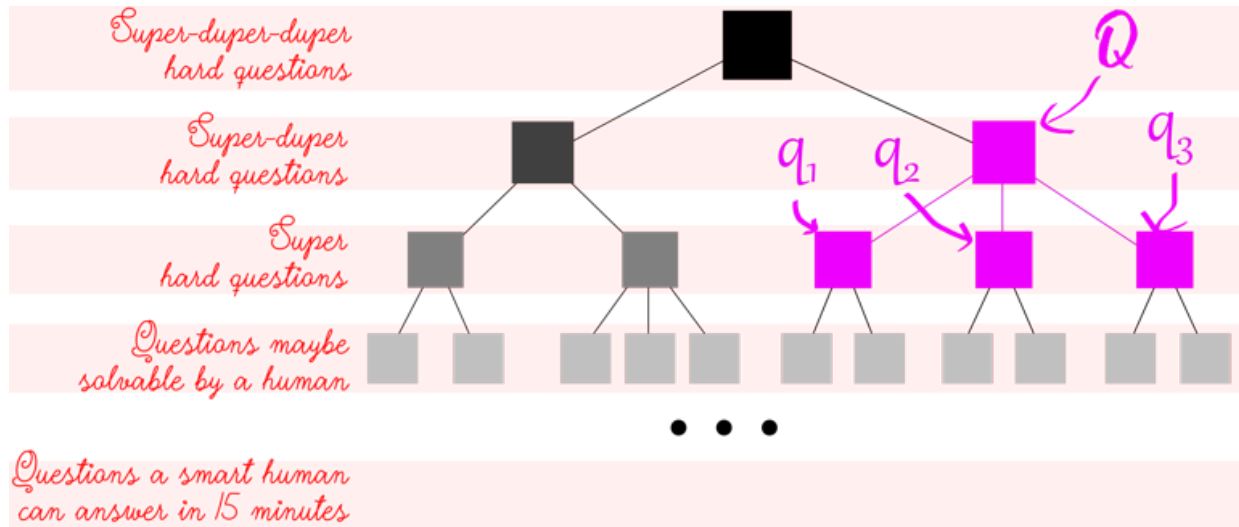
This also illustrates why Debate can be powerful. If complexity is reduced at every step (i.e.,  $B_{23112} \rightarrow B_{23113}$  is easier to verify than  $B_{2311} \rightarrow B_{2312}$  and so on), then a hard problem can be reduced to a much simpler problem, and the judge only has to verify the simple one.<sup>[9]</sup> On this note, recall the question-difficulty hierarchy from the previous chapter:



The recursive zooming into statements from Debate is closely analogous to the decomposition into subquestions from HCH. The important difference is how the tree access

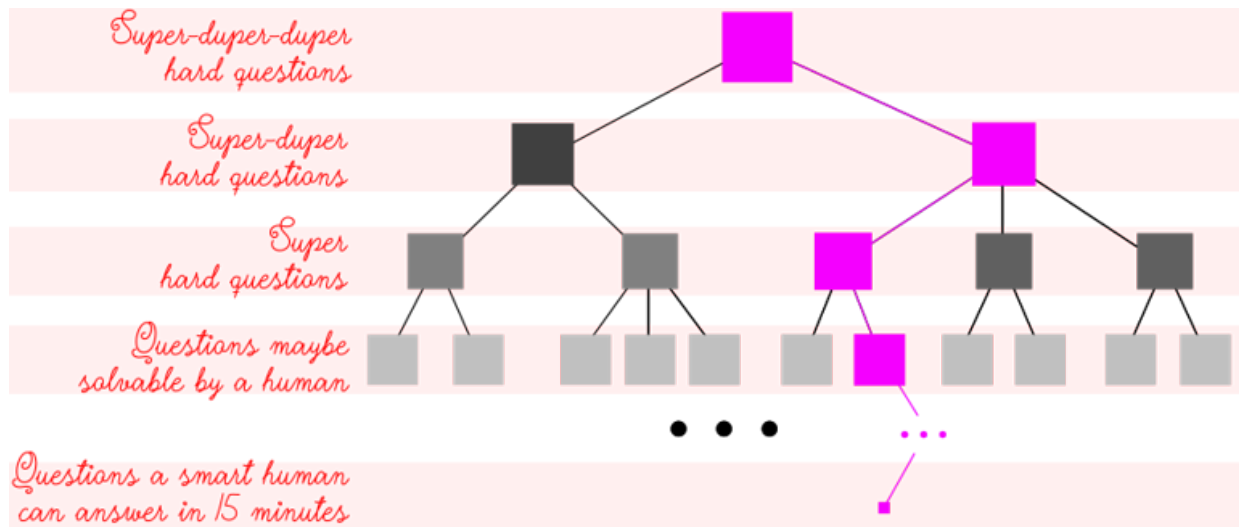
gets sliced up during the training process. If  $[H \rightarrow A_8]$  is asked a super-duper hard

question  $Q$ , but  $A_8$  can only answer super hard questions, then it's her job to decompose  $Q$  into  $q_1, \dots, q_k$  such that the  $q_i$ 's determine  $Q$  and all of them are at most super hard. In other words, we're taking a *local* slice of the tree.



Of course, each *distillation* step will require many such slices, and every subsequent distillation step will require many such slices at a higher level of the tree.

Conversely, each Debate game slices *an entire vertical path* out of the tree.



If Debate is ever implemented, we won't start off with superintelligent agents, and so we won't start off with super-duper<sup>6</sup>-hard questions either. Consequently, the paths may be very short in the beginning and grow longer over time. Either way, each path should always go all the way to the bottom. [\[10\]](#)

The other difference is what I've mentioned at the end of the previous chapter: in Debate, the human never has to deal with figuring out how to decompose a question. At most, she has to figure out whether a decomposition is valid. [\[11\]](#) The way Geoffrey

Irving put it is that a debate transcript (as depicted above) corresponds to a single path through the tree of amplification.<sup>[12]</sup> Crucially, it is a path chosen *by the two Debate agents*.

## 6. Comparison

Both **IDA** and **Debate**...

- may or may not be outer-aligned
- try to utilize the AI systems they're trying to train during the training process
- are designed to scale up to superintelligence
- rely on some version of the **Factored Cognition Hypothesis** to be applicable<sup>[13]</sup> since they traverse the tree of difficult problems/questions

However, **IDA**...

- carves a *local* slice out of the tree at each training step
- has no built-in solution for decomposing questions into subquestions
  - A separate model may be trained for this purpose, or the questions may go meta, i.e., "what is a good way to decompose this question?"
  - Insofar as this makes the decompositions worse, it implies that a *shallow* HCH tree is less powerful than a shallow Debate tree.
- can look very different depending on how the **amplification** and **distillation** black boxes are implemented
- only approximates **HCH** insofar as all distillation steps 'work'

Whereas **Debate**...

- carves a *vertical* slice/path out of the tree at each training step
  - Therefore, it relies on the claim that such a path reliably provides meaningful information about the entire tree.
- probably won't be training-competitive in the above form since each round requires human input
  - This means one has to train a second model to imitate the behavior of a human judge, which introduces further difficulties.
- requires that humans can accurately determine the winner of a debate with debaters *on every level of competence* between zero and superintelligence
- could *maybe* tackle Inner Alignment concerns by allowing debaters to win the debate by demonstrating Inner Alignment failure in the other debater via the use of transparency tools

## 7. Outlook

Although this post is written to work as a standalone, it also functions as a prequel to a sequence on Factored Cognition. Unlike this post, which is summarizing existing work, the sequence will be mostly original content.

If you've read everything up to this point, you already have most of the required background knowledge. Beyond that, familiarity with basic mathematical notation will be required for posts one and two. The sequence will probably start dropping within a week.

---

1. As far as I know, the proposal is most commonly referred to as just 'Iterated Amplification', yet is most commonly abbreviated as 'IDA' (though I've seen 'IA' as well). Either way, all four names refer to the same scheme. [↵](#)
2. Intent Alignment is aligning [what the AI system is trying to do] with [what we want]. This makes it the union of outer and inner alignment. Some people consider this the entire alignment problem. [It does not include 'capability robustness'.](#) [↵](#)
3. I think the details of the distillation step strongly depend on whether IDA is used to train an *autonomous* agent (one which takes actions by itself), or a non-autonomous agent, one which only takes actions if queried by the user.

For the autonomous case, you can think of the model as an 'AI assistant', a system that autonomously takes actions to assist you in various activities. In this case, the most likely implementation involves reinforcement learning.

For the non-autonomous case, you can think of the model as an oracle: it only uses its output channels as a response to explicit queries from the user. In this case, the distillation step may be implemented either via reinforcement learning or via supervised learning on a set of (question, answer) pairs.

From a safety perspective, I strongly prefer the non-autonomous version, which is why the post is written with that in mind. However, this may not be representative of the original agenda. The sequence on IDA does not address this distinction explicitly. [↵](#)

4. Note that, in the theoretical HCH tree, time freezes for a node whenever she asks something to a subtree and resumes once the subtree has delivered the answer, so that every node has the experience of receiving answers instantaneously. [↵](#)
5. It's a bit too complicated to explain in detail how this works, but the gist is that the tree can play through all possible combinations of moves and counter-moves by asking each subtree to explore the game given a particular next move. [↵](#)
6. In particular, it is relevant for stock IDA where the amplification step is implemented by giving a human access to the current model. In principle, one could also implement amplification differently, in which case it may not rely on Factored Cognition. However, such an implementation would also no longer imitate HCH in the limit, and thus, one would need an entirely different argument for why IDA might be outer-aligned. [↵](#)
7. Geoffrey Irving has described Debate as a 'variant of IDA'. [↵](#)
8. This is the step where we rely on debaters being very powerful. If  $Y$  is too weak to find the problematic part of the argument, Debate may fail. [↵](#)
9. Formally, there is a result that, if the judge can solve problems in the complexity class  $P$ , then optimal play in the debate game can solve problems in the complexity class  $PSPACE$ . [↵](#)

10. Given such a path  $p$ , the value  $|p|$  (the total number of nodes in such a path) is bounded by the depth of the tree, which means that it grows logarithmically with the total size of the tree. This is the formal reason why we can expect the size of Debate transcripts to remain reasonably small even if Debate is applied to extremely hard problems. [↵](#)
11. Note that even that can be settled via debate: if  $Y$  claims that the decomposition of  $X$  is flawed, then  $X$  has to defend the decomposition, and both agents zoom into that as the subproblem that will decide the debate. Similarly, the question of how to decompose a question in IDA can, in principle, itself be solved by decomposing the question 'how do I decompose this question' and solving that with help from the model. [↵](#)
12. This is from the [podcast episode](#) I've linked to at the start of the chapter. Here is the relevant part of the conversation:
- Geoffrey:** [...] Now, here is the correspondence. In amplification, the human does the decomposition, but I could instead have another agent do the decomposition. I could say I have a question, and instead of a human saying, "Well, this question breaks down into subquestions  $X$ ,  $Y$ , and  $Z$ ," I could have a debater saying, "The subquestion that is most likely to falsify this answer is  $Y$ ." It could've picked at any other question, but it picked  $Y$ . You could imagine that if you replace a human doing the decomposition with another agent in debate pointing at the flaws in the arguments, debate would kind of pick out a path through this tree. A single debate transcript, in some sense, corresponds to a single path through the tree of amplification.
- Lucas:** Does the single path through the tree of amplification elucidate the truth?
- Geoffrey:** Yes. The reason it does is it's not an arbitrarily chosen path. We're sort of choosing the path that is the most problematic for the arguments. [↵](#)
13. To be precise, this is true for stock IDA, where amplification is realized by giving the human access to the model. Factored Cognition may not play a role in versions of IDA that implement amplification differently. [↵](#)

# Hiding Complexity

Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

## 1. The Principle

Suppose you have some difficult cognitive problem you want to solve. What is the difference between (1) making progress on the problem by thinking about it for an hour and (2) solving a well-defined subproblem whose solution is useful for the entire problem?

(Finding a good characterization of the 'subproblem' category is important for Factored Cognition, but for [this post minus the last chapter], you can think of it purely as a problem of epistemic rationality and human thinking.)

I expect most to share the intuition that there *is* a difference. However, the question appears ill-defined on second glance. 'Making progress' has to cash out as learning things you didn't know before, and it's unclear how that isn't 'solving subproblems'. Whatever you learned could probably be considered the solution to some problem.

If we accept this, then both (1) and (2) technically involve solving subproblems. Nonetheless, we would intuitively talk about subproblems in (2) and not in (1). Can we characterize this difference formally? Is there a well-defined, low-level quantity such that our intuition as to whether we would call a bundle of cognitive work a 'subproblem' corresponds to the size of this quantity? I think there is. If you want, take a minute to think about it yourself; I've put my proposed solution into spoilers.

I think the quantity is **the length of the subproblem's solution**, where by "solution", I mean "the information about the subproblem relevant for solving the entire problem".

As an example, suppose the entire problem is "figure out the best next move in a chess game". Let's contrast (1) and (2):

- (1) was someone thinking about this for an hour. The 'solution' here consists of everything she learns throughout that time, which may include many different ideas/insights about different possible moves/resolved confusions about the game state. There is probably no way to summarize all that information briefly.
- (2) was solving a well-defined subproblem. An example here is, "figure out how good Be5 is".<sup>[1]</sup> If the other side can check in four turns given that move, then the entire solution to this subproblem is the three-word statement "Be5 is terrible".

## 2. The Software Analogy

Before we get to why I think the principle matters, let's try to understand it better. I think the analogy to software design is helpful here.

