Universality and consequentialism within HCH

Paul Christiano Follow
Jan 9 · 9 min read

The scalability of iterated amplification is closely tied to whether very large teams of humans are "ascription universal." If they are, we could hope that their approval would be an adequate training signal for powerful aligned behavior, and that we can approximate this signal with iterated amplification.

There are many reasons that a team of humans could fail to be universal. For example:

1. They could be unable to understand some kinds of effective reasoning, because they can't represent relevant concepts, understand particular inferences, or whatever else.
2. They could make errors that derail an otherwise-universal reasoning process.
3. They could use processes that work "most of the time," but fail catastrophically for a small number of inputs.
4. Errors of type #2 or #3 might produce extreme outputs that tend to propagate through the reasoning process and compound into larger failures (even if they occur with small probability or for a small fraction of inputs).
5. In a large enough computation, errors of type #2 or #3 could eventually give rise to intelligent patterns of behavior that systematically identify and exploit mechanisms by which they could propagate themselves or cause cascading failures. For intuition, note that memetic selection amongst a group of 10^{100} humans could be much stronger than the genetic selection that gave rise to life on Earth.

(This is not an exhaustive, exclusive, or even properly parallel list.)

My current view is that #2 and #3 can probably be corrected by traditional methods, even when compounded by #4: computing carefully or redundantly, sanity-checking outputs, structuring

computations to be more robust to isolated failures, and so on. These problems seem be easier to solve for smarter teams(though also more problematic if left unchecked), and I suspect there is some competence threshold above which they become manageable.

A priori, #2 or #3 compounded by #5 could be harder to cope with:

- An intelligent optimization process could systematically subvert whatever error-correction or robustness mechanisms we try to design.
- For "random" failures, it's easy to imagine that a sufficiently diverse and extensive set of checks can drive failure rates exponentially low, but malicious optimization is a natural source of highly correlated failures.
- Problem #5 potentially becomes *worse* as we move to larger and larger teams, because such teams may give rise to larger and larger adversaries. This means that our defenses are engaged in a race, instead of fighting against a fixed target which we might expect them to eventually overcome. (Absent an adversary, we still have a "race" in that we need to drive failure probabilities lower and lower for larger teams, but if we get exponential or even linear scaling then this seems feasible.)
- Intuitively, I would feel a bit surprised if random errors and propagating patterns managed to totally overwhelm the optimization being applied by a very large team of humans. But I would not feel so surprised if a large team of humans gave rise to other misaligned optimization power, and if that misaligned optimization eventually became the "main affair." This is a little bit like the difference between concerns that could extinguish all life on Earth (which seems quite hard to me) and those that could replace life on Earth with more effective competitors (which feels very possible).

This post

I'm going to argue that problem #5, the appearance of consequentialism within HCH, might be manageable. That is, if HCH would be ascription universal if not for the possibility of malign consequentialism within HCH, then we can probably make it universal even in light of that possibility.

The basic idea is to apply a "sanity check" to each of HCH's computations. If a computation is optimized for negative

consequences, then by ascription universality HCH can identify those consequences, and any output affected by such a malicious computation can be censored. In practice I expect this to be contiguous with the use of reflection to sanity-check large computations (and similar mechanisms may be the best way to protect against more prosaic cascading failures).

The arguments in this post are not tight, and indeed it's not clear they could be totally tight without clarifying the assumption "HCH could be universal if not for the possibility of malicious consequentialism." But I find these arguments suggestive enough that I think the current focus should be on problems #1 and #3—whether it is possible for HCH to robustly understand complex reasoning even in principle—without worrying about the possible emergence of consequentialism. If this is successful, I think there is a good chance that we could address issues #2, #4, and #5.

Filtering

Given a "transcript" τ of HCH(Q, N)'s behavior, i.e. a record of all of the questions, subquestions, and answers, we can ask two questions:

- Info(a, a', τ) : "Which of the two arguments a and a' is more useful for determining whether the transcript τ contains problematic subcomputations?"
- Trouble(a, τ) : "in light of argument a, does transcript τ contain problematic subcomputations?"

