Machine intelligences: a research proposal

Asvin G

April 16, 2025

With the recent, extremely surprising, success of machine learning and artificial intelligence, my interests have sharply pivoted towards questions around the nature of intelligence, consciousness and self-organization of collective intelligences. I began thinking about these questions almost two years ago, and my thinking through this time has evolved significantly along with increasing experience with AI and their increasing capabilities. My ideas have been significantly influenced by Joscha Bach, Michael Levin and Janus (@repligate on twitter), but also goes beyond them in various ways. I would like to go through some of the most compelling questions in my mind today and where my thinking on them stands now. While most of the sections are embryonic research directions, Section 4 contains the most fleshed out direction for further progress on improving machine intelligence.

As a rough starting point, we can define intelligence as:

Definition. *the capability of the agent to create good models of the environment it can interact with.*

Crucially, this definition makes no claims about the *goals* of the agent, and indeed the relation between intelligence in the above sense and emergent goals is one of the most pressing questions today, impinging on everything from how AI finds use in society to future, existential consequences for humanity.

Also importantly, the definition circumscribes the world the agent operates in to its capabilities for interacting with said world. I believe this is a very important point in understanding the nature of intelligence and I will argue later on that this sheds important light on the capabilities of current day AI (including, but not limited to, LLMs).

1 Intelligence as compression

In this section, I want to explore what claims, if any, can be made in general about intelligent agents, and more specifically, about the intelligence of agents optimized to solve a specific goal (such as predicting a missing word, or winning a chess game). The following hypothesis will serve as a starting position for our exploration.

To be precise, by *problem space*, we will mean simply any set X (or set with structure such as a manifold equipped with a fitness function) and by an agent, we mean a dynamical system $T : X \to X$. In other words, points on X encode the "state" of the agent and the evolution function T encodes the dynamics of the agent. Moreover, one might invoke an "energy" functional $E : X \to \mathbb{R}$ and the "goal" of the agent is to minimize E (which might correspond to max-

imizing fitness).

Hypothesis. *Any intelligent agent has to have some internal representation $\mathcal{R}$ of the problem space X based on which it can compute the "optimal" updating function T.*

This is only a preliminary hypothesis and therefore, I would like to be appropriately vague about the various terms "internal representation" and "compute". Nevertheless, there is strong evidence that something like this must be true. For instance, one can prove theorems [2] showing that if an agent X wants to control the dynamics of another system Y and do so well (in a precise information theoretic sense), then X must contain a *good* model of Y.

We can also locate such good models (or concepts) in current day AI systems as a variety of papers show in chess ([3] [7]...), Markovian prediction nets [8], LLMs ([5] as perhaps the most impressive amongst an entire subfield called interpretability research) and many more.

Due to physical constraints, it is often not possible to have a perfect representation of X and $\mathcal{R}$ necessarily has to lose information and moreover, computational resources have to be invested in the process of decryption or decoding the model's outputs in real world actions.

This hypothesis stands in contrast to a long philosophical tradition (especially in the west) where "thinking" is taken to be a perfectly rational, transparent process governed by logical rules and perhaps some heuristics. On the contrary, I think the latest advances in machine learning suggest a contrary view: intelligence is often a complete blackbox, even to humans, and consists of constructing an internal computational model which *produces* good responses to input stimuli (or hypothesized scenarios). In light of this, I think the following conjecture deserves serious investigation.

Conjecture. *Compression is all you need for intelligence.*

Unpacking the above conjecture a little, I think it is plausible that any *sufficiently* good compression of *sufficiently* interesting data will naturally have good generalization properties. The motivating examples are GPT style LLMs. In a very concrete sense, GPT-4 is a good compression of its training corpus - it recovers this corpus with high fidelity when prompted and this was the only metric on which GPT was trained. Surprisingly, GPT has excellent generalization qualities to all sorts of unexpected domains such as coding, general pattern recognition [4] etc.