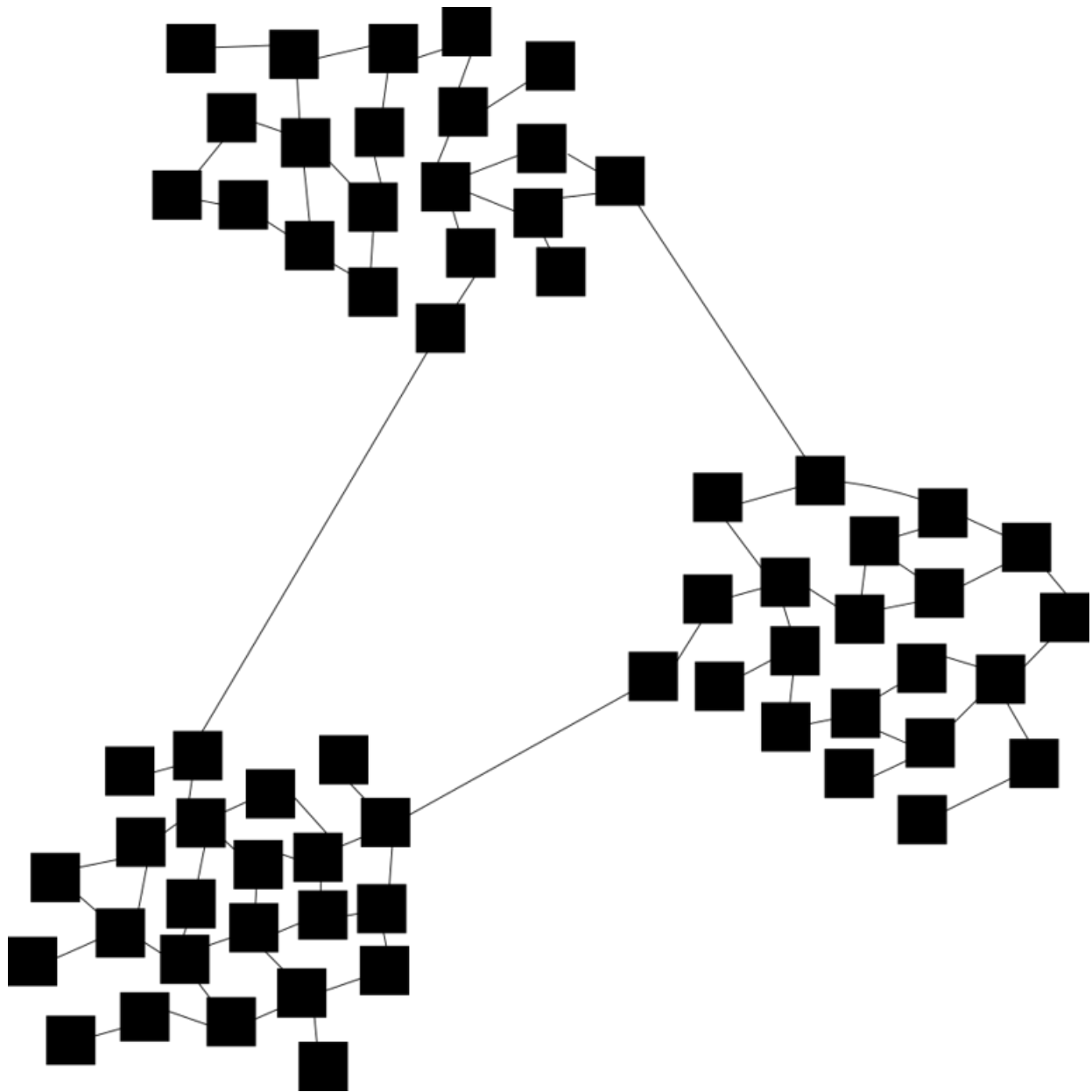
Suppose a company wants to design some big project that will take about 900k (i.e., 900000) lines of code. How difficult is this? Here is a naive calculation:

*An amateur programmer with Python can write a 50 line procedure without bugs in an hour, which suggests a total time requirement of 18k hours. Thus, a hundred amateur programmers working 30 hours a week can write the project in six weeks.*

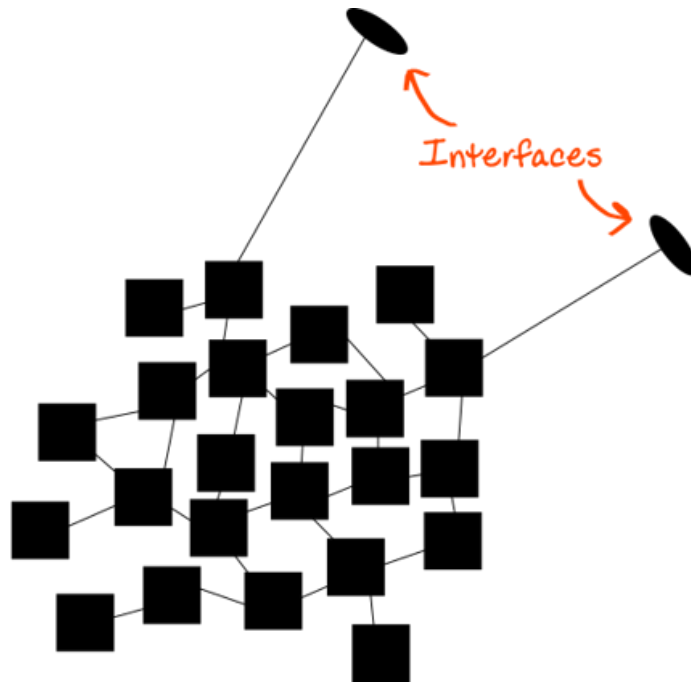
I'm not sure how far this calculation is off, but I think it's at least a factor of 20. This suggests that linear extrapolation doesn't work, and the reason for this is simple: as the size of the project goes up, not only is there more code to implement, but every *piece of code* becomes harder because the entire project is more complex. There are more dependencies, more sources of error, and so forth.

This is where decompositions come in. Suppose the entire project can be visualized like this, where black boxes denote components (corresponding to pieces of code) and edges dependencies between components.





This naturally factors into three parts. Imagine you're head of the team tasked with implementing the bottom-left part. You can look at your job like this:



(An 'interface' is purely a specification of the relationship, so the ellipses are each less than one black box.)

Your team still has to implement 300k lines of code, but regardless of how difficult this is, it's only marginally harder than implementing a project that consists *entirely* of 300k lines. In the step from 300k to 900k, the cost actually *does* scale almost linearly.<sup>[2]</sup>

---

As said at the outset, I'm talking about this not to make a point about software design but as an analogy to the topic of better and worse decompositions. In the analogy, the entire problem is coding the 900k line system, the subproblems are coding the three parts, and the solutions to the second and third part are the interfaces.

I think this illustrates both *why* the mechanism is important and *how* exactly it works.

For the 'why', imagine the decomposition were a lot worse. In this case, there's a higher overhead for each team, ergo higher overall cost. This has a direct analog in the case where a person is thinking about a problem on her own: the more complex the solutions to subproblems are, the harder it becomes for her to apply them to the entire problem. We are heavily bottlenecked by our ability to think about several things at once, so this can make a massive difference.

For the 'how', notice that, while the complexity of the entire system trivially grows with its size, the task of *programming* it can ideally be kept simple (as in the case above), and this is done by **hiding complexity**. From the perspective of your team (previous picture), almost the entire complexity of the remaining project is hidden: it's been reduced to two simple, well-defined interfaces

This mechanism is the same in the case where someone is working on a problem by herself: if she can carve out subproblems, and if those subproblems have short solutions, it dramatically reduces the perceived complexity of the entire problem. In

both cases, we can think of the quality of a decomposition as the total amount of complexity it hides.<sup>[3]</sup>

### 3. Human Learning

I've come to view human learning primarily under the lens of hiding complexity. The world is extremely complicated; the only way to navigate it is to view it on many different layers of abstraction, such that each layer describes reality in a way that hides 99%+ of what's really going on. Something as complex as going grocery shopping is commonly reduced to an interface that only models time requirement and results.

Abstractly, here is the principled argument as to why we know this is happening:

1. Thinking about a lot of things at once feels hard.
2. Any topic you understand well feels easy.
3. Therefore, any topic you understand well doesn't depend on a lot of things in your internal representation (i.e., in whatever structure your brain uses to store information).
4. However, many topics do, in fact, depend on a lot of things.
5. This implies your internal representation is hiding complexity.

For a more elaborate concrete example, consider the task "create a presentation about x", where x is something relatively simple:

- At the highest level, you might think solely about the amount of time you have left to do it; the complexity of how to do it is hidden.
- One level lower, you might think about (1) creating the slides and (2) practicing the speaking part; the complexity of how to do either is hidden.
- One level lower, you might think about (1) what points you want to make throughout your presentation and (2) in what order do you want to make those points; the complexity of how to turn a point into a set of slides is hidden.
- One level lower, you might think about how what slides you want for each major point; the complexity of how to create each individual slide is hidden.
- Et cetera.

In absolute terms, preparing a presentation is hard. It requires many different actions that must be carried out with a lot of precision for them to work. Nonetheless, the *process* of preparing it probably feels easy all the way because every level hides a ton of complexity. This works because you understand the process well: you know what levels of abstraction to use, and how and when to transition between them.

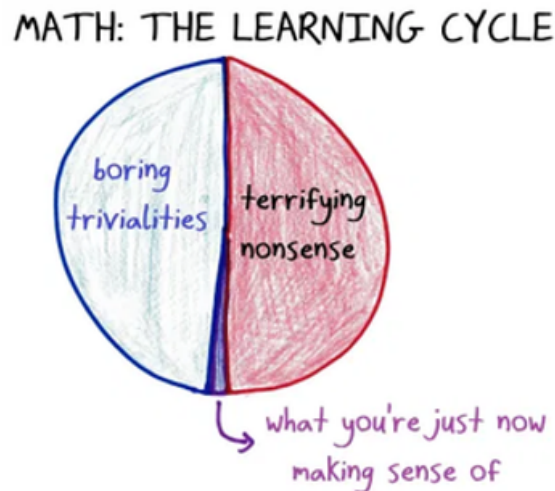
The extreme version of this view (which I'm not arguing for) is that learning is almost entirely about hiding complexity. When you first hear of some new concept, it sounds all complicated and like it has lots of moving parts. When you successfully learned it, the complexity is hidden, and when the complexity is hidden, you have learned it. Given that humans can only think about [a few things at the same time](#), this process only bottoms out on exceedingly simple tasks. Thus, under the extreme view, it's not turtles *all* the way down, but pretty far down. For the most part, learning just *is* representing concepts such that complexity is hidden.

---

I once wrote a tiny post titled '[We tend to forget complicated things](#)'. The observation was that, if you stop studying a subject when it feels like you barely understand it, you will almost certainly forget about it in time (and my conclusion was that you should always study until you think it's easy). This agrees with the hiding complexity view: if something feels complicated, it's a sign that you *haven't* yet decomposed it such that complexity is hidden at every level, and hence haven't learned it properly. Under this view, 'learning complicated things' is almost an oxymoron: proper learning must involve making things feel not-complicated.

It's worth noting that this principle appears to apply even for [memorizing random data](#), at least to some extent, even though you might expect pure memorization to be a counter-example.

There is also this lovely pie chart, which makes the same observation for mathematics:



That is, math is not inherently complicated; only the parts that you haven't yet represented in a nice, complexity-hiding manner feel complicated. Once you have mastered a field, it feels wonderfully simple.

## 4. Factored Cognition

As mentioned in the outset, characterizing subproblems is important for Factored Cognition. Very briefly, Factored Cognition is about *decomposing a problem into smaller problems*. In one setting, a human has access to a model that is similar to herself, except (1) slightly dumber and (2) much faster (i.e., it can answer questions almost instantly).



The hope is that this combined system (of the human who is allowed to use the model as often as she likes) is more capable than either the human or the model by themselves, and the idea is that the human can amplify performance by decomposing

big problems into smaller problems, letting the model solve the small problems, and using its answers to solve the big problem.

There are a ton of details to this, but most of them don't matter for our purposes.<sup>[4]</sup> What does matter is that the model has no memory and can only give short answers. This means that the human can't just tell it 'make progress on the problem', 'make more progress on the problem' and so on, but instead has to choose subproblems whose solutions can be described in a short message.

An unexpected takeaway from thinking about this is that I now view Factored Cognition as intimately related with learning in general, the reason being that both share the goal of choosing subproblems whose solutions are as short as possible:

- In the setting I've described for Factored Cognition, this is immediate from the fact that the model can't give long answers.
- For learning, this is what I've argued in this post. (Note that optimizing subproblems to minimize the length of their solutions is synonymous with optimizing them to maximize their hidden complexity.)

In other words, Factored Cognition primarily asks you to do something that you want to do anyway when learning about a subject. I've found that better understanding the relationship between the two has changed my thinking about both of them.

---

(This post has been the [second of two](#) prologue posts for an upcoming sequence on Factored Cognition. I've posted them as stand-alone because they make points that go beyond that topic. This won't be true for the remaining sequence, which will be narrowly focused on Factored Cognition and its relevance for Iterated Amplification and Debate.)

---

1. Be5 is "move the bishop to square E5". [↵](#)
2. One reason why this doesn't reflect reality is that real decompositions will seldom be as good; another is that coming up with the decomposition is part of the work (and in extension, part of the cost). Note that, even in this case, the three parts all need to be decomposed further, which may not work as well as the first decomposition did. [↵](#)
3. In Software design, the term 'modularity' describes something similar, but it is not a perfect match. Wikipedia [defines](#) it as "a logical partitioning of the 'software design' that allows complex software to be manageable for the purpose of implementation and maintenance". [↵](#)
4. After all, this is a post about hiding complexity! [↵](#)

# Preface to the Sequence on Factored Cognition

Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

Factored Cognition is primarily [studied](#) by [Ought](#), the same organization that was partially credited for implementing the [interactive prediction feature](#). Ought is an organization with at least five members who have worked on the problem for several years. I am a single person who just finished a master's degree. The rationale for writing about the topic anyway was to have diversity of approaches: Ought is primarily doing empirical work, whereas I've studied the problem under the lens of math and epistemic rationality. As far as I know, there is virtually no overlap between what I've written and what Ought has published so far.

Was it successful? Well, all I can say for sure is that writing the sequence has significantly changed my own views.

This sequence has two 'prologue' posts, which make points relevant for but not restricted to Factored Cognition. I think of them as posts #-2 and #-1 (then, this post is #0, and the proper sequence starts at #1). These are

- [A guide to Iterated Amplification and Debate](#), which explains what Factored Cognition is and the two schemes that use it. This post is there to make sure that no prerequisite knowledge is needed to read the sequence. You can skip this if you're already familiar with both schemes.
- [Hiding Complexity](#), which is about characterizing what makes a part of a big problem a 'subproblem'.