Now we'll jointly define filtered-HCH(Q, N) and Filter(τ, N) as follows.

Define Filter(τ, N) as:

- Find the a^* maximizing filtered-HCH(Info(·, ·, τ), N).
  (Interpret $\perp$ as a tie.)
- Run filtered-HCH(Trouble(a^*, τ), N) 3 times, and return the majority vote.
  (Interpret a response of $\perp$ as a judgment that τ is problematic.)

Define filtered-HCH(Q, N) as:

- First call HCH(Q, N) to get an answer A with transcript τ. But every time you would call HCH to answer a subquestion, call

filtered-HCH instead.

- Call Filter(τ, N-1) to test whether this answer is problematic.
- If τ is problematic, return a special symbol ⊥, otherwise return A unmodified. (If you return ⊥, you can eliminate this subtree from the transcript.)

Within HCH, if a question is answered with ⊥, the question-asker needs to respond appropriately. They could try re-asking the question in a different way; if that is unsuccessful they might assume that the question itself represents an attack on HCH and so needs to avoided. What to do next depends on the context—HCH might decide that a particular hypothesis is "malicious" and therefore best dropped from consideration; or that a particular idea is hazardous to think about directly and so it needs to engage in indirect reasoning to try to mitigate the harms; or it might just pass the failure up the tree with some explanation of where the problem appeared.

Note that the pattern of which questions get censored could itself cause trouble if it were maliciously optimized. However, achieving that would require a malicious computation to sometimes decide *not* to get the surrounding computation censored, which shouldn't be possible. That is, if a computation thinks to itself "should I do something malicious in order to cause the computation to be censored?" and concludes "no" in order to cause trouble, then that should already be grounds for censoring.

Informal analysis

Assumption

I'll assume that using HCH to find a maximally useful argument a^* about Q, and then using HCH to answer Q given a^*, is ascription universal—or rather would be universal if not for the possibility of malicious internal consequentialists.

Moreover, I'll assume that it's universal with some small blow-up, i.e. that if we give HCH a budget k N, then it epistemically dominates any computation that could be simulated by HCH with budget N.

Note that this is a very strong assumption. This verifier receives the argument a^* that they'd evaluate most highly, and so this will fail if there is *any* argument that would cause them to give a high score and behave badly. I'm trying to modify HCH to be robust to the possibility

of internal consequentialists, but there are all sorts of other ways that a carefully chosen input could compromise a system that is "mostly" universal, which I'm assuming are already resolved.

Hope

Why might we expect this filtering procedure to fix problems caused by malicious consequentialists?

Then the rough hope is:

- If there is a consequentialist whose behavior is optimized to break HCH, then it "knows" something about why that behavior will cause trouble.
- So by ascription universality, filtered-HCH(Trouble(a^*, τ)) also knows why the behavior within τ will cause trouble.

This is only plausible once the original human H always behaves in the "intended" way and never performs malicious optimization, which I'd hope to achieve by only showing them inputs from some small set. So any malicious consequentialist is implemented "on top of" these humans rather than within.

(Easy mode) Handling stochastic failures

Suppose that HCH(·, N) has internal malice with probability $\varepsilon \ll 1$ on some inputs. This is a special case of the full problem (where errors occur with small probability) and the solution won't be reused to solve the more general problem, so an impatient reader should skip ahead. I'm including it because I feel quite uncertain about the general solution, and I think this simpler solution already shows some of the upside of filtering and has a reasonable chance of being useful.

If we only want to cope with stochastic failures, the easiest approach is to use a slightly modified definition of Filter:

- Use HCH(·, kN) to evaluate Trouble(a^*, τ) and Info(a, a', τ), rather than filtered-HCH(·, N).

We hope that HCH(·, kN) has a probability at most $k\varepsilon$ of encountering malicious internal computations. This seems intuitively plausible, since it is k times larger than HCH(·, N).