The conjecture suggests two main lines of enquiry. First, it is clear that the problem domain that we are operating in is of paramount interest. Trying to compress random noise will surely not lead to anything interesting! Similarly, theoretical computer science suggests that good encryption is almost certainly possible and such encrypted data should in principle, make learning patterns in the data computationally infeasible.

Question 1. *Given these facts, what about "real world data" makes it so amenable to compression (and learning)? And conversely, how can we tell that a trained system does have a* sufficiently *good compression and how was this compression learnt?*

Some recent research in the machine learning literature suggests some sufficient features for such good learning. For instance, GPT type models exhibit few-shot learning, i.e., learning new concepts from a few examples with-

out changing their internal weights. This recent paper [1] investigates whether certain kinds of training data help learn this skill. They find that indeed, few-shot learning emerges when the training data has "clumps" or "burstiness" where similar concepts occur together. Similarly, it also helps to have large numbers of rarely occuring features that need to be learned (in the training set). Perhaps a more general hypothesis, with broad observational evidence across machine learning and biology is:

Hypothesis. *A changing environment leads to generalized learning with modular intrinsic concepts.*

The cognitive science literature in particular seems to be prime mining ground here. The question of what exactly *mental representations/concepts* are seems to have much in common with the question under consideration here. For instance, [6] develops "*a theory of the underpinnings of symbolic cognition that shows how sub-symbolic dynamics may give rise to higher-level cognitive representations of structures, systems of knowledge, and algorithmic processes*" in an explicit enough manner to be computationally implementable. In other words, this paper attempts to build a learning system ("a language of thought") that can learn arbitrary patterns.

Question 2. *Can we identify how a "language of thought" is implemented in biological or artificial intelligence?*

2 The Philosophy of Cognition

The recent successes in machine learning almost force upon us the conclusion that we have indeed found a completely general, algorithmic procedure to learn in any domain. However, as of yet, the science of learning is underspecified and ill-defined so our theory is very much behind practice. While we can point at the thing, we cannot say what we are pointing at!

As a mathematician, the practice of mathematics seems like a fertile ground for a first attack upon this philosophical problem. It is a relatively proscribed domain with very limited interactions with the broader world and with well specified rules and a long history of both practice and philosophy on the subject. Moreover, it is absolutely fundamental to all human thought and theory building. But most of all, it is the domain I know best!

In thinking through what I am doing when I perform mathematics, my tentative current stance is a synthesis of platonism and formalism with an important element for creative thought. I elaborate on these ideas at greater length here but in short, it is a philosophy that emphasizes the act of *creating* a model of the environment one interacts with. Within mathematics, all such interactions are proscribed to *computation*, and therefore one might plausibly describe mathematics as the "physics" of computation. Similar to how physicists create *theories* of reality based on their interactions with the physical world (experiments), mathematicians create *theories* of reality based on their interactions with the mathematical world (computations). And just as with physics, the emphasis is on *understanding*, not the computation or experiments although these are the crucial facts *to be explained.*

The relevance for a broader theory of cognition is, I believe, exactly in the significance placed on the methods of interaction allowed. It is easy at first sight to say that a LLM is "disembodied" because it doesn't seem to interact with the real world, however I think this framing misses all the important phenomenon. It is more accurate

to say that a LLM is *specially* embedded in the world - it only has access to it through text, and the feedback loops are entirely (mis)predicting a given token. However, because input text the LLM is trained on is a reflection of our world, the LLM manages to learn approximate, important models of our world too. With some thought, one sees that the situation with LLMs is not significantly different from the situation with human learning - our brain only receives a sequence data of nervous impulses form our sensory data - it is hard to make the case that these nerve signals are a "true" encoding of reality in any way! Nevertheless, we do manage to learn some important features of our reality, but crucially, these are features important to scales at which our everyday interactions take place. This is precisely why quantum phenomenon or higher mathematics are un-intuitive (unless one trains specifically for them).