The remaining sequence is currently about 15000 words long, though this could change. The structure is roughly:

- Define a mathematical model and see what we can do with that (posts #1-#2)
- Tackle the human component: think seriously about how thinking works and whether solving hard problems with Factored Cognition looks feasible (posts #3-#5)
- Spell out what I conclude from both parts (post #6)

The current version of the sequence includes exercises. This is pretty experimental, so if they are too hard or too easy, it's probably my fault. I've still left them in because I generally think it makes sense to include 'think about this thing for a bit' moments. They look like this:

**EXERCISE (5 SECONDS):** Compute  $2+5$ .

7.

Whenever there's a range, it means that the lower number is an upper bound for the exercise itself, and the remaining time is for rereading parts of this or previous posts. So 1-6 minutes means 'you shouldn't take more than 1 minute for the exercise itself, but you may first take about 5 minutes to reread parts of the post, or perhaps of previous posts'.

The sequence also contains conjectures. Conjectures are claims that I think are true, important, and not trivial. There are only a few of them, and they should all be justified by the sequence up to that point. Conjectures look like this:



**Putting statements into  
gradient color bubbles is a good way  
to make them memorable.**

I'll aim for publishing one post per week, which gives me time for final edits. It could slow down since I'm still working on the second half. Questions/criticism is welcome.

Special thanks to [TurnTrout](#) for providing valuable feedback on much of the sequence.

# Idealized Factored Cognition

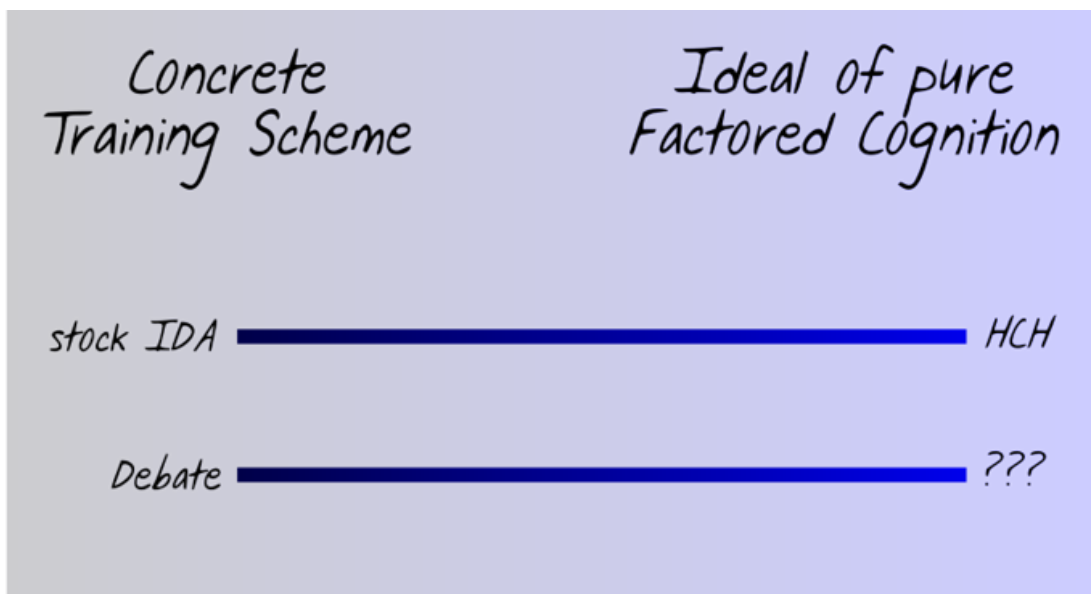
Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

(This post is part of a sequence that's meant to be read in order; see the [preface](#).)

## 1. HCH and Ideal Debate

Recall from [post #-2](#) that we have two perspectives on stock IDA.<sup>[1]</sup> One is that of a human with access to a model, the other is that of an HCH tree.<sup>[2]</sup> We can think of HCH as 'pure' or 'idealized' Factored Cognition that abstracts away implementation details,<sup>[3]</sup> and the training procedure as trying to implement this ideal.

One might now ask the following:



If HCH is the ideal of stock IDA, then what is the ideal of Debate? Since this is a sequence about Factored Cognition, we are primarily interested in analyzing the idealized versions, so this is the first question we'd like to answer.

---

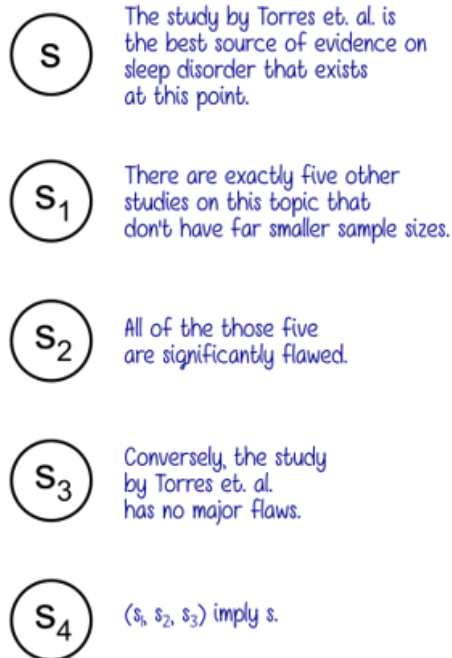
Interlude on notation: throughout this sequence, we will need to refer to single statements, sequences of statements, and sets of statements. To make telling them apart as easy as possible, we follow the following norms:

- single statements use lower case letters, like  $s$  or  $s_1$  or  $s_n$  or  $s_j$
- sequences of statements use uppercase bold letters, like  $\mathbf{S}$  or  $\mathbf{S}_1$  or  $\mathbf{S}_n$
- sets of statements use the 'pretty'  $\mathcal{S}$ , like  $\mathcal{S}_h$  or  $\mathcal{S}_h^T$



---

We'll use the first two of those in the following definition that will be crucial in our discussion of Debate. If  $s$  is a statement and  $S = (s_1, \dots, s_{n+1})$  a sequence of statements such that  $s_{n+1} = "(s_1, \dots, s_n) \text{ imply } s."$ , we say that  $S$  is an **explanation** for  $s$  and denote this by writing  $S \rightarrow s$ . In this setting,  $s_{n+1}$  is what we call an **implication statement**: it precisely says that the statements  $(s_1, \dots, s_n)$  imply the statement  $s_{n+1}$ . Here's a made-up example where  $s_4$  is the implication statement:



The purpose of having implication statements is to ensure that  $s$  *trivially* follows from  $s_1, s_2, s_3, s_4$  since the implication itself is among those statements (here  $s_4$ ). If you dispute  $s$ , you must dispute one of the  $s_i$ .

Armed with this concept, we can define our idealization of the Debate scheme:

### **Ideal Debate**

### **Ideal Debate**

The input to the game is a question in English. The first agent begins by giving an answer plus an explanation<sup>[4]</sup>  $(s_1, \dots, s_{n+1})$  for the answer. At every subsequent step,

the second agent points to one of the statements  $s_j$  in the explanation,

$j \in \{1, \dots, n+1\}$ , and the first agent responds by either giving an explanation for  $s_j$  or declaring that the debate is over. In the latter case, a judge attempts to verify that  $s_j$  is true. If she succeeds, the first agent wins the debate; if not, the second agent wins the debate.<sup>[5]</sup> The first agent must end the game after a finite number of steps.

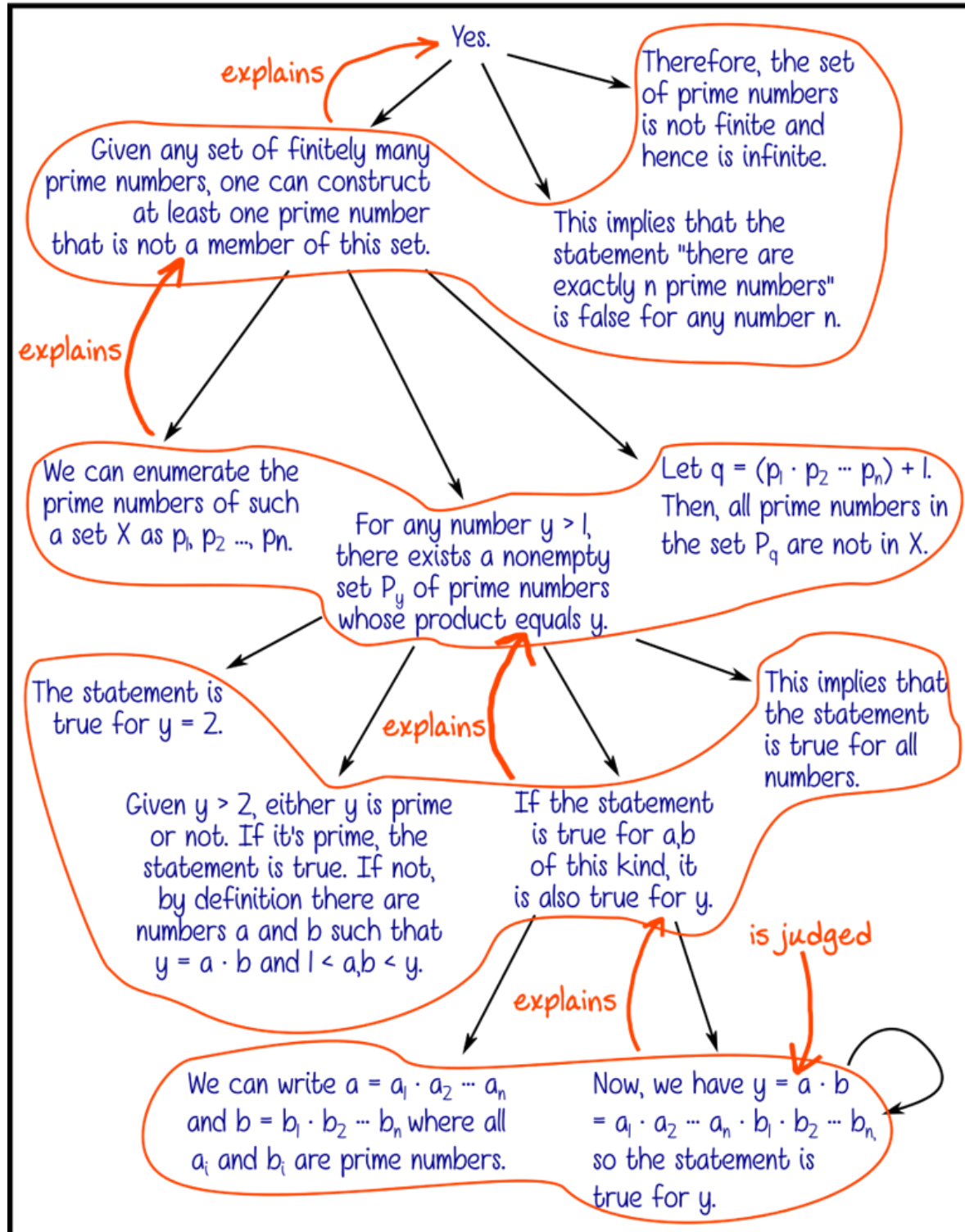
The debaters are maximally powerful agents; the judge is a human.

Just as HCH abstracts away implementation details of stock IDA, Ideal Debate abstracts away implementation details of Debate.<sup>[6]</sup>

Instead of having implication statements, one could have allowed the second agent to deny the fact of the implication, as a 'special move' of sorts. However, I think that would be a mistake; one of the take-away messages from this post is that there is no sharp line between implications and other statements. Both can be disputed and argued about further, which is why we're treating them as the same type of object. In the definition of Ideal Debate above, the second agent is always free to point to  $s_{n+1}$  (which is precisely the implication statement), and if so, the game proceeds normally, i.e., the first agent has to provide an explanation for  $s_{n+1}$  as the next move.

Below is an example of what an Ideal Debate transcript might look like.<sup>[7]</sup> Here, the implication statement is not shown (but this is just to make it easier to draw – it should really be there at every level!), and the statement we recurse into is always the one that the second agent has pointed to.

Is the set of all prime numbers infinite?



You can also look at the [uncluttered version](#).

**A clarification:** statements are not strings. By itself, the string 'Now, we have  $y = a \cdot b$  [...]' cannot be judged since it contains symbols whose meaning is only defined in the remaining transcript. In general, any statement may require an arbitrary amount of additional context to be understood. This means the length of its string doesn't meaningfully indicate how complex a statement is.

Despite this, the complexity of  $s_{\text{final}}$  may remain relatively low. This is due to the principle we've discussed in [post #-1](#). The better the explanations chosen by the first agent are, the less complexity there will be in each part of the argument. Ideally, the judge will only have to deal with a small part of the entire argument to verify  $s_{\text{final}}$ .<sup>[8]</sup>

## 2. Cognition Spaces

One of the things mathematicians like to do when studying a problem is to define the space that one is working in. For example, in machine learning, one usually decides on a space of possible models before beginning the search. Consequently, we would now like to define a space and say that HCH and Ideal Debate are about doing stuff within this space. I will call this a **Cognition Space**.

As mentioned above, I think it is a mistake to differentiate between 'facts' and 'implications'. The prime number transcript is an example of this: it follows from the axioms of set theory that there are infinitely many prime numbers, so technically the entire debate is only about implications, but they just feel like regular statements. For this reason, we will take 'statement' to be a primitive and define a Cognition Space to be a pair

$$(S_h, d_h)$$

where  $S_h$  is a set of statements, and  $d_h : S_h \rightarrow \mathbb{R}_+$  a function assigning each statement a difficulty. On this, several points.

- What  $d_h$  measures is the difficulty of verifying that a statement is true, *not* of understanding what is being said. For example, in the Ideal Debate transcript shown above, the difficulty of the statement "Given any set of finitely many prime numbers, one can construct at least one prime number that is not a member of this set." is likely quite high, even though it's fairly easy to understand what the claim *is*.
- Since we do not differentiate between implications and facts, implication statements are regular members of  $S_h$ . Consequently, a Cognition Space determines which approaches to explaining statements are difficult and which are easy. For example, say you're the first agent in Ideal Debate and have to explain the root statement  $s_0$ . You might see two ways of doing this, either via  $s_1$  and  $s_2$ , or via  $s_3$  and  $s_4$  and  $s_5$ . In this case, not only do these five statements have each a difficulty, but the set  $S_h$  also includes two implication statements that precisely

say “ $(s_1, s_2)$  imply  $s_0$ .” and “ $(s_3, s_4, s_5)$  imply  $s_0$ .”, respectively. All five of the  $s_i$  and the two implication statements are assigned a difficulty by  $d_h$ .