Internal malicious computations can affect Filter(τ, kN) in two ways:

- HCH(Trouble(a, τ), kN) may behave badly with probability $k\varepsilon$. But the only possible "bad" behavior is to give the wrong answer, either a false positive or false negative. If we take a best-of-3 vote, this probability is reduced to $k^2\varepsilon^2$, so we have significantly reduced the failure probability as long as $\varepsilon \ll 1/k^2$ (even if we consider a false positive, leading a subcomputation to be censored, to be as bad as an arbitrary failure).
- HCH(Info(a, a', τ), kN) may behave badly, which affects which argument a^* will be optimal. However, we still get a distribution over computations which is "almost optimal" in the sense that it is preferred to every other argument with probability at least $1/2 - k\varepsilon$ (while the optimal argument would be preferred with probability at least 1/2). It seems reasonable to assume this small difference doesn't affect the universality of HCH (and we can drive this probability exponentially close to 1/2 by ensembling).

This seems like a significant improvement over my previous proposal for reliability amplification because Filter(τ, N) returns yes or no, even if Q itself has complex answers. This lets us use a simple majority vote to improve reliability. In particular, we no longer require that no agent in the ensemble be too much smarter than the others—we have shifted a vague assumption about relative capabilities to a slightly more precise assumption about universality.

This mechanism only fixes errors that occur with sufficiently small probability. But we can apply it throughout a large tree, reducing the error probability frequently enough that it never gets large.

(Hard mode) Handling "bad" questions

The problem is much harder when there are "bad" questions Q for which HCH has a significant probability of encountering malicious consequentialism and failing. If τ is the transcript from a question where HCH failed, Trouble(a^*, τ) may well be another question on which HCH predictably fails (e.g. because the same vulnerability is triggered by the attempt to diagnose the original failure). This could in turn prevent Filter() from censoring the original failure.

In this case we are going to argue inductively based on budgets: we can assume that filtered-HCH(·, N-1) probably has no malicious consequentialism by induction, and hence is universal.

Because we want to use an induction, it's important that we answer Trouble(a^*, τ) and Info(a, a', τ) using a *smaller* budget than was used in τ. But our assumption about universality requires the reasoner to be k times *bigger* than the computation it wants to compete with.

I hope that if there is malicious consequentialism within HCH(Q, N), then it needs to be "small" enough that we can rely on filtered-HCH(Trouble(a^*, τ), N-1) to compete with it. Intuitively:

- None of the unfiltered subquestions in HCH(Q, N) involve any malicious consequentialism, so we hope each subquestion either "behaves as intended" or produces the special symbol ⊥.
- If all subquestions of HCH(Q, N) are behaving as intended, and if the human at the root of HCH(Q, N) is behaving as intended (which we attempted to guarantee by limiting their input space), then any malicious consequentialism is very tightly constrained.
- Intuitively, it's restricted to two options:
  (a) be computed implicitly with "side effects" of the intended behavior, repurpose valid parts of the computation for unintended effects, *etc.*
  (b) make use of the intended behavior of questions
- When using option (a), it feels like the malicious optimization will be significantly less powerful than HCH with the same budget. For example, when I'm answering a question, it seems like most of my useful computational work is going into actually answering the question; if someone wanted to try to do computational work based on details of how I formatted my answer, I think they'd get less done. So we can hope that this part of the malicious consequentialism could be simulated by HCH with total budget much less than N.
- When using option (b), the malicious consequentialism can be very efficient—exactly as efficient as HCH itself, since it's just using that computation in the intended way. But filtered-HCH(Trouble(a^*, τ)) may not need to understand these parts of that computation in order to do its job. Instead it can look at the transcript τ and take the relevant answers at face value, just like the malicious computation is doing. (This relies on the inductive invariant, and even with that isn't obvious, but it looks pretty plausible to me.)

Note that this strategy really leans on the assumption that HCH works well for *every* input. In addition to the fact that a^* is

optimized, Filter calls HCH with a large and potentially malicious transcript τ as its input. The distinction between "untrusted input that you consult when appropriate" and "subcomputation that has a privileged position from which it can cause harm."