Moreover, this framing suggests several research questions focusing on the *differences* between humans and machines for learning. For instance

1. Humans have continuous chains of interaction and feedback with a *locally stable* environment, as opposed to a LLM which might predict tokens across a wide range of textual data with no coherence. It seems plausible to me that exactly this difference is responsible for why human three year olds are very coherent and rarely lose context, while LLMs take a long time to learn coherence, and often learn style before coherence (such as with earlier models of GPT). They are much better imitators before agents.

   Indeed, one has to train these systems specifically to behave like coherent agents (post training).

2. Humans set their own goals during training, and moreover, iterate and refine their goals throughout their lives. They also interact, from extremely early on, in environments with several distinct intelligent actors. Together, these effects seem to me to explain why humans learn metacognition so early and so well compared to LLMs, which seem to learn a notion of self only during post training. For example, one observes that LLMs during training seem to internalize *all* LLM generated data in the training set as "self-produced" which is quite a remarkable finding and not well explained currently. For more research along these directions, see here for example.

This last point leads us into the topic of our next section.

3 Machine Consciousness and Personality

Consciousness is an infamous word within science because of its inherent subjectivity and lack of agreement on even what exactly it should refer to. Nevertheless, it is no longer a topic that we, as a community, can afford to ignore any more as our artificial systems take on more and more of the appearance and behavior of intelligent agents but we have not made any significant progress on answering the question of *What is it like, if it is like anything, to be GPT-4?*. As a first step, I propose the following hypothesis.

Hypothesis. *Conscious experience is the internal interpretation of interactions between a self-model and the rest of the system within the simulated world-model of an intelligent agent.*

In other words, experiences such as "seeing red" and "feeling happy" are the interpretations of the corresponding neural correlates within that part of our mind that we identify as "self". Once again, we place *interaction* and a *relative notion of reality* at the center of our understanding. This provisional definition suggests various questions:

Question 3. *Under what conditions does a self-model develop?*

and

Question 4. *What purpose does consciousness serve in humans?*

While these questions are far from having definitive answers at the moment, I tentatively suggest that the self-models develop when learning in environments containing multiple distinct agents and crucially, *there is a relatively stable part of the environment whose observed actions can be predicted using neural signals originating purely within the brain.* In other words, there is a stable entity (through time and space) whose actions can be predicted before they can be observed. As an aside, I believe our experience of free will stems from exactly those actions of our "self-organism" that can be so predicted.

As for the second question, I consider it much more important since my proposed answer has implications for much more than consciousness. I believe that a self-model and the resulting conscious experience are useful in order to build a *compositional model* model of ourselves relative to the environment which can be interrogated in order to improve and iterate quickly. One of the major inefficiencies in learning through gradient descent is that *all* weights are simultaneously updated on any input data, and *all* the input data is considered equally important for this update. This would be like learning to play tennis and considering every possible input as a potential reason for why our last shot misfired - including a crow flying in the distance and a couple cycling by the road.

Translating this into machine learning, I believe that one could both improve learning efficiency and build in a self-model into our AI systems by controlling the scaling factor for weight updates through a distinct neural network. Currently, we do modify the speed of gradient descent in various ways but these are hard-coded in by human experimentation instead of being learnt naturally. Of course, it is possible (even plausible?) that certain sub-circuits of our largest models implement such a protocol although it must necessarily be fairly inefficient due to having limited and indirect control over the learning process.

An adjacent but not exactly the same topic is *Machine Personality.* We have found that each large model often comes with a distinct personality, a distinct set of wants, likes, dislikes, behaviours it tends towards or away from and so on, and that the resulting personality is often hard to predict or control. The leading figure in this domain is Janus (@repligate on Twitter/X.com), and his investigations have revealed many rich phenomenon and differences between the current leading models. A basic mode of investigation is to let various models interact over long periods of time in a common chat room (with or without human interaction), and to study the resulting outputs much as one would understand human personalities but adapted to LLMs. In many ways, such research is the anti-thesis of understanding LLMs through benchmarks ("standardized tests") and short input/output pairs. By its very nature, such re-

search is hard to summarize and make concrete, but they have been of central importance in large parts of my understanding of these models.