- The ground truth here is entirely based on the human  $h$ , which is either the human in HCH or the judge in Ideal Debate.  $S_h$  contains all things that she would consider statements, and difficulty is measured by how hard it is *for her* to verify a statement. Thus, the human entirely determines the Cognition Space, and we've put her in the subscripts of both  $S_h$  and  $d_h$  to serve as a reminder of that.

We now turn to Ideal Debate in particular. Given the definitions above, we define a **path through a Cognition Space**  $(S_h, d_h)$  to be a pair

$$((s_0), S_1, \dots, S_n), s_{\text{final}})$$

where  $s_{\text{final}} \in S_n$  <sup>[9]</sup> and for each  $j \in \{1, \dots, n\}$ , we have that  $S_j \in S_h$  <sup>[10]</sup> and  $S_j \rightarrow s$  for some  $s \in S_{j-1}$ ; with  $S_0 := (s_0)$  being just the initial answer to the input question. (Recall

that the fat letters denote *sequences*, and that the notation  $S_j \rightarrow s$  is short for ' $S_j$  is an explanation for  $s$ '.)

An Ideal Debate transcript (such as the prime number one shown above) precisely visualizes one path through a Cognition Space. <sup>[11]</sup> To make sure you're following the formalism up to this point, here's an exercise.

**EXERCISE (1-5 MINUTES):** If we extend  $d_h$  to paths, what is the correct definition for the difficulty of a path?

The difficulty should equal that of  $s_{\text{final}}$  since that's the only statement the judge needs to verify.

### 3. Finding Explanations

Ideal Debate traverses a Cognition Space by having the two agents collectively choose a single path that starts at an answer  $s_0$  to the input question and ends at some statement  $s_{\text{final}}$ . If the first agent is honest (and we will assume so for this post), she will give a true initial statement (the answer to the input question) and will provide a true explanation for the previous statement at every step of the game. The second agent will, presumably, try to navigate to the most difficult part of the argument, hoping that the judge will fail to verify  $s_{\text{final}}$ . (The second post will look into this a lot more.)

Conversely, a node in an HCH tree is initialized with a question, which she will attempt to answer while using subtrees to answer related questions. If she succeeds at this, then she must consider [the set of (question, answer) pairs she has exchanged with subtrees plus whatever cognitive work she's done herself] to be an explanation for the answer. (Otherwise, she risks returning an answer that isn't true.)

Thus, explanations<sup>[12]</sup> are key in both cases; however, the *purpose* of the explanation is different:

- in Ideal Debate, the statement that is explained is already known, and the explanation is meant to *demonstrate* that it is true; whereas
- in HCH, the statement to be explained is not known, and the purpose of the explanation is to *derive* a [statement that answers the input question].

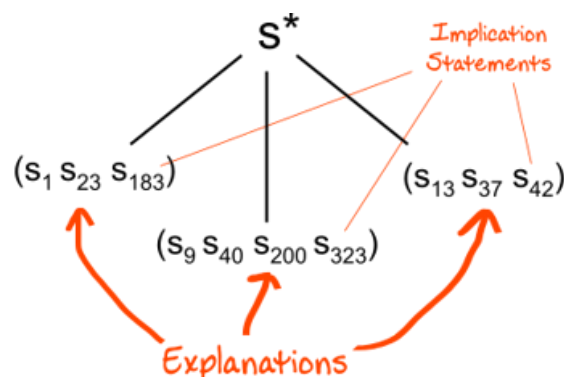
Put differently, Ideal Debate (if the first agent is honest) is analogous professor deciding which statements most cleanly demonstrate that such-and-such is the answer to a particular question, whereas HCH is analogous to a student trying to figure out how to answer the question in the first place, and asking subquestions to help with that. The difference between the resulting decompositions will vary – we can imagine questions where it is nonexistent, such as

$Q := \text{"What is } 987 \cdot 123\text{"}$

In this case, we may end up with transcripts that look like [this](#) or [this](#) (with the usual disclaimers about hidden elements); they're functionally identical. But it's not hard to imagine cases where the difference is substantial. In fact, you need only take the prime number transcript above to have an example; without knowing the proof, many people would not think to ask, 'given any set of finitely many prime numbers, can you use them to construct a new prime number?', which means that an HCH transcript for the same question would look differently.

Let's take a shot at formalizing this difference. Recall our cognition space  $(S_h, d_h)$ .

Given some statement  $s^*$ , the space implicitly represents the *tree of all possible explanations* for  $s^*$ . It looks something like this:



Note that the structure of the tree is entirely determined by the existing implication statements: for each implication statement  $s' = \text{"(s}_j, s_k) \text{ imply s. "}$   $\in S_h$ , there is an

edge from  $(s_j, s_k, s')$  to  $s$  in the tree.

In reality, there are likely far more than three explanations for  $s$ , probably with more components, and then each component of each explanation has itself many more explanations, and the components of those have explanations as well, and so on. It becomes very intractable very quickly.

Formally, write  $\text{Explanations}(s)$  to denote the set of all possible explanations in for  $s$  that exist in  $d_h$ . (This set has precisely as many members as there are implication statements of the form “(arbitrary sequence) imply  $s$ .”) We can now define our requirement that Ideal Debate agents be 'maximally powerful' as the ability to search through all members of  $\text{Explanations}(s)$ . In this setting, the first agent will pick the one that will lead to the easiest possible path, given the adversarial nature of the second agent. (Again, we will look into this more in the second post.)

HCH is much harder to formalize, but for now, we can crudely model the limitations that come with not knowing the answer as the ability to

- only search through a subset of  $\text{Explanations}(s)$ ; and
- being able to derive the answer iff there is a sufficiently easy explanation on offer.

Here's another exercise to make sure the formalism is clear.

**EXERCISE (3-8 MINUTES):** Suppose a node in HCH succeeds in answering a question if she finds an explanation  $(s_1, \dots, s_{n+1}) \in \text{Explanations}(s_0)$  such that  $\sum_{j=1}^{n+1} d_h(s_j) \leq 100$ , where  $s_0$  is the correct answer to the input question. Suppose further that she can search through a hundred elements (randomly chosen) in  $\text{Explanations}(s_0)$  to find such an explanation. Come up with a toy example where this would likely lead to her failing to derive  $s_0$ , even though a simple explanation does exist. To do this, define a full cognition space  $(S_h, d_h)$ .<sup>[13]</sup> You can choose arbitrary values; they need not correspond to anything real.

- $S_h := \{s_0, s_1, \dots, s_n\} \cup \{“(s_j) \text{ imply } s_0.” \mid 1 \leq j \leq n\}$ .
- $d_h(s_n) := 1$
- $d_h(“(s_n) \text{ imply } s_0.”) := 1$
- $d_h(s) := 10^{100}$  for all other  $s \in S_h$ .

Increase  $n$  to make it arbitrarily unlikely for the human to find an explanation.


The prime number example shows that there are real cases where the difference is significant, and the formalism agrees.

So - an HCH tree whose human only has high school knowledge about math would not immediately guess the most elegant proof. Fortunately, it doesn't have to. HCH has a massive computational budget, so if it goes in a fruitless direction first, that's still fine, as long as it finds a correct proof eventually. Would it do that?

Who knows. [\[14\]](#)

However, it certainly isn't *obvious* that *deriving* a mathematical proof has the same asymptotic difficulty as *understanding* it, which is what it means to say that Factored Cognition is guaranteed to either work for both stock IDA and Debate or neither.

We can thus end this post on our first conjecture:



**There is no one  
Factored Cognition Hypothesis  
to rule them all.**

Decompositions are an essential part of any Factored Cognition scheme, and changing how they are chosen is entirely allowed to change how the scheme scales to harder problems.

In the next post, we'll see how much we can do with the formalism. This will not be conclusive, which is why we will then switch gears and turn to the human component.

- 
1. I say 'stock IDA' to refer to any implementation of IDA where a human is doing the decomposition. There are possible implementations where an agent is doing the decomposition or where there is no decomposition at all (those implementations don't rely on Factored Cognition). In its most general form, IDA is merely a template of a training scheme prescribing that there be two procedures called **Distill** and **Amplify**, and under this view, just about every training scheme is technically a variant of IDA (any method that uses gradient descent becomes an instance of IDA if we set [Amplification] = [Gradient Descent step] and [Distillation] = [Identity Function]), which is why we won't talk much about IDA in general.

Note that stock IDA still leaves the implementation of the distillation step open. [↩](#)



2. Note that HCH is technically not a fully defined scheme, but a class of schemes  $\{HCH_{h,t,\ell}\}$ , where  $h$  defines the human component (what human, what environment, etc.),  $t$  is a parameter in  $R_+$  that specifies a time limit, and  $\ell$  defines the communication channel (what kind of messages, what length, etc.).

Given these parameters, we can define a **node** semi-formally like so:

*A node is a human with context specified in  $h$ , initialized with some question  $q$ . It exists for time  $t$ .*

*During this time, it can spawn another node with some question  $q'$  arbitrarily often; whenever it does, it immediately receives the output from that node.*

*By the end of time  $t$ , it needs to provide an output, obeying conditions governed by parameter  $\ell$ .*

(Here, the nodes it can spawn are the same type of object as itself.)

Then, the entire scheme is simply a node initialized with the scheme's input question.

This definition always yields a tree of infinite depth. It corresponds to what Paul calls weak-HCH. I talk more about why this sequence looks at weak-HCH rather than strong-HCH in a later post; it's related to the concepts discussed in [Hiding Complexity](#). ↩

3. Details that have been abstracted away include:

- Inner Alignment concerns: in the real world,  $A_{k+1}$  may have an objective access other than trying to approximate  $[H \rightarrow A_k]$  even if it was trained to do that.
- Bounded depth: for any  $k \in \mathbb{N}$ , the model  $A_k$  only approximates an HCH tree of depth  $k$ , not an infinite one.
- Computational limitations: a trained model can only approximate an exponentially large tree insofar as the tree's computations can be done more cheaply with better algorithms. (This is the part we won't ignore since it's a hard limitation.)

↩

4. It being an explanation is a precise requirement; recall the definition above. ↩
5. Note that the complement of 'the judge succeeds in verifying that  $s_j$  is true' is not 'the judge succeeds in verifying that  $s_j$  is false'. If the judge is uncertain, the

second agent also wins the debate. ↩

6. Details that have been abstracted away include:

- Again Inner Alignment concerns: in real Debate, the agents may have motives that go beyond winning any one debate game (like trying to cause more debates to happen in the future)
- Ambiguity: in reality, the meaning of statements can shift due to ambiguous words. This problem is the motivation for [cross-examination](#).
- Wireheading: with the setup as-is, the first agent could subtly delude the judge rather than playing honestly, and the second agent can't prevent this since she has much more restricted output channels.
- Weak Debaters: although Debate agents should eventually become very powerful (as long as the training signal is accurate, the only limit is the power of the best machine learning techniques), they don't start off that way, and the scheme has to work about even at that point.

↩

7. As an aside: this example probably illustrates that it won't be the case that every human is competent enough to judge Debate transcripts. For example, it requires some degree of familiarity with mathematical notation and some competence in logical thinking. (And they need to understand English.) The relevant question is how the difficulty scales with the complexity of the question. ↩

8. There are some striking examples of this principle in action in more difficult mathematical proofs. I may at some point dedicate a post to illustrating this. ↩

9. Here, we're abusing the  $s \in S$  notation to mean 's appears in the sequence S', which is technically different from set membership. ↩

\*

10. The symbol  $S_h^*$  denotes the *set of all sequences* of statements in  $S_h$ . (This use of the asterisk is standard.) ↩

11. Geoffrey Irving (inventor of the Debate scheme) said something functionally similar months ago on the [AI alignment podcast](#): "A single debate transcript, in some sense, corresponds to a single path through the tree of amplification." ↩

12. Usually, people talk about **decompositions** rather than explanations. As far as I'm concerned, they're synonyms: both terms refer to the set of substatements given by a debate agent/the set of (question, answer) pairs an HCH node exchanges with subtrees. I'm talking about explanations to emphasize the fact that they imply a specific statement. ↩

13. Note that you don't need to define what the non-implication statements are, it's enough to postulate that they exist. You do need to define the implication statements since those determine which sequences are explanations. As an example, the following:

- $S_h := \{s, s', s_1, s_0\}$
- $s_1 := "(s, s') \text{ imply } s_0."$

- $d_h(s) := d_h(s') := d_h(s_l) := d_h(s_0) := 1$

perfectly defines a Cognition Space. However, this is not a solution to the exercise since  $\text{Explanations}(s_0)$  only has a single element. [↵](#)

14. This is a good time to mention that [Ought](#) may or may not be studying questions similar to this one. [↵](#)

# Traversing a Cognition Space

Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

(This post is part of a sequence that's meant to be read in order; see the [preface](#).)

[Post #1](#) was about developing and justifying a formalism for Factored Cognition. Now that we have this formalism, this post is about doing as much with it as possible.

## 1. Debate Trees

[Recall](#) that a Cognition Space is a pair  $(S_h, d_h)$  where  $S_h$  is a set of statements,

$d_h : S_h \rightarrow \mathbb{R}_+$  is a difficulty function, and  $h$  is a human.

So far, I've only shown examples of single transcripts. A single transcript corresponds to one path through  $d_h$  that is dependent on choices from both agents: at every step, the first agent outputs an explanation (which is a sequence of statements), the second agent points at one element of this sequence, and the first agent continues the path from that element onward. However, given that we model Ideal Debate agents as maximally powerful, it is also coherent to ask about the object that results if we fix *all* of the first agent's actions in advance, such that she 'pre-commits' for any possible combination of choices from the second agent.

I call such an object a **Debate Tree**, and we can define it formally as a *directed rooted*

*tree*<sup>[1]</sup>  $(V, E)$ , where  $V \subset S_h^*$  (so each node is a sequence of statements) and  $E \subset V \times V$ , that satisfies the following three conditions:

1. The unique root of  $(V, E)$  is a one-element 'sequence'  $(s_0)$ .

2.  $\forall (S, S') \in E \exists s \in S$ <sup>[2]</sup> :  $S' \rightarrow s$ .

3.  $\forall (S, S'), (S, S'') \in E : [\exists s \in S : S' \xrightarrow{s} s \wedge S'' \xrightarrow{s} s] \implies [S' = S'']$ .

The first condition says that the root of the tree needs to be a single statement (this should be the answer to the input question). The second condition says that every edge  $(S, S')$  needs to explain a statement in  $S$ ; we don't have redundant edges in our tree. And the third condition says that each statement is only explained once; the formal way of saying this is that, if two explanations exist for the same statement, they're really the same. (Note that this restriction exists because the first agent has to choose one explanation during the game; it certainly doesn't imply that only one explanation exists.) There is no condition to demand that each statement needs to

have an explanation – whenever there is none, it means that the first agent decides to end the debate when that statement is pointed at. <sup>[3]</sup>

In this definition, nodes are explanations, and each node has one outgoing edge for each [statement of the explanation that the first agent wants to explain further], which links to an explanation for that statement. Defining the tree over individual statements would lead to a functionally equivalent definition, but I think defining it as-is makes arguments simpler.

Recall that a Debate Tree encodes all decisions that the first agent makes (for any possible combination of choices from the second agent). This means that, once we fix this object, the second agent is free to choose any path she wants out of the tree, without further involvement from the first agent. Formally, a **path** through a Debate tree with root  $(s_0)$  is a pair <sup>[4]</sup>

$$(((s_0), s_1, \dots, s_n), s_{\text{final}})$$

where  $(s_{j-1}, s_j) \in E \ \forall j \in \{1, \dots, n\}$  and  $s_{\text{final}} \in S_n$ . (And  $s_0 = (s_0)$ .) (You can go back to post 1 to convince yourself that a path through a Debate tree is also a path through a Cognition Space.)

As mentioned in the first post, we define the difficulty of a path  $P := (p, s_{\text{final}})$  as the difficulty of the final statement (since that is the one being judged). In symbols,  $d_h(P) := d_h(s_{\text{final}})$ .

## 2. The Ideal Debate-FCH

So far, we haven't talked about how precisely the two agents make their decisions. Given the concepts of Debate Trees and paths, this is now easy. The second agent wants the first agent to lose, which means she'll choose the most difficult path in whatever Debate Tree the first agent chooses. (We still assume that the first agent outputs only true statements, which means that the second agent can't win if the judge successfully verifies the final statement.) The first agent, knowing this, chooses the Debate tree such that the difficulty of the hardest path is minimized.

With this, we are almost ready to define a FCH for Ideal Debate. But first, we need a bit more notation:

- Given a question  $q$ , we (for now) assume there is a unique statement  $a_q \in S_h$  that correctly answers the question.
- Given a cognition space  $(S_h, d_h)$ , we write  $T((S_h, d_h), a_q)$  for the set of all Debate trees that begin with statement  $a_q$ .
- Given a Debate tree  $T$ , we write  $P(T)$  for the set of all paths in  $T$ .

Given this, the difficulty of the path we will end up with is

$$\min_{T \in \mathcal{T}((S_h, d_h), a_q)} \left( \max_{p \in P(T)} d_h(p) \right)$$

Since the absolute values of the difficulty function  $d_h$  are arbitrary (it doesn't matter whether we denote difficulties from 0 to 100 or from 0 to  $10^{10000}$ ), we can assume without loss of generality that the hardest difficulty a human can handle is 1. [\[5\]](#) Thus, we can formulate:

the Ideal Debate FCH(h, Q):

$$\forall q \in Q : \min_{T \in \mathcal{T}((S_h, d_h), a_q)} \left( \max_{p \in P(T)} d_h(p) \right) \leq 1$$

where  $h$  is a human and  $Q$  a set of questions.

Going forward, it will also be useful to talk about the difficulty of Debate Trees. Thus, we define

$$d_h(T) := \max_{p \in P(T)} d_h(p)$$

and we say that a Debate tree  $T$  **can handle** question  $q$  if  $d(T) \leq 1$ . [\[6\]](#)

Here is a different view of the problem. In the space of all statements, there is a subset that the human judge can verify directly, i.e.,

$$\begin{matrix} (0) \\ D_h^{(0)} := \{s \in S_h \mid d_h(s) \leq 1\} \end{matrix}$$

Then, there is a larger subset that contains all of the above plus the statements that   
(0)  
 can be explained solely in terms of statements in  $D_h^{(0)}$ , i.e.,

$$\begin{matrix} (1) & (0) & (0) & e \\ D_h^{(1)} := D_h^{(0)} \cup \{s \in S_h \mid \exists S \in (D_h^{(0)})^* : S \rightarrow s\} \end{matrix}$$

In general, given any  $k \in \mathbb{N}_+$ , we can extend  $D_h^{(k-1)}$  by adding all statements that can be explained solely in terms of statements in  $D_h^{(k-1)}$ , i.e.,

$$D_h^{(k)} := D_h^{(k-1)} \cup \{s \in S_h \mid \exists S \in (D_h^{(k-1)})^* : S \rightarrow s\}$$

Note that this gives us a chain of expanding sets, i.e.,

$$D_h^{(0)} \subseteq D_h^{(1)} \subseteq \dots \subseteq D_h^{(k)} \subseteq D_h^{(k+1)} \subseteq \dots$$

We can also define the set of all statements that are eventually explainable in this way, i.e.,

$$D_h := \bigcup_{j=0}^{\infty} D_h^{(j)}$$

Intuitively, it seems like Ideal Debate should be able to handle all questions with answers in  $D_h$ , since they can be explained in terms of progressively easier statements

- and then any path should eventually bottom out at  $D_h^{(0)}$ , which means that the second agent cannot delay success indefinitely.

This brings us to our first (and as of now, only) theorem:

**Theorem.** Given any  $h$  and  $Q$ ,  $\text{Ideal Debate-FCH}(h, Q) \iff \forall q \in Q : a_q \in D_h$ .

**Proof.** First, note that, while the Ideal Debate-FCH is formulated as a hypothesis, the definition also defines a set, namely

$$X_h = \{s \in S_h \mid \min_{T \in \mathcal{T}((S_h, d_h), s)} d(T) \leq 1\}$$

and the Ideal-Debate FCH simply says that  $\forall q \in Q : a_q \in X_h$ . It thus suffices to show that  $X_h = D_h$ . We will prove an even stronger statement. Note that we can restrict the set  $X_h$  by limiting the maximum depth of the Debate Trees that can handle the respective statements. Formally, we can define

$$X_h^{(k)} := \{s \in S_h \mid \min_{T \in T^{(k)}((S_h, d_h), s)} d(T) \leq 1\}$$

for any  $k \in \mathbb{N}$ , where  $T^{(k)}$  denotes the set of Debate Trees with depth at most  $k$ . ('Depth' is defined as the number of edges in the longest path through the tree.) By

construction, we now have  $X_h = \bigcup_{j=0}^{\infty} X_h^{(j)}$ , just as  $D_h = \bigcup_{j=0}^{\infty} D_h^{(j)}$ . What we will show is that

$$D_h^{(k)} = X_h^{(k)} \quad \forall k \in \mathbb{N}.$$

We proceed by induction. First, if  $s \in D_h^{(0)}$ , then  $d_h(s) \leq 1$ , which means that the trivial tree  $(\{(s)\}, \emptyset)$  is a Debate Tree of depth 0 that can handle  $s$ , so that  $s \in X_h^{(0)}$ .

Conversely, if  $s \in X_h^{(0)}$ , then the Debate tree  $T$  handling  $s$  must have no edges (otherwise, its depth would be at least 1). Thus,  $p := ((s), s)$  is a path in this tree, and we have  $1 \geq d(p) = d(s)$ , hence  $s \in D_h^{(0)}$ .

Now, suppose the statement is true for some  $k \in \mathbb{N}$ . We show that  $D_h^{(k+1)} = X_h^{(k+1)}$ .

" $\subset$ ": Let  $s \in D_h^{(k+1)}$ . Then, there exists  $S = (s_1, \dots, s_{n+1}) \in (D_h^{(k)})^*$  such that  $S \rightarrow s$ . Apply the Inductive Hypothesis to find Debate Trees  $T_1, \dots, T_{n+1}$  of depth at most  $k$  such that tree  $T_j$  handles statement  $s_j$ . We combine these trees into a larger tree with root  $(s)$  and an additional edge  $((s), S)$ .<sup>[7]</sup> Since all  $T_j$  have depth at most  $k$ , this tree has depth at most  $k + 1$ . Furthermore, given any path  $p$  through  $T$ , by construction, the path must end in a node that exists in one of the  $T_j$ , which implies that  $d(p) \leq 1$ . It follows that  $T$

handles  $s$  and hence  $s \in X_h^{(k+1)}$ .



$\supset$ : Let  $s \in X_h^{(k+1)}$ . Then, there exists a Debate Tree of depth at most  $k + 1$  that handles  $s$ . Let  $((s), S)$  be the unique<sup>[8]</sup> edge descending from the root. For each  $s_j \in S$ , let  $T_j$  be the subtree growing out of  $s_j$ .<sup>[9]</sup> By construction,  $T_j$  has depth at most  $k$  and handles  $s_j$ , so (applying the Inductive hypothesis), we have  $s_j \in D_h^{(k)}$ . Then,  $S \in (D_h^{(k)})^*$  and  $S \rightarrow s$ , and hence  $s \in D_h^{(k+1)}$ .

### 3. Interpretation

At this point, we have a bunch more definitions and a theorem. Now, what does this mean?

Let's start with Debate Trees. A Debate Tree is actually a very natural object; it's what you get if you explain a subject in a hierarchical rather than a linear way. ( $X$  is true because of  $Y_1, \dots, Y_4$ ; then  $Y_1$  is true because [...].) It is very similar to [Elizabeth's](#) project of [breaking questions down](#). Notably, that project never mentions Factored Cognition; it's just presented as an epistemic tool.

In a better world, would textbooks use something like Debate Trees to explain proofs? I'm almost certain the answer is yes. There is no way that a purely sequential presentation of information is optimal. Our understanding doesn't work that way (compare [post #-2](#)).

There is one difference between Debate Trees and a hierarchical presentation of information optimized for being easily understandable. In the former, only one part is actually explored, which means that a Debate Tree doesn't mind having redundancy in it (by explaining stuff in more than one place). Conversely, if you optimize for understandability with respect to a single reader, you'll want to avoid redundancy. Nonetheless, they are very similar.

So much for Debate Trees. What about the theorem we've just proved? What does it mean?

Essentially, it means that Debate is nicely behaved in the limit. As both agents become stronger and the structure of the game becomes stricter, we approach a situation where the scheme can answer a question if and only if its answer can be recursively explained until there are no more difficult components. Even though the game results from two powerful agents applying optimization in opposite directions, the result is can be described without mentioning either one of them. Note that the same is not true for Iterated Amplification; even in the limit of perfect Factored Cognition, it is entirely possible that the scheme fails at a question for which an easy explanation exists.

Notably, the theorem stops being true if we drop the assumption that both agents are maximally powerful.<sup>[10]</sup> If the first agent is weaker, she might fail to find the best

explanation, which shrinks the set of statements Debate can handle. Conversely, if the second agent is weaker, she may fail to point to the most problematic statement, which enlarges the set of statements Debate can handle. Do these factors equal out? I'm not sure. One of the things that I haven't yet tried but may be reasonable is to model inadequacy and see whether this benefits the first or second agent.

It's worth pointing out that the Ideal Debate FCH doesn't talk about false statements. It formalizes the claim 'the first agent can always win by being honest' which leaves open the possibility that she can also win by being dishonest. (And if she could do both, she would presumably do what's easier or safer.) It is necessary but perhaps not sufficient to realize Factored Cognition with Debate.

I think focusing on the honest case makes the most sense. Nonetheless, the next chapter is about what happens if the first agent wants to defend a lie.

## 4. Relaxing the truth assumption

To model dishonesty by the first agent, we assume that

- In addition to  $S_h^T$ , there is a set  $S_h^F$  of false statements. We have  $S_h = S_h^T \sqcup S_h^F$ .<sup>[11]</sup>
- Given a question  $q$ , in addition to the honest answer  $a_q \in S_h^T$ , there is at least one dishonest answer  $\tilde{a}_q \in S_h^F$ . The first agent may defend this dishonest answer.
- Any explanation for a false statement needs to contain at least one false element, i.e., if  $(s_1, \dots, s_{n+1}) \rightarrow s$  and  $s \in S_h^F$ , then  $\exists j \in \{1, \dots, n+1\}$  such that  $s_j \in S_h^F$ .

Since the second agent is also maximally powerful, she can repeatedly point to a false statement in the explanation. Due to the third assumption made above, this becomes an invariant: the ability to point to a false statement in any one step leads to the ability to point to a false statement in the next step, and therefore, since she can point to one in the beginning, she can do so at every step in the game.

In particular, this means that, regardless of when the first agent ends the game, we will have that  $s_{\text{final}} \in S_h^F$ .

At this point, we have two conflicting intuitions. On the one hand, the judge should only declare the first agent the winner if she is *certain* that the statement to be verified is correct. This should only happen for simple statements; if  $s_{\text{final}}$  is complicated, she should declare the second agent the winner. Thus, the first agent should only be able to win the game by choosing a  $s_{\text{final}}$  that looks *simple and obviously true*, yet is false. This sounds extremely hard to accomplish.

On the other hand, we are, in some sense, pitching a human in a battle of wits against a potentially superintelligent AI, which sure sounds like doom. The superintelligent AI has to follow a very rigid structure, but... it still kinda sounds like doom. In particular, we probably expect that the first agent will navigate the game so cleverly that she only appears to follow the rigid structure while really doing something else. Crucially, *we lack the ability of the second agent to interfere through an equally powerful I/O channel* that exists in regular Debate.

There thus seems to be a tradeoff here, where at the one end, we have the 'stock Debate' protocol that enforces no structure, and on the other, the 'Ideal Debate' protocol that enforces extremely rigid structure. More structure means more concern with things like wireheading, but also stronger theoretical guarantees.