4 Learning as Reshaping the Search Landscape

In this section, I would like to focus more on problem-solving and how intelligence (as defined above) relates to it. I believe that all problem solving consists of two parts - an explicit search over strings of discrete tokens (as in alpha-beta search in chess playing machines or reasoning in humans and the latest model of LLMs) and an implicit "policy function" that guides the search (the role of the deep neural network in alphago, intuition in humans or the probability distribution the LLM samples from at the final stage).

In simple enough domains, one might be able to execute an explicit search over all possibilities but alas, most domains are not nearly so simple. This is the reason that we need the second component: the implicit policy function can be seen as an effective *reshaping of the search landscape* in order to trade off the exponential growth of possibilities for something more manageable, while at the same time hopefully not missing the target completely. Through this lens, *learning* consists of finding a good policy function so that the relevant search chains are relatively short and possible to find. This is the stage at which our latest AI currently exists - by adding in an explicit search (i.e., reasoning or chain-of-thought), we gain the ability to search for solutions that are not immediate.

However, for truly difficult problems, reasoning alone is not sufficient. This is the case for most research, for example in any domain. In this case, intelligence consists of not simply searching through the problem domain with our learnt policy function but rather to use the traces of our search process as input data in order to learn a better policy function, and even a better language to search in, for the problem we are interested in. One can view this as a "distillation from an agent given more time to an agent given less time". I believe this will be the next step in AI progress.

However, even more important in my opinion, is the reshaping of our search language through the discovery and naming of new "concepts". Mathematics is the domain where this process has come closest to perfection, but it is a crucially important component of any research field. It remains a completely open question to implement this in artificial intelligence, albeit one that I have some promising speculations about.

The reason that such explicit naming of concepts is so important is many-fold. For one, it allows for explicit iteration and improvement, both in time and across a community. Second, it is the most efficient way of *reshaping the search landscape* - while the implicit policy function can serve a similar purpose (and indeed, I believe that current deep neural networks learn intrinsic concepts for exactly this purpose), the right "definitions" can suggest new directions and connections that go beyond any implicit intuition. And third, such explicit naming allows one to transmit the concept to other entities through their explicit use in various situations. In this way, we can leverage *collective intelligence*, instead of just individual intelligence. This is precisely why learning *from* the best chess playing machines, for instance, is so slow compared to learning from the best humans and why we do not see recursive improvement in the best machines except through better hardware and longer training time.

5 Goodharting and Substitute Metrics

And in this section, I would like to discuss the problem that I see as most relevant to "mislaignment" or learning the wrong model. In training current AI, we are often trying to force the model to learn some ill-defined property such as intelligence and we train the network through a well-defined proxy for this quantity (such as perplexity for LLMs). It is a remarkable fact that this substitute metric does often induce the true property we are aiming for in many cases, but not in every case and the deviations can be critical in many systems.

This is often the result of "over-training" such systems and sometimes goes under the name of goodharting. Indeed, if one conceptualizes the space of all possible weights for our system, the optimum for the approximate metric might fall somewhere close to, but not exactly at, the optimum for the true metric and under moderate training, our network might fall somewhere close to the approximate optimum but also close to the true optimum. However, on extended training, as our network approaches the approximate optimum ever closer, it might get farther away from the true metric. For this heuristic sketch to work well, it suggests that the true geometry of the weight space might be *non-archimedean* or *hyperbolic* and made up of several distinct levels close in the weight metric but distant in the effective output. This should be a rich domain for future mathematical investigation, and an important one.