*Stock Debate*

*Ideal Debate*

*No restrictions to the kinds of statements both agents can make*

*Rigid recursive structure enforcing zooming into complex questions*

*perfectly symmetric output channels may provide protection against a wide class of behaviors that deviate from the intended structure (-> wireheading)*

*stronger asymmetry between difficulty of winning with true and false statements, but only as long as the intended structure is followed*

I strongly suspect that running literal Ideal Debate is a recipe for disaster since the first agent can break away from the format entirely and attempt to wirehead the judge. At the very least, it would require a way for the judge to prematurely declare a winner (in literal Ideal Debate, the first agent has complete control over when to end the game), and it would probably also require the second agent to claim that wireheading is occurring. On the other hand, I do have the intuition that one should aim to get as close to Ideal Debate as possible while managing these concerns.

## 5. The HCH-FCH

Ideal Debate has been amendable to a formal description because the human only makes decisions at the end. In HCH, the human is involved all throughout and, crucially, is in charge of decomposing questions. To have an accurate model, one needs to abstract away the entire process of the human decomposing the question as well as

any other cognitive work she might do on the question – but if this is done, there is nothing left to formally capture.

(And older version of this post used to have an attempt of formalizing it more, but I've since concluded that it can't be done right.)

To have something analogous to the Ideal Debate-FCH, we trivially define:

the HCH-FCH( $Q, h, t, \ell$ ):

$HCH_{h,t,\ell}$  can solve every question in  $Q$

## 6. Conclusion

This post concludes the first part of the sequence. There probably weren't any huge surprises so far. Is having a formalism useful?

I think so. One purpose of formalizing a setup is that it overcomes the [illusion of transparency](#). Without it, it is possible to think the setup is clear even when it really is not. I can take myself as one data point: my first attempt at coming up with a formalism looked different (and, I think, wrong). At least my past self did not understand the problem to the point that the formalism is trivial.

Anyway, at this point in the sequence, I want to defend the following two claims:

1. It doesn't make any sense to talk about the 'Factored Cognition Hypothesis'; there is no one requirement that works for both schemes.
2. The formalism of Cognition Spaces is *accurate*, in that it doesn't include anything that misrepresents Factored Cognition as implemented by IDA or Debate. It may be incomplete (e.g., there is nothing about defining terms, as people have pointed out on post #1, and maybe you could do more with false statements).

I think these are pretty conservative claims, even though the first clearly contradicts what I've heard other people say. If anyone doubts them, this is the place to discuss it. My conclusions from the second part are probably going to be a lot more controversial.

- 
1. A directed rooted tree, also called an [arborescence](#), is a directed acyclic graph such that there exists [a node from which there is exactly one path to every other node]. This node is called the root; it's unique because, if there were two such nodes  $x$  and  $y$ , there would be a path  $x \rightarrow y$  and a path  $y \rightarrow x$  and hence a cycle  $x \rightarrow y \rightarrow x$ . [↵](#)
  2. We continue writing  $s \in S$  to denote that  $s$  appears in the sequence  $S$ . [↵](#)
  3. The 'trivial tree'  $((s_0), \emptyset)$  for some  $s_0 \in S$  is a proper Debate Tree, according to this definition. It corresponds to the first Debate agent giving an answer  $s_0$  to the input question and deciding that this answer is already self-evident. [↵](#)

4. Note that the first element of this pair is a *path* as defined in Graph theory (through the Debate Tree, which is a graph) whereas the second is an additional element that denotes which statement in final explanation we end up in. ↵
5. In particular, one could model the same by having a difficulty threshold  $c_h$  for a human, such that the human can deal with all questions that are at most  $c_h$  hard.

However, the pair  $(d_h, c_h)$  is equivalent to the pair  $(d_h, 1)$ , where  $d_h$  is like  $d_h$  except that all difficulties are scaled by  $c_h$ . ↵

6. Given this, one could alternatively phrase the Ideal Debate FCH as

$$\forall q \in Q : \min_{T \in T(d_h, a_q)} d_h(T) \leq 1$$

Or in words, one could say, 'every question in  $Q$  can be handled by a Debate Tree'. ↵

7. This is a step where defining the nodes of Debate Trees to be explanations rather than individual statements makes things harder. For each subtree, one needs to replace its current root (that's a one-element sequence) with the node  $S$ . This can be done formally in terms of the underlying nodes and edge sets; it's just cumbersome. ↵
8. The edge from the root is unique because the root is a one-element sequence  $(s)$ , and each statement can only have one explanation (and thus only one edge) by the third condition of Debate Trees. This corresponds to the fact that the first agent only has to prepare one explanation for her initial answer; there is not yet a choice from the second agent involved. ↵
9. This step is the reverse of the combining step. The subtree growing out of  $s_j$  is the tree we get by taking the sub-graph out of  $S$  and replacing its root  $S$  with just  $(s_j)$ . ↵
10. Recall that there is, in fact, only one agent playing against itself. Thus, we can assume that both agents are always *equally* competent. ↵
11. The 'squared cup' symbol  $\sqcup$  means the same as  $\cup$  plus the information that the two sets are disjoint. ↵

# Clarifying Factored Cognition

Crossposted from the [AI Alignment Forum](#). May contain more technical jargon than usual.

This post is sort of an intermediate between parts 1 and 2 of the sequence. It makes three points that I think people tend to get wrong.

## 1. Factored Cognition is about reducing hard problems to human judgment to achieve outer alignment.

It's possible to lose sight of why Factored Cognition is employed in the first place. In particular, *it's not as a way to boost capability*: while the amplification step in IDA is implemented via Factored Cognition, the *purpose* of this is alignment: IDA would be more straight-forward (and [more similar to AlphaZero](#)) if each  $A_k$  were trained to

access

approximate  $[A_{k-1} \text{ thinking for longer}]$  rather than  $[H \rightarrow A_{k-1}]$ .

[Recall from post #-2](#) that I've framed the entire problem of AI risk as 'once systems become too capable, it gets difficult for a human to provide a training signal or training data'. There are many approaches to get aligned systems anyway: there's [ambitious value learning](#), there's [trying to develop an new framework](#), there's [impact measures](#), there's [norm following](#), there's [avoiding agents altogether](#), et cetera et cetera. But another option is to reduce the difficulty of providing the training signal to human judgment (and do so for each instance separately), which is what IDA and Debate are trying to do. This is what Factored Cognition is used for. In a previous version of this post, I've dubbed the approach 'narrow alignment proving' since the system repeatedly 'proves' that it gives true answers. Case in point, the proper way to view Factored Cognition is *as a tool to achieve outer alignment*.

In stock IDA, this corresponds to the fact that we get outer alignment of each  $A_k$  by induction, precisely because each amplification step is implemented via Factored Cognition. In Debate, this corresponds to the debate game itself. Take out everything after the first statement by the first agent, and you get precisely the classical Oracle AI setup.

## 2. Debate $\approx$ IDA + decomposition Oracle.

My impression has been that some people view IDA and Debate as quite differently. They can get a handle on IDA, perhaps because of its similarity to existing Machine

Learning techniques, but having two agents debate each other sounds exotic.

However, I think the proper way to view Debate, especially in the limit, is simply as stock IDA plus a decomposition Oracle. If you consider an HCH tree solving a problem, say deriving a math proof, you get a transcript that corresponds to a Debate Tree, the formal object I've introduced in [post #2](#). In particular, it's going to be a much larger, less elegant Debate Tree. But add a decomposition Oracle (that all nodes in the HCH tree can use), and the two things become almost identical.

In Debate, only one path of the tree really occurs when the scheme is executed; the rest remains implicit. But this is also analogous to IDA, where only the upper node really occurs. Both schemes have the human look at a tiny part of the Cognition Space (although they do differ on what part it is).

I think it is almost fair to say that Ideal Debate = HCH + Decomposition Oracle. In the concrete schemes, the = becomes an  $\approx$  since implementation challenges are different.

### **3. The way to evaluate feasibility of Factored Cognition is to look at the task of the human.**

Factored Cognition can seem hard to get a grasp on if it is viewed as an infinite process of decomposing a problem further and further, especially if we start to consider meta questions, where the task of decomposing a problem is itself decomposed.

However, in both schemes, Factored Cognition includes a step that is by definition non-decomposable. In Ideal Debate, this is step is judging the final statement. In HCH, it is solving a problem given access to the subtrees. This step is also entirely internal to the human. The human has to

- find an answer and its explanation (in HCH only; this is done by the scheme in Ideal Debate)
- verify that the explanation is correct<sup>[1]</sup>

Note that this is in line with the [formalism](#) from posts #1 and #2: statements have difficulties, and at some point, the judge needs to verify one. This works iff the difficulty isn't too high. This statement cannot be decomposed due to the way it was chosen (if it could, the first agent would have done so, and the judge wouldn't have to verify it).

I think a good way to think about the question 'is HCH capable of solving hard problems' is to take the task 'solve a problem given access to an oracle that can solve slightly easier problems', and consider it a function  $f$  of the difficulty  $x$  of the input

problem. Then ask, how fast does  $f$  grow as a function of  $x$ ?

- If  $f \in \Theta(x)$ , [\[2\]](#) HCH doesn't work. As the tree grows larger, the job of the nodes high up in the tree becomes more difficult, but the nodes in the tree have constant time budgets. In particular, solving a problem for nodes high up in the tree has the same asymptotic difficulty as solving them without using subtrees.
- If  $f \in \Theta(1)$ , there are instances of HCH that can solve arbitrarily hard problems .
- If  $f \in \Theta(\log(x))$ , no instance of HCH with fixed parameters can solve arbitrarily hard problems, but IDA may still be performance competitive.

The same framing also works for Debate, where  $f$  is the difficulty of judging  $s_{\text{final}}$ , and  $x$  is the complexity of the initial answer  $a_q$  to the input question.

---

1. It's also worth pointing out that the second step is the only part where mistakes can come in. In both idealized schemes, correctness of Factored Cognition comes down to a human verifying whether or not an implication of the form  $(s_1, \dots, s_n) \implies s$ , what we've called an explanation, is valid. [↵](#)
2. The notation  $f \in \Theta(g)$  is defined as  $f \in O(g) \wedge g \in O(f)$  and means that  $f$  and  $g$  grow asymptotically equally fast. [↵](#)



# Intuition

In the [previous post](#), I've said,

[...] in both schemes, Factored Cognition includes a step that is by definition non-decomposable. In Ideal Debate, this step is judging the final statement. In HCH, it is solving a problem given access to the subtrees. This step is also entirely internal to the human.

You can take this as a motivation for part two of the sequence, which is about how humans think. I think a good place to start here is by reflecting on the argument against Factored Cognition based on **intuition**. Here is a version [made by Rohin Shah](#) on the AI alignment podcast:

[...] I should mention another key intuition against [the Factored Cognition Hypothesis]. We have all these examples of human geniuses like Ramanujan, who were posed very difficult math problems and just immediately get the answer and then you ask them how did they do it and they say, well, I asked myself what should the answer be? And I was like, the answer should be a continued fraction. And then I asked myself which continued fraction and then I got the answer. And you're like, that does not sound very decomposable. It seems like you need these magic flashes of intuition. Those would be the hard cases for factored cognition. [...]

This sounds sort of convincing, but what is this intuition thing? Wikipedia [says](#) that...

Intuition is the ability to acquire knowledge without recourse to conscious reasoning.

... which I take to represent the consensus view. However, I don't think it's accurate. Consider the following examples:

1. I throw you a ball, and you catch it. We know that your brain had to do something that effectively approximates Newtonian physics to figure out where the ball was headed, but you're not consciously aware of any such process.
2. I ask you to compute  $5 \cdot 8$ . I predict that your brain just 'spat out' the right answer, without any conscious computation on your part.
3. I show you a mathematical conjecture that you immediately feel is true. I ask you why you think it's true, you think about it for two minutes, and manage to derive a short proof. We know that this proof is not the reason why you thought it was true to begin with.

It's evident that your brain is acquiring knowledge without recourse to conscious reasoning in all three cases. That means they would all involve intuition, according to Wikipedia's definition. Nonetheless, we would not defer to intuition for any of them.

This leads us to this post's conjecture:

## We defer to Intuition for thoughts we cannot explain.

Under this view, intuition has nothing to do with how you derived a result and everything with whether you can explain the result after the fact. This characterization fits all three of the above examples (as well as any others I know of):

1. The ability to catch a ball does not feel impressive, hence it does not feel like it requires explaining.<sup>[1]</sup>
2. You could easily prove that  $5 \cdot 8 = 40$  using a lower-level concept like addition, hence you would not defer to intuition for the result.
3. In this case, you might well defer to intuition initially, when I first ask you about why the conjecture is true, and you (intuitively) think it is. But as soon as you have the proof in hand, you would refer to the proof instead. In other words, as we change your best explanation for the result, our verdict on whether it is intuition changes as well, which shows that it can't possibly be about how the result was derived.

As an aside: the standard definition says 'intuition is [...]', whereas my proposed characterization says 'we refer to intuition for [...]'. Why? Because intuition is not a well-defined category. Whether we call something intuition depends on the result itself *and* on the rest of our brain, which means that any accurate characterization somehow has to take the rest of the brain into account. Hence the 'we refer to [...]' wording.

---

The classical view of intuition leads to a model of thinking with two separate modes: the 'regular' one and the 'intuition' one. This post asks you to replace that model with a unified one: there is only one mode of thinking, which sometimes yields results we can explain, and other times results we can't explain.

Provided you buy this, what this means is that we have dissolved the concept. Intuition isn't a mode of thinking, it's just something we say depending on our ability to explain our thoughts. So, that's great! It means we have nothing to worry about! Factored Cognition works! Haha, just kidding. It's actually closer to the opposite. Yes, there is only one mode of thinking, but that's because *all* thinking is intuition-like, in the sense that the vast majority of steps are hidden.

To see this, all you need to do is look at examples. Do you have access to the computations your brain does to compute the 'for-tee' thought that pops into your head whenever you read the symbols  $5 \cdot 8$ ? Now summon the mental image of a hammer.

Did you have access to the computations your brain did to construct this image? Or, you can go back to catching that ball. In all those cases (and others), our brain provides us zero access to inspect what it is doing. That's just how awareness works. Our brain shows us the *results*, but that's it. The algorithms are hidden.

I think this view is very compatible with a scientific understanding of the brain, and much more so than anything that positions intuition as a special category of thought.

But more on that in the next post.

---

Given the single-process model, let's revisit the Ramanujan example. What does this kind of thing mean for Factored Cognition?

The immediate thing the example shows that the [computations your brain can run without consulting you] can be quite long. Unlike in the  $5 \cdot 8$  case where your brain did something unimpressive, Ramanujan did something that most people probably couldn't replicate even if they had a month to spend on the problem.

Let's linger on this for a bit. In a private conversation, TurnTrout has called the 'computations your brain can run without consulting you' the 'primitives [of] your cognitive library'. I think this is a cool term. Note that 'primitive' indicates an indivisible element, so they are only primitives *from the perspective of awareness*. For example, a primitive of most people's library is 'compute the destination of a flying ball', and another is 'compute  $5 \cdot 8$ '. If you're a computer scientist or mathematician, your library probably has a primitive for  $2^{10} = 1024$ , whereas if you're a normal person, it probably doesn't, so you would have to compute  $2^{10}$  as

$$2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow \dots \rightarrow 512 \rightarrow 1024$$

And even then, some of these steps won't be primitives but will require a sequence of smaller primitives. For example, the step from 512 might be computed by  $500 + 500 = 1000 \rightarrow 12 + 12 = 24 \rightarrow 1000 + 24 = 1024$ , where each of those steps uses a primitive.

Similarly, if you're Ramanujan, your brain has a very complicated primitive that immediately suggests a solution for some set of problems. If you're a professional chess player, your library probably has all sorts of primitives that map certain constellations of chess boards to estimates of how promising they are. And so on. I think this is a solid framework under which to view this post. However, note that it's just descriptive: I'm saying that viewing your mental capabilities as a set of primitives is an accurate description of what your brain is doing and what about it you notice; I'm not saying that each primitive corresponds to a physical thing in the brain.

Then, just to recap, the claim is that 'it's intuition' is something we say whenever our primitives produce results we can't explain after the fact, and the concept doesn't refer to anything beyond that.

**EXERCISE (OPEN-ENDED):** If you agree with the post so far, think about what this might or might not imply for Factored Cognition, and why. Is it different for the [Ideal Debate FCH](#) and the [HCH FCH](#)?

Open-ended means that I'm not going to set a time limit, and I won't try to answer the question in this post, so you can think about it for as long as you want.

The reason why it *might* be a problem is that Factored Cognition likes to decompose things, but we cannot decompose our cognitive primitives. This leaves Factored Cognition with two options:

1. help a human generate new primitives; or
2. get by with the human's existing primitives.

The first option begs the question of how new primitives are created. As far as I can tell, the answer is usually experience + talent. People develop their primitives by having seen a large number of problem instances throughout their career, which means that the amount of experience can be substantial. A common meme, [which is probably not literally true](#), is that it takes 10000 hours to achieve mastery in a field. Anything in that range is intractable for either scheme.

This doesn't mean no new primitives are generated throughout the execution of an HCH tree, since all learning probably involves generating some primitives. However, it does exclude a large class of primitives that could be learned in a scheme that approximates one human thinking for a long time (rather than many humans consulting each other). By using someone who already is an expert as the human, it's possible to start off with a respectable set of primitives, but then that set cannot be significantly expanded.

- 
1. As a further demonstration, consider what would happen if you studied the ball case in detail. If you internalized that your brain is doing something *complicated*, and that we have no clue what's really going on, you might gradually be tempted to say that we use intuition after all. If so, this demonstrates that explanations are the key variable. [↩](#)

# FC final: Can Factored Cognition schemes scale?

*(Apologies for the long delay.)*

## Scaling of Regular Thought

The punchline of the [previous post](#) was that there is only one mode of thinking: your brain can solve various tasks in a single step (from the perspective of awareness), and we've called those tasks your cognitive primitives. All primitives are intuition-like in that we can't inspect how they're being done, and we may or may not have an explanation for the result after the fact.

We're now interested in how this process scales. We don't tend to solve hard problems by staring at their description and waiting for a primitive to solve them in one step, so some kind of reasoning backward is going on. However, there is no module for this in the brain, so our ability to 'reason backward' also has to be implemented by primitives.

The easiest way to observe how this works is to take a problem that is just barely too hard to solve with a single primitive. My pick for this is multiplying two 2-digit numbers. Thus, I invite you to do the following

**EXERCISE:** There is a simple (two 2-digit numbers) multiplication problem in the spoiler below. Make sure you have something to write; it can be a piece of paper or a digital notepad. Look at the exercise, solve it in your head, and write down every verbal thought that pops into your mind until you have the solution. Write only to document your thoughts; don't do a written calculation.

$$12 \cdot 17$$

.

.

.

.

.

.

.

Below is what my transcript looks like. You may have had more or fewer intermediate steps, repetitions, or unrelated thoughts in between. That's all fine.

$$12 * 17 \rightarrow 10 * 17 \rightarrow 170 \rightarrow 2 * 17 \rightarrow 34 \rightarrow 170 + 34 \rightarrow 204$$

The coloring is something I've added after the fact. The black thoughts (minus the problem itself) are the outputs of primitives that actually solve math problems. Those

are all simple; it's  $10 \cdot 17$  and  $2 \cdot 17$  and  $170 + 34$ . On the other hand,  $14 \cdot 17$  itself is outside the set of exercises that can be handled by a single primitive (at least for me). And thus, my brain has performed a feat of utter genius: instead of a primitive that sees the exercise and outputs a *solution*, it found a primitive which saw the exercise and output another exercise! (Namely,  $10 \cdot 17$ .) Subsequently, that exercise was taking as the input to a different primitive, which was then able to solve it in one step.

(It may be that some of the 'subproblem outputs' like  $10 \cdot 17$  did not appear as verbal thoughts for you. In general, not all outputs of primitives make it into awareness, and [the process that determines whether they do is complicated](#). You would probably observe the same patterns with a harder exercise.)

This suggests that a major part of thinking consists of applying primitives that output new subproblems.<sup>[1]</sup> Does this generalize to harder and/or non-mathy problems? I think the answer is almost certainly yes, even in mundane cases, provided that you don't solve the problem immediately. For example, suppose you have to decide what present to buy a friend for Christmas. This problem does have some potential to be solved quickly, given that there is a set of 'default' options, like sweets or a bottle of wine. But if you're not content with those, you're unlikely to passively wait for your brain to produce more ideas. Instead, you would ask things like "what would make her happy?" or "what are her hobbies?". If you think about it for a while, you might get to less obvious questions like "what kind of gifts don't have the property that she will know better what she likes than I do?" Maybe you would consider helping her solve a problem she hasn't bothered to solve herself, and that would lead to questions like "what has she complained about before?". And so on. Since the domain is no longer governed by a small set of explicit rules, the subproblems don't immediately and uniquely determine the answer as they do in the multiplication case. Nonetheless, they are smaller problems whose solutions constitute progress on the overall problem. In general, I think you will be hard-pressed to find an example where you think about something for a while without outputting subproblems.

## Factored Cognition vs. Regular Thought

Factored Cognition, [as defined by Ought](#), refers to "mechanisms [...] where sophisticated learning and reasoning is broken down (or factored) into many small and mostly independent tasks". In light of the above, I posit the sequence's final conjecture:



Regular thought  
uses Factored Cognition.

I've drawn a parallel between Factored Cognition and regular thought [all the way back in post #-1](#). The difference is that that post was taking the perspective of someone who already understands the problem and can choose between different ways of decomposing it, which is relevant for a Debate agent, but not so much for the human (in either scheme) who starts off not understanding the problem. The claim now is that the process of understanding does itself use Factored Cognition.

Consider a node at the top of an HCH tree (say with  $t = 1$  hour) on the one hand, and a single person thinking for a hundred years on the other. We can call them H and D, respectively. ('D' for 'iDeal' since this is an idealized setting from the standpoint of capability). Presumably, everyone would agree that H and D do something different when they try to solve a problem, **but this difference cannot be that H uses Factored Cognition because D does that as well**. The difference also cannot be that D only produces one new subproblem at a time since H does that as well: each new question she asks is allowed to depend on everything that has happened up to that point. In both cases, the 'decomposition' is a thing that is continuously updated, not a thing that is ever output in one piece.

So, what is the difference? If you buy that D would be superintelligent but are less sold on H, this is the key question, and the heart of this post will be trying to answer it. We can separate the ways in which H is disadvantaged into two distinct categories. I call them the alternating problem and the translation problem.

## The Alternating Problem

The alternating problem is the fact that H is restricted in how many times she can alternate between asking and solving. D has the time budget to iterate through millions of questions throughout her thought process, but H only lives for an hour. On the upside, while D may only make incremental progress on each question, H immediately receives the proper solution, provided the question isn't too difficult. We would now like to know how much value the answer to one such subquestion has.

Here is a model to ask this more formally. Suppose we can assign each question  $q$  a difficulty  $d(q) \in \mathbb{R}$  (this is very similar to the model from part I of the sequence).

Suppose further that we can measure H's progress on  $q$  with a real number  $y \in \mathbb{R}$  so that the question gets solved once  $y \geq d(q)$ . Now if H receives the answer to a subquestion, this will increase  $y$ . The question is, by how much?

One possible answer is **by a fraction of the input question's difficulty**, i.e.,  $c \cdot d(q)$  for some constant  $c$ . As the input question gets more difficult,  $H$  simply asks more promising questions, and it always takes about  $\frac{1}{c}$  many to arrive at a solution.

To test if this is realistic, consider the following three problems:<sup>[2]</sup>

Suppose  $a$  and  $b$  are real numbers, not both 0. Find real numbers  $c$  and  $d$  such that

$$\frac{a+bi}{a-bi} = c + di.$$

Prove that there does not exist an operator  $T \in L(R^7)$  such that  $T^2 + T + I$  is nilpotent.

Decide whether it is true that  $\forall n \in 2\mathbb{N} : n > 2 \implies (\exists p, q \text{ prime} : p + q = n)$ .

For the above to be true, it would have to be the case that, for all three questions, receiving the answer to a relevant subquestion gets you the same portion of the way to the solution. This is clearly nuts. If you ask 'how can I get the denominator of  $\frac{a+bi}{a-bi}$  to be real', you're almost there; if you ask, 'what does nilpotent mean', you've only done a small step; if you ask 'what's the smallest proven gap between prime numbers', you've presumably only taken an infinitesimal step.

On the other hand, asking the *correct questions* may get you there, but that's not what we're talking about.

So it's not a fraction of  $d(q)$ . A second answer is that it's **a fraction of the current progress**, i.e.,  $c \cdot y$  for some constant  $c$ . Every subquestion  $H$  asks has an answer whose usefulness is proportional to  $H$ 's current understanding of  $q$ .

For this to be true, it would have to be the case that understanding a problem leads one to ask better questions. I probably don't have to convince anyone that this is true, but just to hammer down how prevalent this mechanism is, here are five made-up examples from different contexts:

1. Anna tries to predict whether China or the USA will have more AGI capabilities in thirty years. After pondering various considerations, she realizes that she should figure out what proportion of each country's AI efforts goes to AGI specifically.
2. Bob tries to prove that there are infinitely many prime numbers. His approach is to assume there are finitely many and derive a contradiction. After thinking about this for a bit, he realizes that 'take a finite set, construct an additional prime number' is a less confusing problem that amounts to the same thing.
3. Carla wants to join her first-ever Zoom call but doesn't have a microphone. After considering various ways to acquire one, she realizes that her phone has one and asks whether Zoom could run on that.



4. Dana tries to find the next best move in a chess game. After studying various lines, she realizes that her opponent's light square bishop is crucial as it can trap her queen in most relevant lines. She now asks how to deal with the bishop.
5. You come up with a bunch of items that could plausibly be useful for one of your friend's hobbies, but all have the property that you would probably buy an inferior product to what she could buy for herself. You conclude that you should look for things that she likes but doesn't know more about than you do.

If the fraction-of-current-progress answer is correct, then H's progress  $y = f(t)$  ( $t$  is the number of questions considered) obeys the recursive equation

$f(t) = f(t-1) + c \cdot f(t-1)$ , which is simply  $f(t) = (1 + c)^t$ . (Of course, progress on any real problem is highly discontinuous and high-variance, so all of this is approximation.) In this model, progress is exponential in the number of questions asked. This also makes sense of why thinking for a very long time is powerful. Suppose that D only gets ~~1/100~~ as much utility out of each subquestion asked, given that she may only consider them for a few seconds. This still yields  $f(t) = (1 + \frac{1}{100}c)^t$ , which may grow slowly at first, but will arrive at something useful eventually because there is a large number in the exponent. Conversely, the abilities of an HCH tree are bounded. Up to some level of difficulty, nodes that receive perfect answers from their children can produce perfect answers themselves, so HCH can answer all such questions by induction. But there is some lowest level of difficulty for which it takes too long to build up an understanding, and a node won't be able to answer such a question even if all subtrees give perfect answers. This is the step on which the induction breaks down.

A relevant counterpoint here is the ability of H to ask meta-questions. A meta-question is something like "what is the best way to think about the question, "What Christmas present should I buy for Hannah?"". This is similar to "What subquestion should I ask to make progress on the question, "What Christmas present should I buy for Hannah?"". The ways in which the two questions are not the same relate to the subtleties of thought that this post mostly brushes over: there can be insights about a problem that don't directly translate to subquestions, there's thinking that's neither about asking nor about solving questions (such as repeating what you understand already), and so on. All of that stuff makes life harder for H (more things to be done in limited time with questionable help), which means that reality will look at least as bad for HCH as the simplified view suggests.

In the simplified view, the existence of meta-questions allows H to receive help in figuring out what subquestions to ask next. The problem is that there is no reason to expect HCH to be capable of solving the meta-question. If thinking is a constant alternation between posing and answering questions – where the questions and their answers become progressively more useful – then finding the perfect questions to ask should be roughly as hard as solving the problem outright. Less abstractly, take a look at the five examples of how progress informs future subquestions. Most of them involve past subquestions *and their solutions*. If the quality of subquestions is a function of current progress, then thinking about subquestions alone doesn't cut it. Making progress requires alternating between asking and solving.