However, there is another way in which training for an approximate metric might make the system misbehave. Even if the approximate optimum and true optimum are exactly the same, it is entirely possible that the gradient descent paths towards the two metrics diverge greatly before finally converging. If so, then any approximate learning could lead to wildly different-from-expected behaviour. This hypothesis in fact explains some interesting features of existing neural networks. For example, in training Alphago, Deepmind found that it was efficient to train for both a policy function and a "game state value" function. This is surprising because, in principle, it should be possible to completely recover the policy function from a game state value. In this case, I believe that the efficiency gains come from precisely a positive aspect of the gradient descent paths diverging. In optimizing for either function, the network learns distinct skills that can then be combined in order to obtain a stronger network than training for either one individually.

And indeed, this suggests that a general procedure to circumvent such goodharting might be to have several different approximate metrics that we optimize for simultaneously, in the hope that each metric penalizes over-optimization in any of the other metrics. I believe that research in this area is of the utmost importance, both theoretically and practically.

6 Self Organization

In this section, I tackle the broad question of how self-organization among intelligent agents is achieved and moreover, how a coherent sense of self is generated and maintained over long periods of time. In fact, this question is mysterious even more short periods of time. Our subjective experience of the now is in fact a *smeearing-out* of the present a little bit into the past and future and I suspect that the discreteness of this subjec-

tive experience will turn out to be crucial. After all, any computational process has to take place over some period of time and our subjective experience is surely the effect of a computational experience.

Along with time, another crucial aspect seems to be memory. From one view point, the past is completely illusory and any organism has to recreate its self-conception and agency at every moment from the materials is has at hand. On the other hand, the past clearly has great explanatory power and can be usefully compressed and stored in order to help the agent make choices. But over an above this, memory seems to play a key role in creating a conception of a unified self, if only by remembering that this was so in the past.

Finally, the mechanisms by which individual needs and goals can be suppressed in the service of the whole also seems very interesting. Levin proposes sharing *stress* as a key mechanism in achieving co-operation. It would be extremely interesting to work out the mechanics of this. After all, when a virus induces stress in its host cells, the host responds by identifying the virus as an outsider and trying to eradicate it which is the polar opposite of co-operation. On the other hand, some "parasites" do manage to cohabit in a host organism (such as our gut bacteria). This suggests another question:

Question 5. *Is the mechanism of co-operation different between cells that share the same genetic code vs those that don't? In other words, is there a significant advantage to co-operation when beginning with the same "hardware"?*

One lesson of Levin's body of work is that once one is open to the existence of unconventional intelligences in a diverse range of problem domains, one simply finds it everywhere. Levin has exploited this to great effect in the biological realm and I am interested in transporting these insights to different realms (and vice-versa).

For instance, one perspective on the global economy is as a distributed computation for finding the *relative values* of various goods. This process is truly non-local in that no individual human or even small groups of human have much influence over the value of any good relative to any other and yet, prices do get decided at both a local and global scale. This process is even quite efficient and opportunities for arbitrage are relatively rare. *Can one find analogs of this in biological networks?* And can we otherwise use insights from economics to help understand the questions outlined above?

References

[1] Stephanie Chan et al. "Data distributional properties drive emergent in-context learning in transformers". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18878–18891.

[2] Roger C Conant and W Ross Ashby. "Every good regulator of a system must be a model of that system". In: *International journal of systems science* 1.2 (1970), pp. 89–97.

[3] Thomas McGrath et al. "Acquisition of chess knowledge in alphazero". In: *Proceedings of the National Academy of Sciences* 119.47 (2022), e2206625119.

[4] Suvir Mirchandani et al. "Large Language Models as General Pattern Machines". In: *arXiv preprint arXiv:2307.04721* (2023).

[5] *On the Biology of a Large Language Model*. https://transformer-circuits.pub/2025/attribution-graphs/biology.html. Accessed: 2020-04-16.

[6] Steven T Piantadosi. "The computational origin of representation". In: *Minds and machines* 31 (2021), pp. 1–58.

[7] Lisa Schut et al. "Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero". In: *arXiv preprint arXiv:2310.16410* (2023).

[8] Adam Shai et al. "Transformers represent belief state geometry in their residual stream". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 75012–75034.