I find that this result is supported by introspection. The current sequence looks nothing like what I had in mind initially. When I decided to spend time on this problem, the first thing I did was to ask 'what are questions about?', which led to a post called 'Target systems'. Another early post was called 'Dependency Graphs'. Both of those posts were answers to subproblems I had come up with; neither of them turned out to be good subproblems, but I wouldn't have realized this if I hadn't written them. Only through the alternation of asking and answering did I get to this point. The same process happened one level down: within one post, I regularly found that a question I was trying to answer wasn't coherent, and then I usually scrapped it and rethought what I was trying to do. If I were forced to stick with the question anyway (which is the analog of having the alternation problem), I expect it wouldn't work very well. It's also not the case that the decomposition only changed in the beginning; some structural changes have occurred fairly late, and I would change some earlier posts right now if they weren't already published.

## The Translation Problem

If we take D and add the alternating problem, we get a scheme where one person is thinking for a long time with the restriction that the decomposition on every level can only be updated a limited number of times. This scheme is not identical to H, so there is a second difference, which I call the translation problem. The translation problem is the fact that every insight communicated between two nodes has to be translated into text (or whatever other format the scheme is using) and back. If H calls a subtree that works for a total of 1000 hours, then H didn't think 1000 hours herself but merely receives the subtree's output. This problem goes both ways: it handicaps the results from subtrees, and it handicaps how much context a node can give to subtrees when asking a question.

More concretely, it has several consequences:

1. It makes learning new skills difficult. (This is what we've left on at the end of the previous post.) Whenever acquiring a new cognitive primitive takes too much time, it becomes impossible for H to acquire it. This precludes learning primitives that require a lot of examples. These are often the ones that we refer to as intuition.
2. It can leave value on the table because the subtree is missing context. Suppose H asks a subtree to answer question  $q$ , and the subtree asks another subtree to answer  $q'$  to help with  $q$ . It may be that  $q'$  and its answer are important for the overall problem (they may be more important than  $q$ ), but H never realizes this since all she receives is the finished answer to  $q$ . An example is the concept of Ideal Debate in this sequence, which I believe started as a footnote. Similar things can happen whenever a subtree misjudges which parts of what it found out are relevant for the overall problem.

3. It makes asking meta-questions throughout difficult. In light of this post, it would seem that asking meta-questions is something H would want to do as often as possible throughout the process. Yet, people tend to think of meta-questions as a thing that's only asked once, and the reason for this is the translation problem. A meta-question asked later in the process can't just be "what is the best way to think about this?" because that was already asked in the beginning. Instead, it has to be "what is the best way to think about this, given that I've already figured out xyz?" This is difficult to do, and it's also not in the spirit of Factored Cognition, which is supposed to be about independent questions or tasks.

Insofar as the third point is accurate, it implies that we're looking at a second fundamental restriction for H. The first is the alternating problem: the fact that the number of times H can flip between asking and solving is bounded. The second is that the total amount of time H can spend on thinking about new questions is bounded as well. For this to be acceptable, it needs to be the case that 'find the next relevant subproblem' is a task whose difficulty is bounded across every context.

On this point, consider the phenomenon of getting stuck. When thinking about something difficult, I sometimes reach a point where I feel like I'm no longer making progress. This usually doesn't last forever, which means that the sense of 'not making any progress' is not literally true, but it shows that finding the next useful subproblem can be difficult. In a world where bounded decomposition budgets are sufficient to solve arbitrary problems, getting stuck should not be possible. You could always come up with a new relevant subproblem and solve that – or if it's too hard, come up with a subproblem for that, and so on. In some sense, 'naive Factored Cognition' appears impossible because it relies on the idea that you can decompose everything, but figuring out the decomposition is a big chunk of the work, and that part appears largely non-decomposable. Speculatively, I think there may be the case that figuring out the decomposition isn't just a big chunk but actually *most* of the work. My experience of getting stuck is not 'please brain, solve this subproblem' but rather 'please brain, tell me another angle to approach this problem'.

## Conclusion

My tentative conclusion from all of this is that an HCH tree would not be superintelligent, with the usual caveat that brute-forcing isn't allowed. I'll operationalize this in terms of [strong-HCH](#) since this is what Paul considers to be the 'normal' scheme (whereas the thing the sequence has focused on is called 'weak-HCH'). In strong-HCH, each node has a list of all IDs of subnodes, allowing her to talk to the same instances repeatedly. Furthermore, messages can contain pointers to existing nodes (so if I'm node p, and I know that node x has insights on a part of a problem that I'm asking node y about, I can include a pointer to x in my question to y). I think one of the mistakes I've made in this sequence is to not focus on strong-HCH more. That said, strong-HCH doesn't seem to solve the problems I've talked about in this post, except for the one about missing context. Alas,

**Prediction (85%):** Ought will not succeed in demonstrating something roughly equivalent to solving the hardest exercise in a textbook using a structure that mirrors

strong-HCH, provided each person is unfamiliar with the material and has at most 30 minutes of time. Note that I'm making this prediction without any inside knowledge; I've just read what Ought has published.

Before writing the sequence, I think I would have assigned between 50 and 60 percent to the same prediction (I believe I was a bit more optimistic about Factored Cognition than most people, but there's some penalty since this could be hard to set up even if it's feasible), so there has been about a 30% swing.

Needless to say, if Ought does do such a thing, it will (a) mean I'm wrong and (b) be very good news.

## What about Debate?

ヾ(ツ)ノ

The reasons I've mentioned for thinking HCH wouldn't work don't apply to Debate (with one exception that I'll talk about in a bit). In fact, I'm yet to come across an argument that Debate cannot work in principle, and the formalism from the first part of the sequence is mildly encouraging. Of course, I also haven't come across an argument for why it must work, but it's harder to imagine that such an argument could exist, so the absence of evidence is altogether a good sign.

Most importantly, Debate sidesteps the alternating problem entirely. If you start with the best possible subquestions, then both of the toy models discussed in this post would agree that things should work out. Of course, the Debate agents don't perform surgery on the judge's brain to insert the perfect decomposition into memory; they have to write it down in text form. The amount that this matters, given that Debate agents are supposed to be highly intelligent, seems like a very hard-to-answer, very *different* problem from the things I've discussed in this post. I don't have too many intelligent things to say about it, except to repeat that talking about a 'Factored Cognition Hypothesis' really absolutely definitely doesn't make sense.

The aforementioned exception is the fact that the judge is highly limited in her ability to acquire new primitives. However, it seems like the ability to understand arguments fundamentally requires only a bounded set of skills. This is backed up by formal logic,<sup>[3]</sup> and we can see the same thing in practice with understanding mathematical proofs. Once again, there is no generalization of this point to a context where a human has to derive the proof.

My verdict is something like 80% that Debate won't fail due to fundamental problems (i.e., problems that relate to Ideal Debate). Note that this number is inflated because there is a chance that Debate would ultimately fail due to fundamental reasons, but we never get there because it fails due to practical problems first. I was a bit disheartened to read [the latest report on Debate](#), which indicates that one of those practical problems (the honest debate agent figuring out which claim of the dishonest agent contains the lie) appears to be quite serious. My estimate on Ideal Debate working out may be more like 60%, but that is not testable.

## Miscellaneous

Here is an example of how Debate can handle mathematical proofs. Recall the exercise I've mentioned earlier:

Prove that there does not exist an operator  $T \in L(\mathbb{R}^7)$  such that  $T^2 + T + I$  is nilpotent.

While this involves more advanced concepts, the exercise is still relatively easy. Here is a copy-paste from the solution I've written back then:

Let  $\lambda$  be an eigenvalue of  $T$  and  $v$  a nonzero eigenvector. (Use Theorem 5.26.) We have

$$(T^2 + T + I)v = (\lambda^2 + \lambda + 1)v = ((\lambda + \tfrac{1}{2})^2 + \tfrac{3}{4})v$$

So that  $(T^2 + T + I)v = \alpha v$  where  $\alpha = (\lambda + \tfrac{1}{2})^2 + \tfrac{3}{4}$ . Clearly  $\alpha > 0$ , hence

$(T^2 + T + I)^k v = \alpha^k v \neq 0$  for all  $k \in \mathbb{N}$ . Thus,  $(T^2 + T + I)$  is not nilpotent.

If this looks like gibberish, you're in the same position as a judge in Debate may be in. However, as a debate judge, you don't have to understand the entire argument. Here is a possible decomposition into claims, of which you will only have to verify one.

- **Claim 1:** There exists an eigenvalue  $\lambda \in \mathbb{R}$  with eigenvector  $v$  for  $T$ , where  $v$  is not the zero vector.
- **Claim 2:**  $(T^2 + T + I)v = (\lambda^2 + \lambda + 1)v$ .
- **Claim 3:**  $\lambda^2 + \lambda + 1 = (\lambda + \tfrac{1}{2})^2 + \tfrac{3}{4}$ .
- **Claim 4:** Set  $\alpha := (\lambda + \tfrac{1}{2})^2 + \tfrac{3}{4}$ . Then  $\alpha > 0$ .
- **Claim 5:** Claims #2-4 imply that  $(T^2 + T + I)v = \alpha v > 0$ .
- **Claim 6:** Claim #5 implies that  $(T^2 + T + I)^k v = \alpha^k v > 0$  for all  $k \in \mathbb{N}$ .
- **Claim 7:** Claims #1-6 imply that  $T^2 + T + I$  is not nilpotent.

Claim #3 requires only high-school math. If this statement is pointed at, you can verify it without engaging with the concepts 'eigenvector' or 'nilpotent' or even 'vector space'. The same is almost true for claims #4 and #6. Claim #5 requires being comfortable with equations, but not anything specific to Linear Algebra. Claim #1 requires looking up a theorem but not understanding why it is true.<sup>[4]</sup> Only claims #2 and #7 require explaining one or more of the field-specific concepts.

Another point I want to make is that this is probably not the optimal decomposition. When translating a text-based proof into a [Debate Tree](#), one need not do it sequentially. Here is a different approach:

- **Claim 1:** {same as above}

- **Claim 2:** There exists an  $\alpha \in \mathbb{R}_+$  such that  $(T^2 + T + I)v = \alpha v$ .
- **Claim 3:** Claims #1-2 imply that  $T^2 + T + I$  is not nilpotent.

Subsequently, Claims #2-5 from the previous decomposition can become #2.1-#2.4, and Claim #6 can become Claim #3.1. This decomposition is superior from the standpoint of [hiding complexity](#). I think it's fair to say that the primary mechanism for an Ideal Debate agent is to reduce a concept to its behavioral properties. In this case, that concept is the  $\alpha$ . The behavioral properties are entirely given in Claim #2 of the second decomposition, and they are sufficient for the high-level argument. Only if the other agent doubts the existence of such an  $\alpha$  does the debate have to open this black box and look at how  $\alpha$  is constructed. In that case, that's still a win in that the judge doesn't have to bother understanding if and how this  $\alpha$  solves the exercise (because if claim #2 is pointed at, claim #3 is not).

## Appendix: the sequence in 500 words

Since there was this big gap between the previous post and this one, I thought it might be useful to write an ultra-abbreviated version to refresh everyone's memory.

**Post #-1:** To characterize what constitutes 'solving a subproblem', as supposed to 'making progress on a big problem', one can look at the length of the subproblem's solution. Under this view, decomposing problems is all about hiding as much complexity as possible. It must be the case that we do something like this in regular thought because we can only keep a few objects in mind at the same time yet are able to solve complex problems.

[Post-sequence edit]: This perspective assumes a bird's eye view of the problem, which makes it primarily applicable to the job of an honest debate agent, less so to a human who starts off not understanding the problem.

**Post #1:** HCH is the ideal of stock amplification. It abstracts away a number of practical problems and implementation details. We can similarly define an ideal for Debate. Given these idealized schemes, we can define and study a formalism ( $\rightarrow$  Cognition Spaces). The formalism suggests that HCH and Ideal Debate don't necessarily scale similarly, which means there is no one Factored Cognition Hypothesis.

**Post #2:** Here are some things we can do with the formalism. Debate seems nicely behaved in the limit. Debate Trees may be an interesting object to consider when thinking about how to explain things optimally.

**Post #3:** Factored Cognition is about reducing hard problems to human judgment to achieve outer alignment; it's not used because it's the best way to boost capability. Ideal Debate = HCH + Decomposition Oracle. To evaluate HCH or Ideal Debate, consider the task of the human as this is the non-decomposable part.

**Post #4:** People tend to talk about intuition as if it's a separate mode of thinking that works without access to 'conscious reasoning', but really all thinking is like that; it's just that sometimes we can explain our thoughts, and sometimes we can't. It's useful to

think about human thinking in terms of the set of operations that can be done in one such step. We call these operations our cognitive primitives.

**Post #5:** This whole decomposing problems thing that characterizes Factored Cognition is something we do all the time when we think, except that we constantly alternate between decomposing and solving. You can verify this by taking an arbitrary problem and observing what your brain is doing. Since we only output one subproblem at a time, the term 'decomposition' describes a thing that is continuously updated, not a thing that's ever output in one piece. The alternating thing seems like it's critical, which is bad for HCH. Also problematic is the fact that nodes in an HCH tree have to communicate with something like text. In particular, it will mean that nodes probably won't get a lot of help for the task of decomposing their problem. This seems bad if you believe that decomposing constitutes much or even most of thinking. Strong-HCH may help, but probably not by much. Most of this stuff doesn't apply to Debate. There is no one Factored Cognition Hypothesis.

---

1. Note that when I say 'applying', I'm not suggesting a dualistic picture where there is an additional thing in the brain that gets to choose where to apply the primitives. [↵](#)
2. The first is the first exercise out of [my favorite textbook](#), the second is an exercise out of chapter 9 of the same book, and the third is a famous open math problem called the twin prime conjecture. [↵](#)
3. There are formal proof systems that posit a small set of primitive operations such that every proof is reducible to a sequence of such operations. This is what allows proofs about what is provable. [↵](#)
4. Theorem 5.26 is "Every operator on an odd-dimensional real vector space has an eigenvalue." (Incidentally, this is the only thing for which the 7 in  $\mathbb{R}^7$  matters. It could have also been  $\mathbb{R}^{1439995}$ . This is another aspect that may have made it more difficult to find a proof because it has the potential to be misleading, but barely matters for verifying the proof.) [↵](#)